



KEMENTERIAN  
PENDIDIKAN  
MALAYSIA

# ASAS SAINS KOMPUTER

Tingkatan 3





# **RUKUN NEGARA**

**Bahawasanya Negara Kita Malaysia**  
mendukung cita-cita hendak;

Mencapai perpaduan yang lebih erat dalam kalangan  
seluruh masyarakatnya;

Memelihara satu cara hidup demokrasi;

Mencipta satu masyarakat yang adil di mana kemakmuran negara  
akan dapat dinikmati bersama secara adil dan saksama;

Menjamin satu cara yang liberal terhadap  
tradisi-tradisi kebudayaannya yang kaya dan pelbagai corak;

Membina satu masyarakat progresif yang akan menggunakan  
sains dan teknologi moden;

MAKA KAMI, rakyat Malaysia,  
berikrar akan menumpukan  
seluruh tenaga dan usaha kami untuk mencapai cita-cita tersebut  
berdasarkan prinsip-prinsip yang berikut:

**KEPERCAYAAN KEPADA TUHAN  
KESETIAAN KEPADA RAJA DAN NEGARA  
KELUHURAN PERLEMBAGAAN  
KEDAULATAN UNDANG-UNDANG  
KESOPANAN DAN KESUSILAN**

KURIKULUM STANDARD SEKOLAH MENENGAH (KSSM)

# ASAS SAINS KOMPUTER TINGKATAN 3

## **Penulis**

Wan Zaleha binti Wan Jusoh  
Mathan a/l Balaraman  
Looi Kam Fatt  
Yap Thin Peng

## **Editor**

Yeoh Joue Huey  
Izyani binti Ibrahim

## **Pereka Bentuk**

Wong Shir Yen

## **Ilustrator**

Ai Khen bin Wong



**PAN ASIA PUBLICATIONS SDN. BHD.**  
**2018**



# KEMENTERIAN PENDIDIKAN MALAYSIA

NO. SIRI BUKU: 0074

KPM2018 ISBN 978-967-466-319-3

Cetakan Pertama 2018

© Kementerian Pendidikan Malaysia

Hak Cipta Terpelihara. Mana-mana bahan dalam buku ini, tidak dibenarkan diterbitkan semula, disimpan dalam cara yang boleh dipergunakan lagi, ataupun dipindahkan dalam sebarang bentuk atau cara, baik dengan elektronik, mekanik, penggambaran semula mahupun dengan cara perakaman tanpa kebenaran terlebih dahulu daripada Ketua Pengarah Pelajaran Malaysia, Kementerian Pendidikan Malaysia. Perundingan tertakluk kepada perkiraan royalti atau honorarium.

Diterbitkan untuk Kementerian Pendidikan Malaysia oleh:

Pan Asia Publications Sdn. Bhd. (226902-X)

No. 2-16, Jalan SU 8,

Taman Perindustrian Subang Utama,

Seksyen 22, 40300 Shah Alam,

Selangor Darul Ehsan, Malaysia.

Tel: +603-5614 4168

Faks: +603-5614 4268

E-mel: [enquiry@panasiapub.com](mailto:enquiry@panasiapub.com)

Laman web: [www.panasiapub.com](http://www.panasiapub.com)

Reka Letak dan Atur Huruf:

Pan Asia Publications Sdn. Bhd. (226902-X)

Muka Taip Teks: Berling LT Std

Saiz Muka Taip Teks: 11 poin

Dicetak oleh:

Percetakan Rina Sdn. Bhd.

Lot 45, Persiaran Mewah,

Bandar Tun Razak,

56000 Cheras, Kuala Lumpur.

## Penghargaan

Penerbitan buku teks ini melibatkan kerjasama banyak pihak. Sekalung penghargaan dan terima kasih ditujukan kepada semua pihak yang terlibat:

- Jawatankuasa Penambahbaikan Pruf Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Pembetulan Pruf Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Naskhah Sedia Kamera, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Pegawai-pegawai Bahagian Buku Teks dan Bahagian Pembangunan Kurikulum, Kementerian Pendidikan Malaysia.
- Yusuf & Francisco 2009
- Sekolah Menengah Kebangsaan Indahpura 1
- MIT Media Lab @ L. Barry Hetherington

# Kandungan

Pendahuluan	v
Ciri-ciri Istimewa dalam Buku Ini dan Fungsinya	vi
Penggunaan-penggunaan Peta Pemikiran i-THINK Semasa Pembelajaran	viii
Penerangan Teknik-teknik dalam Pemikiran Komputasional	viii

## BAB 1

### KONSEP ASAS PEMIKIRAN KOMPUTASIONAL 1

1.1 Pembangunan Atur Cara	2
1.1.1 Menggunakan Konsep Pemikiran Komputasional dalam Pembangunan Atur Cara	2
1.1.2 Pelaporan Penggunaan Teknik Pemikiran Komputasional bagi Setiap Fasa Pembangunan Atur Cara	18
1.1.3 Menghasilkan Satu Projek Mini Berdasarkan Situasi Penyelesaian Masalah dengan Merujuk Fasa Pembangunan Atur Cara	23
Uji Kendiri 1.1	25
Rumusan Grafik 1	26
Semak Semula 1	26
Pentaksiran 1	27

## BAB 2

### PERWAKILAN DATA 29

2.1 Kriptografi dalam Keselamatan Data	30
2.1.1 Kriptografi dalam Pengkomputeran	30
2.1.2 Proses Sifer ( <i>Cipher</i> )	32
2.1.3 Menghasil dan Menterjemah Mesej Menggunakan Kaedah Sifer	34
2.1.4 Membanding Kaedah Sifer	45
2.1.5 Pemilihan Kaedah Sifer Terbaik	48
2.1.6 Kaedah Sifer dalam Penyelesaian Masalah	51
Uji Kendiri 2.1	55
Rumusan Grafik 2	56
Semak Semula 2	56
Pentaksiran 2	57

## BAB 3

### ALGORITMA 60

3.1 Pembangunan Algoritma	62
3.1.1 Ciri-ciri <i>Search</i> dan <i>Sort</i>	62
3.1.2 (i) Menulis Pseudokod dan Melukis Carta Alir bagi <i>Linear Search</i>	73

3.1.2 (ii) Menulis Pseudokod dan Melukis Carta Alir bagi <i>Binary Search</i>	77
3.1.3 (i) Menulis Pseudokod dan Melukis Carta Alir bagi <i>Bubble Sort</i>	81
3.1.3 (ii) Menulis Pseudokod dan Melukis Carta Alir bagi <i>Bucket Sort</i>	86
3.1.4 Mengesan dan Membaiki Ralat dalam Pseudokod dan Carta Alir	90
3.1.5 Pengecaman Corak dalam Algoritma <i>Search</i> dan <i>Sort</i>	101
3.1.6 Algoritma bagi Gabungan <i>Search</i> dan <i>Sort</i>	102
3.1.7 Algoritma bagi Gabungan <i>Search</i> atau <i>Sort</i>	104
Uji Kendiri 3.1	107
Rumusan Grafik 3	108
Semak Semula 3	108
Pentaksiran 3	109

## BAB 4

### KOD ARAHAN

**112**

4.1 Pangkalan Data dan SQL	114
4.1.1 Maksud dan Penggunaan Pangkalan Data dan <i>Structured Query Language</i> (SQL)	114
4.1.2 Entiti dan Atribut	118
4.1.3 Kekunci Primer dan Kekunci Asing	121
4.1.4 Kekardinalan ( <i>Cardinality</i> ) antara Entiti dalam Hubungan	124
4.1.5 Membina Pangkalan Data	127
4.1.6 Menghasilkan Borang dan Memasukkan Data Melalui Borang	140
4.1.7 Penggunaan Pernyataan-pernyataan SQL	150
4.1.8 Penggunaan Pernyataan-pernyataan SQL yang Melibatkan Ungkapan <i>Boolean</i>	157
4.1.9 Menjana Laporan Berdasarkan Hasil Pertanyaan ( <i>Query</i> )	161
Uji Kendiri 4.1	165
4.2 Struktur Kod Arahan	167
4.2.1 Fungsi <i>Function</i> dan <i>Procedure</i> dalam Atur Cara	167
4.2.2 Penggunaan Pernyataan <i>Function</i>	170
4.2.3 Menulis Pernyataan <i>Function</i> dan <i>Procedure</i>	176
4.2.4 Menulis Atur Cara yang Melibatkan <i>Function</i> dan <i>Procedure</i>	182
4.2.5 Menguji Atur Cara dan Membaiki Ralat	189
4.2.6 Menghasilkan Atur Cara yang Melibatkan Gabungan Struktur Kod Arahan bagi Menyelesaikan Masalah	200
Uji Kendiri 4.2	210
Rumusan Grafik 4	211
Semak Semula 4	212
Pentaksiran 4	213
JAWAPAN	217
GLOSARI	229
INDEKS	231
SENARAI RUJUKAN	232

# Pendahuluan

Buku Teks Asas Sains Komputer Tingkatan 3 ini ditulis berdasarkan Dokumen Standard Kurikulum dan Pentaksiran (DSKP) Asas Sains Komputer Tingkatan 3 yang diterbitkan oleh Kementerian Pendidikan Malaysia bagi Kurikulum Standard Sekolah Menengah (KSSM). Fokus kandungan buku teks ini menjurus kepada penguasaan ilmu pengetahuan, kemahiran dan nilai yang perlu diterapkan dalam diri setiap murid. Standard Kandungan, Standard Pembelajaran dan Standard Prestasi yang terdapat dalam DSKP menjadi rujukan bagi pembinaan teks yang berkualiti seiring dengan perkembangan teknologi dan menjadikan pembelajaran murid-murid lebih bermakna.

Kandungan dalam buku teks ini dapat memantapkan lagi proses pengajaran dan pembelajaran guru serta murid. Aktiviti dan tugas yang dibina berdasarkan pembelajaran berasaskan inkuiri dapat membentuk pemikiran murid agar berfikir secara lebih kritis dan kreatif. Konsep pemikiran komputasional yang ditekankan di dalam buku teks ini pula bertujuan untuk meningkatkan kemahiran murid dalam penyelesaian masalah, mereka bentuk dan mencipta aplikasi baharu.

Pelbagai bentuk dan jenis soalan, aktiviti serta tugas yang berorientasikan Pembelajaran Abad Ke-21 disediakan agar pemikiran murid lebih luas, matang dan bertanggungjawab dalam menyelesaikan sesuatu masalah. Melalui Pembelajaran Abad Ke-21 ini, murid-murid secara tidak langsung dapat meningkatkan keyakinan diri dan seterusnya mampu bersaing di peringkat global.

Elemen Kemahiran Berfikir Aras Tinggi (KBAT) juga diberi penekanan dalam latihan dan tugas bagi membentuk murid yang berdaya saing serta mampu melontarkan pandangan, membuat keputusan, mencipta dan berinovasi. Bagi memastikan murid mampu menangani cabaran pada masa akan datang, tugas yang bersifat Elemen Merentas Kurikulum (EMK) juga disediakan.

Selain itu, pendekatan STEM (*Science, Technology, Engineering and Mathematics*) turut didedahkan supaya murid-murid dapat meneroka persekitaran melalui inkuiri dan seterusnya dapat menyelesaikan masalah dalam kehidupan seharian. Pendekatan STEM ini diterapkan melalui tugas dan projek yang dapat dilakukan oleh murid secara individu, berpasangan atau berkumpulan.

## Ciri-ciri Istimewa dalam Buku Ini dan Fungsinya



QR code pada kulit depan buku mengandungi huraian tema buku, biodata penulis serta maklumat dan fakta yang dikemas kini (sekiranya ada).



Menyediakan pelbagai jenis aktiviti pembelajaran abad ke-21 untuk membantu murid meningkatkan kemahiran.



Membekalkan murid dengan aktiviti yang melibatkan perisian komputer.

### Contoh 1.1

Memberikan contoh penyelesaian terhadap masalah dengan mengaplikasikan konsep dan teori yang dipelajari.



Imbasan QR code yang mengandungi maklumat lanjut dan video tutorial tentang topik yang dipelajari. Pautan laman sesawang diberikan untuk memudahkan murid mengakses maklumat.

### Tahukah Anda?

Mengemukakan fakta atau maklumat tambahan yang bertujuan untuk menambahkan dan meluaskan pengetahuan murid.



Menyediakan soalan yang merangsang minda murid untuk berfikir dan meningkatkan kemahiran murid tentang topik yang dipelajari.



### Kerjaya Bidang Sains Komputer

Mencadangkan kerjaya-kerjaya dalam bidang sains komputer yang ditawarkan dalam kehidupan sebenar yang dapat diceburi oleh murid.





Menyediakan maklumat tentang topik-topik yang pernah dipelajari oleh murid.



Soalan dan aktiviti yang diberikan ialah jenis KBAT.

**STEM**

Tugasan yang memerlukan penerapan pendekatan STEM untuk menyelesaikan masalah.



Menyediakan fail Python bagi murid memuat turun dan mencuba sendiri atur cara yang diterangkan dalam buku ini.

**E-PRAKTIS**



Menyediakan praktis tambahan berbentuk kuiz dalam talian untuk mengukuhkan lagi pemahaman murid.



Menyediakan latihan formatif sebagai pengukuhan konsep kepada murid pada akhir setiap subtopik.

**RUMUSAN GRAFIK 1**

Rumusan ringkas tentang bab dalam bentuk peta minda.

**SEMAK SEMULA 1**

Senarai semak disediakan untuk mengetahui kemahiran-kemahiran yang perlu dikuasai oleh murid.

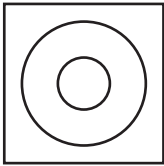
**PENTAKSIRAN 1**

Menyediakan soalan yang pelbagai berbentuk objektif, struktur dan penyelesaian masalah bagi menguji tahap kefahaman murid terhadap topik yang telah dipelajari.

## Penggunaan-Penggunaan Peta Pemikiran i-THINK Semasa Pembelajaran

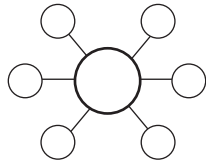
### Peta Bulatan

Sesuai digunakan semasa aktiviti sumbang saran untuk menyenaraikan idea



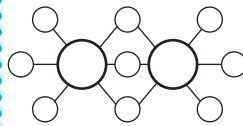
### Peta Buih

Sesuai digunakan untuk menunjukkan ciri-ciri sesuatu item



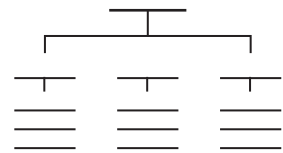
### Peta Buih Berganda

Sesuai digunakan untuk membuat perbandingan dan perbezaan



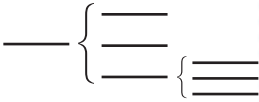
### Peta Pokok

Sesuai digunakan untuk mengelaskan item



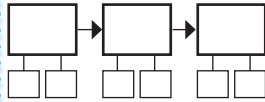
### Peta Dakap

Sesuai digunakan untuk menganalisis bahagian sesuatu item



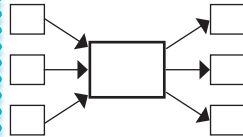
### Peta Alir

Sesuai digunakan untuk membuat urutan sesuatu proses



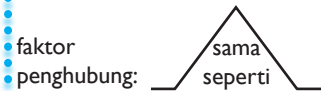
### Peta Pelbagai Alir

Sesuai digunakan untuk menunjukkan punca dan kesan sesuatu proses atau situasi



### Peta Titi

Sesuai digunakan untuk membuat analogi dengan faktor penghubung



## Penerangan Teknik-teknik dalam Pemikiran Komputasional

### Teknik Leraian

Memecahkan masalah yang besar kepada bahagian-bahagian masalah yang lebih kecil.



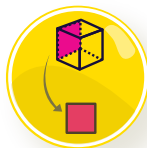
### Teknik Pengecaman Corak

Mengenal pasti ciri-ciri persamaan corak pada bahagian-bahagian masalah yang kecil



### Teknik Peniskalaan

Mengenal pasti aspek-aspek penting dan meninggalkan aspek-aspek kurang penting.



### Teknik Pengitlakan

Menghasilkan model bagi menyelesaikan masalah yang serupa.



# BAB 1

# KONSEP ASAS PEMIKIRAN KOMPUTASIONAL

Pemikiran komputasional telah membantu manusia bagi menyelesaikan masalah secara sistematik. Selaras dengan perkembangan teknologi yang pesat, konsep pemikiran komputasional perlu diberi penekanan yang serius malah ilmu pengaturcaraan juga perlu dikuasai dengan baik bagi melahirkan generasi pencipta teknologi yang unggul.

## Standard Kandungan

### 1.1 Pembangunan Atur Cara

Tonton video ini untuk mengetahui tentang gambaran Bab 1.



[goo.gl/Y36Pft](https://goo.gl/Y36Pft)

Imbas  
Saya

## Kata Kunci

- Fasa pembangunan atur cara
- Leraian
- Pengecaman corak
- Pengitlakan
- Peniskalaan

**Teknik Leraian** membolehkan jaguh catur menentukan pergerakan langkah yang boleh dibuat dan menganalisis setiap langkah itu.

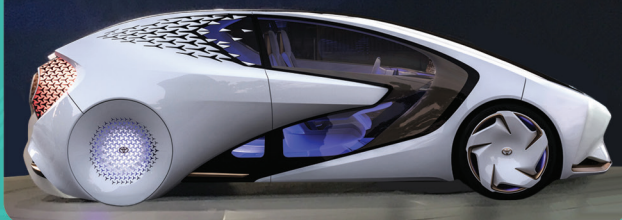


**Teknik Pengecaman Corak** digunakan untuk membandingkan corak perubahan cuaca.



Pemikiran  
Komputasional

**Teknik Pengitlakan** melibatkan penghasilan model, formula dan algoritma seperti penghasilan kenderaan konsep dalam industri kereta.



**Teknik Peniskalaan** digunakan untuk mengenal pasti ciri-ciri penting sesuatu virus untuk memahami kelakuannya.



# 1.1 PEMBANGUNAN ATUR CARA

## Imbas Kembali

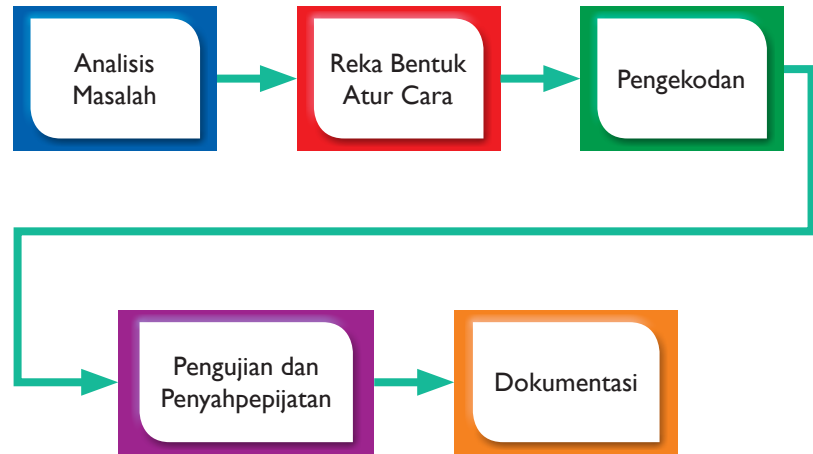
Algoritma digunakan untuk menyelesaikan sesuatu masalah secara langkah demi langkah. Algoritma yang terdiri daripada pseudokod dan carta alir akan digunakan dalam fasa reka bentuk atur cara.



## Kerjaya Bidang Sains Komputer

**Pembangun perisian (software developer)** bertanggungjawab dan terlibat dalam fasa-fasa pembangunan atur cara.

Pembangunan atur cara terdiri daripada lima fasa, iaitu fasa analisis masalah, fasa reka bentuk atur cara, fasa pengekodan, fasa pengujian dan penyahpejijatan serta fasa dokumentasi. Rajah 1.1 menunjukkan fasa-fasa pembangunan atur cara.



Rajah 1.1 Fasa-fasa pembangunan atur cara

## Standard Pembelajaran

Murid boleh:

- 1.1.1** Menggunakan konsep pemikiran komputasional dalam fasa pembangunan atur cara bagi membangunkan atur cara aritmetik:
- Analisis masalah
  - Reka bentuk atur cara
  - Pengekodan
  - Pengujian dan penyahpejijatan
  - Dokumentasi

## 1.1.1

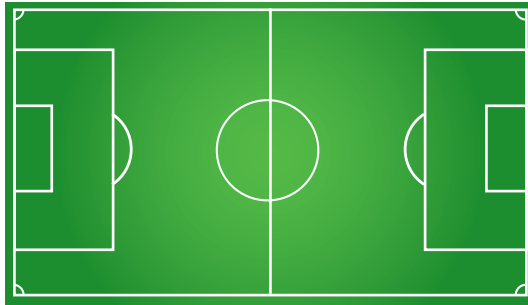
### Menggunakan Konsep Pemikiran Komputasional dalam Pembangunan Atur Cara

**Pemikiran komputasional** ditakrifkan sebagai satu proses pemikiran bertujuan untuk menyelesaikan masalah oleh manusia sendiri berbantuan mesin atau kedua-duanya sekali dengan menggunakan konsep asas sains komputer. Proses penyelesaian masalah ini melibatkan usaha memecahkan masalah kepada komponen yang lebih kecil dan mencari penyelesaian secara sistematik.

Pernyataan berikut menunjukkan suatu masalah yang diketengahkan bagi membincangkan tentang penggunaan konsep pemikiran komputasional dalam pembangunan atur cara.

**Anda diminta untuk membina atur cara mudah bagi mengira luas dan perimeter sebuah padang bola yang berbentuk segi empat tepat.**

Rajah 1.2 menunjukkan contoh sebuah padang bola berbentuk segi empat tepat.



Rajah 1.2 Padang bola berbentuk segi empat tepat

**(i) Fasa Analisis Masalah**

Dalam fasa ini, pengatur cara akan:

- (a) Mengenal pasti masalah, keperluan sistem dan perisian serta sasaran pengguna.
- (b) Mengenal pasti input, proses dan output bagi atur cara yang akan dibina.
- (c) Menemu bual, membuat soal selidik dan pemerhatian bagi mengetahui keperluan pelanggan.

**>> Teknik Leraian**

Teknik leraian digunakan untuk menganalisis masalah dengan memecahkan masalah yang besar kepada bahagian-bahagian yang lebih kecil. Bahagian-bahagian kecil ini ialah masalah kecil yang lebih mudah diselesaikan. Rajah 1.3 menunjukkan penggunaan teknik leraian dalam fasa analisis masalah.



Rajah 1.3 Penggunaan teknik leraian dalam fasa analisis masalah

**Tahukah Anda?**



Pada tahun 1843, Augusta Ada Lovelace telah menterjemah dan menerbitkan satu artikel dalam jurnal sains tentang mesin *Analytical Engine*. Dalam artikel yang sama, beliau juga menerangkan cara menulis atur cara untuk mengarahkan mesin memproses huruf dan simbol dan mengulangi pelaksanaan satu set tindakan. Beliau digelar sebagai pengatur cara komputer yang pertama. (Sumber: [goo.gl/zu0IIj](http://goo.gl/zu0IIj))

**Uji Minda 1.1**

Berapakah bilangan teknik pemikiran komputasional yang sesuai digunakan dalam setiap satu fasa pembangunan atur cara?

Teknik-teknik pemikiran komputasional



[goo.gl/XY2hRj](https://goo.gl/XY2hRj)



Video kitar hayat pembangunan atur cara



[goo.gl/AjFc8C](https://goo.gl/AjFc8C)



## Tahukah Anda?

Berdasarkan Kementerian Pendidikan Malaysia, fokus Pembestarian Sekolah (2016–2020) menekankan kepada peningkatan kemahiran dan kompetensi digital serta pemikiran komputasional murid melalui KSSR dan KSSM.

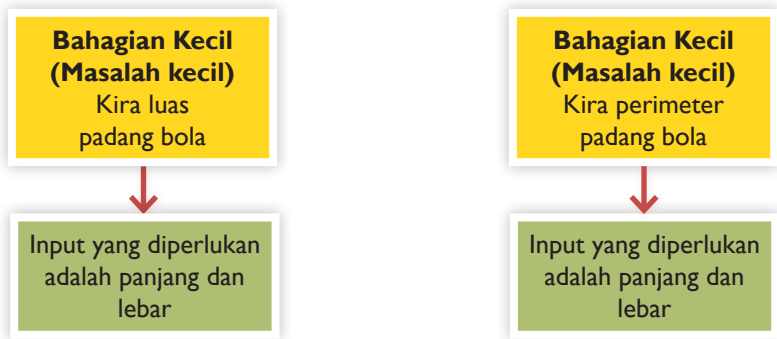
## (ii) Fasa Reka Bentuk Atur Cara

Dalam fasa ini, pengatur cara akan:

- Menulis pseudokod.
- Melukis carta alir.
- Mereka bentuk antara muka pengguna.

### >> Teknik Pengecaman Corak

Selepas meleraikan masalah besar, setiap bahagian kecil atau masalah kecil akan dianalisis untuk mengecam corak-corak yang tertentu sekiranya ada. Pengecaman corak ini dapat membantu dalam mencari penyelesaian masalah dengan lebih efisien. Rajah 1.4 menunjukkan corak yang wujud dalam pengiraan luas dan perimeter sebuah padang bola. Didapati bahawa pengiraan bagi luas dan perimeter tersebut menggunakan pemboleh ubah yang sama, iaitu panjang dan lebar.



Rajah 1.4 Penggunaan teknik pengecaman corak dalam fasa reka bentuk atur cara

### >> Teknik Peniskalaan

Setelah corak-corak dicamkan, corak-corak ini diteliti untuk meninggalkan aspek-aspek kurang penting dan memberikan penekanan terhadap aspek-aspek penting yang dapat membantu dalam penyelesaian masalah. Pemboleh ubah merupakan jenis corak yang telah dicamkan manakala panjang dan lebar merupakan aspek penting yang perlu diberikan penekanan. Panjang dan lebar padang bola yang berbentuk segi empat tepat serta formula matematik yang berkenaan diperlukan untuk mendapatkan luas dan perimeter padang bola tersebut. Formula untuk mengira luas dan perimeter sebuah padang bola adalah seperti berikut:

$$\text{Luas padang bola} = \text{panjang} \times \text{lebar}$$

$$\text{Perimeter padang bola} = (\text{panjang} + \text{lebar}) \times 2$$

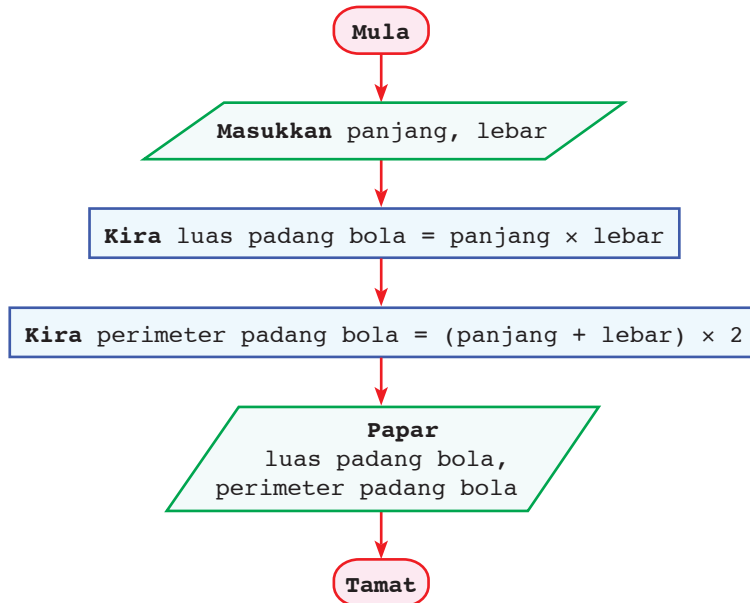
### ►► Teknik Pengitlakan

Berdasarkan maklumat yang diperoleh daripada leraian dan peniskalaan masalah yang dilakukan, algoritma yang diperlukan untuk menyelesaikan masalah pengiraan luas dan perimeter padang bola boleh ditentukan. Algoritma komputer biasanya terdapat dalam bentuk pseudokod atau carta alir. Aspek penting yang perlu ditekankan semasa menulis algoritma ialah langkah-langkah yang perlu dilaksanakan untuk menghasilkan pseudokod dan carta alir yang betul. Bagi carta alir, penggunaan simbol yang betul untuk sesuatu langkah merupakan aspek yang penting kerana simbol-simbol tersebut merujuk kepada suatu fungsi yang tertentu.

Rajah 1.5 dan Rajah 1.6 masing-masing menunjukkan pseudokod dan carta alir bagi mengira luas dan perimeter sebuah padang bola.

- 1 **Mula**
- 2 **Masukkan** panjang, lebar
- 3 **Kira**
  - 3.1 luas padang bola = panjang × lebar
  - 3.2 perimeter padang bola = (panjang + lebar) × 2
- 4 **Papar** luas padang bola, perimeter padang bola
- 5 **Tamat**

Rajah 1.5 Pseudokod bagi mengira luas dan perimeter padang bola



Rajah 1.6 Carta alir bagi mengira luas dan perimeter padang bola

Definisi dan contoh pseudokod



[goo.gl/9x0aME](https://goo.gl/9x0aME)



**Uji Minda 1.2**

Adakah pemikiran komputasional merupakan perkara yang baharu atau telah lama wujud? Berikan alasan kepada jawapan anda.

**Uji Minda 1.3**

Adakah pseudokod mempunyai cara penulisan piawai? Berikan alasan kepada jawapan anda.

**Uji Minda 1.4**

Nyatakan maksud bagi simbol carta alir berikut:

	?
	?
	?
	?

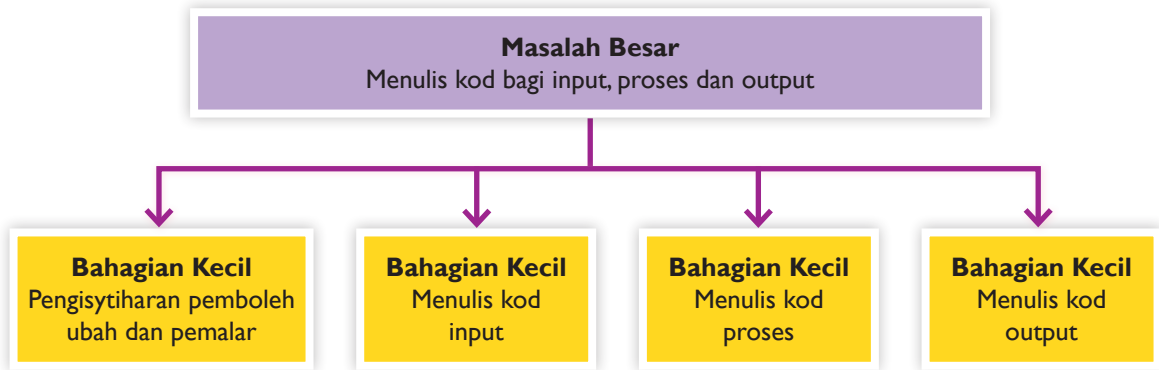
### (iii) Fasa Pengekodaan

Dalam fasa ini, pengatur cara akan:

- Memilih bahasa pengaturcaraan, misalnya Python dan perisian pengaturcaraan, misalnya Pyscripter dan IDLE.
- Mengekod atur cara berdasarkan pseudokod dan carta alir.

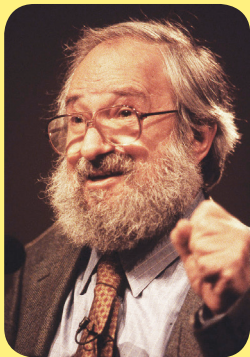
#### >> Teknik Leraian

Semasa fasa pengekodaan, anda boleh menggunakan teknik leraian untuk mengenal pasti komponen masalah ketika menulis kod arahan. Rajah 1.7 menunjukkan penggunaan teknik leraian bagi mengatasi masalah penulisan kod arahan.



Rajah 1.7 Penggunaan teknik leraian bagi mengatasi masalah penulisan kod arahan

### Tahukah Anda?



Seymour Papert merupakan individu yang pertama memperkenalkan pemikiran komputasional untuk pembelajaran. (Sumber: [goo.gl/SbdAP](http://goo.gl/SbdAP))

#### >> Teknik Pengecaman Corak

Teknik pengecaman corak memberikan penekanan kepada pemboleh ubah bagi input, iaitu panjang dan lebar kerana kedua-duanya menunjukkan corak yang sama. Nilai bagi pemboleh ubah pula merupakan corak yang berbeza kerana pengguna boleh memasukkan sebarang nilai yang ingin diuji bagi mendapatkan luas dan perimeter sebuah padang bola. Jadual 1.1 menunjukkan persamaan dan perbezaan pemboleh ubah.

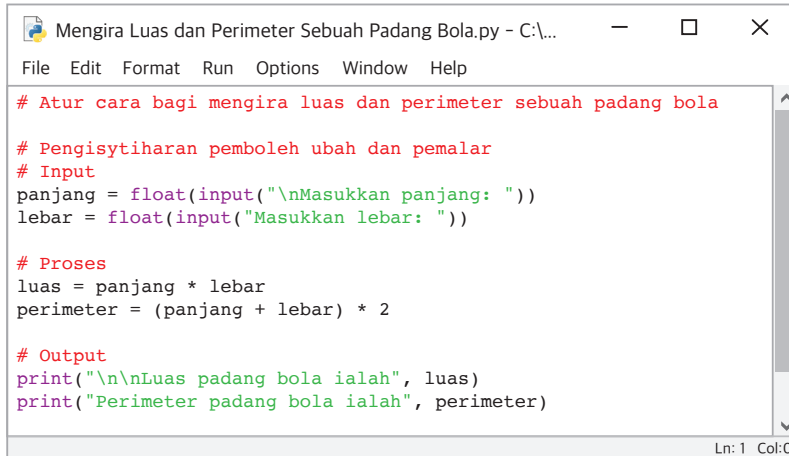
Jadual 1.1 Persamaan dan perbezaan pemboleh ubah

Persamaan	Perbezaan
Pemboleh ubah (a) panjang (b) lebar	Nilai luas dan perimeter padang bola (a) Luas = panjang $\times$ lebar (b) Perimeter = (panjang + lebar) $\times$ 2



## ►► Teknik Peniskalaan

Aspek-aspek penting bagi setiap bahagian masalah kecil adalah berlainan. Contohnya penggunaan pemboleh ubah yang betul, kod arahan input, formula yang digunakan dan kod arahan output perlu diambil kira semasa penulisan kod arahan. Rajah 1.8 menunjukkan kod arahan bagi input, proses dan output dalam bahasa pengaturcaraan Python.



```

Mengira Luas dan Perimeter Sebuah Padang Bola.py - C:\...
File Edit Format Run Options Window Help

# Atur cara bagi mengira luas dan perimeter sebuah padang bola

# Pengisytiharan pemboleh ubah dan pemalar
# Input
panjang = float(input("\nMasukkan panjang: "))
lebar = float(input("Masukkan lebar: "))

# Proses
luas = panjang * lebar
perimeter = (panjang + lebar) * 2

# Output
print("\n\nLuas padang bola ialah", luas)
print("Perimeter padang bola ialah", perimeter)
Ln: 1 Col:0
  
```

**Rajah 1.8** Contoh kod arahan bagi input, proses dan output dalam bahasa pengaturcaraan Python

## (iv) Fasa Pengujian Atur Cara dan Penyahpejatan Ralat

Dalam fasa ini, pengatur cara akan:

- Menguji atur cara yang telah ditulis bagi mengesan dan membetulkan sebarang ralat yang terdapat pada atur cara.
- Mengenal pasti ralat-ralat yang ada. Terdapat tiga jenis ralat, iaitu:
  - Ralat sintaks (*syntax error*)  
Ralat yang disebabkan oleh penggunaan sintaks sesuatu bahasa pengaturcaraan yang tidak betul dalam penulisan satu atur cara.
  - Ralat masa larian (*runtime error*)  
Ralat yang disebabkan oleh kemasukan data yang tidak menepati kehendak arahan. Ralat ini akan menyebabkan pelaksanaan atur cara terhenti secara tiba-tiba dengan mengeluarkan paparan mesej ralat yang tertentu.
  - Ralat logik (*logical error*)  
Ralat yang disebabkan oleh kesilapan logik pengatur cara sehingga menyebabkan atur cara menghasilkan output yang salah.

Jenis-jenis data dalam pengaturcaraan Python



[goo.gl/KXzHhn](https://goo.gl/KXzHhn)



Penulisan kod arahan terbahagi kepada tiga bahagian, iaitu input, proses dan output.

## Tahukah Anda?

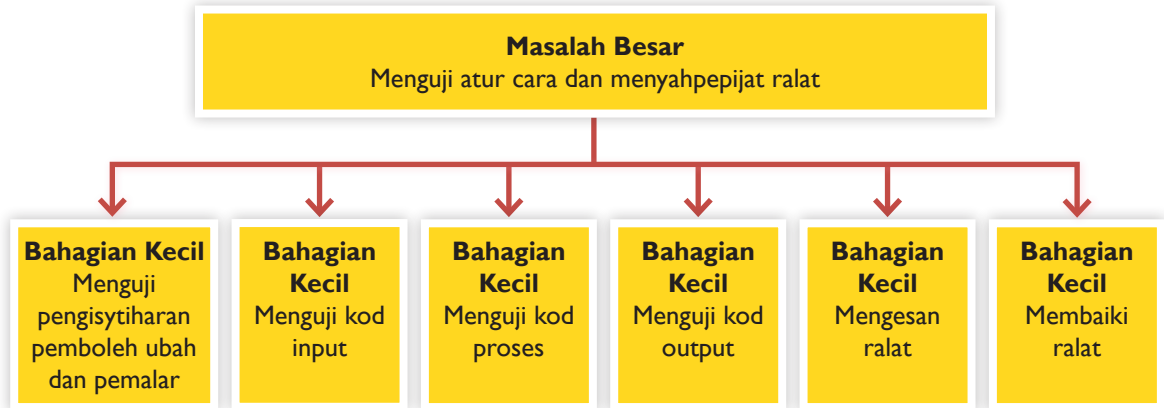
Terdapat dua lagi teknik pemikiran komputasional selain empat teknik yang telah anda pelajari di Tingkatan 1. Teknik komputasional tersebut ialah **penaakulan logik** (*logical reasoning*) dan **penilaian** (*evaluation*).

## Uji Minda 1.5

Apakah jenis ralat yang terhasil apabila pengatur cara menulis formula yang salah semasa membuat pengekodan?

## ►► Teknik Leraian

Selepas meleraikan masalah besar kepada bahagian-bahagian kecil, pengujian atur cara dan penyahpejatan ralat dapat dilakukan secara berperingkat. Rajah 1.9 menunjukkan penggunaan teknik leraian bagi pengujian atur cara dan penyahpejatan ralat secara berperingkat.



Rajah 1.9 Penggunaan teknik leraian bagi menyamak atur cara secara berperingkat

## Tahukah Anda?



Guido van Rossum ialah pencipta bahasa pengaturcaraan Python.

## ►► Teknik Pengecaman Corak

Tujuan utama proses pengujian kod arahan adalah untuk memastikan atur cara berfungsi dengan baik dan memaparkan output yang betul. Semasa proses pengujian, jenis mesej ralat yang diterima atau cara sesuatu atur cara bertindak perlu dikenal pasti. Setiap ralat yang ditemui perlu dikenal pasti ciri-ciri persamaan dan perbezaannya. Pengecaman corak-corak ini dapat membantu anda dalam mengenal pasti jenis-jenis ralat yang ditemui, iaitu ralat sintaks, ralat masa larian atau ralat logik.

Pengecaman corak boleh dibuat berdasarkan mesej ralat yang dipaparkan semasa pengujian. Paparan mesej ralat itu sendiri merupakan corak yang dicamkan. Persamaan corak pada mesej ralat membolehkan anda mengenal pasti jenis ralat tersebut. Jadual 1.2 menunjukkan corak paparan mesej ralat bagi setiap jenis ralat.

Jadual 1.2 Corak paparan mesej ralat

Jenis ralat	Contoh paparan mesej ralat
Ralat sintaks	Semua paparan mesej ralat mempunyai perkataan <i>invalid syntax</i> .
Ralat masa larian	Semua paparan mesej ralat akan menyatakan baris yang mana kod arahan adalah salah. Contohnya <i>line 13</i> .
Ralat logik	Semua paparan akan menunjukkan data input dan maklumat output tetapi outputnya tidak betul.

Jadual 1.3 menunjukkan punca-punca lazim dan contoh kesalahan yang menyebabkan ralat.

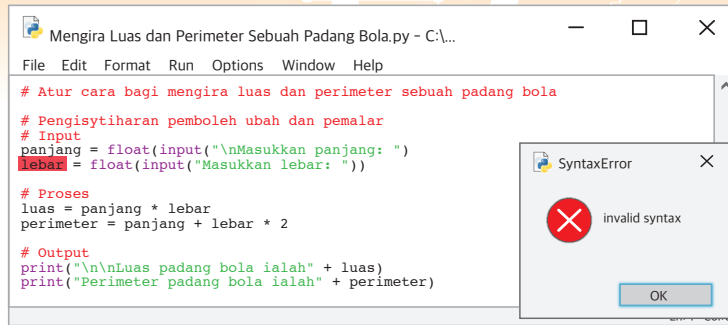
**Jadual 1.3** Punca-punca lazim dan contoh kesalahan yang menyebabkan ralat

Jenis ralat	Punca ralat	Contoh kesalahan
Ralat sintaks	<ol style="list-style-type: none"> <li>1. Kesalahan mengeja kata kunci.</li> <li>2. Kesalahan atau tertinggal tanda-tanda seperti tanda kurung, tanda titik bertindih, tanda petik dan lain-lain.</li> <li>3. Kesalahan menggunakan sintaks bahasa pengaturcaraan.</li> <li>4. Kesalahan menggunakan inden.</li> </ol>	<pre># Atur cara bagi mengira luas dan perimeter sebuah padang bola # Pengisytiharan pemboleh ubah dan pemalar # Input panjang = float(input("\nMasukkan panjang: ")) lebar = float(input("Masukkan lebar: "))</pre> <p>Tiada simbol hash (#) pada penulisan komen.</p> <p>Tiada tanda petik (") sebelum tanda kurung.</p>
Ralat masa larian	<ol style="list-style-type: none"> <li>1. Kesalahan memberikan atau tidak memberikan jenis data kepada pemboleh ubah.</li> <li>2. Kesalahan menggunakan pemboleh ubah atau pemalar yang tidak wujud.</li> <li>3. Kesalahan melakukan pembahagian sifar.</li> <li>4. Kesalahan menggunakan operator matematik dan operator logik pada pemboleh ubah yang berjenis data <i>string</i>.</li> </ol>	<pre># Atur cara bagi mengira luas dan perimeter sebuah padang bola # Pengisytiharan pemboleh ubah dan pemalar # Input panjang = float(input("\nMasukkan panjang: ")) lebar = float(input("Masukkan lebar: ")) # Proses luas = panjang * tinggi perimeter = (panjang * lebar) * 2</pre> <p>Kesalahan menggunakan pemboleh ubah "tinggi".</p>
Ralat logik	<ol style="list-style-type: none"> <li>1. Kesalahan membuat pengiraan.</li> <li>2. Kesalahan ungkapan <i>boolean</i>.</li> <li>3. Kesalahan menggunakan operator.</li> </ol>	<pre># Atur cara bagi mengira luas dan perimeter sebuah padang bola # Pengisytiharan pemboleh ubah dan pemalar # Input panjang = float(input("\nMasukkan panjang: ")) lebar = float(input("Masukkan lebar: ")) # Proses luas = panjang * tinggi perimeter = (panjang * lebar) * 2</pre> <p>Kesalahan menggunakan operator "*".</p>

## ➤➤ Teknik Peniskalaan

Teknik peniskalaan digunakan semasa penyahpepijatan dengan memberikan penekanan kepada aspek penting untuk menyahpepijat ralat, iaitu mengetahui sintaks bahasa pengaturcaraan, cara untuk mengesan ralat atau kedudukan ralat dan logik sesuatu atur cara.

Penyahpepijatan ialah proses mencari dan memperbaiki ralat yang berlaku. Pelaksanaan kod arahan dilakukan oleh penterjemah Python sebaris demi sebaris bermula daripada kod input. Rajah 1.10 menunjukkan paparan mesej ralat sintaks.

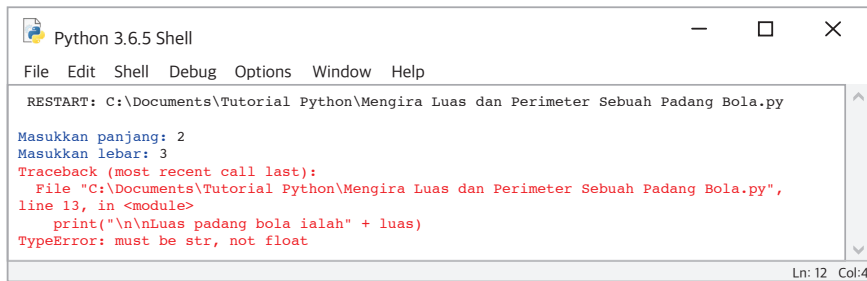


**Rajah 1.10** Paparan mesej ralat sintaks

Baris di mana penterjemah Python berhenti akan ditandakan merah seperti dalam Rajah 1.10. Kejadian ralat sintaks berada di atas baris yang ditandakan merah. Punca ralat sintaks ini adalah disebabkan oleh kekurangan satu tanda kurung. Ralat boleh dibaiki dengan menaipkan simbol “)” seperti yang ditunjukkan di bawah.

```
panjang = float(input("\nMasukkan panjang: "))
```

Setelah ralat sintaks dibaiki, atur cara dilaksanakan sekali lagi. Atur cara terhenti secara tiba-tiba selepas memasukkan input bagi panjang dan lebar. Rajah 1.11 menunjukkan paparan mesej ralat masa larian.



**Rajah 1.11** Paparan mesej ralat masa larian

Mesej ralat dalam Rajah 1.11 menerangkan kesalahan mencantumkan pemboleh ubah berjenis data *float* dalam ayat yang berjenis data *string*. Kejadian ini berlaku di baris 13. Jadual 1.4 menunjukkan kod arahan yang menyebabkan ralat dan cara membaiki ralat ini.

**Jadual 1.4** Kod arahan yang menyebabkan ralat dan cara membaiki ralat masa larian

Kod arahan yang mengandungi ralat	Penerangan
<pre>print("\n\nLuas padang bola ialah" + luas) print("Perimeter padang bola ialah" + perimeter)</pre>	Dalam <i>string</i> , simbol (+) digunakan untuk mencantumkan dua <i>string</i> . Jika nilai pemboleh ubah bukan <i>string</i> , ralat akan berlaku.
Cara membaiki ralat	Penerangan
<pre>print("\n\nLuas padang bola ialah", luas) print("Perimeter padang bola ialah", perimeter)</pre>	Simbol koma (,) bertindak sebagai pemisah bagi dua nilai yang perlu dipaparkan, iaitu teks yang berjenis data <i>string</i> dan pemboleh ubah yang berjenis data <i>float</i> .

Setelah ralat masa larian dibaiki, atur cara dilaksanakan sekali lagi. Rajah 1.12 menunjukkan output yang diperolehi apabila nilai panjang dan lebar dimasukkan. Nilai perimeter yang dikira oleh atur cara adalah salah.

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
RESTART: C:\Documents\Tutorial Python\Mengira Luas dan Perimeter Sebuah Padang Bola.py
Masukkan panjang: 2
Masukkan lebar: 3
Luas padang bola ialah 6.0
Perimeter padang bola ialah 8.0
Ln: 12 Col:4
```



Rajah 1.12 Paparan output dengan nilai perimeter yang salah

Ralat logik ini terjadi disebabkan oleh pengiraan perimeter yang tidak tepat. Jadual 1.5 menunjukkan kod arahan yang bermasalah serta cara memperbaiki ralat logik ini.

Jadual 1.5 Semakan manual terhadap formula pengiraan perimeter

<pre>perimeter = panjang + lebar * 2             = 2 + 3 * 2             = 2 + 6             = 8</pre>	Pengkodan tanpa tanda kurung ( ) menghasilkan output yang salah.
<pre>perimeter = (panjang + lebar) * 2             = (2 + 3) * 2             = 5 * 2             = 10</pre>	Pengkodan dengan tanda kurung ( ) menghasilkan output yang betul.

Setelah kesemua ralat dibaiki, paparan kod arahan dan contoh output yang diperolehi adalah seperti dalam Rajah 1.13.

```
Mengira Luas dan Perimeter Sebuah Padang Bola.py...
File Edit Format Run Options Window Help
# Atur cara bagi mengira luas dan perimeter sebuah padang bola
# Pengisytiharan pemboleh ubah dan pemalar
# Input
panjang = float(input("\nMasukkan panjang: "))
lebar = float(input("Masukkan lebar: "))
# Proses
luas = panjang * lebar
perimeter = (panjang + lebar) * 2
# Output
print("\n\nLuas padang bola ialah", luas)
print("Perimeter padang bola ialah", perimeter)
Ln: 10 Col:13
```

(a) Kod arahan setelah semua ralat dibaiki

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
RESTART: C:\Documents\Tutorial Python\Mengira Luas dan Perimeter Sebuah Padang Bola.py
Masukkan panjang: 2
Masukkan lebar: 3
Luas padang bola ialah 6.0
Perimeter padang bola ialah 10.0
>>>
```

(b) Contoh output

Rajah 1.13 Paparan kod arahan dan contoh output

## ►► Teknik Pengitlakan

Teknik pengitlakan melibatkan pembinaan model bagi proses pengujian dan penyahpepijatan atur cara. Rajah 1.14 dan 1.15 menunjukkan senarai semak yang boleh anda bina selepas proses pengujian dan penyahpepijatan atur cara yang telah anda lakukan.

### LANGKAH-LANGKAH MENGUJI DAN MENYAHPEPIJAT

#### A Pengujian dan penyahpepijatan secara manual

(a) Semak kesesuaian nama pemboleh ubah.

(b) Sintaks kod arahan adalah betul.

(c) Logik dan susunan kod-kod arahan adalah betul.

(d) Formula matematik adalah betul.

(e) Gunakan sampel data untuk menguji atur cara.

Rajah 1.14 Senarai semak proses pengujian dan penyahpepijatan secara manual

#### B Pengujian dan penyahpepijatan semasa pelaksanaan atur cara

(a) Laksanakan atur cara.

(b) Baca dan fahami mesej ralat.

(c) Kenal pasti lokasi ralat dan baiki ralat.

(d) Ulang langkah (a) hingga (c) sehingga tiada mesej ralat.

(e) Semak output menggunakan beberapa set sampel data.

Rajah 1.15 Senarai semak proses pengujian dan penyahpepijatan atur cara

Kesimpulannya, senarai semak ini jelas menunjukkan langkah-langkah yang perlu dilakukan semasa membuat pengujian dan penyahpepijatan ralat. Penyelesaian masalah secara langkah demi langkah ini dapat digunakan untuk menyelesaikan masalah lain yang serupa.

**(v) Fasa Dokumentasi**

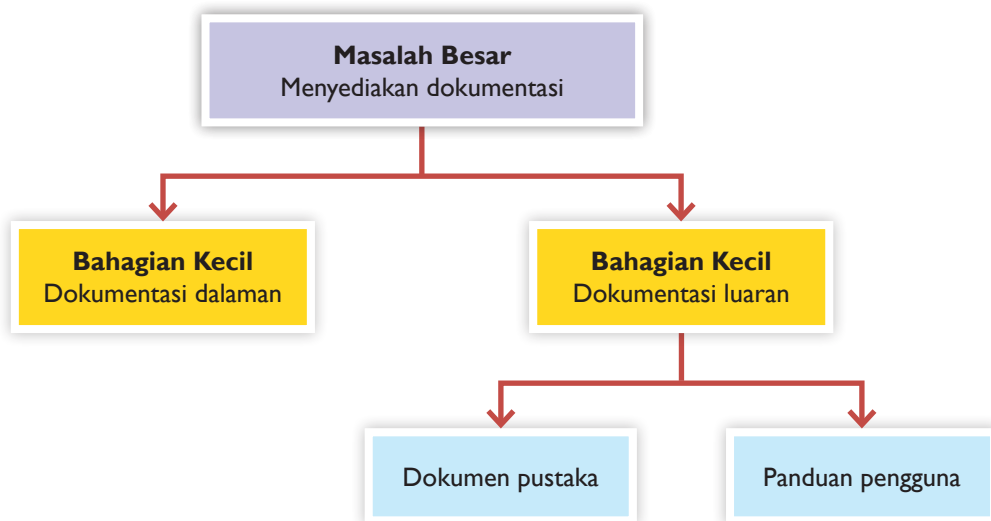
Dalam fasa ini, pengatur cara akan menyediakan dokumentasi dalaman dan dokumentasi luaran. Jadual 1.6 menunjukkan definisi serta tujuan dokumentasi dalaman dan dokumentasi luaran disediakan.

**Jadual 1.6** Definisi dan tujuan dokumentasi dalaman dan dokumentasi luaran disediakan

	Dokumentasi Dalaman	Dokumentasi Luaran
<b>Definisi</b>	Dokumen yang akan dijadikan rujukan oleh pengatur cara baharu dalam sebuah organisasi.	<ul style="list-style-type: none"> <li>Dokumen yang akan digunakan oleh pengguna akhir sesuatu atur cara.</li> <li>Terbahagi kepada dokumen pustaka dan panduan pengguna.</li> </ul>
<b>Tujuan</b>	<ul style="list-style-type: none"> <li>Membantu pengatur cara baharu untuk memahami atur cara yang telah dibangunkan.</li> <li>Memudahkan penyenggaraan sesuatu aplikasi atau atur cara.</li> </ul>	<ul style="list-style-type: none"> <li>Dokumen pustaka memberikan maklumat tentang perisian, perkakasan, algoritma atur cara dan kod arahan atur cara kepada pengguna akhir.</li> <li>Panduan pengguna memberikan maklumat kepada pengguna akhir tentang pemasangan dan cara pelaksanaan atur cara yang telah dibangunkan.</li> </ul>

**>> Teknik Leraian**

Demi memastikan dokumentasi atur cara adalah lengkap, langkah pertama yang harus diambil adalah meleraikan dokumentasi yang perlu disediakan kepada jenis dan bahagian-bahagian yang kecil. Rajah 1.16 menunjukkan penggunaan teknik leraian semasa menyediakan dokumentasi.



**Rajah 1.16** Fasa dokumentasi menggunakan teknik leraian

Teknik leraian membolehkan anda menentukan jenis dan komponen-komponen penting yang perlu ada dalam dokumentasi yang disediakan.

## ➤➤ Teknik Pengecaman Corak

Teknik pengecaman corak boleh digunakan apabila dokumentasi dalaman dan dokumentasi luaran memberikan maklumat penting tentang atur cara kepada pengguna. Justeru, maklumat penting tentang pembangunan dan penggunaan atur cara merupakan corak yang telah dicamkan melalui teknik ini. Setelah corak dicamkan, aspek-aspek penting bagi corak tersebut perlu dianalisis.

Maklumat penting yang perlu berada dalam dokumentasi dalaman dan dokumentasi luaran adalah berlainan. Ini bergantung kepada pengguna yang akan menggunakannya.

## ➤➤ Teknik Peniskalaan

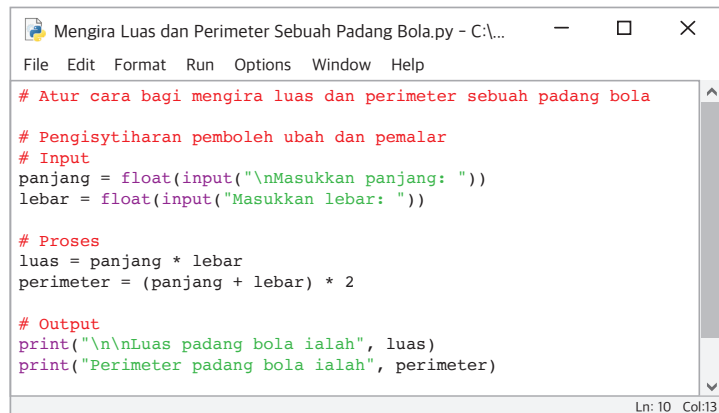
Pada peringkat ini, penggunaan teknik peniskalaan adalah sangat sesuai. Penulisan komen perlu diberikan perhatian kerana penulisan komen akan membantu pengatur cara mengingat semula atau membantu pengatur cara lain memahami maksud segmen kod yang ditulis. Justeru, ini memudahkan kerja menyenggara atur cara apabila sebarang perubahan perlu dilakukan kelak. Rajah 1.17 menunjukkan penulisan komen pada atur cara.

### Tahukah Anda?

Cara penulisan komen:

Komen tunggal  
# Komen sebaris

Komen berbilang  
"""Komen yang  
melibatkan  
lebih daripada  
satu baris"""



```
File Edit Format Run Options Window Help
# Atur cara bagi mengira luas dan perimeter sebuah padang bola
# Pengisytiharan pemboleh ubah dan pemalar
# Input
panjang = float(input("\nMasukkan panjang: "))
lebar = float(input("Masukkan lebar: "))
# Proses
luas = panjang * lebar
perimeter = (panjang + lebar) * 2
# Output
print("\n\nLuas padang bola ialah", luas)
print("Perimeter padang bola ialah", perimeter)
```

Rajah 1.17 Penulisan komen pada atur cara

## ➤➤ Teknik Pengitlakan

Selepas mengenal pasti aspek-aspek penting yang dilakukan semasa teknik peniskalaan, anda perlu menghasilkan satu dokumentasi yang sebenar. Rajah 1.18 menunjukkan contoh dokumen pustaka yang diperlukan oleh seorang pengatur cara dan Rajah 1.19 menunjukkan contoh panduan pengguna.

Kedua-dua dokumen yang dihasilkan dapat dijadikan panduan pada masa depan jika pengatur cara menghadapi masalah menyediakan dokumentasi.

Penulisan komen dalam atur cara Python



[goo.gl/7WUL5J](https://goo.gl/7WUL5J)





## DOKUMEN PUSTAKA (LIBRARY DOCUMENTATION)

Keperluan Sistem Pengendalian	Windows/Mac OS X/Linux
Keperluan Perisian	Python 3.6.5
Keperluan Ingatan	Minimum 2GB

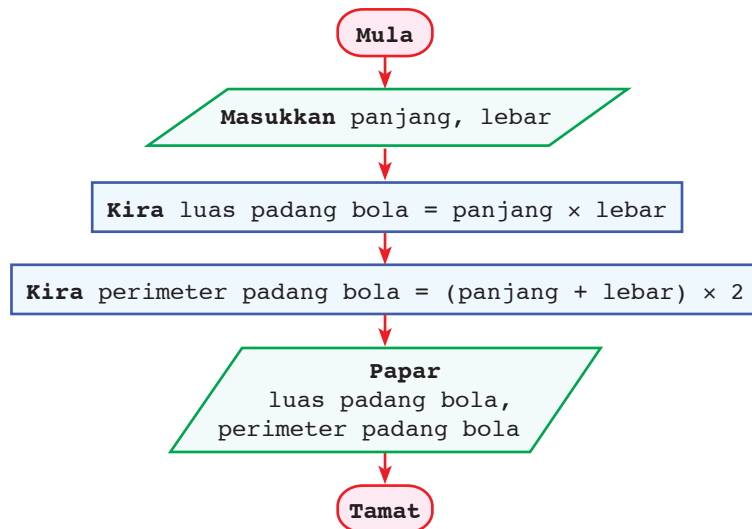
### Pseudokod

```

1 Mula
2 Masukkan panjang, lebar
3 Kira
  3.1 luas padang bola = panjang × lebar
  3.2 perimeter padang bola = (panjang + lebar) × 2
4 Papar luas padang bola, perimeter padang bola
5 Tamat

```

### Carta Alir



### Kod arahan

```

Mengira Luas dan Perimeter Sebuah Padang Bola.py - C:\...
File Edit Format Run Options Window Help
# Atur cara bagi mengira luas dan perimeter sebuah padang bola
# Pengisytiharan pemboleh ubah dan pemalar
# Input
panjang = float(input("\nMasukkan panjang: "))
lebar = float(input("Masukkan lebar: "))
# Proses
luas = panjang * lebar
perimeter = (panjang + lebar) * 2
# Output
print("\n\nLuas padang bola ialah", luas)
print("Perimeter padang bola ialah", perimeter)
Ln: 10 Col:13

```

Rajah 1.18 Contoh dokumen pustaka (*library documentation*)

Rajah 1.19 menunjukkan contoh panduan pengguna yang dapat membantu pengguna memuat turun dan membuka aplikasi Python.

## PANDUAN PENGGUNA

### Memuat turun perisian

1. Taipkan alamat URL berikut: <https://www.python.org/downloads/>.
2. Paparan berikut akan diperhatikan, pilih versi Python 3.6.5 dan klik *Download*.

Looking for a specific release?

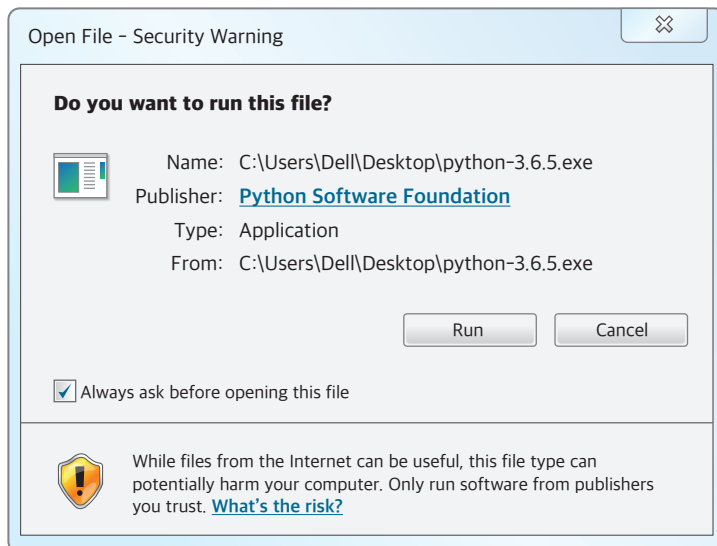
Python releases by version number:

Release version	Release date		Click for more
<a href="#">Python 2.7.15</a>	2018-05-01	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.5</a>	2018-03-28	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.4.8</a>	2018-02-05	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.5</a>	2018-02-05	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.4</a>	2017-12-19	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.3</a>	2017-10-03	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.3.7</a>	2017-09-19	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 2.7.14</a>	2017-09-16	<a href="#">Download</a>	<a href="#">Release Notes</a>

3. Pengguna *Windows* sila klik pada versi *Windows x86 executable installer*. Fail bernama *python-3.6.5.exe* akan mula dimuat turun ke dalam folder *Downloads*.

### Memasang perisian

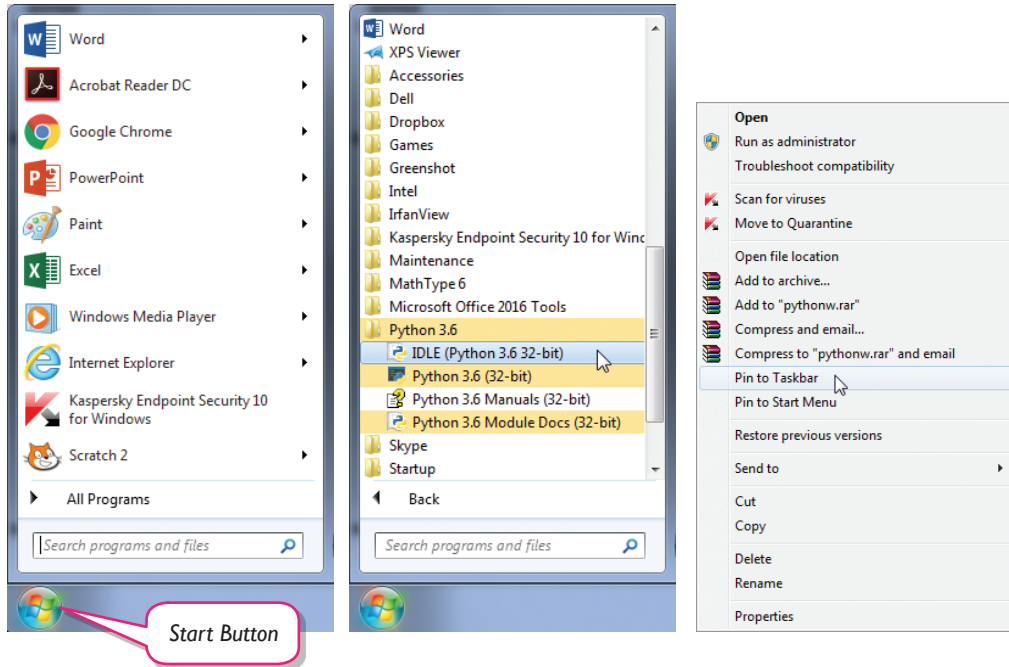
1. Dwiklik pada fail *python-3.6.5.exe* yang siap dimuat turun.
2. Tetingkap *Open File – Security Warning* akan muncul dan klik *Run*. Pilih *Install Now*.



3. Selepas siap memasang perisian, tutup semua tetingkap.

## PANDUAN PENGGUNA

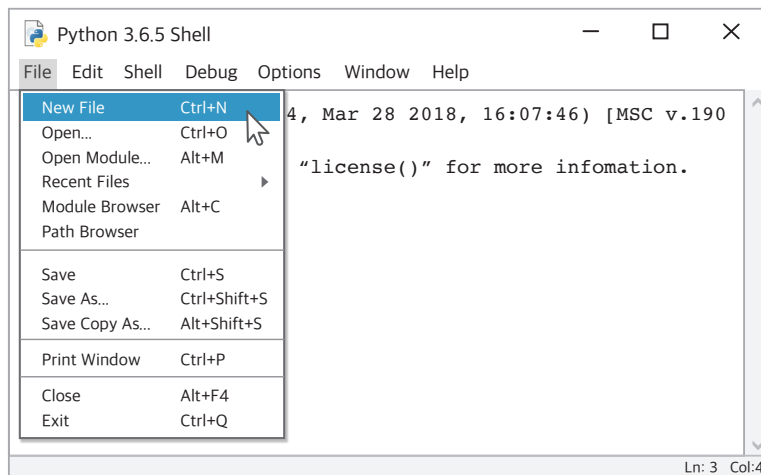
4. Klik *Start Button* >> *All Program* >> *Python 3.6* >> *Python 3.6 32-bit*, klik kanan pada tetikus dan pilih >> *Pin to taskbar*.



5. Ikon Python akan muncul pada *Taskbar*. Klik pada ikon Python.



6. Python 3.6.5 Shell akan dipaparkan. Klik *File* >> *New File*.



Rajah 1.19 Contoh panduan pengguna



### Implementasi Pemikiran Komputasional dalam Pembangunan Atur Cara



Kaedah Pembentangan Hasil Sendiri

Pada setiap bulan, anda akan menerima bil elektrik dengan jumlah bayaran tertentu. Tahukah anda bagaimana elektrik yang telah digunakan ini dikira? Bina satu atur cara bagi mengira bayaran bil elektrik berpandukan maklumat di bawah. Gunakan teknik-teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara untuk menyelesaikan masalah ini.

Penggunaan elektrik	Kadar (RM)
1-200 kWj (sebulan)	0.218
201 kWj ke atas (sebulan)	0.492

Berdasarkan pernyataan masalah di atas, lakukan aktiviti yang berikut:

1. Kaji pernyataan masalah tersebut dan bincangkan penggunaan teknik-teknik pemikiran komputasional dalam setiap fasa pembangunan atur cara untuk mengira bayaran bil elektrik.
2. Tuliskan penyelesaian kepada masalah dalam kertas masing-masing.
3. Setiap kumpulan membentangkan penyelesaian di hadapan kelas.
4. Guru membuat rumusan pelajaran dengan murid.

**STEM**

### Standard Pembelajaran

Murid boleh:

- 1.1.2 Membuat pelaporan penggunaan teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara.



### Pelaporan Penggunaan Teknik Pemikiran Komputasional bagi Setiap Fasa Pembangunan Atur Cara

Pelaporan membawa maksud perihal melaporkan sesuatu perkara secara terurus dan terancang berdasarkan format tertentu. Pelaporan bertujuan untuk memberikan maklumat yang berguna tentang penggunaan teknik pemikiran komputasional dalam menyelesaikan sesuatu masalah supaya penggunaan teknik-teknik tersebut adalah jelas dan lebih ketara. Dengan membuat pelaporan, cara-cara untuk mengurus maklumat secara sistematik dapat dipelajari dan penggunaan teknik pemikiran komputasional dalam menyelesaikan masalah dapat ditunjukkan dengan jelas.

Berikut ialah contoh pelaporan penggunaan teknik pemikiran komputasional bagi setiap fasa semasa pembangunan atur cara. Contoh pelaporan ini adalah berdasarkan pembinaan atur cara untuk mengira luas dan perimeter sebuah padang bola yang anda pelajari dalam Subtopik 1.1.1.

## CONTOH PELAPORAN

## Teknik Leraian



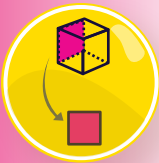
## Fasa 1: Analisis Masalah

- (a) Masalah besar  
Membina atur cara mudah bagi mengira luas dan perimeter sebuah padang bola
- (b) Bahagian kecil/Masalah kecil  
(i) Kira luas padang bola  
(ii) Kira perimeter padang bola

Teknik  
Pengecaman  
Corak

## Fasa 2: Reka Bentuk Atur Cara

	Persamaan	Perbezaan
Pengiraan luas padang bola	Pemboleh ubah yang sama, iaitu panjang dan lebar.	Dapatkan nilai-nilai panjang dan lebar, kemudian kira: Luas = panjang $\times$ lebar
Pengiraan perimeter padang bola		Dapatkan nilai-nilai panjang dan lebar, kemudian kira: Perimeter = (panjang + lebar) $\times$ 2

Teknik  
Peniskalaan

Penentuan aspek-aspek penting terdiri daripada:



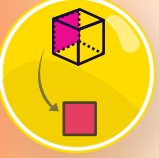

Pengiraan luas padang bola	(a) Nilai-nilai panjang dan lebar. (b) Formula matematik: Luas = panjang $\times$ lebar
Pengiraan perimeter padang bola	(a) Nilai-nilai panjang dan lebar. (b) Formula matematik: Perimeter = (panjang + lebar) $\times$ 2



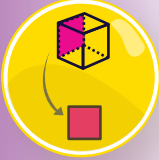

Teknik  
Penggantian

Algoritma dalam bentuk pseudokod

- 1 **Mula**
- 2 **Masukkan** panjang, lebar
- 3 **Kira**
  - 3.1 luas padang bola = panjang  $\times$  lebar
  - 3.2 perimeter padang bola = (panjang + lebar)  $\times$  2
- 4 **Papar** luas padang bola, perimeter padang bola
- 5 **Tamat**

Teknik Leraian	Fasa 3: Pengekodaan									
	(a) Masalah besar Mengekod algoritma, mengira luas dan perimeter sebuah padang bola (b) Bahagian-bahagian kecil: (i) Pengisytiharan pemboleh ubah dan pemalar (ii) Input (iii) Proses (iv) Output									
<b>Teknik Pengecaman Corak</b> 	Corak yang dapat dicamkan bagi setiap bahagian kecil ialah penggunaan jenis data <i>float</i> . <table border="1" data-bbox="404 575 1270 929"> <tr> <td data-bbox="404 575 692 687">(i) Pengisytiharan pemboleh ubah dan pemalar</td> <td data-bbox="692 575 1270 687">Pemboleh ubah untuk menyimpan nilai</td> </tr> <tr> <td data-bbox="404 687 692 799">(ii) Input</td> <td data-bbox="692 687 1270 799">Pegguna memasukkan satu nilai numerik untuk panjang dan satu nilai numerik untuk lebar</td> </tr> <tr> <td data-bbox="404 799 692 880">(iii) Proses</td> <td data-bbox="692 799 1270 880">Kira luas = panjang × lebar Kira perimeter = (panjang + lebar) × 2</td> </tr> <tr> <td data-bbox="404 880 692 929">(iv) Output</td> <td data-bbox="692 880 1270 929">Papar luas dan perimeter padang bola</td> </tr> </table>		(i) Pengisytiharan pemboleh ubah dan pemalar	Pemboleh ubah untuk menyimpan nilai	(ii) Input	Pegguna memasukkan satu nilai numerik untuk panjang dan satu nilai numerik untuk lebar	(iii) Proses	Kira luas = panjang × lebar Kira perimeter = (panjang + lebar) × 2	(iv) Output	Papar luas dan perimeter padang bola
(i) Pengisytiharan pemboleh ubah dan pemalar	Pemboleh ubah untuk menyimpan nilai									
(ii) Input	Pegguna memasukkan satu nilai numerik untuk panjang dan satu nilai numerik untuk lebar									
(iii) Proses	Kira luas = panjang × lebar Kira perimeter = (panjang + lebar) × 2									
(iv) Output	Papar luas dan perimeter padang bola									
<b>Teknik Peniskalaan</b> 	Penentuan aspek-aspek penting terdiri daripada: <table border="1" data-bbox="404 977 1270 1723"> <tr> <td data-bbox="404 977 668 1232">(i) Pengisytiharan pemboleh ubah dan pemalar</td> <td data-bbox="668 977 1270 1232"> <ul style="list-style-type: none"> <li>• Pemboleh ubah untuk menyimpan nilai:               <ul style="list-style-type: none"> <li>– Panjang</li> <li>– Lebar</li> <li>– Luas</li> <li>– Perimeter</li> </ul> </li> <li>• Semua pemboleh ubah yang terlibat diisytihar sebagai <i>float</i>.</li> </ul> </td> </tr> <tr> <td data-bbox="404 1232 668 1406">(ii) Input</td> <td data-bbox="668 1232 1270 1406">           Kod arahan mesej dorongan (<i>prompt message</i>):  <pre>panjang = float(input ("\n Masukkan panjang: ")) lebar = float(input ("Masukkan lebar: "))</pre> </td> </tr> <tr> <td data-bbox="404 1406 668 1518">(iii) Proses</td> <td data-bbox="668 1406 1270 1518">           Kod arahan mengira luas dan perimeter:  <pre>luas = panjang * lebar perimeter = (panjang + lebar) * 2</pre> </td> </tr> <tr> <td data-bbox="404 1518 668 1723">(iv) Output</td> <td data-bbox="668 1518 1270 1723">           Kod arahan memaparkan nilai luas dan perimeter:  <pre>print ("\n\nLuas padang bola ialah", luas) print ("Perimeter padang bola ialah", perimeter)</pre> </td> </tr> </table>		(i) Pengisytiharan pemboleh ubah dan pemalar	<ul style="list-style-type: none"> <li>• Pemboleh ubah untuk menyimpan nilai:               <ul style="list-style-type: none"> <li>– Panjang</li> <li>– Lebar</li> <li>– Luas</li> <li>– Perimeter</li> </ul> </li> <li>• Semua pemboleh ubah yang terlibat diisytihar sebagai <i>float</i>.</li> </ul>	(ii) Input	Kod arahan mesej dorongan ( <i>prompt message</i> ): <pre>panjang = float(input ("\n Masukkan panjang: ")) lebar = float(input ("Masukkan lebar: "))</pre>	(iii) Proses	Kod arahan mengira luas dan perimeter: <pre>luas = panjang * lebar perimeter = (panjang + lebar) * 2</pre>	(iv) Output	Kod arahan memaparkan nilai luas dan perimeter: <pre>print ("\n\nLuas padang bola ialah", luas) print ("Perimeter padang bola ialah", perimeter)</pre>
(i) Pengisytiharan pemboleh ubah dan pemalar	<ul style="list-style-type: none"> <li>• Pemboleh ubah untuk menyimpan nilai:               <ul style="list-style-type: none"> <li>– Panjang</li> <li>– Lebar</li> <li>– Luas</li> <li>– Perimeter</li> </ul> </li> <li>• Semua pemboleh ubah yang terlibat diisytihar sebagai <i>float</i>.</li> </ul>									
(ii) Input	Kod arahan mesej dorongan ( <i>prompt message</i> ): <pre>panjang = float(input ("\n Masukkan panjang: ")) lebar = float(input ("Masukkan lebar: "))</pre>									
(iii) Proses	Kod arahan mengira luas dan perimeter: <pre>luas = panjang * lebar perimeter = (panjang + lebar) * 2</pre>									
(iv) Output	Kod arahan memaparkan nilai luas dan perimeter: <pre>print ("\n\nLuas padang bola ialah", luas) print ("Perimeter padang bola ialah", perimeter)</pre>									

Teknik Leraian	Fasa 4: Pengujian dan Penyahpepijatan									
	<p>(a) Masalah besar Mengenal pasti dan memperbaiki ralat pada keseluruhan atur cara</p> <p>(b) Bahagian-bahagian kecil</p> <ul style="list-style-type: none"> <li>(i) Menguji pengisytiharan pemboleh ubah dan pemalar</li> <li>(ii) Menguji kod input</li> <li>(iii) Menguji kod proses</li> <li>(iv) Menguji kod output</li> </ul>									
Teknik Pengecaman Corak 	<p>(i) Menguji pengisytiharan pemboleh ubah dan pemalar</p>	Berdasarkan mesej ralat dan cara atur cara bertindak semasa pengujian, jenis-jenis ralat dapat dikenal pasti, iaitu: <ul style="list-style-type: none"> <li>• Ralat sintaks</li> <li>• Ralat masa larian</li> <li>• Ralat logik</li> </ul>								
<p>(ii) Menguji kod input</p>	<p>(iii) Menguji kod proses</p>									
<p>(iii) Menguji kod proses</p>	<p>(iv) Menguji kod output</p>									
<p>(iv) Menguji kod output</p>										
Teknik Peniskalaan 	Penentuan aspek-aspek penting terdiri daripada: <table border="1" data-bbox="422 1093 1288 1468"> <tbody> <tr> <td data-bbox="422 1093 732 1222"> <p>(i) Menguji pengisytiharan pemboleh ubah dan pemalar</p> </td> <td data-bbox="732 1093 1288 1222">           Semua pemboleh ubah diisytiharkan dengan betul dari segi:           <ul style="list-style-type: none"> <li>• Jenis data</li> <li>• Kesesuaian nama dan nama dieja dengan betul</li> <li>• Diberikan nilai permulaan (<i>initialised</i>)</li> </ul> </td> </tr> <tr> <td data-bbox="422 1222 732 1315"> <p>(ii) Menguji kod arahan input</p> </td> <td data-bbox="732 1222 1288 1315">           Pastikan pemboleh ubah panjang dan lebar digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna         </td> </tr> <tr> <td data-bbox="422 1315 732 1408"> <p>(iii) Menguji kod arahan proses</p> </td> <td data-bbox="732 1315 1288 1408"> <ul style="list-style-type: none"> <li>• Formula yang digunakan adalah betul</li> <li>• Logik/susunan adalah betul</li> </ul> </td> </tr> <tr> <td data-bbox="422 1408 732 1468"> <p>(iv) Menguji kod arahan output</p> </td> <td data-bbox="732 1408 1288 1468"> <ul style="list-style-type: none"> <li>• Output yang dipaparkan adalah tepat dan betul</li> <li>• Format output adalah seperti yang dikehendaki</li> </ul> </td> </tr> </tbody> </table>		<p>(i) Menguji pengisytiharan pemboleh ubah dan pemalar</p>	Semua pemboleh ubah diisytiharkan dengan betul dari segi: <ul style="list-style-type: none"> <li>• Jenis data</li> <li>• Kesesuaian nama dan nama dieja dengan betul</li> <li>• Diberikan nilai permulaan (<i>initialised</i>)</li> </ul>	<p>(ii) Menguji kod arahan input</p>	Pastikan pemboleh ubah panjang dan lebar digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna	<p>(iii) Menguji kod arahan proses</p>	<ul style="list-style-type: none"> <li>• Formula yang digunakan adalah betul</li> <li>• Logik/susunan adalah betul</li> </ul>	<p>(iv) Menguji kod arahan output</p>	<ul style="list-style-type: none"> <li>• Output yang dipaparkan adalah tepat dan betul</li> <li>• Format output adalah seperti yang dikehendaki</li> </ul>
<p>(i) Menguji pengisytiharan pemboleh ubah dan pemalar</p>	Semua pemboleh ubah diisytiharkan dengan betul dari segi: <ul style="list-style-type: none"> <li>• Jenis data</li> <li>• Kesesuaian nama dan nama dieja dengan betul</li> <li>• Diberikan nilai permulaan (<i>initialised</i>)</li> </ul>									
<p>(ii) Menguji kod arahan input</p>	Pastikan pemboleh ubah panjang dan lebar digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna									
<p>(iii) Menguji kod arahan proses</p>	<ul style="list-style-type: none"> <li>• Formula yang digunakan adalah betul</li> <li>• Logik/susunan adalah betul</li> </ul>									
<p>(iv) Menguji kod arahan output</p>	<ul style="list-style-type: none"> <li>• Output yang dipaparkan adalah tepat dan betul</li> <li>• Format output adalah seperti yang dikehendaki</li> </ul>									
Teknik Pengitlakan 	Menghasilkan satu senarai semak yang dapat digunakan semasa pengujian dan penyahpepijatan sesuatu atur cara.									

Teknik Leraian	Fasa 5: Dokumentasi	
	(a) Masalah besar	Menyediakan dokumentasi
	(b) Bahagian-bahagian kecil	(i) Dokumentasi dalaman
		(ii) Dokumentasi luaran <ul style="list-style-type: none"> <li>• Dokumen pustaka</li> <li>• Panduan pengguna</li> </ul>
	(a) Dokumentasi dalaman	Maklumat penting tentang atur cara merupakan corak yang telah dicamkan. Komen tentang tujuan dan fungsi dilakukan oleh setiap bahagian/barisan kod arahan.
	(b) Dokumentasi luaran (Dokumen Pustaka)	Maklumat teknikal tentang atur cara untuk tujuan rujukan, penyelenggaraan dan penambahbaikan.
	(c) Dokumentasi luaran (Panduan Pengguna)	Maklumat tentang cara menggunakan atau melaksanakan arahan
	Penentuan aspek-aspek penting terdiri daripada:	
	(a) Dokumentasi dalaman	Penulisan komen untuk menerangkan fungsi setiap bahagian atau sesuatu baris kod arahan
	(b) Dokumentasi luaran (Dokumen Pustaka)	Maklumat yang diperlukan oleh seorang pengatur cara: <ul style="list-style-type: none"> <li>• Tarikh</li> <li>• Nama pengatur cara</li> <li>• Tujuan/kegunaan atur cara</li> <li>• Keperluan sistem pengendalian</li> <li>• Keperluan perkakasan</li> <li>• Keperluan perisian</li> <li>• Algoritma (pseudokod dan carta alir)</li> <li>• Kod arahan (<i>source code</i>)</li> </ul>
(c) Dokumentasi luaran (Panduan Pengguna)	Maklumat yang diperlukan oleh seorang pengguna seperti: <ul style="list-style-type: none"> <li>• Nama dan tujuan atur cara</li> <li>• Keperluan sistem pengendalian, perkakasan dan perisian</li> <li>• Cara penggunaan</li> </ul>	
	Teknik Pengitlakan Menghasilkan dokumentasi dalaman dan dokumentasi luaran berdasarkan aspek-aspek penting yang dikenal pasti.	





## Aktiviti 1.2



## Aktiviti Berpasangan


Bina Pelaporan Penggunaan Teknik Pemikiran Komputasional bagi Setiap Fasa Pembangunan Atur Cara



Kaedah  
Think-Pair-Share

Anda telah membina satu atur cara untuk mengira bayaran bil elektrik pada Aktiviti 1.1. Kini, anda dikehendaki menghasilkan pelaporan penggunaan teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara.

Berdasarkan pernyataan di atas, anda dikehendaki menjalankan aktiviti yang berikut:

1. Kenal pasti teknik-teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara anda.
2. Kongsikan hasil dapatan anda bersama-sama pasangan anda dan hasilkan satu pelaporan penggunaan teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara. 
3. Bentangkan hasil pelaporan kumpulan anda dengan bantuan perisian persembahan.

1.1.3

### Menghasilkan Satu Projek Mini Berdasarkan Situasi Penyelesaian Masalah dengan Merujuk Fasa Pembangunan Atur Cara

Anda telah mempelajari cara menggunakan konsep pemikiran komputasional dalam pembangunan atur cara aritmetik dan membuat satu pelaporan tentang penggunaan teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara. Kebiasaannya, projek pembangunan atur cara dilakukan secara berkumpulan. Mari lakukan Aktiviti 1.3 untuk menghasilkan satu projek mini secara berkumpulan.

### Standard Pembelajaran

Murid boleh:

- 1.1.3** Menghasilkan satu projek mini secara berkumpulan berdasarkan situasi dalam penyelesaian masalah berdasarkan fasa pembangunan atur cara.

## Tahukah Anda ?



Pada Oktober 2017, Kementerian Pendidikan Malaysia dan *Malaysian Digital Economy Corporation* (MDEC) telah mengadakan kempen *Hour of Code*. Kempen ini bertujuan untuk mendedahkan dan meningkatkan kesedaran murid terhadap pengekodan (*coding*), pengaturcaraan dan sains komputer. Oleh itu, murid mampu meningkatkan kemahiran penyelesaian masalah selain mendedahkan murid untuk menghadapi pasaran kerja baharu berasaskan ekonomi digital.



## Aktiviti 1.3



### Aktiviti Kumpulan


#### Penghasilan Satu Atur Cara



Kaedah  
Gallery Walk

1. Kaji pernyataan masalah berikut.

Anda ingin memasang sebuah tangki air yang baharu bagi menampung keperluan penggunaan air yang semakin meningkat. Oleh hal yang demikian, anda diminta untuk membina satu atur cara bagi mengira jumlah luas permukaan dan isi padu sebuah tangki air berbentuk silinder.

2. Bincangkan bersama-sama rakan anda tentang penggunaan teknik-teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara.
3. Hasilkan satu pelaporan yang lengkap menggabungkan teknik pemikiran komputasional dalam pembangunan atur cara secara sistematik. 
4. Tampilkan pelaporan kumpulan anda pada dinding kelas.
5. Murid-murid diminta membaca dan menulis komen tentang pelaporan kumpulan lain. Komen ini perlu ditampal berhampiran dengan pelaporan tersebut.



## Praktis Amali 1.1

#### Membina Atur Cara yang Dibincangkan dalam Aktiviti 1.3

1. Lakukan praktis amali ini dalam kumpulan.
2. Gunakan perisian aplikasi Python dan lancarkan satu fail baharu.
3. Taipkan segmen kod yang dihasilkan dalam Aktiviti 1.3. Laksanakan atur cara anda dan baiki ralat jika ada.
4. Gunakan perisian pemprosesan perkataan yang sesuai untuk menghasilkan setiap laporan yang berikut:
  - (a) Dokumen Pustaka (*Library Documentation*)
  - (b) Panduan Pengguna
  - (c) Pelaporan penggunaan teknik pemikiran komputasional dalam pembangunan atur cara
5. Cetak semua laporan yang dihasilkan.
6. Di dalam kelas, susun semua laporan di atas meja.
7. Semua murid diminta bergerak untuk melihat laporan-laporan kumpulan yang lain.

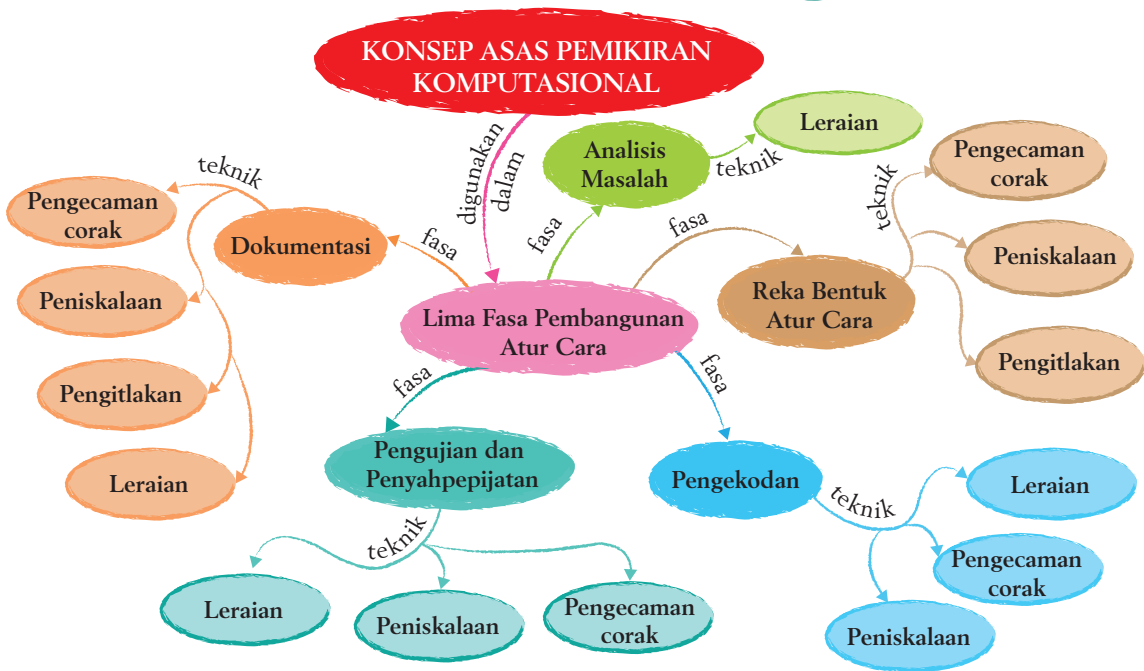
## Uji Kendiri 1.1

1. Lengkapkan rajah di bawah dengan fasa-fasa pembangunan atur cara yang betul.



2. Dalam fasa , pengatur cara akan menemu bual, membuat soal selidik dan pemerhatian bagi mengetahui keperluan pelanggan.
3. Pada fasa reka bentuk atur cara, pengatur cara akan menulis  dan melakar .
4. Semasa melaksanakan fasa , pseudokod akan ditukarkan kepada kod arahan yang difahami oleh komputer.
5. Nyatakan jenis-jenis ralat yang wujud semasa pelaksanaan atur cara.
6. Nyatakan jenis-jenis dokumentasi yang wujud dalam fasa dokumentasi.
7. Nyatakan teknik-teknik pemikiran komputasional yang boleh digunakan dalam pembangunan atur cara.
8. Teknik  digunakan untuk memecahkan masalah yang besar kepada bahagian-bahagian yang lebih kecil.
9. Teknik  digunakan dengan memberikan penekanan kepada aspek-aspek penting dan meninggalkan aspek-aspek kurang penting.
10. Teknik  digunakan dengan meneliti ciri-ciri persamaan corak terhadap sesuatu masalah.
11. Terangkan maksud serta berikan contoh yang sesuai bagi menerangkan setiap ralat yang berikut:
  - (a) Ralat sintaks
  - (b) Ralat logik
  - (c) Ralat masa larian
12. Nyatakan **dua** cara yang digunakan dalam Python untuk membezakan pernyataan komen dengan barisan kod arahan.
13. Jelaskan kepentingan *library documentation* dalam pembangunan atur cara.
14. Apakah kepentingan dokumentasi dalaman?

# RUMUSAN GRAFIK 1



## SEMAK SEMULA 1

Pada akhir bab ini, saya dapat...

menggunakan konsep pemikiran komputasional dalam fasa analisis masalah bagi membangunkan atur cara aritmetik.

menggunakan konsep pemikiran komputasional dalam fasa reka bentuk atur cara bagi membangunkan atur cara aritmetik.

menggunakan konsep pemikiran komputasional dalam fasa pengekoden bagi membangunkan atur cara aritmetik.

menggunakan konsep pemikiran komputasional dalam fasa pengujian dan penyahpejitan bagi membangunkan atur cara aritmetik.

menggunakan konsep pemikiran komputasional dalam fasa dokumentasi bagi membangunkan atur cara aritmetik.

membuat pelaporan menggunakan teknik pemikiran komputasional bagi setiap fasa pembangunan atur cara.

menghasilkan satu projek mini secara berkumpulan berdasarkan situasi dalam penyelesaian masalah berdasarkan fasa pembangunan atur cara.

# PENTAKSIRAN 1

## Soalan Objektif

1. Fasa pembangunan atur cara terdiri daripada \_\_\_\_\_.
  - A analisis masalah, reka bentuk atur cara, pengujian dan penyahpejijatan, pengekodan, dokumentasi
  - B analisis masalah, reka bentuk atur cara, pengekodan, pengujian dan penyahpejijatan, dokumentasi
  - C analisis masalah, reka bentuk atur cara, pengujian dan penyahpejijatan, dokumentasi, pengekodan
  - D analisis masalah, reka bentuk atur cara, pengekodan, dokumentasi, pengujian dan penyahpejijatan
2. Dalam fasa analisis masalah, pengatur cara akan mengenal pasti input, proses dan \_\_\_\_\_.
  - A output
  - B storan
  - C perisian
  - D perkakasan
3. Dalam fasa reka bentuk atur cara, pengatur cara akan \_\_\_\_\_, melukis carta alir dan mereka bentuk antara muka pengguna.
  - A melukis papan cerita
  - B menulis kod arahan
  - C melakar reka bentuk
  - D menulis pseudokod
4. Pernyataan yang manakah menerangkan dokumentasi dalaman dengan betul?
  - A Dokumentasi dalaman bertujuan membantu pengguna menggunakan aplikasi yang telah dibina.
  - B Dokumentasi dalaman disediakan untuk membantu pengatur cara memahami aplikasi yang telah dibina.
  - C Dokumentasi dalaman bertindak sebagai manual pengguna.
  - D Dokumentasi dalaman mengandungi proses pemasangan perisian secara langkah demi langkah.
5. Seorang pengatur cara telah mengalami masalah apabila paparan output yang terhasil adalah salah.
 

Apakah jenis ralat yang boleh dikaitkan dengan situasi di atas?

  - A Ralat sintaks
  - B Ralat masa larian
  - C Ralat logik
  - D Ralat pelaksanaan
6. Teknik \_\_\_\_\_ digunakan apabila masalah yang besar perlu dipecahkan kepada bahagian-bahagian masalah yang lebih kecil.
  - A pengecaman corak
  - B peniskalaan
  - C leraian
  - D pengitlakan
7. Raimi ditugaskan untuk mencari persamaan dan perbezaan antara sebuah basikal dengan sebuah motosikal. Raimi telah menggunakan teknik \_\_\_\_\_.
  - A leraian
  - B pengecaman corak
  - C peniskalaan
  - D pengitlakan
8. Atur cara yang dibina merupakan model yang dapat membantu menyelesaikan masalah yang serupa. Pembinaan model ini merujuk kepada teknik \_\_\_\_\_.
  - A leraian
  - B pengitlakan
  - C peniskalaan
  - D pengecaman corak

9. Arissa ingin membuat pengiraan harga bagi suatu barang selepas diskaun. Bagi menyelesaikan masalah ini, Arissa mengambil kira aspek-aspek penting berkaitan pengiraan diskaun. Apakah teknik pemikiran komputasional yang digunakan oleh Arissa?
- A Leraian  
B Pengecaman corak  
C Pengitlakan  
D Peniskalaan
10. Apakah kelebihan penggunaan teknik-teknik pemikiran komputasional dalam pembangunan atur cara?
- I Menyelesaikan masalah secara sistematik  
II Menghasilkan atur cara dengan cekap  
III Menghasilkan atur cara yang menepati kehendak pengguna
- A I sahaja      C II dan III  
B I dan II      D I, II dan III

### Soalan Berstruktur

1. Kod arahan berikut ialah atur cara bagi mencari jumlah dan purata untuk dua nombor.

```

1 # Atur cara untuk mencari jumlah dan purata bagi dua nombor
2 # Minta pengguna masukkan input
3 nombor1 = float(input("Masukkan nombor pertama: "))
4 nombor2 = float(input("Masukkan nombor kedua: "))
5 jumlah = nombor1 + nombor2
6 purata = jumlah/2
7
8 print("\nJumlah bagi dua nombor ialah", jumlah)
9 print("Purata bagi dua nombor ialah", purata)

```

Berdasarkan kod arahan di atas, jawab soalan-soalan yang berikut:

- (a) Nyatakan semua pemboleh ubah bagi input.  
(b) Kenal pasti pernyataan-pernyataan yang mengandungi ralat.  
(c) Baiki ralat yang anda kesan di (b), kemudian tulis sintaks yang betul.

2. Aimah diminta menulis kod arahan Python bagi atur cara yang melakukan operasi tambah, tolak, darab dan bahagi bagi dua nombor. Penulisan kod mesti mengikut kehendak yang dinyatakan berikut:

- (a) (i) Meminta input daripada pengguna  
(ii) Formula yang digunakan  
(iii) Output yang akan dihasilkan  
(b) Lakarkan paparan output yang akan diperolehi.

### Soalan Penyelesaian Masalah

Anda dikehendaki menyelesaikan masalah untuk mengira luas dan lilitan sebuah bulatan. Anda perlu:

- (a) Membuat pelaporan penggunaan teknik pemikiran komputasional bagi fasa pembangunan atur cara:
- (i) Analisis masalah  
(ii) Reka bentuk atur cara  
(iii) Pengkodan  
(iv) Pengujian dan penyahpejatan  
(v) Dokumentasi
- (b) Membina atur cara mudah

**STEM**

**E-PRAKTIS**



# BAB 2

# PERWAKILAN DATA

Sejak zaman purba hingga zaman moden, kriptografi telah digunakan oleh manusia untuk menjaga privasi mesej yang ingin dihantar. Pelbagai kaedah sifer diwujudkan untuk menjamin keselamatan data dan memastikan mesej hanya dapat dibaca dan difahami oleh penerimanya.

## Standard Kandungan

### 2.1 Kriptografi dalam Keselamatan Data

## Kata Kunci

- Kriptografi
- Penyulitan (*encryption*)
- Nyahsulit (*decryption*)
- Kunci kod rahsia
- Sifer (*cipher*)

Guru Sejarah telah mengajar bahawa salah satu cara menghantar utusan adalah melalui burung merpati. Apakah yang akan terjadi sekiranya burung tersebut ditangkap oleh pihak musuh?

Ya, pasti maklumat dalam utusan akan terbongkar sekiranya jatuh kepada pihak musuh.

Tidak, kerana maklumat dalam utusan ditulis dalam bentuk yang hanya difahami oleh pihak penghantar dan penerima tertentu sahaja. Contohnya di Mesir, tulisan Hieroglif yang berupa simbol digunakan. Ini merupakan contoh penggunaan kriptografi.

Tonton video ini untuk mengetahui tentang gambaran Bab 2.



[goo.gl/kqTFPo](https://goo.gl/kqTFPo)

Imbas  
Saya

# 2.1

## KRIPTOGRAFI DALAM KESELAMATAN DATA

Penerangan kriptografi



[goo.gl/wSyf3N](http://goo.gl/wSyf3N)

Imbas Saya

Dalam era teknologi maklumat dan komunikasi, kebanyakan maklumat dikongsi melalui Internet dalam bentuk digital. Penghantaran suatu dokumen tidak lagi bergantung kepada perkhidmatan pos tetapi lebih kepada penggunaan rangkaian komputer, contohnya melalui e-mel atau media sosial. Pernahkah anda terfikir bagaimana sesuatu mesej atau data yang dihantar secara elektronik diterima oleh penerima tanpa dicuri, diubah atau ditokok tambah oleh pihak lain?

### 2.1.1 Kriptografi dalam Pengkomputeran

#### Standard Pembelajaran

- Murid boleh:
- 2.1.1 Menerangkan maksud dan kepentingan kriptografi dalam pengkomputeran.

Kriptografi (*cryptography*) berasal daripada bahasa Yunani yang terdiri daripada dua suku kata, iaitu “*kriptos*” dan “*graphein*”. “*Kriptos*” bermaksud sembunyi manakala “*graphein*” bermaksud untuk tulis. Oleh itu, kriptografi boleh dihuraikan sebagai kajian tentang teknik kerahsiaan atau dikenali sebagai keselamatan komunikasi data. Tujuan utama kriptografi dicipta adalah untuk melindungi informasi daripada terdedah dan dipintas oleh pihak lain.

Gambar foto 2.1 menunjukkan satu contoh tulisan Hieroglif yang digunakan oleh golongan elit Tamadun Mesir Purba. Tulisan Hieroglif ini ialah satu contoh kriptografi.

#### Tahukah Anda?

Dari zaman dahulu hingga sekarang, kriptografi dianggap sama seperti penyulitan. Sebenarnya, kriptografi bukan hanya sekadar penyulitan tetapi kriptografi merangkumi pelbagai aspek mengenai keselamatan maklumat.



Gambar foto 2.1 Tulisan Hieroglif

Pada zaman moden hari ini, kita sangat bergantung kepada sistem pengkomputeran digital. Situasi ini telah menyebabkan kebanyakan maklumat penting atau sulit individu, masyarakat dan negara disimpan dalam bentuk digital yang boleh diakses melalui laman sesawang. Cara penyimpanan dalam bentuk digital ini telah membuka ruang untuk **penceroobohan maklumat** berlaku.

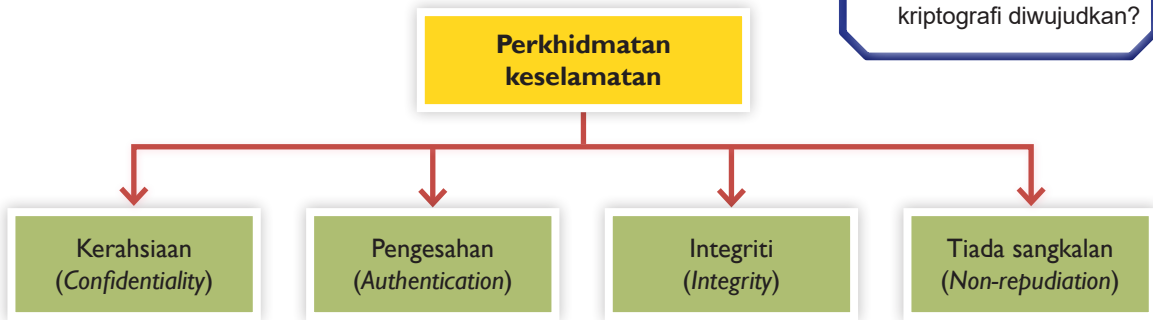


Bagi mengatasi masalah pencerobohan maklumat, kriptografi telah menjadi satu keperluan besar untuk melindungi kepentingan maklumat dalam pengkomputeran. Rajah 2.1 menunjukkan empat kepentingan perkhidmatan keselamatan data kriptografi.



### Uji Minda 2.1

1. Berikan satu contoh aplikasi yang menggunakan kriptografi.
2. Mengapakah kriptografi diwujudkan?



Rajah 2.1 Kepentingan perkhidmatan keselamatan data kriptografi

**Kerahsiaan** merujuk kepada kesulitan sesuatu maklumat yang dihantar melalui rangkaian komputer yang perlu dilindungi supaya maklumat tersebut tidak diketahui oleh pihak lain selain penerima. **Pengesahan** pula berkaitan dengan pengenalpastian pihak-pihak yang terlibat dalam suatu komunikasi. Maklumat atau data yang diterima perlu dikenal pasti agar maklumat dihantar oleh pihak yang disahkan. Contohnya, nama pengguna dan kata laluan bagi perbankan Internet hanya boleh diakses oleh pengguna yang sah sahaja.

**Integriti** ialah perkhidmatan keselamatan yang mengenal pasti sebarang perubahan terhadap maklumat atau data oleh pihak lain. Contohnya, transaksi perbankan dalam talian menggunakan kriptografi bagi mengelakkan pihak lain melakukan perubahan sebarang maklumat dalam transaksi seperti jumlah wang, nombor akaun dan sebagainya. **Tiada sangkalan** digunakan untuk membuktikan bahawa penghantar dan penerima maklumat tidak menafikan bahawa mereka menghantar dan menerima maklumat tersebut.

Oleh itu, penggunaan kriptografi penting untuk menjaga kerahsiaan maklumat serta mengelakkan sebarang penyamaran atau penipuan berlaku.



## Aktiviti 2.1

### Aktiviti Berpasangan

#### Memahami Kepentingan Kriptografi



Kaedah  
Peta i-Think

1. Anda dan rakan perlu menyenaraikan kepentingan kriptografi dalam pengkomputeran.
2. Kumpulkan semua kepentingan yang telah disenaraikan dan bincangkan bersama-sama rakan anda.
3. Catatkan hasil perbincangan anda dan rakan menggunakan peta pemikiran yang sesuai.
4. Lakukan pembentangan di hadapan kelas dan adakan sesi soal jawab bersama-sama dengan murid yang lain.

## Standard Pembelajaran

Murid boleh:

- 2.1.2** Menunjuk cara langkah demi langkah proses sifer (*cipher*) yang melibatkan:
- penyulitan (*encryption*)
  - nyahsulit (*decryption*)

## Tahukah Anda?

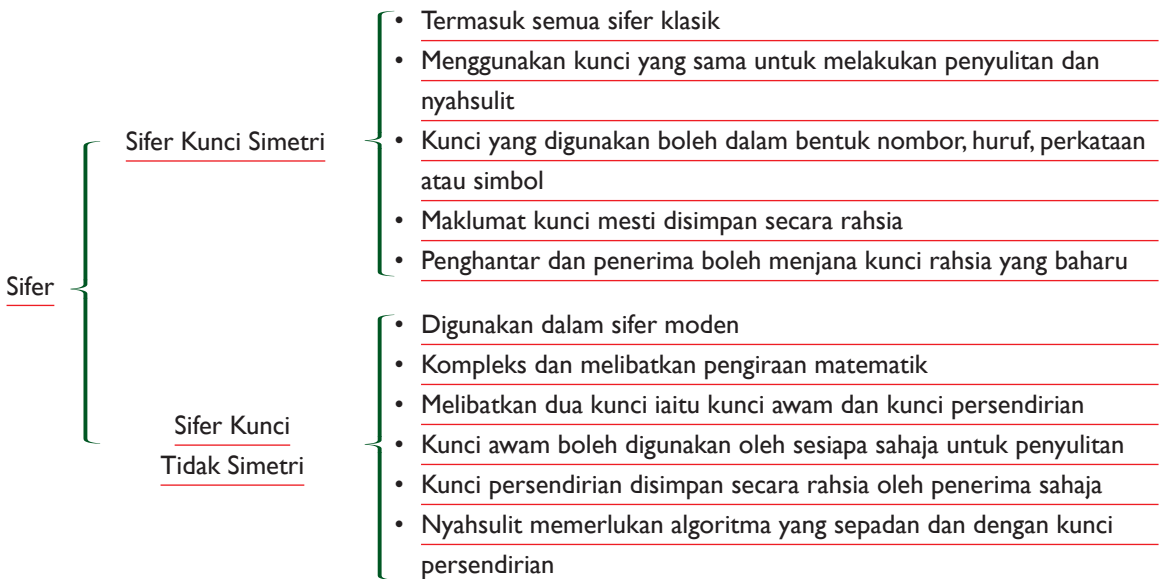
Sifer kunci simetri menggunakan satu kunci yang sama bagi kedua-dua proses penyulitan dan nyahsulit manakala sifer kunci tidak simetri menggunakan dua kunci berbeza bagi proses penyulitan dan nyahsulit.

## 2.1.2 Proses Sifer (*Cipher*)

Anda telah mempelajari bahawa kriptografi ialah penyamaran demi menjaga kerahsiaan dan keselamatan sesuatu data. Satu cara untuk mencapai tujuan ini adalah dengan melakukan penyulitan (*encryption*).

Dalam kriptografi, pasangan algoritma digunakan untuk melakukan penyulitan dan nyahsulit. Pasangan algoritma ini dikenali sebagai algoritma kriptografi atau sifer. Sesuatu **teks biasa** (*plaintext*) akan melalui proses **penyulitan** (*encryption*) yang menyebabkan data berubah menjadi bentuk yang tidak bermakna, iaitu **teks sifer** (*ciphertext*) sebelum dihantar kepada penerima. Hanya penerima yang berhak sahaja dapat melakukan proses **nyahsulit** (*decryption*), iaitu menukarkan kembali teks sifer menjadi teks biasa. Setiap proses penyulitan dan proses nyahsulit dapat dilengkapi dengan penggunaan **kunci** (*key*) yang hanya diketahui oleh penghantar dan penerima. Teks sifer tidak boleh dinyahsulit oleh pihak yang tidak berhak tanpa kunci tersebut.

Algoritma kriptografi atau sifer yang dicipta pada zaman dahulu dikenali sebagai sifer klasik. Pada masa itu, sifer klasik digunakan untuk penyulitan dan nyahsulit teks tulisan. Kini, perkembangan teknologi komputer telah membolehkan ciptaan sifer yang sangat kompleks digunakan untuk menyulitkan sebarang bentuk data. Rajah 2.2 menunjukkan dua jenis sifer.



Rajah 2.2 Dua jenis sifer

**(i) Penyulitan (Encryption)**

Penyulitan ialah satu komponen penting dalam bidang kriptografi. Penyulitan ialah satu proses penukaran teks biasa kepada teks sifer, iaitu teks dalam bentuk yang tidak bermakna apabila dibaca. Penukaran ini dilakukan dengan menggunakan algoritma dan kunci penyulitan. Rajah 2.3 menunjukkan proses penyulitan.



**Rajah 2.3** Proses penyulitan

Langkah-langkah penyulitan adalah seperti berikut.

1. Pengirim menghantar mesej (teks biasa) kepada penerima.
2. Mesej ini akan disulitkan menggunakan satu kunci dan algoritma penyulitan.
3. Teks sifer, iaitu teks yang tidak bermakna akan terhasil.

**(ii) Nyahsulit (Decryption)**

Apabila suatu teks biasa telah melalui proses penyulitan dan menghasilkan teks sifer, proses nyahsulit akan berlaku terhadap teks sifer untuk mendapatkan kembali teks biasa yang asal. Secara umumnya, nyahsulit ialah proses untuk mengembalikan teks sifer menjadi teks biasa yang dapat dibaca dan mempunyai maksud. Seperti proses penyulitan, nyahsulit juga menggunakan suatu algoritma dan kunci nyahsulit. Rajah 2.4 menunjukkan proses nyahsulit.



**Rajah 2.4** Proses nyahsulit

Langkah-langkah nyahsulit adalah seperti berikut.

1. Teks sifer yang telah disulitkan akan dinyahsulit menggunakan satu kunci dan algoritma nyahsulit.
2. Teks sifer akan menjadi teks biasa.
3. Penerima akan menerima teks biasa yang dapat dibaca.



**Uji Minda 2.2**

Adakah proses penyulitan sama dengan proses nyahsulit? Terangkan.

## Standard Pembelajaran

Murid boleh:

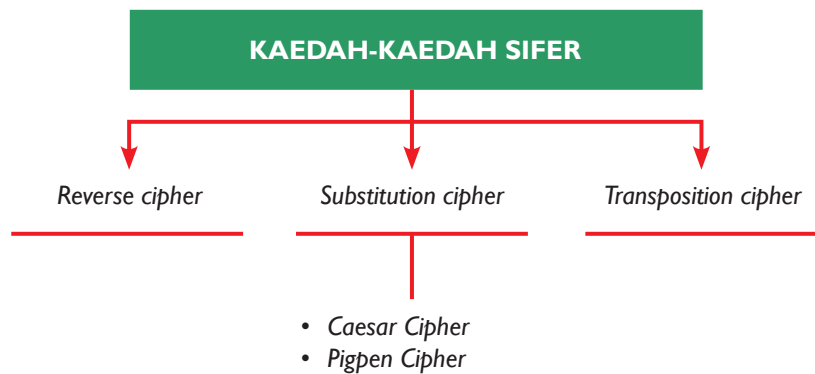
- 2.1.3** Menghasil dan menterjemah mesej menggunakan kaedah sifer seperti berikut:
- Reverse cipher*
  - Substitution cipher*  
(*Caesar Cipher*, *Pigpen Cipher*)
  - Transposition cipher*

2.1.3

## Menghasil dan Menterjemah Mesej Menggunakan Kaedah Sifer

Julius Caesar telah memperkenalkan satu kaedah sifer sekitar 100 tahun Sebelum Masihi (SM). Sifer yang dicipta oleh Julius Caesar ini bertujuan untuk menghantar mesej dan pesanan kepada semua pegawainya dengan menggunakan huruf ketiga daripada huruf-huruf Roman yang asal. Selain itu, kod Morse juga dibangunkan sebagai suatu sifer. Kod Morse menggunakan telegraf untuk menghantar pesanan semasa peperangan. Sebuah mesin yang dikenali sebagai Enigma, iaitu mesin yang menggunakan sistem sifer dan kunci yang sangat kompleks telah digunakan semasa Perang Dunia ke-2. Kesemua contoh ini menggunakan kaedah sifer untuk menghantar mesej kepada penerima tanpa diketahui oleh pihak lain.

Dalam topik ini, kita akan mempelajari cara-cara untuk menghasilkan dan menterjemah mesej menggunakan kaedah sifer. Kaedah-kaedah sifer yang akan dipelajari adalah seperti ditunjukkan dalam Rajah 2.5.



Rajah 2.5 Kaedah-kaedah sifer yang akan dipelajari



### Uji Minda 2.3

Adakah kunci penyulitan atau kunci nyahsulit diperlukan bagi *Reverse cipher*?

### (i) *Reverse cipher*

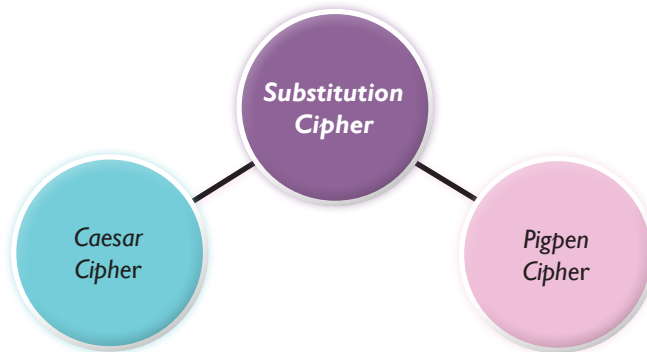
*Reverse cipher* ialah satu kaedah sifer yang paling mudah. Hal ini demikian kerana kaedah ini menggunakan cara songsangan untuk menyulitkan mesej. Terdapat tiga jenis *Reverse cipher*. Jadual 2.1 menunjukkan cara-cara dan penerangan kepada tiga jenis *Reverse cipher*.





(ii) **Substitution cipher**

Dalam *Substitution cipher*, mesej disulitkan dengan menggantikan satu unit teks biasa dengan satu unit teks yang lain (sifer). Satu unit bermaksud satu abjad, pasangan abjad atau kumpulan abjad. Rajah 2.6 menunjukkan dua contoh *Substitution cipher*.

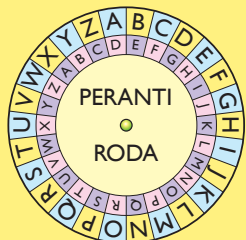


Rajah 2.6 Dua contoh *Substitution cipher*

*Monoalphabetic substitution* ialah jenis *Substitution cipher* yang paling mudah. *Caesar Cipher* dan *Pigpen Cipher* terdiri daripada *monoalphabetic substitution* kerana proses penyulitan dilakukan dengan menggantikan setiap abjad teks biasa dengan abjad yang lain.

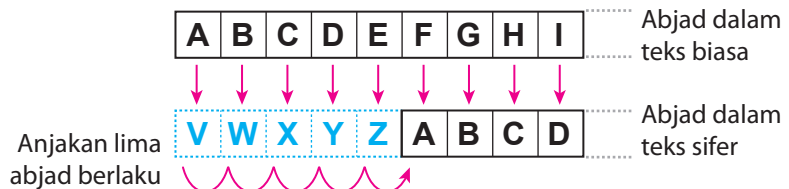
**Tahukah Anda?**

*Caesar Cipher* telah diperkenalkan oleh Julius Caesar. Beliau menggunakan peranti roda (*wheel device*) yang mempunyai 26 abjad, masing-masing di bahagian dalam dan luar roda. Bahagian dalam roda boleh diputar supaya setiap huruf pada bahagian dalam roda dipadankan dengan huruf pada bahagian luar roda.



►► **Caesar Cipher**

*Caesar Cipher* ialah salah satu kaedah sifer yang paling awal digunakan oleh manusia untuk menyulitkan mesej. *Caesar Cipher* juga disebut *shift cipher* (sifer anjakan). Penyulitan teks biasa (*plaintext*) dilakukan dengan menggantikan setiap abjad dalam teks itu dengan abjad yang lain dalam susunan abjad, iaitu dengan menganjukkan abjad dalam susunan abjad. Bilangan anjakan ini dikenali sebagai kunci dan boleh diwakili dengan sebarang simbol. Contohnya,  $K = 5$  bermaksud anjakan sebanyak lima tempat ke kanan dilakukan pada senarai abjad. Rajah 2.7 menunjukkan contoh anjakan  $K = 5$ .



Rajah 2.7 Susunan abjad dengan kunci,  $K = 5$

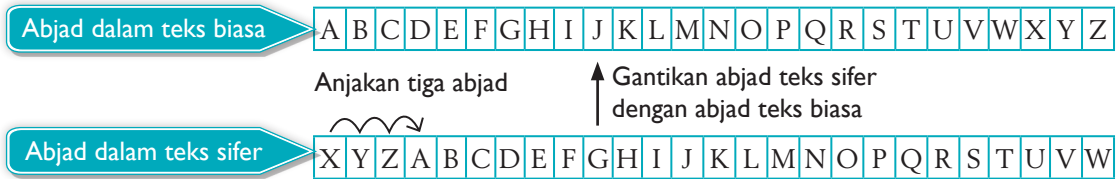
Maka, abjad 'A' dalam teks biasa akan digantikan dengan abjad 'V', abjad 'B' akan digantikan dengan abjad 'W' dan seterusnya. Pengirim mesej akan menentukan nilai  $K$  yang akan dijadikan sebagai kunci penyulitan.

**Contoh 2.1** Nyahsilit menggunakan *Caesar Cipher*.

Amir menerima satu mesej berikut. Dia diberitahu bahawa kaedah sifer yang digunakan ialah *Caesar Cipher* dan kunci ialah 3.

GRJMX AF HXKQFK

Kunci,  $K = 3$  bermaksud anjakan berlaku sebanyak tiga abjad ke kanan. Berikut menunjukkan susunan abjad dalam teks biasa dan susunan abjad dalam teks sifer selepas anjakan.



Tulis semula abjad-abjad dalam mesej sifer. Gunakan jadual yang dibina di atas, gantikan abjad-abjad teks sifer dengan abjad-abjad teks biasa. Mesej sulit akan dinyahsilit seperti berikut.

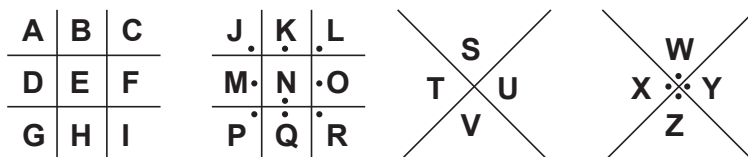
G	R	J	M	X		A	F		H	X	K	Q	F	K
↓	↓	↓	↓	↓		↓	↓		↓	↓	↓	↓	↓	↓
J	U	M	P	A		D	I		K	A	N	T	I	N

Oleh itu, mesej yang dihantar ialah JUMPA DI KANTIN.

➤➤ *Pigpen Cipher*

*Pigpen Cipher* dikategorikan sebagai *Substitution cipher* geometri yang mudah. Setiap abjad digantikan dengan simbol grafik yang sepadan seperti yang ditunjukkan dalam Rajah 2.8. *Pigpen Cipher* adalah mudah digunakan kerana penyulitan dan nyahsilit dapat dilakukan dengan mudah jika simbol-simbol grafik diketahui.

Garis sempadan yang menghasilkan simbol grafik unik dalam *Pigpen Cipher* ialah kunci bagi sifer ini. Dengan mengetahui kunci ini, proses penyulitan dan nyahsilit dapat dilakukan dengan mudah.



**Rajah 2.8** Garis sempadan yang berbeza membentuk simbol grafik yang berbeza

**Uji Minda 2.4**  
Nyatakan perbezaan antara *Caesar Cipher* dan *Pigpen Cipher*.

Jadual 2.2 menunjukkan abjad dan perwakilan simbol *pigpen* yang berpadanan.

**Jadual 2.2** Abjad dan perwakilan simbol *pigpen* yang berpadanan

A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	

Berdasarkan Jadual 2.2, mesej SELAMAT PAGI dapat disulitkan menggunakan *Pigpen Cipher* dengan mudah seperti yang ditunjukkan dalam Jadual 2.3.

**Jadual 2.3** Penyulitan menggunakan *Pigpen Cipher*

Abjad	S	E	L	A	M	A	T
Simbol							
Abjad	P	A	G	I			
Simbol							

## Tahukah Anda

*Pigpen Cipher* mempunyai variasi berlainan yang melibatkan penggunaan nombor 0 hingga 9. Variasi *Pigpen Cipher* ini dicipta oleh Thomas Hunter II. Berikut menunjukkan grid variasi *Pigpen Cipher* ini.

(Sumber: <https://thomashunter.name/blog/extended-pigpen-cipher-to-include-numbers/>)

A	B	C
D	E	F
G	H	I

	J	
1		2
K	3	L
4		5
	M	

N	O	P
Q	R	S
T	U	V

	W	
6		7
X	8	Y
9		0
	Z	



## Contoh 2.2 Nyahsulisit menggunakan Pigpen Cipher.

Anda menerima teks sifer yang berikut. Nyahsulisit mesej ini.

Teks sifer:



### Uji Minda 2.5

Adakah kunci penyulitan atau kunci nyahsulisit digunakan dalam Pigpen Cipher?

Rujuk Jadual 2.2, teks dapat dinyahsulisit dengan mudah.

Teks biasa (selepas nyahsulisit):

**SERONOK BELAJAR**

### (iii) Transposition cipher

*Transposition cipher* ialah suatu kaedah penyulitan yang berbeza dengan *Substitution cipher*. *Substitution cipher* menggantikan abjad dalam teks biasa dengan abjad lain atau simbol. Manakala *Transposition cipher* mengubah kedudukan abjad dalam teks biasa. Abjad-abjad yang terkandung dalam teks biasa adalah tetap. Ini bermaksud, sifer ini melakukan perubahan terhadap kedudukan abjad-abjad dalam teks sahaja. Perubahan yang berlaku ini juga dikenali sebagai pemutaran atau pencampuran (*scrambling*). Penyulitan dan nyahsulisit menggunakan *Transposition cipher* ini memerlukan penggunaan kunci. Anda akan mempelajari dua jenis *Transposition cipher*, iaitu *Columnar Transposition* dan *Rail Fence Cipher*.

#### ►► Columnar Transposition

*Columnar Transposition* ialah satu contoh sifer transposisi yang menyulitkan satu mesej dengan menulisnya dalam jadual secara mendatar (*horizontal*) dengan bilangan lajur (*column*) yang telah ditetapkan. Satu perkataan dipilih untuk dijadikan kunci penyulitan. Bilangan abjad dalam kunci akan menentukan bilangan lajur dalam jadual manakala abjad-abjad yang ada dalam kunci akan menentukan cara bagi mengubah urutan abjad-abjad dalam mesej. Setiap abjad dalam mesej akan ditulis pada setiap petak dalam jadual.

Transposition cipher



[goo.gl/a66kQ2](https://goo.gl/a66kQ2)



Columnar  
Transposition



[goo.gl/JBmMF5](https://goo.gl/JBmMF5)



## Contoh 2.3 Penyulitan menggunakan *Columnar Transposition*.

Langkah-langkah untuk menyulitkan teks biasa:  
MAKLUMAT MESYUARAT AKADEMIK.

### Langkah 1

Kenal pasti kunci yang ingin digunakan. Katakan perkataan BOLA digunakan sebagai kunci. Bina satu jadual seperti di bawah. Kemudian, isikan baris yang seterusnya dengan nombor berdasarkan susunan abjad, iaitu abjad A = 1, B = 2, L = 3 dan O = 4.

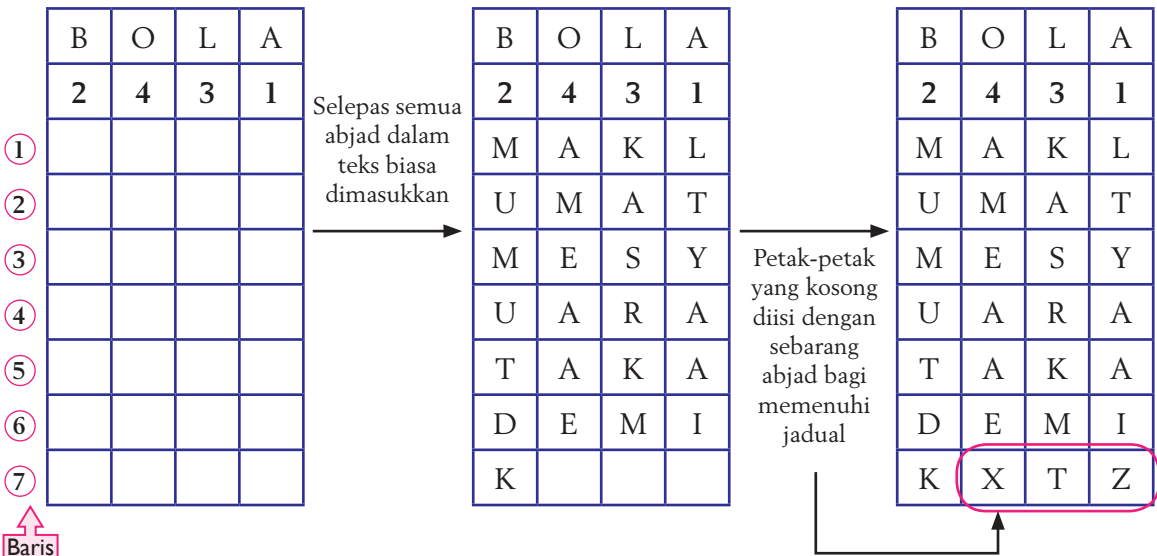
B	O	L	A
2	4	3	1

### Langkah 2

Gunakan formula yang berikut untuk menghitung bilangan baris yang diperlukan dalam jadual.

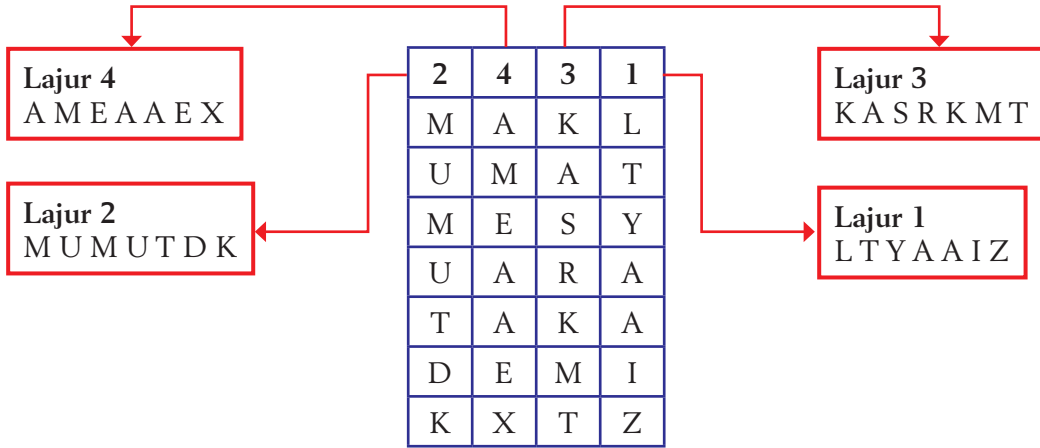
$$\begin{aligned}\text{Bilangan baris} &= \frac{\text{Bilangan abjad dalam teks biasa}}{\text{Bilangan abjad dalam kunci}} \\ &= \frac{25}{4} \\ &= 6.25 \\ &= 7 \text{ baris}\end{aligned}$$

Kembangkan jadual dalam langkah 1 seperti jadual berikut. Kemudian, isikan semua abjad dalam teks biasa ke dalam petak-petak kosong dari kiri ke kanan baris demi baris.



**Langkah 3**

Teks sifer akan ditulis berdasarkan jadual yang dihasilkan dalam langkah 2. Mulai dari lajur 1, abjad ditulis dari atas ke bawah. Ini diikuti dengan lajur-lajur yang seterusnya.



Teks sifer dihasilkan dengan mencantumkan semua abjad dari lajur 1, 2, 3 dan 4.

**Teks sifer anda ialah: L T Y A A I Z M U M U T D K K A S R K M T A M E A A E X**

**Contoh 2.4** Nyahsulisit menggunakan *Columnar Transposition*.

Ramli menerima teks sifer dan kunci daripada pelatih bola sepaknya seperti yang ditunjukkan di bawah:

Teks sifer: E I E A L A N P K I A K S A D M O P A E P N S X R N E B E X

Kunci: PINTU

**Langkah 1**

Bina satu jadual dan isikan abjad-abjad kunci seperti di bawah. Kemudian, isikan baris yang seterusnya dengan nombor berdasarkan susunan abjad, iaitu abjad I = 1 dan seterusnya.

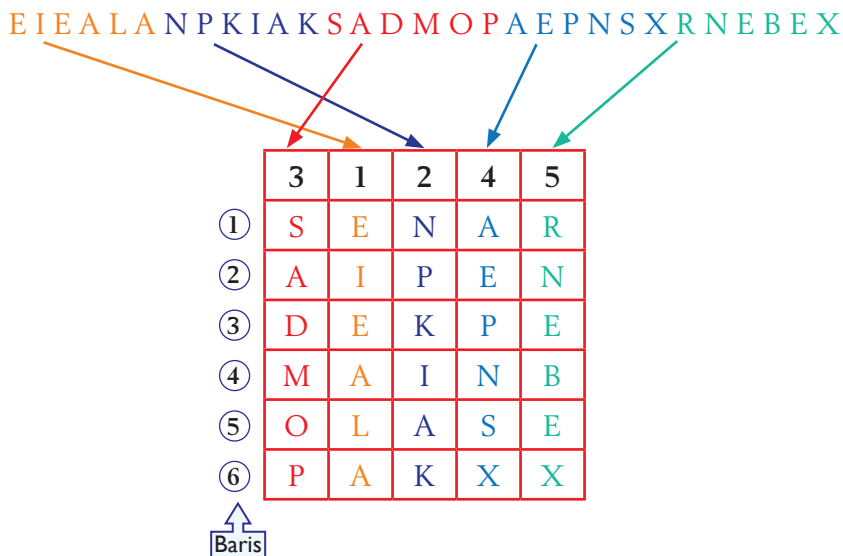
P	I	N	T	U
3	1	2	4	5

**Langkah 2**

Bilangan baris yang diperlukan dapat diperolehi daripada formula yang berikut:

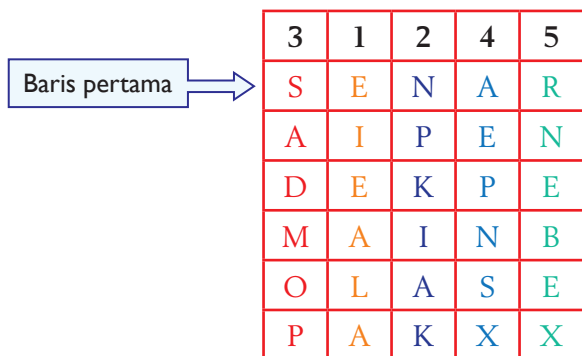
$$\begin{aligned}
 \text{Bilangan baris} &= \frac{\text{Bilangan abjad dalam teks sifer}}{\text{Bilangan abjad dalam kunci}} \\
 &= \frac{30}{5} \\
 &= 6 \text{ baris}
 \end{aligned}$$

Kembangkan jadual dalam langkah 2 dengan tambahan enam baris. Kemudian, isikan abjad-abjad teks sifer ke dalam jadual di bawah secara menegak (*vertical*) berdasarkan kepada urutan nombor.



### Langkah 3

Berdasarkan jadual dari langkah 2, anda dapat menuliskan teks biasa dari kiri ke kanan secara mendatar (*horizontal*) bermula dengan baris pertama.



SENARAI PENDEK PEMAIN BOLA SEPAK X X

Abjad yang digunakan untuk mengisi petak yang kosong

Teks biasa selepas dinyahsulit: SENARAI PENDEK PEMAIN BOLA SEPAK

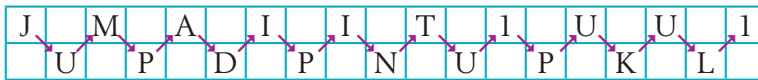
Perkara penting semasa nyahsulis teks sifer kepada teks biasa menggunakan *Columnar Transposition* ialah mengetahui bilangan lajur yang digunakan oleh penghantar mesej. Penerima mesej perlu mengira bilangan abjad dalam perkataan kunci untuk mendapatkan bilangan lajur pada jadual. Kemudian, abjad-abjad dalam teks sifer diisi semula ke dalam jadual secara menegak dan teks biasa akan diperolehi dari kiri ke kanan secara mendatar.

➤ ➤ **Rail Fence Cipher**

*Rail Fence Cipher* juga disebut sebagai *zig-zag cipher*. Cara penyulitan adalah seperti namanya, iaitu abjad-abjad dalam teks biasa ditulis semula mengikut susunan baris demi baris dan mengikut corak *zig-zag* seperti pagar. *Rail Fence Cipher* perlu dinyatakan kunci untuk penyulitan dan nyahsulis berlaku. Nilai kunci merujuk kepada bilangan baris yang digunakan. Rajah 2.9 menunjukkan penyulitan teks biasa JUMPA DI PINTU 1 PUKUL 1 menggunakan *Rail Fence Cipher* dengan kunci = 2.



Kunci = 2 bermaksud dua baris digunakan. Isi abjad-abjad teks biasa dalam corak *zig-zag* baris demi baris.



Teks sifer ditulis secara melintang mulai daripada baris pertama, iaitu **JMAIIT1UUIUPDPNUPKL**

**Rajah 2.9** Penyulitan teks biasa menggunakan *Rail Fence Cipher*

**Tahukah Anda?**

Ruang kosong diabaikan semasa penyulitan dan nyahsulis menggunakan *Rail Fence Cipher*.

**Contoh 2.5** Nyahsulis menggunakan *Rail Fence Cipher*.

Rizal telah menerima teks sifer KMDENAIUICMRAGMREL berserta kunci = 3 daripada seorang rakan. Kaedah yang digunakan oleh rakannya ialah *Rail Fence Cipher*. Bantu Rizal nyahsulis mesej ini.

**Langkah 1**

Kunci yang diterima mewakili baris bagi jadual, iaitu 3. Bagi menentukan bilangan lajur, hitungkan jumlah abjad dalam teks sifer, iaitu 18. Ini bermaksud 18 lajur diperlukan. Bina satu jadual yang mempunyai tiga baris dan 18 lajur, seperti yang ditunjukkan di bawah.


## Langkah 2

Tandakan petak menggunakan simbol “-” supaya menjadi corak *zig-zag*. Petak yang bertanda akan digunakan untuk mengisi abjad-abjad dalam teks sifer yang diberikan. Langkah ini dapat mengelakkan kesilapan mengisi abjad teks sifer.

-				-				-				-				-	
	-		-		-		-		-		-		-		-		-
		-				-				-				-			

## Langkah 3

Pindahkan teks sifer KMDENAIUICMRAGMREL ke dalam jadual. Bermula dengan baris pertama, abjad diisi satu demi satu seperti yang ditunjukkan di bawah

K				M				D				E				N	
	-		-		-		-		-		-		-		-		-
		-				-				-				-			

## Langkah 4

Selepas baris pertama, abjad-abjad yang seterusnya diisi di baris kedua dan baris ketiga sehingga semua abjad selesai diisi seperti yang ditunjukkan di bawah.

K				M				D				E				N	
	A		I		U		I		C		M		R		A		G
		M				R				E				L			

## Langkah 5

Baca mesej yang dinyahsulit mengikut corak *zig-zag* seperti yang ditunjukkan di bawah.

K				M				D				E				N	
	A		I		U		I		C		M		R		A		G
		M				R				E				L			

Mesej yang diperoleh ialah KAMIMURIDCEMERLANG. Ruang kosong diisi sendiri semasa membaca mesej, iaitu KAMI MURID CEMERLANG.



## Aktiviti 2.3

### Aktiviti Kumpulan


#### Proses Penyulitan dan Nyahsulit



Kaedah  
Table Talkers

1. Guru memberikan satu mesej kepada setiap kumpulan seperti berikut.

“SEMOGA BERJAYA DALAM PEPERIKSAAN”

2. Guru memegang empat keping kad yang bertulis kaedah sifer yang perlu dipilih oleh setiap kumpulan seperti berikut. Setiap kumpulan perlu memilih sekeping kad sahaja. 

*Pigpen Cipher*

*Columnar Transposition*

*Reverse cipher*

*Caesar Cipher*

3. Setiap kumpulan melakukan penyulitan terhadap teks yang diberikan.
4. Setelah mesej disulitkan, setiap kumpulan perlu menukar mesej sulit masing-masing dengan kumpulan lain.
5. Setiap kumpulan perlu melakukan nyahsulit teks sifer kumpulan lain dan catatkan langkah-langkah nyahsulit yang dilakukan. Bincangkan bersama-sama ahli kumpulan.

2.1.4

### Membanding Kaedah Sifer

Sebelum seorang pengirim menghantar sesuatu mesej sulit, pengirim perlu mempertimbangkan kekuatan dan kelemahan kaedah sifer yang dipilih. Tujuannya adalah untuk memastikan mesej sulit yang dihantar oleh pengirim selamat, iaitu tidak mudah diceroboh dan mesej sulit tersebut boleh dinyahsulit oleh penerima. Dalam subtopik sebelum ini, anda telah mempelajari tiga jenis kaedah sifer, iaitu *Reverse cipher*, *Substitution cipher* dan *Transposition cipher*.


### Standard Pembelajaran

Murid boleh:

- 2.1.4** Membanding beza kaedah sifer yang telah dipelajari.

## Contoh 2.6 Perbandingan untuk kaedah-kaedah sifer yang dipelajari.

Cikgu Ramesh meminta murid-murid kelasnya untuk membandingkan kaedah-kaedah sifer *Columnar Transposition*. Teks biasa yang perlu disulitkan ialah JAGA KEBERSIHAN KELAS.

Kaedah sifer	Penyulitan	
<i>Reverse cipher</i>	<ul style="list-style-type: none"> <li>Tidak perlu penggunaan kunci.</li> <li>Pilih cara songsangan, iaitu songsangkan keseluruhan mesej.</li> <li>Teks sifer yang dihasilkan: SALEK NAHISREBEK AGAJ</li> </ul>	
<i>Caesar Cipher</i>	<ul style="list-style-type: none"> <li>Kunci = 5 digunakan.</li> <li>Lukiskan jadual penggantian abjad berdasarkan anjakan 5 abjad ke kanan.</li> <li>Gantikan setiap abjad dalam teks biasa dengan abjad lain berdasarkan jadual yang dihasilkan.</li> <li>Teks sifer yang dihasilkan: EVBV FZWZMNDCVI FZGVN</li> </ul>	
<i>Pigpen Cipher</i>	<ul style="list-style-type: none"> <li>Bina jadual perwakilan abjad bagi simbol <i>pigpen</i>.</li> <li>Tukarkan teks biasa kepada teks sifer berdasarkan kepada jadual yang dibina.</li> <li>Teks sifer yang dihasilkan:</li> </ul> 	
<i>Rail Fence Cipher</i>	<ul style="list-style-type: none"> <li>Kunci = 4 digunakan.</li> <li>Lukiskan jadual yang mempunyai empat baris. Isikan abjad-abjad dalam teks biasa ke dalam jadual berdasarkan corak <i>zig-zag</i>.</li> <li>Baris demi baris abjad dalam jadual ditulis semula.</li> <li>Teks sifer yang dihasilkan: JBASAEHNAGKRIKLASE</li> </ul>	
<i>Columnar Transposition</i>	<ul style="list-style-type: none"> <li>Kunci = TIKAR</li> <li>Tentukan urutan abjad dalam kunci yang diberikan, iaitu TIKAR =&gt; 5,2,3,1,4</li> <li>Lukiskan satu jadual yang mempunyai lima lajur. Bilangan baris diperoleh dengan formula:  <math display="block">\text{Baris} = \frac{\text{Bilangan abjad dalam teks biasa}}{\text{Bilangan abjad dalam kunci}} = \frac{19}{5} = 3.8.</math>                     Maka, bilangan baris diperlukan ialah 4.</li> <li>Masukkan semua abjad dalam teks biasa ke dalam jadual baris demi baris. Jika terdapat petak yang kosong, isikan dengan sebarang abjad.</li> <li>Tuliskan abjad-abjad secara menegak berdasarkan nombor pada lajur untuk membentuk teks sifer.</li> <li>Teks sifer yang dihasilkan: ARNSABHLGEEAAKSKXJEIE</li> </ul>	

### Uji Minda 2.6

Berdasarkan contoh di atas, kaedah sifer yang manakah paling kuat dan yang manakah paling lemah? Mengapa?





yang telah dipelajari, iaitu *Reverse cipher*, *Caesar Cipher*, *Pigpen Cipher*, *Rail Fence Cipher* dan

Nyahsulit	Kekuatan	Kelemahan
<ul style="list-style-type: none"> <li>• Songsangkan abjad-abjad dalam teks sifer.</li> <li>• Boleh dibaca daripada teks sifer.</li> </ul>	<ul style="list-style-type: none"> <li>• Mudah digunakan.</li> </ul>	<ul style="list-style-type: none"> <li>• Tidak menggunakan kunci.</li> <li>• Mesej mudah dinyahsulit dengan hanya menyongsangkannya.</li> </ul>
<ul style="list-style-type: none"> <li>• Lukiskan jadual penggantian abjad dengan anjakan 5 abjad ke kanan.</li> <li>• Nyahsulitkan abjad demi abjad berdasarkan jadual penggantian yang disediakan.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan kunci.</li> <li>• Abjad dalam teks biasa digantikan dengan abjad lain.</li> </ul>	<ul style="list-style-type: none"> <li>• Kaedah cuba-jaya boleh digunakan untuk meramal nilai kunci.</li> </ul>
<ul style="list-style-type: none"> <li>• Bina jadual perwakilan abjad bagi simbol <i>pigpen</i>.</li> <li>• Tukarkan teks sifer kepada teks biasa berdasarkan jadual yang dibina.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan kunci.</li> <li>• Abjad dalam teks biasa digantikan dengan simbol lain.</li> </ul>	<ul style="list-style-type: none"> <li>• Mudah dinyahsulit jika mengetahui cara perwakilan simbol.</li> </ul>
<ul style="list-style-type: none"> <li>• Lukiskan satu jadual dengan bilangan baris ialah nilai kunci dan bilangan lajur ialah jumlah abjad dalam teks sifer.</li> <li>• Tandakan '-' dalam petak yang perlu diisi abjad dengan corak <i>zig-zag</i>. Semua abjad dalam teks sifer diisi ke dalam petak yang ditandakan baris demi baris.</li> <li>• Teks mesej boleh dibaca mengikut corak <i>zig-zag</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan kunci, iaitu baris.</li> <li>• Bilangan baris sukar diramal.</li> </ul>	<ul style="list-style-type: none"> <li>• Abjad-abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.</li> </ul>
<ul style="list-style-type: none"> <li>• Kunci = TIKAR</li> <li>• Tentukan urutan abjad dalam kunci yang diberikan, iaitu TIKAR =&gt; 5,2,3,1,4</li> <li>• Lukiskan satu jadual yang mempunyai lima lajur. Bilangan baris diperolehi dengan formula:  <math display="block">\text{Baris} = \frac{\text{Bilangan abjad dalam teks biasa}}{\text{Bilangan abjad dalam kunci}}</math> <math display="block">= \frac{19}{5} = 3.8.</math> Maka, bilangan baris diperlukan ialah 4.</li> <li>• Berdasarkan urutan abjad kunci, pindahkan semua abjad dalam teks sifer ke dalam jadual.</li> <li>• Teks mesej dibaca secara mendatar, mulai dengan baris pertama.</li> </ul>	<ul style="list-style-type: none"> <li>• Menggunakan kunci.</li> <li>• Nombor pada lajur yang berdasarkan kunci sukar diramal.</li> </ul>	<ul style="list-style-type: none"> <li>• Abjad-abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.</li> </ul>



## Aktiviti 2.4

Aktiviti Kumpulan

### Perbandingan Kaedah Sifer



Kaedah  
*Hot Seat*

1. Guru akan memberikan dua keping kad yang tertulis kaedah sifer untuk setiap kumpulan.
2. Bersama-sama ahli kumpulan, bincangkan tentang kekuatan dan kelemahan kedua-dua kaedah sifer tersebut.
3. Seorang ahli kumpulan dipilih menjadi “pakar” bagi kumpulan masing-masing.
4. Setiap “pakar” akan duduk di atas kerusi yang disediakan, iaitu *hot seat*.
5. “Pakar” perlu menjawab soalan-soalan yang dikemukakan oleh murid-murid.
6. “Pakar” daripada setiap kumpulan lain mengambil giliran untuk duduk di *hot seat* dan menjawab pertanyaan yang diajukan.

### Standard Pembelajaran

Murid boleh:

- 2.1.5** Memilih kaedah sifer terbaik yang telah dipelajari berdasarkan situasi yang telah diberikan.

### 2.1.5

### Pemilihan Kaedah Sifer Terbaik

Pemilihan penggunaan kaedah sifer untuk penyulitan sesuatu mesej adalah bergantung kepada situasi. Pengirim boleh memilih kaedah sifer yang hendak digunakan mengikut tahap kerahsiaan sesuatu mesej.

Jika sesuatu mesej itu kurang penting atau tahap kerahsiaannya rendah, kaedah *Reverse cipher* atau *Pigpen Cipher* boleh digunakan. Kedua-dua kaedah sifer ini mudah digunakan untuk menyulit atau menyahsulit mesej.

Bagi mesej yang dikategorikan sebagai tahap kerahsiaan sederhana dan tinggi, kaedah *cipher* yang sesuai ialah *Caesar Cipher*, *Rail Fence Cipher* dan *Columnar Transposition* kerana kaedah-kaedah sifer ini memerlukan kunci untuk menyulit dan menyahsulit mesej.

Contoh 2.7, 2.8 dan 2.9 menerangkan pemilihan situasi yang menggunakan kaedah sifer yang sesuai bagi menyulit dan menyahsulit mesej mengikut tahap kerahsiaan mesej.

Peranan sifer dalam keselamatan data



[goo.gl/xRK5jd](https://goo.gl/xRK5jd)



**Contoh 2.7** Pemilihan kaedah sifer yang sesuai untuk mesej yang mempunyai tahap kerahsiaan rendah.

Salim dan Hock Chuan ialah rakan sekelas. Mereka saling membantu semasa mengulang kaji pelajaran. Mereka juga selalu berkomunikasi melalui mesej yang dihantar di dalam kelas untuk menetapkan masa mengulang kaji pelajaran. Setelah mereka mempelajari kaedah-kaedah sifer, mereka ingin menyulitkan mesej mereka kerana tidak ingin diganggu semasa mengulang kaji.

Penentuan tahap kerahsiaan:

1. Mesej yang perlu disulitkan ialah tempat dan masa mengulang kaji. Tahap kerahsiaan adalah rendah.
2. Masa untuk menyulit dan menyahsulit mesej tidak boleh mengambil masa yang lama kerana dilakukan di dalam kelas.

Pilihan kaedah sifer yang sesuai ialah kaedah *Reverse cipher*. Hal ini kerana kaedah ini boleh dilakukan dengan mudah tanpa kegunaan kunci.

**Contoh 2.8** Pemilihan kaedah sifer yang sesuai untuk mesej yang mempunyai tahap kerahsiaan sederhana.

Cikgu Faizal mengadakan mesyuarat tentang pemilihan pengawas sekolah bersama-sama dengan beberapa orang guru. Keputusan mesyuarat yang diperoleh perlu disimpan sebaiknya sehingga hari perhimpunan sekolah. Cikgu Faizal memasukkan nama-nama pengawas yang terpilih di dalam satu fail. Nama fail tersebut disulitkan supaya tidak ada sesiapa yang mengetahui bahawa fail tersebut mengandungi nama-nama pengawas yang terpilih.

Penentuan tahap kerahsiaan:

Mesej yang perlu disulitkan ialah nama fail yang mengandungi nama-nama pengawas. Tahap kerahsiaan adalah sederhana.

Pilihan kaedah sifer yang sesuai ialah kaedah *Caesar Cipher* atau *Rail Fence Cipher* yang mempunyai kunci.

## Contoh 2.9

Pemilihan kaedah sifer untuk mesej yang mempunyai kerahsiaan tinggi.

Seorang pegawai di cawangan sebuah bank ingin menghantar maklumat kepada pegawai bank induk mengenai status pendapatan suku tahunan. Maklumat ini diklasifikasikan sebagai terlalu sulit. Oleh itu, pegawai cawangan mengambil keputusan untuk melakukan penyulitan maklumat tersebut sebelum disampaikan kepada pegawai bank induk.

Penentuan tahap kerahsiaan:

1. Mesej yang perlu disulitkan ialah status pendapatan suku tahunan bank. Tahap kerahsiaan adalah tinggi.
2. Masa yang diperlukan untuk menyulit dan menyahsulit adalah lama kerana maklumat tersebut amat penting.

Kaedah yang paling sesuai digunakan ialah kaedah sifer yang mempunyai kunci, seperti *Columnar Transposition* yang rumit untuk dinyahsulit. Kaedah *Caesar Cipher* dan *Rail Fence Cipher* juga boleh digunakan.

Berdasarkan ketiga-tiga situasi dalam Contoh 2.7, 2.8 dan 2.9, pengirim perlu memilih kaedah yang mempunyai kunci bagi maklumat yang diklasifikasikan sebagai tahap sederhana dan tinggi. Hal ini demikian kerana, kaedah sifer yang mempunyai kunci lebih selamat serta sukar diceroboh oleh pihak lain selain sukar untuk dinyahsulit melainkan dengan penggunaan kunci.



## Aktiviti 2.5

Aktiviti Kumpulan

### Pemilihan Kaedah Sifer Terbaik



Kaedah  
*Brainstorming*

1. Kaji pernyataan masalah berikut.

Puan Aminah ialah seorang guru di sebuah sekolah menengah. Beliau telah ditugaskan untuk menyenaraikan dan menyimpan semua kata laluan yang telah ditetapkan untuk perkakasan wifi di seluruh kawasan sekolah tersebut di dalam satu fail. Senarai kata laluan ini akan menjadi rujukan jika diperlukan. Maka, Puan Aminah telah mengambil keputusan untuk menyimpan semua kata laluan tersebut dalam bentuk teks sifer bagi menjamin keselamatan maklumat.

2. Lakukan aktiviti sumbang saran untuk mendapatkan kaedah sifer yang sesuai bagi tahap kerahsiaan mesej pernyataan masalah di atas. Seorang murid dipilih sebagai pencatat.
3. Berdasarkan hasil sumbang saran yang diperolehi, bincangkan bersama-sama rakan anda bagi memilih kaedah sifer yang paling sesuai untuk digunakan.
4. Berikan justifikasi terhadap kaedah sifer yang dipilih.

## 2.1.6

## Kaedah Sifer dalam Penyelesaian Masalah

Dalam dunia yang serba moden ini, manusia sangat bergantung kepada pelbagai kemudahan sedia ada seperti Internet, rangkaian komputer, pangkalan data, telefon pintar, kad debit, kad kredit dan sebagainya. Kemudahan seperti ini kadangkala memberikan peluang kepada pihak lain untuk mencerooboh maklumat yang terkandung seperti kata laluan e-mel, nombor pin kad debit, maklumat peribadi, rahsia syarikat dan sebagainya. Salah satu cara untuk melindungi maklumat-maklumat penting ini daripada digodam adalah dengan menggunakan kaedah sifer.

Peggunaan kaedah sifer yang kompleks akan menyukarkan pihak lain mencerooboh maklumat yang ingin dilindungi. Walau bagaimanapun, kaedah sifer hanya salah satu daripada kaedah keselamatan siber (*cyber security*) yang digunakan untuk menjaga keselamatan data dan sistem komputer.

Setiap individu boleh menghasilkan kaedah sifer mereka yang unik untuk kegunaan sendiri mengikut keperluan. Dalam proses penghasilan kaedah sifer yang baharu, dua perkara perlu dititikberatkan, iaitu kelebihan dan kelemahan sifer yang akan dibangunkan. Sebagai contoh, dalam *Rail Fence Cipher*, rahsia kepada sifer ini ialah membina satu jadual berdasarkan satu nombor yang disebut sebagai kunci. Sekiranya rahsia *Rail Fence Cipher* terbocor, maka maklumat dapat dinyahsulitkan dengan mudah.

### Standard Pembelajaran

Murid boleh:

- 2.1.6** Menghasilkan kaedah sifer bagi menyelesaikan masalah dalam kehidupan seharian.


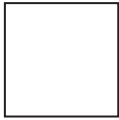







### Contoh 2.10 Penghasilan kaedah sifer yang baharu.

Safri, selaku Pengerusi Kelab Komputer pada tahun ini telah diminta untuk mencadangkan nama pengganti yang berpotensi menjawat jawatan pengerusi kelab tersebut bagi tahun hadapan. Dalam proses menamakan calon-calon yang sesuai, beliau harus memastikan bahawa maklumat ini adalah sulit sehingga disampaikan kepada guru penasihat kelab. Bagi memastikan maklumat sulit ini tidak dicerooboh, Safri telah mencipta satu kaedah sifer baharu yang berjenis *Substitution cipher*.

Jadual 2.4 menunjukkan simbol-simbol yang akan digunakan oleh Safri dan guru penasihatnya untuk menyulit dan nyahsulit maklumat berkaitan pemilihan pengerusi baharu. Kaedah sifer baharu ini menggunakan nombor 1 hingga 6 sebagai kunci dan setiap kunci mewakili simbol 1 yang unik dan simbol 2 yang umum.

Jadual 2.4








Kunci (k)	Simbol 1	Simbol 2
1		
2		
3		
4		
5		
6		

Safri akan mula menyulitkan nama-nama calon yang dicadangkan melalui langkah-langkah berikut.

### Langkah 1

Andaikan, Safri menggunakan kunci  $(k) = 2$  untuk menyulitkan maklumat yang hendak dihantar, maka beliau akan membina satu jadual asas yang berikut berdasarkan kunci pilihan. Baris pertama dimulai dengan  $k = 2$  dan diikuti dengan nombor  $k$  yang seterusnya iaitu 3 hingga 6 dan diulangi semula dengan  $k = 1$ .

Jadual 2.5

Kunci (k)	Simbol 1	Simbol 2
2		
3		
4		
5		
6		
1		

### Uji Minda 2.7

Jika kunci = 5, binakan jadual asas yang perlu dihasilkan.

**Langkah 2**

Setiap abjad (A – Z) diwakili dengan simbol daripada Jadual 2.4. Simbol-simbol bagi empat abjad pertama (A – D) dibentuk dengan menggabungkan simbol 1 dan 2 dari baris pertama, iaitu  $k = 2$ . Abjad A diwakili dengan melukiskan simbol 1 ke dalam kotak simbol 2. Abjad B diwakili dengan melukiskan simbol 1 yang dipusing  $90^\circ$  mengikut arah jam ke dalam kotak simbol 2. Abjad C diwakili dengan melukiskan simbol 1 yang dipusing  $180^\circ$  mengikut arah jam ke dalam kotak simbol 2. Abjad D diwakili dengan melukiskan simbol 1 yang dipusing  $270^\circ$  mengikut arah jam ke dalam kotak simbol 2. Jadual 2.6 menunjukkan hasil lukisan yang akan diperolehi.

**Jadual 2.6**

A		B		C		D	
---	--	---	--	---	--	---	--

Bagi 4 abjad yang seterusnya (E – H), ulangi proses yang sama dengan menggunakan simbol 1 dan simbol 2 bagi  $k = 3$ . Langkah ini diulangi sehingga semua simbol daripada semua kunci (1 – 6) habis digunakan. Anda akan mendapati bahawa hanya abjad Y dan Z yang belum diwakili dengan simbol. Abjad Y akan diwakili oleh simbol 1 dari  $k = 2$ . Manakala abjad Z akan diwakili oleh simbol 2 sahaja. Jadual 2.7 menunjukkan abjad dan perwakilan simbol yang berpadanan.

**Jadual 2.7**

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		

### Langkah 3

Menyulitkan teks biasa kepada teks sifer dengan merujuk kepada Jadual 2.7.

Teks biasa 1:

KHAIRUL

Teks sifer 1:



Teks biasa 2:

AMIRAH

Teks sifer 2:



Maka, Safri akan menggunakan teks sifer 1 dan 2 sebagai nama calon untuk berkongsi dengan guru penasihatnya.



## Aktiviti 2.6

Aktiviti Kumpulan

### Mencipta Kaedah Sifer Baharu



1. Berdasarkan Contoh 2.10, hasilkan satu kaedah sifer baharu yang lain.
2. Anda boleh memilih sama ada ingin menggunakan kaedah yang berasaskan *Substitution cipher* atau *Transposition cipher*.
3. Sediakan satu folio digital bagi penyulitan dan nyahsulit yang dihasilkan dengan menggunakan perisian persembahan.
4. Bentangkan folio digital yang disediakan di hadapan kelas.



## Praktis Amali 2.7

### Permainan Sifer dengan Codemoji

1. Layari laman sesawang ini <https://codemoji.org/#/welcome>.
2. Laman sesawang ini menunjukkan kaedah sifer yang baharu dengan menggunakan simbol *emoji*.
3. Lakukan permainan penyulitan dan nyahsulit dengan menggunakan kaedah *codemoji* bersama-sama rakan anda.



## Uji Kendiri 2.1

1. Terangkan maksud kriptografi (*cryptography*) dan kepentingannya.
2. Mengapakah kriptografi dianggap penting pada zaman moden ini?
3. Nyatakan **empat** kepentingan perkhidmatan keselamatan data kriptografi. Kemudian, padankan setiap kepentingan dengan penerangan yang betul.

Kepentingan perkhidmatan keselamatan data kriptografi	Penerangan
	Pengenalpastian pihak-pihak yang terlibat dalam suatu komunikasi.
	Kesulitan sesuatu maklumat yang dihantar melalui rangkaian komputer terjamin.
	Perkhidmatan keselamatan yang mengenal pasti sebarang perubahan terhadap maklumat atau data oleh pihak lain.
	Membuktikan penghantar dan penerima maklumat tidak menafikan bahawa mereka menghantar dan menerima maklumat tersebut.

4. Terangkan maksud bagi istilah berikut yang digunakan dalam proses sifer.
 

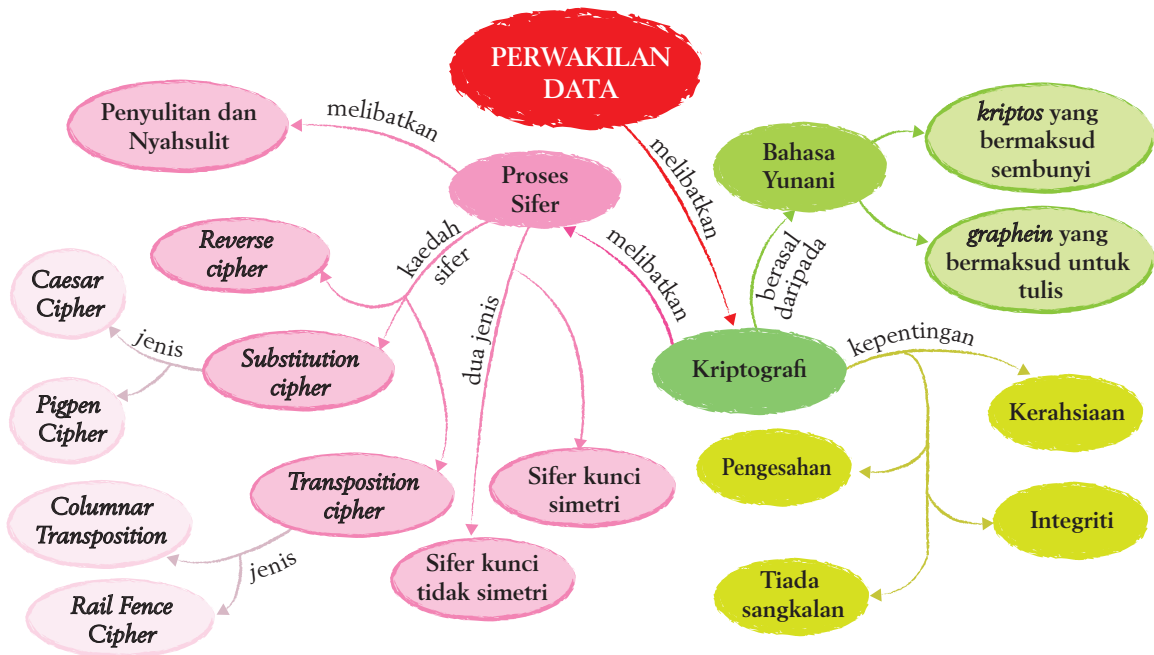
(a) Teks biasa ( <i>plaintext</i> )	(c) Penyulitan	(e) Kunci
(b) Teks sifer ( <i>ciphertext</i> )	(d) Nyahsulit	(f) Sifer ( <i>cipher</i> )
5. Sulitkan mesej SAYA TELAH BERJAYA dengan menggunakan *Reverse cipher*.
6. Aminah menerima satu mesej sulit dan tip daripada ibunya seperti berikut.

Mesej:  
MR MJUJV KNTJB KNAJB

Tip:  
 $K = (30 - (90 \div 30)) \div 3$

Bantu Aminah menyahsulit mesej tersebut untuk mengetahui tempat ibunya menyimpan wang untuk membeli roti.

7. Dengan menggunakan kaedah *Pigpen Cipher*, laksanakan proses penyulitan untuk mesej SAYA AKAN PERGI BERKELAH.
8. Nathan perlu menyulitkan mesej "SAYA TELAH MENERIMA MESEJ ANDA" dengan menggunakan kaedah *Rail Fence Cipher*, kunci = 4. Anda perlu membantu Nathan untuk menyulitkan dan menyahsulitkan mesej balasan "JAPUJMAEGMALMTNPJA" yang diterima oleh Nathan.
9. Bandingkan kekuatan dan kelemahan kaedah-kaedah sifer yang telah digunakan di soalan 6, 7 dan 8.







## SEMAK SEMULA 2

Pada akhir bab ini, saya dapat...

menerangkan maksud kriptografi.	
menerangkan kepentingan kriptografi dalam pengkomputeran.	
menghuraikan langkah-langkah proses sifer melibatkan penyulitan ( <i>encryption</i> ).	
menghuraikan langkah-langkah proses sifer melibatkan nyahsulit ( <i>decryption</i> ).	
menghasilkan mesej dan menterjemah mesej menggunakan kaedah <i>Reverse cipher</i> .	
menghasilkan mesej dan menterjemah mesej menggunakan <i>Caesar Cipher</i> .	
menghasilkan mesej dan menterjemah mesej menggunakan <i>Pigpen Cipher</i> .	
menghasilkan mesej dan menterjemah mesej menggunakan kaedah <i>Transposition cipher</i> .	
mencipta kaedah sifer baharu untuk menyelesaikan masalah.	

# PENTAKSIRAN 2

## Soalan Objektif

- Apakah komponen utama semasa melakukan kriptografi?
  - Kunci
  - Teks biasa
  - Teks sifer
  - Algoritma
  - I dan II
  - II dan III
  - II, III dan IV
  - Semua di atas
- Apakah kunci bagi *Caesar Cipher* jika abjad pertama bermula dengan abjad F?
  - 7
  - 6
  - 5
  - 4
- Apakah simbol grafik bagi abjad B dalam *Pigpen Cipher*?
  - 
  - 
  - 
  - 
- Teliti teks biasa berikut.

SELAMAT MAJU JAYA

Apakah teks sifer yang terhasil apabila teks biasa ini disulitkan menggunakan *Caesar Cipher* dengan kunci 2?

- UGNCPCVPCLWLCAC
- UGMCOCVOCLWLCAC
- UGNCOCVOCLWLCAC
- UHNCPCLWLCAC

- Teliti mesej berikut.

SELAMAT DATANG

Apakah teks sifer yang terhasil apabila mesej di atas disulitkan menggunakan *Rail Fence Cipher* dengan kunci 4?

- SATGEADNLMAAT
- STGEADNLMAAAT
- STEAGDNLMAAAT
- SGTAEADNLMAAT

- Apakah teks sifer yang mungkin terhasil apabila mesej "PERTANDINGAN MELUKIS" disulitkan dengan menggunakan pelbagai jenis teknik *Reverse cipher*?
  - NATREPNAGNID SIKULEM
  - SIKULEM NAGNIDNATREP
  - ULEMSIK NAGNIDNATREP
  - NAGNIDNATREP SIKULEM
  - I dan II
  - I dan III
  - II dan IV
  - III dan IV

- Rajah berikut menunjukkan satu teks sifer yang telah disulitkan menggunakan *Pigpen Cipher*.



Apakah teks biasa yang terhasil apabila teks sifer di atas dinyahsulit?

- BELAJAR
- BELAYAR
- NELAYAN
- PELAJAR

8. Suraj ingin menyulitkan suatu mesej menggunakan *Columnar Transposition*. Dia menggunakan kunci "SIHAT" untuk menyulitkan mesej tersebut. Berapakah bilangan lajur (*column*) yang digunakan oleh Suraj?
- A 4                      C 6  
B 5                      D 7

9. Darren menyulitkan suatu mesej untuk dihantar kepada Khairul. Mesej yang dihantar itu telah diceroboh oleh pihak lain kerana Darren menggunakan kaedah sifer yang menggunakan simbol. Apakah kemungkinan teknik sifer yang digunakan oleh Darren?
- A *Caesar Cipher*  
B *Rail Fence Cipher*  
C *Pigpen Cipher*  
D *Columnar Transposition*

10. Yee Mei ingin menggunakan *Rail Fence Cipher* untuk menyulitkan suatu mesej yang akan dihantar kepada penerima. Dia telah memindahkan satu teks sifer kepada bentuk seperti di bawah.

T						S			
	E					A		I	
		R				K			H
			I		A				X
				M					X

- Apakah kunci yang perlu digunakan oleh penerima untuk menyahsulitkan mesej tersebut?
- A  $K = 3$   
B 5 baris  
C 13 lajur  
D Perkataan BENIH

### Soalan Berstruktur

1. Apakah tujuan utama kriptografi dilakukan?
2. Apakah hubungan antara kriptografi dengan penyulitan dan sifer?
3. Jelaskan perbezaan antara kunci simetri dan kunci tidak simetri.
4. Lakukan penyulitan pada teks biasa "KEPUTUSAN PEPERIKSAAN AKHIR TAHUN" menggunakan teknik *Reverse cipher*.
5. Anatasia ingin menggunakan teknik *Caesar Cipher* berdasarkan jadual susunan abjad teks biasa dan teks sifer untuk penyulitannya.

Abjad dalam teks sifer

W X Y Z A B C D E F G H I J K L M N O P Q R S T U V

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Abjad dalam teks biasa

Berdasarkan senarai abjad di atas,

- (a) nyatakan kunci yang digunakan.
- (b) sulitkan mesej "KUNCI DI BAWAH KARPET".
- (c) nyahsulitkan teks sifer "EJPACNEPEIWGHQIWP".

### E-PRAKTIS



6. Anda telah menerima satu teks sifer daripada rakan anda seperti berikut.



JSIRRJMUADADAAMYANEMXPATMKLXADAEIAX

Mesej asal bagi teks sifer itu telah disulitkan menggunakan *Columnar Transposition* dengan lajur 5. Anda perlu menyahsulitkan teks sifer tersebut untuk mengetahui apakah mesej yang rakan anda ingin sampaikan.

- Berapakah bilangan baris yang akan anda gunakan?
- Nyahsulitkan teks sifer tersebut.
- Adakah terdapat huruf yang tidak digunakan dalam teks tersebut? Mengapa?

### Soalan Penyelesaian Masalah



Ahmad dan Mei Ling ialah pengerusi dan bendahari bagi koperasi sekolah. Setiap hari, bendahari akan menyimpan kutipan wang di dalam peti wang yang disimpan dalam pejabat koperasi sekolah. Pin keselamatan peti wang dicatatkan dan disimpan dalam satu fail koperasi yang tersembunyi. Hal ini bagi memudahkan pengerusi koperasi membuat laporan dan guru koperasi mengeluarkan wang untuk dimasukkan ke dalam bank. Pada suatu hari, peti wang tersebut hilang. Pencuri yang mencuri wang tersebut tidak dapat ditangkap. Oleh yang demikian, Ahmad dan Mei Ling telah membuat keputusan untuk mengambil tindakan bagi mencegah segala kes kecurian yang bakal berlaku. Bantu mereka dengan,

- mencadangkan satu kaedah *Substitution cipher* dan satu kaedah *Transposition cipher* yang sesuai digunakan oleh Ahmad dan Mei Ling. Mengapakah kedua-dua jenis sifer dicadangkan sesuai untuk digunakan? Berikan justifikasi anda.
- melakukan penyulitan menggunakan satu kaedah sifer yang baharu.

Teks biasa yang perlu disulitkan ialah nama fail koperasi yang tersembunyi, iaitu

AKAUN AMANAH KOPERASI SEKOLAH

# BAB 3

# ALGORITMA

Keluarga Encik Fong sedang membuat persiapan untuk pergi bercuti ke Kundasang, Sabah. Mereka perlu menempah tiket kapal terbang dari Kuala Lumpur ke Kota Kinabalu. Mereka juga perlu menempah bilik hotel di Kundasang. Bagaimanakah *apps* (aplikasi) dalam telefon pintar dapat membantu keluarga Encik Fong untuk menempah tiket kapal terbang dan bilik hotel?

## Standard Kandungan

### 3.1 Pembangunan Algoritma

Cuti sekolah ini saya bercadang nak bawa anak-anak kita pergi bercuti ke Sabah.

Bagus juga. Sudah lama kita tidak pergi bercuti bersama anak-anak. Kawan saya kata Kundasang tempat yang menarik. Cuaca dan pemandangannya seperti di luar negara. Mari kita tengok tiket penerbangan untuk ke sana semasa cuti sekolah.

Tonton video ini untuk mengetahui tentang gambaran Bab 3.



[goo.gl/gpvvaA](http://goo.gl/gpvvaA)



Fri, 2 Mar 2018			
129 <sup>MYR</sup> 00			
Fare			
06:15 → 08:55	2H 40M AK 5110	ADULT 129 <sup>MYR</sup> 00	KID 129 <sup>MYR</sup> 00
KUL → BKI			
4 Seat(s) left at this fare			
07:20 → 09:55	2H 35M AK 5108	ADULT 159 <sup>MYR</sup> 00	KID 159 <sup>MYR</sup> 00
KUL → BKI			

*Apps* (aplikasi) telefon pintar dan enjin gelintar yang digunakan oleh ahli keluarga Encik Fong menggunakan teknik carian (*search*) dan isihan (*sort*). Kedua-dua teknik ini digunakan bagi mendapatkan maklumat yang diinginkan dengan lebih mudah. Kita akan mempelajari dengan lebih lanjut berkaitan teknik carian dan isihan dalam bab ini.

## Kata Kunci

- *Search*
- *Sort*
- *Linear search*
- *Binary search*
- *Bubble sort*
- *Bucket sort*

Ayah kata nak bawa kita pergi bercuti ke Kundasang.

Mari kita cari tempat-tempat menarik yang boleh kita lawati di sana nanti.



Wah, seronoknya! Mesti banyak tempat menarik di Kundasang.

# 3.1 PEMBANGUNAN ALGORITMA

## Imbas Kembali

Algoritma ialah salah satu teknik penyelesaian masalah. Algoritma boleh diwakili dengan pembinaan pseudokod dan carta alir.

Algoritma ialah satu siri langkah atau tindakan yang jelas yang perlu dilaksanakan untuk menghasilkan output yang dikehendaki. Dalam bidang pengkomputeran, pengatur cara akan membina atur cara berpandukan algoritma yang dibangunkan terlebih dahulu. Pembangunan algoritma merupakan satu teknik pengitlakan yang melibatkan pembinaan langkah-langkah penyelesaian masalah.

Tugas ahli-ahli dalam bidang sains komputer adalah mengkaji dan membangunkan algoritma dengan tujuan untuk menyelesaikan masalah-masalah pengaturcaraan. Antara faedah yang diperolehi daripada kajian serta pembangunan algoritma yang dilakukan adalah seperti berikut:

1. Memahami bagaimana komputer memproses urutan tindakan-tindakan yang perlu dilaksanakan.
2. Mempelajari cara menulis algoritma yang teratur, mudah dibaca dan difahami.
3. Mempelajari cara menulis algoritma dengan cekap.
4. Mempelajari cara melakukan penambahbaikan pada algoritma.

Dalam bab ini, anda akan mempelajari algoritma bagi dua jenis masalah pengaturcaraan, iaitu algoritma *search* dan algoritma *sort*.

Search dan Sort dalam e-mel



[goo.gl/noGM8v](http://goo.gl/noGM8v)

Imbas Saya

## Standard Pembelajaran

Murid boleh:

- 3.1.1** Mengenal pasti ciri-ciri *search* (*linear, binary*) dan *sort* (*bubble, bucket*)

### 3.1.1 Ciri-ciri Search dan Sort

Zulaikha ingin mendapatkan sebuah buku di perpustakaan untuk digunakan sebagai rujukan semasa membuat tugasan. Bagaimanakah Zulaikha mencari buku yang dikehendakinya di perpustakaan? Terdapat lebih daripada satu cara atau teknik *search* boleh digunakan bagi mencari buku-buku di perpustakaan bergantung kepada bagaimana buku-buku tersebut disusun.

#### (i) Search

*Search* ialah proses untuk mendapatkan suatu item tertentu yang terkandung dalam satu senarai. Dalam bidang pengkomputeran, *search* amat penting untuk membantu manusia mendapatkan maklumat yang dikehendaki dengan lebih cepat. Manusia boleh mengarahkan aplikasi pemprosesan kata untuk mencari satu perkataan yang berulang dalam suatu senarai.

Dalam bab ini, dua jenis *search* akan dipelajari, iaitu *linear search* dan *binary search*.

## Tahukah Anda?

*Search* dan *sort* hanya akan berlaku apabila terdapat satu senarai atau pangkalan data.



**Linear search**

*Linear search* ialah satu teknik untuk mendapatkan item yang dikehendaki dalam satu senarai linear. Carian akan bermula dengan item pertama yang terdapat dalam senarai. Jika item pertama bukan item yang dikehendaki, carian akan diteruskan dengan item kedua. Jika item kedua bukan item yang dikehendaki, carian akan diteruskan dengan item ketiga dan seterusnya sehingga item yang dikehendaki diperolehi. Jika item yang dikehendaki tidak diperolehi sehingga carian selesai, maka carian akan ditamatkan tanpa hasil.

Penerangan *linear search*



[goo.gl/CWLfLM](http://goo.gl/CWLfLM)

**Imbas Saya**

**Contoh 3.1** Pencarian satu item menggunakan *linear search*.

Erina dan kakaknya, Esma ingin membeli buah mangga untuk membuat puding. Mereka ke sebuah pasar raya yang menjual buah-buahan. Terdapat beberapa bahagian yang diletakkan dengan pelbagai jenis buah-buahan di sepanjang ruang jualan buah-buahan. Erina dan Esma berjalan di sepanjang ruang jualan buah-buahan itu untuk mencari buah mangga.

Buah-buahan yang terdapat dalam setiap bahagian adalah seperti berikut:

Bahagian pertama	Bahagian kedua	Bahagian ketiga	Bahagian keempat	Bahagian kelima
				

**Langkah 1**

Carian bermula di bahagian pertama. Adakah buah mangga berada di bahagian pertama?

Bahagian pertama	Bahagian kedua	Bahagian ketiga	Bahagian keempat	Bahagian kelima
				

**Langkah 2**

Bahagian pertama mengandungi buah pisang, bukan buah mangga. Carian akan menuju ke bahagian kedua. Adakah buah mangga berada di bahagian kedua?

Bahagian pertama	Bahagian kedua	Bahagian ketiga	Bahagian keempat	Bahagian kelima
				

### Langkah 3

Bahagian kedua mengandungi buah manggis, bukan buah mangga. Carian akan menuju ke bahagian ketiga. Adakah buah mangga berada di bahagian ketiga?

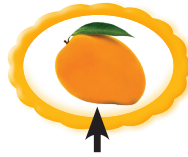
Bahagian pertama



Bahagian kedua



Bahagian ketiga



Bahagian keempat



Bahagian kelima



### Langkah 4

Bahagian ketiga mengandungi buah mangga. Carian dihentikan.

**Contoh 3.2** Pencarian data dalam satu senarai menggunakan *linear search*.

Cikgu Aiman ingin mencari nama murid Kelas 3 Utarid yang memperoleh 90 markah dan ke atas dalam ujian bulanan. Kelas 3 Utarid terdiri daripada 27 orang murid.

Perhatikan senarai nama Kelas 3 Utarid yang berikut. Carian dimulakan dengan nama pertama dalam senarai. Alif Qusyairi mendapat 98 markah. Catat nama Alif Qusyairi. Kemudian, lihat nama kedua dan nama seterusnya sehingga nama terakhir. Nama murid yang mendapat 90 markah dan ke atas akan dicatatkan.

	A	B	C	D
1				
2		<b>NAMA MURID</b>	<b>MARKAH</b>	
3	1	ALIF QUSYAIRI BIN MUAZ	98	
4	2	AMRUTHA A/P RAJU	76	
5	3	DHESIGAN A/L SUBRA	90	
6	4	MANJIT SINGH	78	
7	5	MARK ANAK AGAN	64	
8	6	MATHEW ANAK JAMES	88	
9	7	MOEY KAM SENG	90	
10	8	MOHAMAD AIMAN NAIM BIN SYUKRI	80	
11	9	MOHAMAD HAIDHAR BIN MARZUKI	75	
12	10	MONISA ANAK BALANGAN	90	
13	11	MUHAMMAD HAIKAL BIN MOHD FAISAL	72	
14	12	MUHAMMAD DANISH BIN ABDUL RAHMAN	88	
15	13	MUHAMMAD IFFAT BIN ROZAIDI	78	
16	14	MUHAMMAD INDRA IRFAN BIN NOOR SYAFIQ	68	
17	15	MUHAMMAD RAMADHAN BIN SULAIMAN	60	
18	16	MUHAMMAD ZULHILMI BIN MOHD AMIRUDDIN	82	
19	17	NG YOKE LAN	72	
20	18	NOR AINA NAJWA BINTI MOHD RASHID	74	
21	19	NUR AIN SHAZWANI BINTI AMIR FIRDAUS	98	
22	20	NUR ALIA FARIESYA BINTI ZUHRI	96	
23	21	NURUL ASNA BINTI GHAFFAR	78	
24	22	NURUL NAJWA BINTI NORDIN	86	
25	23	NURUL NATASHA FITRI BINTI MOHD FAKHRI	74	
26	24	RUBHITRA ALANG	84	
27	25	SAFIYYA MAISARA BINTI SHAHRIZAL	70	
28	26	SARVINA KAUR	80	
29	27	TAN MOOI YONG	82	
30				

Carian pertama

Carian terakhir

Nama murid yang disenaraikan oleh Cikgu Aiman ditunjukkan dalam Jadual 3.1.

**Jadual 3.1** Senarai nama murid yang mendapat 90 markah dan ke atas


	NAMA MURID	MARKAH
1	ALIF QUSYAIRI BIN MUAZ	98
2	DHESIGAN A/L SUBRA	90
3	MOEY KAM SENG	90
4	MONISA ANAK BALANGAN	90
5	NUR AIN SHAZWANI BINTI AMIR FIRDAUS	98
6	NUR ALIA FARIESYA BINTI ZUHRI	96

Algoritma *linear search* ialah satu algoritma yang sangat mudah. Item-item dalam senarai tidak perlu diisih mengikut urutan terlebih dahulu apabila *linear search* ingin diaplikasikan. Walau bagaimanapun, *linear search* ini tidak begitu efisien kerana teknik ini akan menyemak setiap item yang terdapat dalam senarai. Cara penyemakan ini memerlukan masa yang panjang, terutamanya jika senarai mengandungi terlalu banyak item. Jika item yang dicari berada di akhir senarai atau item yang dicari tidak berada dalam senarai, carian akan tetap dilakukan sehingga item yang terakhir.

### Binary search

*Binary search* ialah teknik carian yang melibatkan keputusan dwipilihan. Sebelum melakukan *binary search*, item-item dalam senarai perlu diisih dalam urutan menaik. Kemudian, item yang berada di tengah senarai akan disemak. Jika item di tengah ialah item yang dikehendaki, maka carian akan tamat. Jika tidak, item yang dikehendaki dibandingkan dengan item di tengah senarai itu. Jika item yang dikehendaki lebih kecil, maka item di tengah senarai serta semua item di sebelah kanannya diabaikan. Jika item yang dikehendaki lebih besar, maka item di tengah senarai serta semua item di sebelah kirinya diabaikan. Proses ini diulang pada senarai item yang tinggal dan berterusan sehingga item yang dikehendaki diperolehi. Jika item yang dikehendaki tidak diperolehi apabila carian selesai, maka carian akan ditamatkan tanpa hasil.

*Binary search* adalah lebih efisien dan menjimatkan masa berbanding dengan *linear search* kerana *binary search* tidak perlu menyemak setiap item dalam senarai. *Binary search* ini juga sesuai digunakan pada senarai dengan item yang banyak.

 **Uji Minda 3.1**


Fikirkan satu situasi harian yang melibatkan *linear search*.

Penerangan *binary search*



[goo.gl/fgtBUX](https://goo.gl/fgtBUX)

**Imbas Saya**

 **Uji Minda 3.2**

Fikirkan satu situasi harian yang melibatkan *binary search*.

## Tahukah Anda?

Indeks dalam sesuatu senarai adalah penentu kedudukan bagi setiap item dalam senarai tersebut. Dalam bidang pengkomputeran, indeks bagi sesuatu senarai bermula dengan 0.

## Tahukah Anda?

Dalam *binary search*, titik pertengahan yang menghasilkan nombor perpuluhan akan dibaca oleh komputer sebagai nombor bulat terkecil bagi indeks tersebut. Contohnya 13 bahagi 5 ialah 2.6. Maka, komputer akan membaca nilai item di indeks 2.

### Contoh 3.3

Pencarian satu nombor menggunakan *binary search*.

Diberikan satu senarai nombor. Nombor yang dicari ialah 35.

12	30	35	39	42	48	50	53	62
----	----	----	----	----	----	----	----	----

**Langkah 1** Letakkan indeks pada setiap item dalam senarai.

Indeks dalam satu senarai bermula dengan 0.

Indeks	0	1	2	3	4	5	6	7	8
	12	30	35	39	42	48	50	53	62

**Langkah 2** Kenal pasti item pertengahan dalam senarai.

Untuk mencari item pertengahan, formula yang digunakan adalah

$$\frac{\text{Indeks terkecil} + \text{Indeks terbesar}}{2} = \frac{0 + 8}{2} = 4$$

Maka, item pertengahan dalam senarai ini berada pada indeks 4, iaitu 42.

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62

Item  
pertengahan

**Langkah 3** Bandingkan item yang dicari dengan item pertengahan. Didapati  $35 < 42$ .

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62

$35 < 42$

**Langkah 4** Oleh sebab  $35 < 42$ , maka item 42 dan item-item selepasnya diabaikan. Kenal pasti item pertengahan dalam senarai yang tinggal.

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62

Item  
pertengahan

**Langkah 5** Bandingkan 35 dengan 30. Didapati  $35 > 30$ .

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62



**Langkah 6** Oleh sebab  $35 > 30$ , maka item 30 dan item sebelumnya diabaikan. Kenal pasti item pertengahan dalam senarai yang tinggal.

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62



**Langkah 7** Bandingkan 35 dengan 35. Didapati  $35 = 35$ .

0	1	2	3	4	5	6	7	8
12	30	35	39	42	48	50	53	62

**Langkah 8** Carian selesai.

**Contoh 3.4** Pencarian satu aksara menggunakan *binary search*.

Shalina perlu menggunakan Buku Panduan Telefon untuk mendapatkan nombor telefon rumah Maisen. Shalina perlu mencari nama bapa Maisen, iaitu James di dalam Buku Panduan Telefon.

**Langkah 1** Andaikan halaman tengah yang dibuka ialah 'M'. Perhatikan 'J' berada sebelum 'M'.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z



Maka, separuh daripada halaman buku, iaitu bermula daripada 'M' hingga 'Z' diabaikan.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Langkah 2** Halaman tengah bahagian pertama dibuka dan halaman 'F' diperolehi.



A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Disebabkan huruf 'J' yang dicari berada selepas huruf 'F', maka halaman huruf 'F' dan semua halaman sebelumnya diabaikan.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Langkah 3** Jika halaman yang dibuka ialah 'I'.

dibuka

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Maka halaman huruf 'G', 'H' dan 'I' boleh diabaikan kerana berada sebelum huruf 'J'.

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Langkah 4** Jika halaman yang dibuka ialah 'K', maka halaman-halaman 'K' dan 'L' boleh diabaikan dan Shalina perlu membuat carian pada halaman sebelum 'K', iaitu 'J'.

dibuka

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Carian dijumpai

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z



### Aktiviti 3.1



Aktiviti Kumpulan

Perbandingan antara *linear search* dan *binary search*

1. Perhatikan senarai nombor berikut.

25	10	35	45	5	30	15	40	20
----	----	----	----	---	----	----	----	----

2. Cari nombor 5 dengan menggunakan

- linear search*.
- binary search*.

3. Cari nombor 40 dengan menggunakan

- linear search*.
- binary search*.

4. Bandingkan bilangan langkah yang digunakan dalam soalan 2 dan 3. Beri justifikasi anda.

**(ii) Sort**

*Sort* ialah proses mengisih atau menyusun item-item dalam suatu senarai linear mengikut urutan yang tertentu. Dalam bidang perkomputeran, *sort* ialah teknik yang penting untuk mendapatkan maklumat dengan cepat dan tepat.

Antara contoh *sort* dalam kehidupan seharian ialah mengisih surat-surat di pejabat pos mengikut negeri, menyusun senarai nama murid mengikut abjad dan menyusun murid-murid ke dalam barisan mengikut rumah sukan.

Dalam bab ini, dua jenis *sort* akan dipelajari, iaitu *bubble sort* dan *bucket sort*.



**Bubble sort**

*Bubble sort* ialah satu teknik pengisihan yang mudah untuk mengisih item-item dalam senarai mengikut urutan menaik atau menurun. Teknik isihan ini bermula dengan membandingkan dua item pertama dalam senarai. Bagi isihan yang melibatkan urutan menaik, item pertama dengan item kedua akan bertukar tempat jika nilai item pertama lebih besar daripada nilai item kedua. Bagi isihan yang melibatkan urutan menurun, item pertama dengan item kedua akan bertukar tempat jika nilai item pertama lebih kecil daripada nilai item kedua. Perbandingan dua item bersebelahan akan berterusan pada item-item seterusnya sehingga ke item terakhir.

**Contoh 3.5** Isihan mengikut urutan menurun menggunakan *bubble sort*.

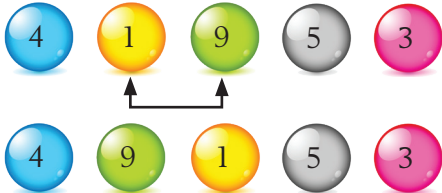


**Langkah 1**



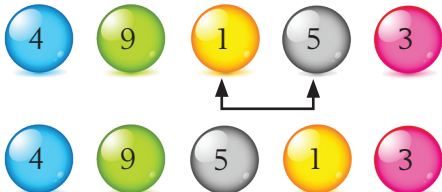
Bandingkan dua item pertama dalam senarai, iaitu 4 dengan 1. Oleh sebab  $4 > 1$ , maka tiada sebarang penukaran tempat dilakukan.

**Langkah 2**



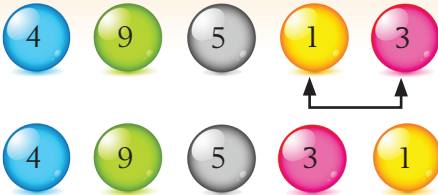
Bandingkan dua item seterusnya, iaitu 1 dengan 9. Perhatikan bahawa  $1 < 9$ . Maka, penukaran tempat perlu dilakukan antara 1 dengan 9.

**Langkah 3**



Seterusnya, bandingkan 1 dengan 5. Penukaran tempat perlu dilakukan kerana  $1 < 5$ .

#### Langkah 4

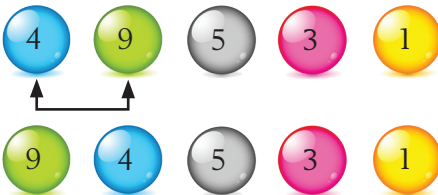


Dua item seterusnya yang perlu dibandingkan ialah 1 dengan 3. Oleh sebab  $1 < 3$ , maka kedua-dua item perlu saling bertukar.

Perhatikan bahawa kesemua item telah melalui proses *bubble sort* tetapi senarainya masih belum berada dalam urutan menurun. Maka, proses perbandingan diulang semula.

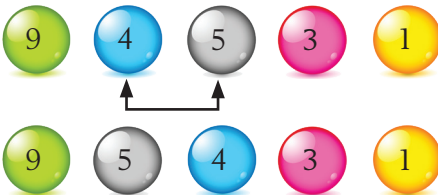


#### Langkah 5



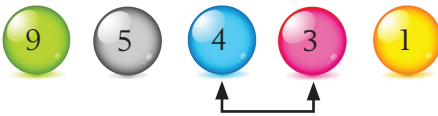
Bandingkan semula item pertama dengan item kedua. Perhatikan  $4 < 9$ . Maka kedua-dua item perlu saling bertukar.

#### Langkah 6



Bandingkan pula 4 dengan 5. Perhatikan  $4 < 5$ , maka kedua-dua item itu perlu saling bertukar.

#### Langkah 7



Seterusnya, bandingkan 4 dengan 3. Kedua-dua item ini berada pada tempat yang betul dan tidak perlu ditukar.

#### Langkah 8



Bandingkan dua item yang terakhir. Oleh sebab  $3 > 1$ , maka kedua-dua item itu tidak perlu ditukar.

Perhatikan bahawa kesemua item telah diisih mengikut urutan menurun. Maka, proses *bubble sort* dihentikan.





**Bucket sort**

*Bucket sort* ialah satu teknik yang mengasingkan item-item dalam senarai tertentu ke dalam baldi (*bucket*), seterusnya item-item di dalam baldi akan diisih dan disusun semula ke dalam senarai. Bilangan baldi yang diperlukan bergantung kepada pengatur cara dan bilangan item dalam senarai yang perlu diisih.

**Contoh 3.6** Isihan mengikut urutan menaik menggunakan *bucket sort*.



**Langkah 1**

Sediakan tiga buah baldi dan tetapkan julat yang sesuai bagi item yang akan diletakkan ke dalam setiap baldi.



Item A hingga C



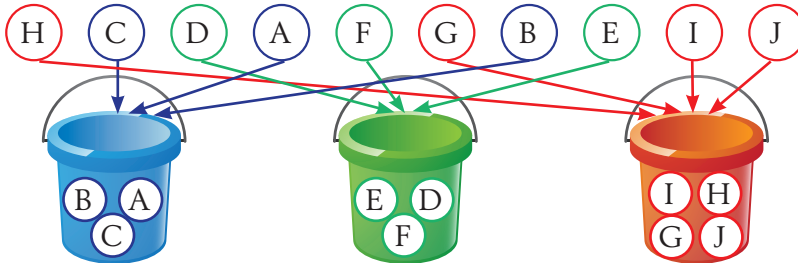
Item D hingga F



Item G hingga J

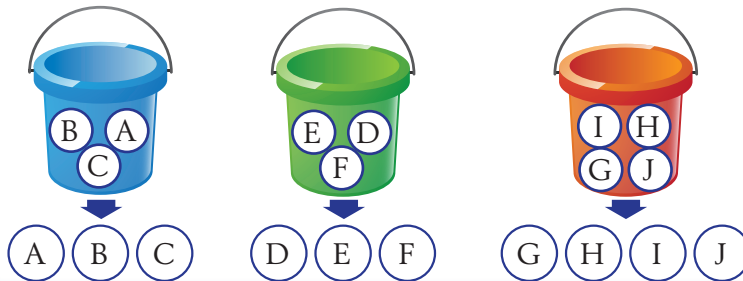
**Langkah 2**

Masukkan item-item yang perlu diisih ke dalam baldi mengikut julat yang betul.



**Langkah 3**

Isih item-item di dalam setiap baldi secara berasingan dan masukkan semula item-item yang telah diisih ke dalam senarai.



**Uji Minda 3.3**

Diberikan satu senarai, 4, 7, 2, 5, 6. Isihkan senarai ini dalam urutan menaik dengan menggunakan *bubble sort*. Tunjukkan langkah kerja anda.

Penerangan *bucket sort*

[goo.gl/ugNmPD](http://goo.gl/ugNmPD)

**Imbas Saya**

**Uji Minda 3.4**

Nyatakan satu aktiviti harian yang menggunakan *bucket sort*.

**Tahukah Anda?**

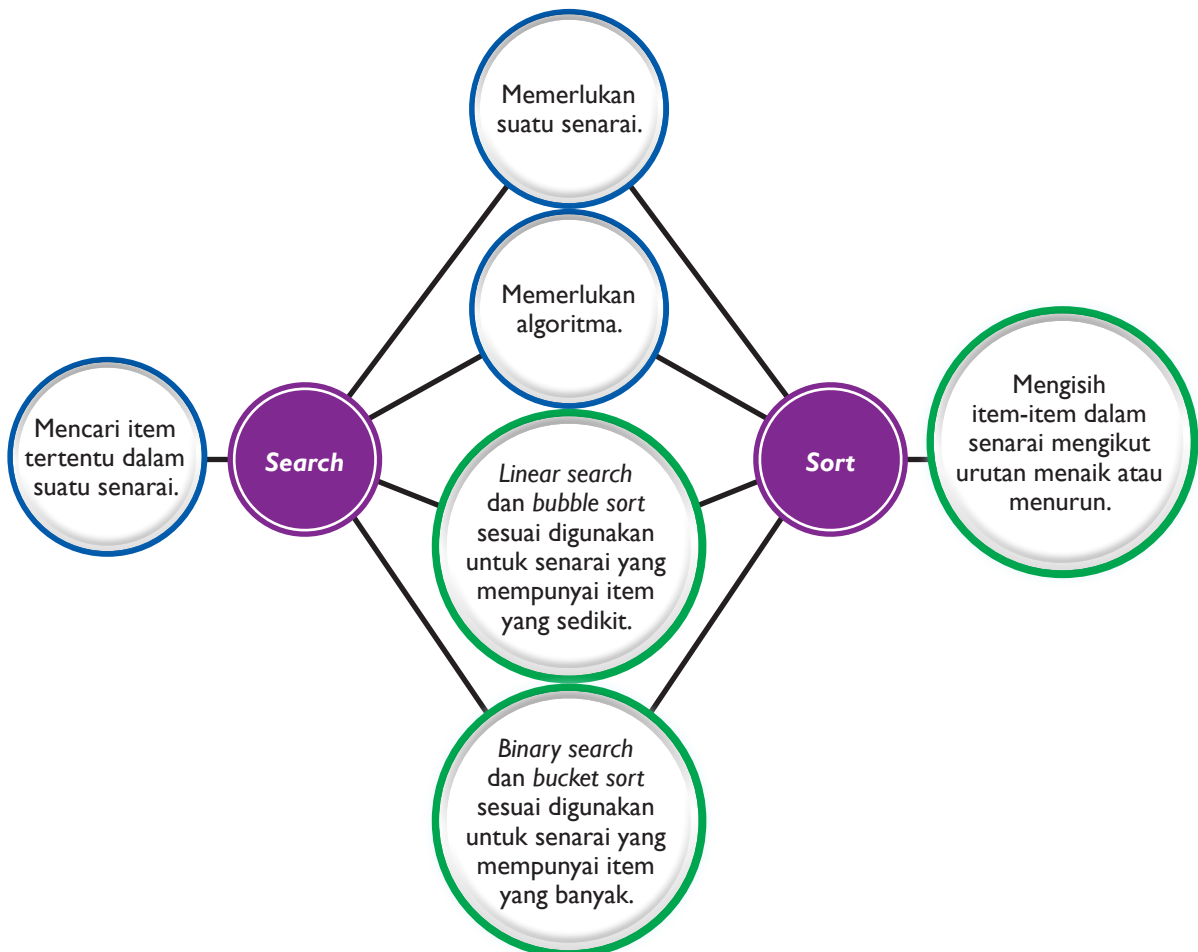
Item-item di dalam setiap baldi boleh diisih menggunakan kaedah *bubble sort*.



Membuat perbandingan antara kaedah *bubble sort* dan *bucket sort*

1. Seorang ahli dari setiap kumpulan akan mengukur ketinggian setiap ahli kumpulan masing-masing.
2. Kemudian, catatkan ketinggian setiap ahli kumpulan dalam bentuk jadual.
3. Isih rekod ketinggian yang telah dicatat itu mengikut tertib menaik menggunakan (a) *bubble sort*.  
(b) *bucket sort*.
4. Bincang bersama-sama ahli kumpulan tentang perbandingan antara kaedah *bubble sort* dan *bucket sort* yang dilakukan.
5. Bentangkan dapatan kumpulan masing-masing di hadapan kelas.

Daripada kedua-dua kaedah *search* dan *sort* yang telah dipelajari, terdapat beberapa ciri persamaan dan perbezaan antara kedua-duanya. Rajah 3.1 menunjukkan persamaan dan perbezaan ciri-ciri bagi *search* dan *sort*.



Rajah 3.1 Persamaan dan perbezaan ciri-ciri *search* dan *sort*

### 3.1.2(i) Menulis Pseudokod dan Melukis Carta Alir bagi *Linear Search*

Algoritma ialah satu set langkah yang teratur untuk dilakukan bagi mencapai sesuatu hasil yang diinginkan. Algoritma wujud dalam kebanyakan aktiviti harian seperti memasak, membasuh pakaian, menganjurkan sesuatu majlis dan menyelesaikan sesuatu masalah. Jika langkah-langkah dalam algoritma yang telah dibina tidak diikuti dengan betul, hasil atau output yang diperoleh berkemungkinan tidak seperti yang dikehendaki.

Algoritma boleh diwakilkan dalam bentuk pseudokod atau carta alir. Pseudokod ialah perwakilan algoritma dalam bahasa yang boleh difahami manakala carta alir adalah perwakilan algoritma dalam bentuk grafik.

Oleh itu, sebelum menyelesaikan sesuatu masalah khususnya dalam bidang pengkomputeran, pseudokod dan carta alir perlu dibina supaya set langkah yang dihasilkan jelas dan mudah difahami. Pembinaan algoritma juga memudahkan seseorang untuk mengesan ralat yang ada sekiranya output yang diperoleh mempunyai kesilapan selain dapat memastikan output yang dihasilkan adalah output yang diinginkan.

**Contoh 3.7** Langkah-langkah pembinaan pseudokod dan carta alir bagi *linear search* untuk mencari satu nombor dalam senarai yang rawak (*random list*).

Carian suatu nombor menggunakan senarai [10, 82, 5, 18, 27, 15, 44, 100, 42, 99].

#### Langkah 1

Memahami proses *linear search* dengan membina algoritma

Algoritma bagi proses *linear search* dirumuskan seperti Rajah 3.2.

1. Mula dengan item pertama dalam senarai.
2. Bandingkan item carian dengan item pertama.
3. Jika nilai item carian sama dengan nilai item pertama, carian dihentikan.
4. Jika nilai item carian tidak sama dengan nilai item pertama, ulang Langkah 2 hingga Langkah 4 dengan item yang berikutnya dalam senarai.
5. Carian tamat apabila item carian diperoleh atau apabila semua item telah disemak.

Rajah 3.2 Algoritma proses *linear search*

### Standard Pembelajaran

Murid boleh:

- 3.1.2** Menulis pseudokod dan melukis carta alir yang menunjukkan:
- (i) *linear search*

### Imbas Kembali

Semasa di Tingkatan 1, anda telah mempelajari pembangunan algoritma yang melibatkan struktur kawalan pilihan, struktur kawalan ulangan dan cara mengesan ralat pada algoritma dalam penyelesaian masalah.

Semasa di Tingkatan 2 pula, anda telah mempelajari pembangunan algoritma yang melibatkan struktur kawalan pilihan bersarang, struktur kawalan ulangan dan cara mengesan serta membaiki ralat dalam pseudokod dan carta alir dalam penyelesaian masalah.

Algoritma dalam *linear search*



[goo.gl/mXnQgH](http://goo.gl/mXnQgH)



## Uji Minda 3.5

Anda ingin membeli hadiah sempena sambutan Hari Guru. Anda mengunjungi sebuah kedai cenderamata. Gunakan *linear search* dan huraikan cara anda mendapatkan hadiah yang diinginkan.

### Langkah 2

### Menulis pseudokod dan melukis carta alir

Pemboleh ubah yang berkaitan perlu diketahui secara kasar terlebih dahulu. Pemboleh ubah yang diperlukan adalah seperti berikut:

#### Katakan:

- **L** ialah senarai yang mengandungi 10 item dengan nilai-nilai  $L_0 = 10, L_1 = 82, \dots, L_{n-1} = 99$ .
- **T** ialah pemboleh ubah yang mewakili nilai item carian.
- **i** ialah pemboleh ubah yang mewakili kedudukan item dalam senarai **L**.
- **n** ialah pemboleh ubah yang mewakili bilangan item dalam senarai **L**.

Rajah 3.3 dan Rajah 3.4 menunjukkan pseudokod dan carta alir yang dibina.

## Tahukah Anda?

Dalam Contoh 3.7, pemboleh ubah **L**, **T**, **i** dan **n** masing-masing daripada perkataan *List*, *Target*, *index* dan *number*. Huruf lain juga boleh digunakan untuk mewakili pemboleh ubah.

Sebenarnya, hampir semua bahasa pengaturcaraan membekalkan *search* dan *sort* dalam pustaka masing-masing. Namun demikian, mempelajari pembinaan algoritma membantu kamu memahami bagaimana masalah dapat diselesaikan dan bagaimana cara komputer memproses data. Pengetahuan ini akan memberikan faedah kepada kamu untuk menyelesaikan masalah seharian yang serupa dengan cara pengecaman corak.

### 1 Menulis pseudokod

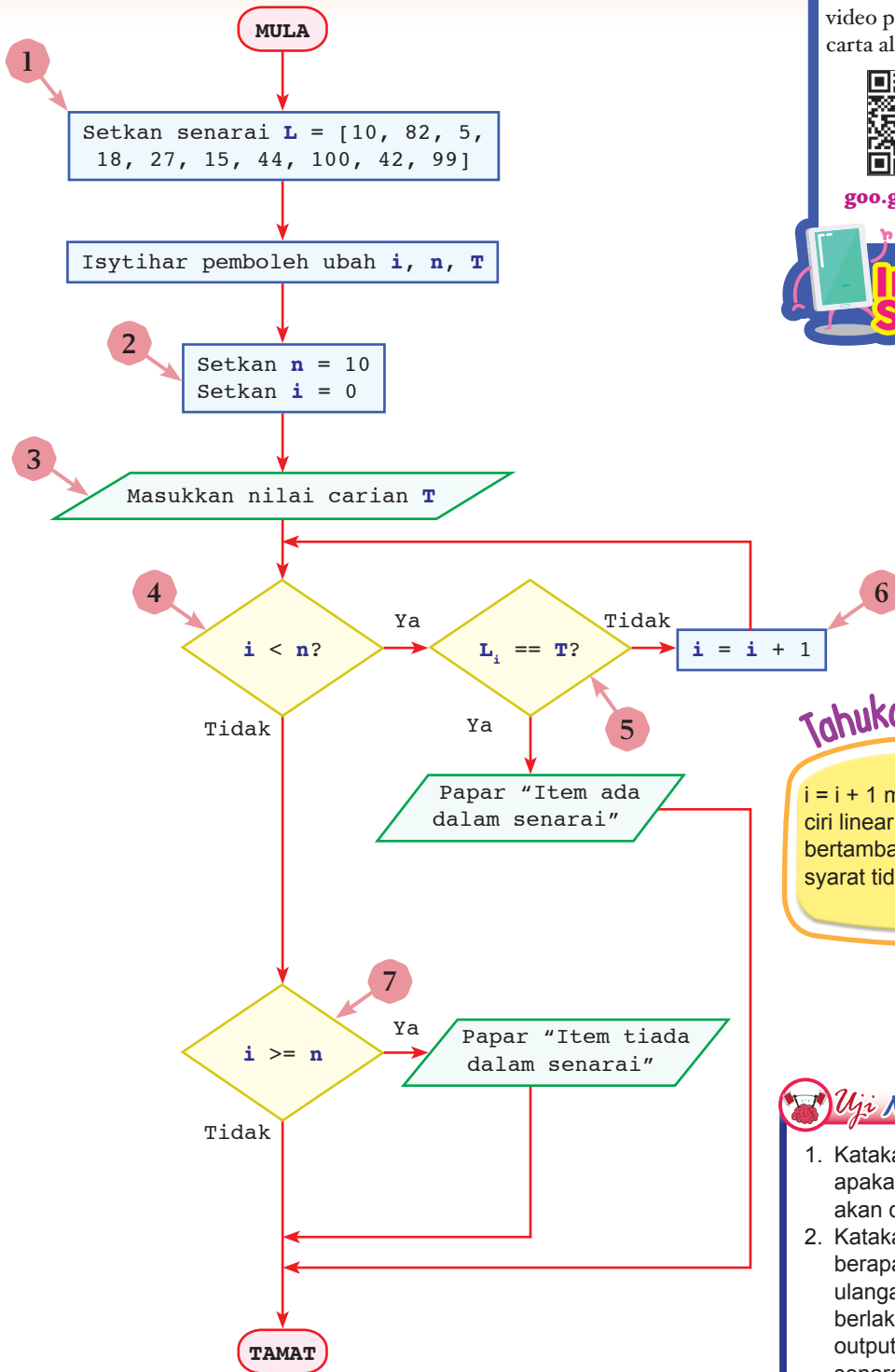
```
1 Mula
2 Setkan senarai L = [10, 82, 5, 18, 27, 15, 44, 100, 42, 99]
3 Isytihar pemboleh ubah i, n, T
4 Setkan n = 10
5 Setkan i = 0
6 Masukkan nilai carian T
7 for i < n
  7.1 Jika  $L_i == T$ 
    7.1.1 Papar "Item ada dalam senarai"
    7.1.2 Langkau ke Langkah 9
  7.2 Jika tidak
    7.2.1 Kira  $i = i + 1$ 
  7.3 Tamat jika
8 Jika  $i \geq n$ 
  8.1 Papar "Item tiada dalam senarai"
  8.2 Tamat jika
9 Tamat
```

Rajah 3.3 Pseudokod bagi *linear search*



Mengapakah kita perlu mempelajari pembinaan algoritma? Adakah kita perlu mengkod semua algoritma *search* dan *sort* ini?

2 Melukis carta alir



Imbas QR code untuk menonton video penerangan carta alir ini.

[goo.gl/9wT4uv](https://goo.gl/9wT4uv)

**Imbas Saya**

**Tahukah Anda?**

$i = i + 1$  menunjukkan ciri linear di mana indeks bertambah 1 selagi syarat tidak dipenuhi.

**Uji Minda 3.6**

1. Katakan  $T = 90$ , apakah output yang akan dipaparkan?
2. Katakan  $T = 44$ , berapakah bilangan ulangan yang akan berlaku sehingga output item dalam senarai dipaparkan?

Rajah 3.4 Carta alir bagi linear search

Simbol	Penerangan
1	Senarai <b>L</b> diumpukkan dengan nilai-nilai [10, 82, 5, 18, 27, 15, 44, 100, 42, 99].
2	<b>n</b> yang mewakili bilangan item dalam senarai diumpukkan dengan nilai 10. Pemboleh ubah <b>i</b> diumpukkan dengan nilai 0, iaitu kedudukan item yang pertama dalam senarai.
3	Pengguna memasukkan satu nombor carian yang diumpukkan kepada pemboleh ubah <b>T</b> .
4	Selagi syarat ( $i < n$ ) benar, carian akan dilakukan pada setiap item dalam senarai.
5	Jika item pada kedudukan tertentu, iaitu $L_i$ didapati sama dengan nilai <b>T</b> , mesej "Item ada dalam senarai" akan dipaparkan dan carian akan dihentikan.
6	Jika tidak, carian akan diteruskan dengan item yang berikutnya, iaitu $i = i + 1$ dan proses ini akan diulang sehingga nilai <b>T</b> dijumpai dalam senarai atau sehingga kesemua item telah disemak.
7	Pada akhir carian, jika $i \geq n$ , ini bermaksud kesemua item dalam senarai telah disemak dan item carian tidak dijumpai. Maka, mesej "Item tiada dalam senarai" dipaparkan dan carian dihentikan.



### Aktiviti 3.3



#### Aktiviti Kumpulan

#### Menghasilkan Pseudokod dan Carta Alir

1. Perhatikan maklumat di bawah.

Senarai nombor: 1.3 3.7 4.8 1.5 2.5

Item carian: 1.5

Kaedah carian: *Linear search*

2. Berdasarkan maklumat di atas, lakukan perbincangan bersama-sama ahli kumpulan anda untuk menghasilkan pseudokod dan lukiskan carta alir bagi *linear search*.
3. Kemudian, catatkan semua hasil perbincangan pada kertas A3.
4. Setiap kumpulan perlu memilih seorang wakil untuk membentangkan hasil kerja kumpulan di hadapan kelas. Ahli kumpulan yang lain perlu menanyakan soalan.
5. Ulang langkah 4 sehingga semua ahli kumpulan selesai membuat pembentangan.

3.1.2(ii)

## Menulis Pseudokod dan Melukis Carta Alir bagi *Binary Search*

Pembelajaran tentang teknik *binary search* adalah penting kerana situasi harian yang dilakukan kadangkala mengaplikasikan *binary search*. Antara situasi yang mengaplikasikan *binary search* ialah carian nombor telefon berdasarkan nama dalam buku panduan telefon, carian makna bagi perkataan tertentu dalam kamus, carian kad rekod pesakit di klinik mengikut nombor kad pengenalan serta carian fail penuntut di kolej.

Algoritma perlu dibina bagi memudahkan proses *binary search*. Berikut langkah-langkah pembinaan algoritma bagi mencari item carian menggunakan *binary search*.

### Contoh 2.8

Langkah-langkah pembinaan pseudokod dan carta alir bagi *binary search* untuk mencari satu abjad.

Carian suatu abjad dalam senarai A hingga Z.

#### Langkah 1

Memahami *binary search* dengan membina algoritma

Rajah 3.5 menunjukkan algoritma bagi *binary search*.

1. Pastikan item-item dalam senarai yang diberi telah diisih mengikut urutan menaik.
2. Lihat item yang berada di tengah senarai.
3. Bandingkan item carian dengan item yang berada di tengah senarai.
4. Jika nilai item carian sama dengan nilai item yang berada di tengah senarai, carian dihentikan.
5. Jika nilai item carian kurang daripada nilai item yang berada di tengah senarai, abaikan item di tengah senarai dan item-item selepasnya. Kemudian, lihat pada senarai yang tinggal.
6. Jika nilai item carian lebih daripada nilai item di tengah senarai, abaikan item di tengah senarai dan item-item sebelumnya. Kemudian, lihat pada senarai yang tinggal.
7. Ulang Langkah 2 hingga Langkah 6 sehingga item carian dijumpai atau apabila carian selesai tanpa menjumpai item carian.

Rajah 3.5 Algoritma proses *binary search*

## Standard Pembelajaran

Murid boleh:

- 3.1.2 Menulis pseudokod dan melukis carta alir yang menunjukkan:
- (ii) *binary search*

Algoritma dalam *binary search*



[goo.gl/VwNFJm](https://goo.gl/VwNFJm)



## Imbas Kembali

Senarai perlu diisih terlebih dahulu sebelum melakukan *binary search*.

## Uji Minda 3.7

Anda berada di sebuah perpustakaan awam. Anda ingin mencari senaskhah buku rujukan yang mempunyai nombor rujukan 334. Terangkan cara untuk anda mendapatkan buku tersebut dengan menggunakan teknik *binary search*.

## Langkah 2

## Menulis pseudokod dan melukis carta alir

Terdapat beberapa pemboleh ubah yang terlibat dalam *binary search* ini.

### Tahukah Anda?

$j$  ialah pemboleh ubah bagi indeks terakhir dalam senarai. Untuk mendapatkan nilai  $j$ , bilangan item dalam senarai perlu ditolak dengan 1 kerana indeks bermula dengan nilai 0.

#### Katakan:

- $L$  ialah senarai yang mengandungi 26 item A hingga Z.
- $n$  ialah pemboleh ubah yang mewakili bilangan item dalam senarai  $L$ .
- $i$  ialah pemboleh ubah yang mewakili kedudukan item dalam senarai.
- $m$  ialah pemboleh ubah yang mewakili kedudukan item pertengahan dalam senarai.
- $b$  ialah pemboleh ubah yang mewakili nilai item carian.
- $j$  ialah  $n - 1$ .

Rajah 3.6 dan Rajah 3.7 menunjukkan pseudokod dan carta alir yang dibina.

### 1 Menulis pseudokod

### Tahukah Anda?

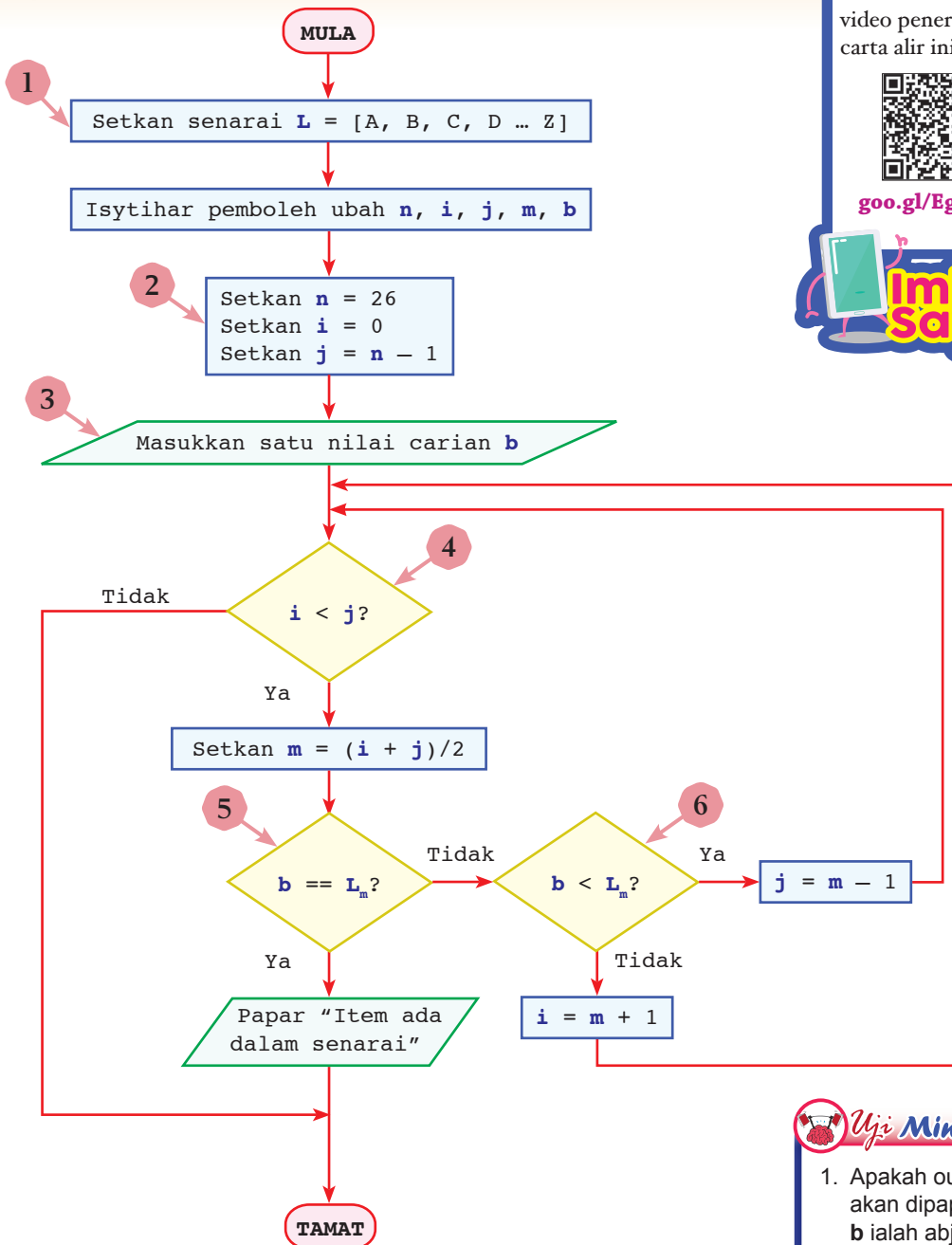
Semasa penulisan pseudokod, kegunaan inden adalah amat penting, terutamanya jika banyak struktur kawalan digunakan.

```
1 Mula
2 Setkan senarai  $L = [A, B, C, D \dots Z]$ 
3 Isytihar pemboleh ubah  $n, i, j, m, b$ 
4 Setkan  $n = 26$ 
5 Setkan  $i = 0$ 
6 Setkan  $j = n - 1$ 
7 Masukkan satu nilai carian  $b$ 
8 while  $i < j$ 
  8.1 Setkan  $m = (i + j)/2$ 
  8.2 Jika  $b == L_m$ 
    8.2.1 Papar "Item ada dalam senarai"
    8.2.2 Keluar gelung
  8.3 Jika tidak
    8.3.1 Jika  $b < L_m$ 
      8.3.1.1 Kira  $j = m - 1$ 
    8.3.2 Jika tidak
      8.3.2.1 Kira  $i = m + 1$ 
    8.3.3 Tamat jika
  8.4 Tamat jika
9 Tamat
```

Rajah 3.6 Pseudokod bagi *binary search*



2 Melukis carta alir



Imbas QR code untuk menonton video penerangan carta alir ini.



[goo.gl/EgjDoM](https://goo.gl/EgjDoM)



Rajah 3.7 Carta alir bagi binary search

Uji Minda 3.8

1. Apakah output yang akan dipaparkan jika **b** ialah abjad Q?
2. Katakan **b** ialah abjad F, berapakah bilangan ulangan yang akan berlaku sehingga output item dalam senarai dipaparkan?

Simbol	Penerangan
1	Senarai <b>L</b> diumpukkan dengan nilai-nilai [A, B, C, hingga Z].
2	Pemboleh ubah <b>i</b> diumpukkan dengan nilai 0, iaitu kedudukan item pertama dalam senarai. Pemboleh ubah <b>n</b> diumpukkan dengan nilai 26 yang mewakili bilangan item dalam senarai. Pemboleh ubah <b>j</b> diumpukkan dengan nilai $(n - 1)$ .
3	Pengguna perlu memasukkan satu item carian, iaitu <b>b</b> .
4	Selagi <b>i</b> kurang daripada <b>j</b> , indeks pertengahan <b>m</b> ditentukan dengan menggunakan formula $m = (i + j)/2$ .
5	Jika $b == L_m$ , mesej "Item ada dalam senarai" akan dipaparkan dan carian akan dihentikan.
6	Jika $b < L_m$ , kedudukan item terakhir dalam senarai akan dikira semula, iaitu $j = m - 1$ . Jika $b > L_m$ , kedudukan item pertama dalam senarai akan dikira semula, iaitu $i = m + 1$ . Proses ini akan berulang sehingga item carian dijumpai atau sehingga semua item dalam senarai telah disemak.

### Aktiviti 3.4 Aktiviti Kumpulan

#### Menghasilkan Pseudokod dan Carta Alir

**PA K-21** Kaedah Pembentangan Hasil Sendiri

1. Perhatikan maklumat berikut.

Senarai nombor: (10) (20) (30) (40) (50) (60) (70)

Item carian: (30)

Kaedah carian: *Binary search*

- Berdasarkan maklumat di atas, lakukan perbincangan bersama-sama ahli kumpulan anda untuk menghasilkan langkah-langkah bagi mencari nilai 30 dalam senarai di atas.
- Tuliskan pseudokod dan lukiskan carta alir yang dihasilkan pada sehelai kertas.
- Setiap kumpulan perlu memilih seorang wakil untuk membentangkan hasil kerja kumpulan di hadapan kelas.



## Aktiviti 3.5

### Aktiviti Berpasangan

#### Permainan Meneka Nombor

1. Pilih satu nombor daripada 1 hingga 100 dan tuliskan di atas sehelai kertas. Jangan dedahkan kepada rakan anda.
2. Minta rakan anda meneka nombor yang anda tulis menggunakan *linear search*.
3. Catatkan bilangan tekaan rakan anda setelah nombor sebenar berjaya diteka.
4. Sekarang, rakan anda pula menulis satu nombor di atas sehelai kertas dan anda akan meneka nombor tersebut menggunakan *binary search*.
5. Catatkan bilangan tekaan anda setelah nombor sebenar berjaya diteka.
6. Apakah kesimpulan yang dapat dibuat terhadap kedua-dua carian yang digunakan?

### 3.1.3(i)

#### Menulis Pseudokod dan Melukis Carta Alir bagi *Bubble Sort*

*Sort* merupakan satu tugas (*task*) yang sering digunakan untuk menyelesaikan masalah sama ada dalam pengaturcaraan atau dalam kehidupan seharian. *Sort* dalam bidang pengkomputeran merujuk kepada penyusunan data seperti nama, nombor, rekod dan fail dalam tertib tertentu. Data yang telah diisih menjadikan carian lebih cekap, percantuman lebih efisien dan memudahkan pemrosesan data dalam tertib yang dikehendaki. Dalam set data yang besar, adalah lebih efisien untuk menghapuskan atau mencantumkan data duplikasi. Apabila data telah diisih, pengesanan data yang berulang lebih mudah untuk dihapuskan atau dicantumkan.

*Bubble sort* boleh diaplikasikan dalam pelbagai jenis situasi seperti menyenaraikan markah-markah yang diperoleh murid mengikut urutan menaik atau menurun. Sebagai contoh, markah murid yang paling tinggi akan sentiasa bergerak menuju ke permulaan senarai diikuti dengan markah kedua tertinggi dan seterusnya. Selain itu, *bubble sort* juga boleh digunakan untuk mengisih umur peserta-peserta yang mengambil bahagian dalam acara larian amal supaya peserta-peserta boleh dikategorikan mengikut peringkat seperti peringkat remaja, belia dan warga emas.

### Standard Pembelajaran

Murid boleh:

- 3.1.3** Menulis pseudokod dan melukis carta alir yang menunjukkan:
- (i) *bubble sort*

Algoritma dalam  
*bubble sort*



[goo.gl/5xjBAH](https://goo.gl/5xjBAH)



## Tahukah Anda?

Item-item dalam senarai akan saling bertukar kedudukan sehingga item-item berada pada urutan yang betul seperti buih yang lebih ringan akan terapung di permukaan air. *Bubble sort* mendapat nama daripada konsep ini.

## Contoh 3.9

Langkah-langkah pembinaan pseudokod dan carta alir bagi *bubble sort* daripada satu senarai nombor yang rawak.

Semasa hari sukan sekolah, acara lempar cakera telah dipertandingkan. 10 orang murid daripada empat buah rumah sukan layak untuk menyertai peringkat akhir acara lempar cakera itu. Guru penyelaras, Cikgu Azmi ingin mengisih senarai peserta acara lempar cakera mengikut jarak lontaran dalam urutan menurun untuk menentukan pemenang bagi acara tersebut. Rajah 3.8 menunjukkan jarak lontaran (dalam m) bagi 10 orang peserta acara lempar cakera.

28.2	26	26.5	27.2	30.4	27	31.4	29.1	27.4	27.8
------	----	------	------	------	----	------	------	------	------

**Rajah 3.8** Jarak lontaran (dalam m) peserta acara lempar cakera

## Langkah 1

Memahami proses *bubble sort* dengan membina algoritma



## Uji Minda 3.9

Berdasarkan Contoh 3.9, apakah yang akan berlaku jika terdapat dua orang murid yang memperoleh jarak lontaran yang sama?

Rajah 3.9 menunjukkan algoritma bagi *bubble sort*.

1. Semak item pertama dengan item kedua dalam senarai.
2. Bandingkan kedua-dua item tersebut.
3. Jika kedua-dua item tersebut tidak mengikut urutan, tukarkan kedudukan kedua-duanya.
4. Ulang langkah 2 hingga langkah 3 dengan item kedua dan item ketiga, kemudian item ketiga dengan item keempat dan seterusnya sehingga ke item terakhir.
5. Semak senarai untuk melihat sama ada semua item berada pada kedudukan yang betul.
6. Ulang langkah 1 hingga 5 sehingga semua item dalam senarai telah diisih mengikut urutan.

**Rajah 3.9** Algoritma proses *bubble sort*

Perhatikan langkah 4 dan langkah 6 pada algoritma dalam Rajah 3.9. Kedua-dua langkah ini memerlukan gelung kerana item-item bersebelahan akan sentiasa dibandingkan sehingga dua item terakhir dan item-item dalam senarai akan sentiasa diisih sehingga kesemua item berada dalam urutan yang betul.

## Langkah 2

## Menulis pseudokod dan melukis carta alir

Pemboleh ubah yang diperlukan semasa menulis pseudokod dan melukis carta alir adalah seperti berikut:

## Katakan:

- **L** ialah senarai yang mengandungi 10 item dengan nilai-nilai  $L_0 = 28.2, L_1 = 26, \dots, L_{n-1} = 27.8$ .
- **n** ialah pemboleh ubah yang mewakili bilangan item dalam senarai **L**.
- **i** ialah pemboleh ubah yang mewakili bilangan lelaran yang berlaku dalam gelung.
- **j** ialah pemboleh ubah yang mewakili kedudukan item dalam senarai **L**.
- **temp** ialah pemboleh ubah yang menyimpan data secara sementara semasa membuat pertukaran kedudukan item.

Rajah 3.10 dan Rajah 3.11 menunjukkan pseudokod dan carta alir yang dibina.

## 1 Menulis pseudokod

```

1  Mula
2  Setkan senarai L = [28.2, 26, 26.5, 27.2,
30.4, 27, 31.4, 29.1, 27.4, 27.8]
3  Isytihar pemboleh ubah i, j, n, temp
4  Setkan n = 10
5  Setkan i = 0
6  Setkan temp = 0
7  for i < n - 1
7.1 Setkan j = 0
7.2 for j < n - i - 1
7.2.1 Jika Lj > Lj+1
7.2.1.1 temp = Lj
7.2.1.2 Lj = Lj+1
7.2.1.3 Lj+1 = temp
7.2.2 Tamat jika
7.2.3 Kira j = j + 1
7.3 Kira i = i + 1
8  Papar senarai
9  Tamat

```

Pemboleh ubah temp diambil daripada perkataan *temporary*

## Uji Minda 3.10

Anda mempunyai segelas susu dan secawan kopi. Bagaimanakah cara anda menukar air di dalam gelas dan cawan itu supaya gelas mengandungi kopi dan cawan mengandungi susu?

Rajah 3.10 Pseudokod bagi *bubble sort*

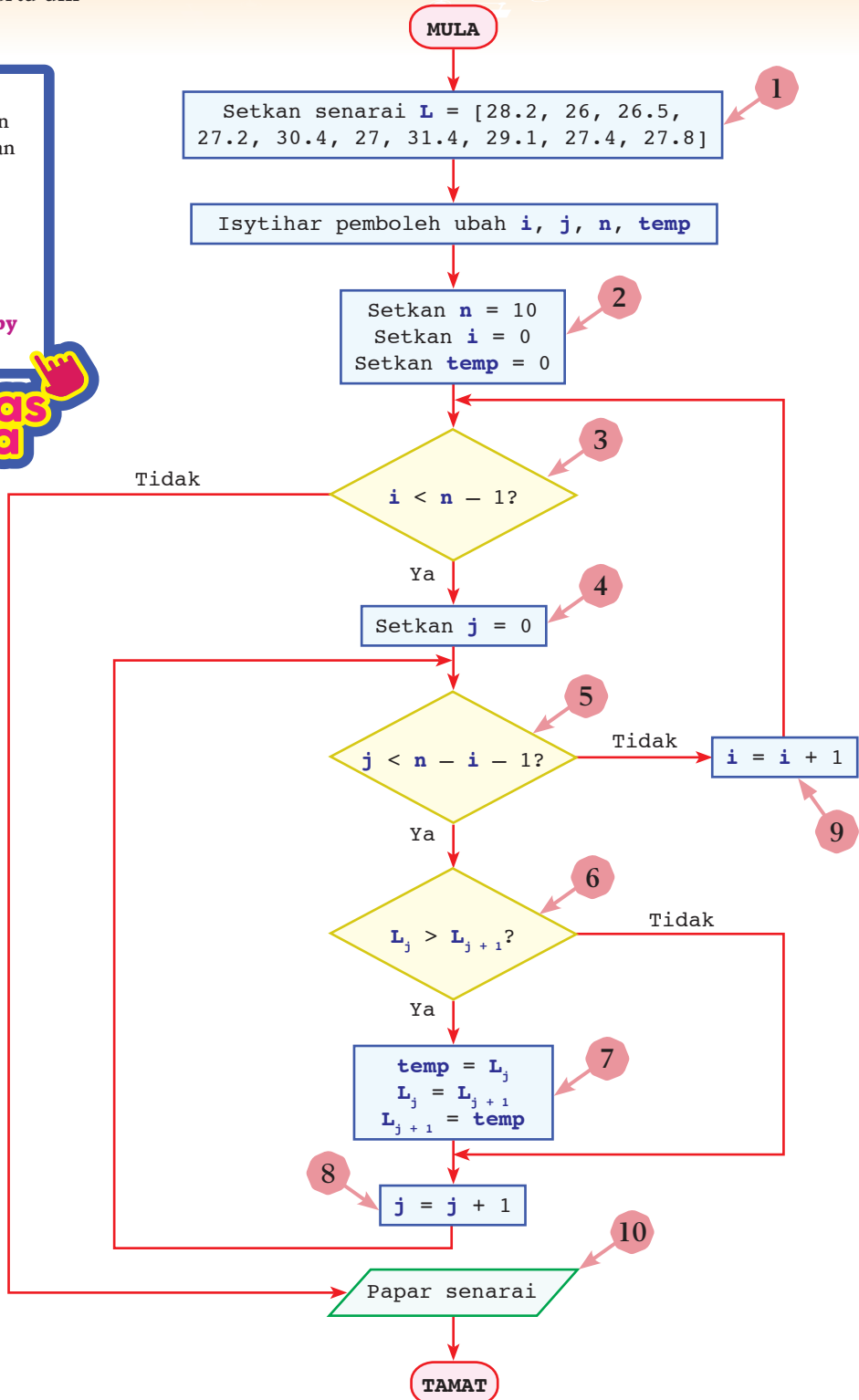
## 2 Melukis carta alir

Imbas QR code untuk menonton video penerangan carta alir ini.



[goo.gl/87kCby](https://goo.gl/87kCby)

Imbas Saya



Rajah 3.11 Carta alir bagi bubble sort

Simbol	Penerangan
1	Senarai <b>L</b> diumpukkan dengan nilai-nilai [28.2, 26, 26.5, 27.2, 30.4, 27, 31.4, 29.1, 27.4, 27.8].
2	Pemboleh ubah <b>n</b> diumpukkan dengan nilai 10 yang mewakili bilangan item dalam senarai. Pemboleh ubah <b>i</b> diumpukkan dengan nilai 0 yang mewakili bilangan lelaran pertama dalam gelung. Pemboleh ubah <b>temp</b> diumpukkan dengan nilai 0 yang mewakili data untuk disimpan secara sementara semasa membuat pertukaran kedudukan item.
3	Gelung <i>for</i> digunakan untuk menentukan bilangan lelaran yang berlaku.
4	Pemboleh ubah <b>j</b> diumpukkan dengan nilai 0 yang mewakili kedudukan item pertama dalam senarai.
5	Gelung <i>for</i> digunakan untuk memastikan setiap pasangan item bersebelahan dibandingkan, bermula dengan $L_0$ dan $L_1$ , $L_1$ dan $L_2$ , dan seterusnya.
6	Jika $L_j$ lebih besar daripada $L_{j+1}$ , maka kedudukan kedua-dua item akan ditukar.
7	Semasa proses penukaran kedudukan, satu pemboleh ubah <b>temp</b> digunakan untuk menukar kedudukan dua item tersebut tanpa perlu menjadikan nilai kedua-dua item itu sama.
8	Gelung <i>for</i> akan sentiasa berulang sehingga item-item bersebelahan dibandingkan.
9	Gelung <i>for</i> akan sentiasa berulang sehingga semua item telah diisih mengikut urutan.
10	Jika tiada lagi penukaran kedudukan berlaku, item-item dalam senarai yang telah diisih akan dipaparkan.



## Aktiviti 3.6



### Aktiviti Kumpulan

#### Menghasilkan Pseudokod dan Carta Alir bagi *Bubble Sort*



Kaedah *Shout Out & Gallery Walk*

- Rajah di bawah menunjukkan satu pseudokod bagi *bubble sort* yang tidak lengkap.

```
Mula
Setkan senarai = [ ]
Isytihar pemboleh ubah i, k, temp
Setkan k = bilangan item dalam senarai
for i < n - 1
.
.
Tamat
```

- Dalam kumpulan, murid digalakkan untuk memberikan idea dan cadangan secara lisan semasa sesi sumbang saran.
- Seorang ahli kumpulan perlu mencatatkan semua idea dan cadangan yang diberikan.
- Berdasarkan hasil yang diperolehi, lakukan perbincangan bagi melengkapkan pseudokod yang diberikan.
- Kemudian, lukiskan carta alir berdasarkan pseudokod yang telah dilengkapkan.
- Setiap kumpulan harus bergerak ke kumpulan lain untuk membuat penilaian dengan memberikan komen, pandangan serta persoalan bagi hasil kerja yang disediakan oleh kumpulan lain.

## Standard Pembelajaran

Murid boleh:

- 3.1.3** Menulis pseudokod dan melukis carta alir yang menunjukkan:  
(ii) *bucket sort*

**3.1.3(ii)**

## Menulis Pseudokod dan Melukis Carta Alir bagi *Bucket Sort*

*Bucket sort* ialah suatu isihan yang menyusun item-item dalam suatu senarai mengikut kategori. Item-item akan dimasukkan ke dalam baldi mengikut kategori yang ditetapkan. Kemudian, item-item di dalam setiap baldi akan diisih mengikut urutan menaik atau menurun. Item-item di dalam setiap baldi yang telah diisih akan dicantumkan semula ke dalam senarai.

### Contoh 3.10

Langkah-langkah pembinaan pseudokod dan carta alir bagi *bucket sort* daripada satu senarai nama yang rawak.



Qistina ialah Setiausaha Rumah Sukan Seri Usaha di sekolahnya. Dia dikehendaki mengisih senarai nama ahli jawatankuasa rumah sukan itu mengikut urutan abjad. Ahli jawatankuasa Rumah Sukan Seri Usaha terdiri daripada 15 orang. Qistina mengaplikasikan teknik *bucket sort* untuk mengisih senarai itu.

#### Langkah 1

Memahami proses *bucket sort* dengan membina algoritma

Rajah 3.12 menunjukkan algoritma bagi *bucket sort*.

1. Wujudkan beberapa buah baldi untuk menyimpan item-item daripada senarai secara sementara.
2. Tentukan julat bagi setiap baldi supaya item-item dapat dimasukkan ke dalam baldi.
3. Masukkan setiap item secara sementara ke dalam baldi berdasarkan julat yang ditetapkan.
4. Isih item-item di dalam setiap baldi.
5. Setelah item-item dalam semua baldi diisih, cantumkan item-item tersebut dan masukkan semula ke dalam senarai.

#### Rajah 3.12 Algoritma proses *bucket sort*

Dalam algoritma *bucket sort*, item-item dalam senarai akan diisih sebanyak dua kali supaya item-item berada dalam urutan yang dikehendaki. Isihan pertama berlaku apabila item-item dalam senarai diisih mengikut julat pada baldi. Isihan kedua berlaku apabila item-item dalam setiap baldi diisih mengikut urutan.

### Tahukah Anda?

Penggunaan *bucket sort* amat berguna untuk mengisihkan satu senarai item yang amat besar. *Bucket sort* perlu digunakan bersama-sama dengan isihan lain untuk melakukan pengisihan item-item dalam setiap baldi.



Daripada algoritma dalam Rajah 3.12, beberapa perkara perlu dikenal pasti terlebih dahulu untuk memudahkan proses menulis pseudokod dan melukis carta alir. Pemboleh ubah yang perlu diketahui adalah seperti berikut:

**Katakan:**

- Senarai **L** mengandungi item-item yang perlu diisih, iaitu senarai nama ahli.
- **i** ialah pemboleh ubah yang mewakili kedudukan item dalam senarai **L**.
- **n** ialah pemboleh ubah yang mewakili bilangan item dalam senarai **L**.
- **nB** ialah pemboleh ubah yang mewakili bilangan baldi yang diperlukan.
- **j** ialah pemboleh ubah yang mewakili kedudukan baldi.

**Langkah 2**

**Menulis pseudokod dan melukis carta alir**

**1** Menulis pseudokod

```

1  Mula
2  Setkan senarai L = [Hilmi, Anas, Siti, ..., Maisara]
3  Setkan senaraiBaharu[ ]
4  Isytihar pemboleh ubah i, j, n, nB
5  Setkan n = 15
6  Setkan nB = bilangan baldi
7  Wujudkan baldi kosong
8  Masukkan item ke dalam baldi
9  Setkan i = 0
10 for i < n
    10.1 Masukkan senarai[i] ke dalam Baldi
    10.2 Kira i = i + 1
11 Setkan j = 0
12 for j < nB
    12.1 Jika Baldi[j] kosong
        12.1.1 Kira j = j + 1
    12.2 Jika tidak
        12.2.1 Isih item
        12.2.2 Kira j = j + 1
13 Cantum dan masukkan item-item ke dalam senaraiBaharu[ ]
14 Papar senaraiBaharu[ ]
15 Tamat

```

Gunakan *bubble sort* atau jenis isihan lain yang sesuai untuk mengisih item di dalam baldi.

**Rajah 3.13** Pseudokod bagi *bucket sort*

**Tahukah Anda** ?

Bilangan baldi, nB boleh ditentukan dengan menggunakan formula:

$$nB = \text{had atas } \sqrt{n}$$

Contohnya, kiraan bilangan baldi bagi 20 item dalam senarai ialah had atas  $\sqrt{20} = 4.472$ . Maka, bilangan baldi yang diperlukan ialah 5.

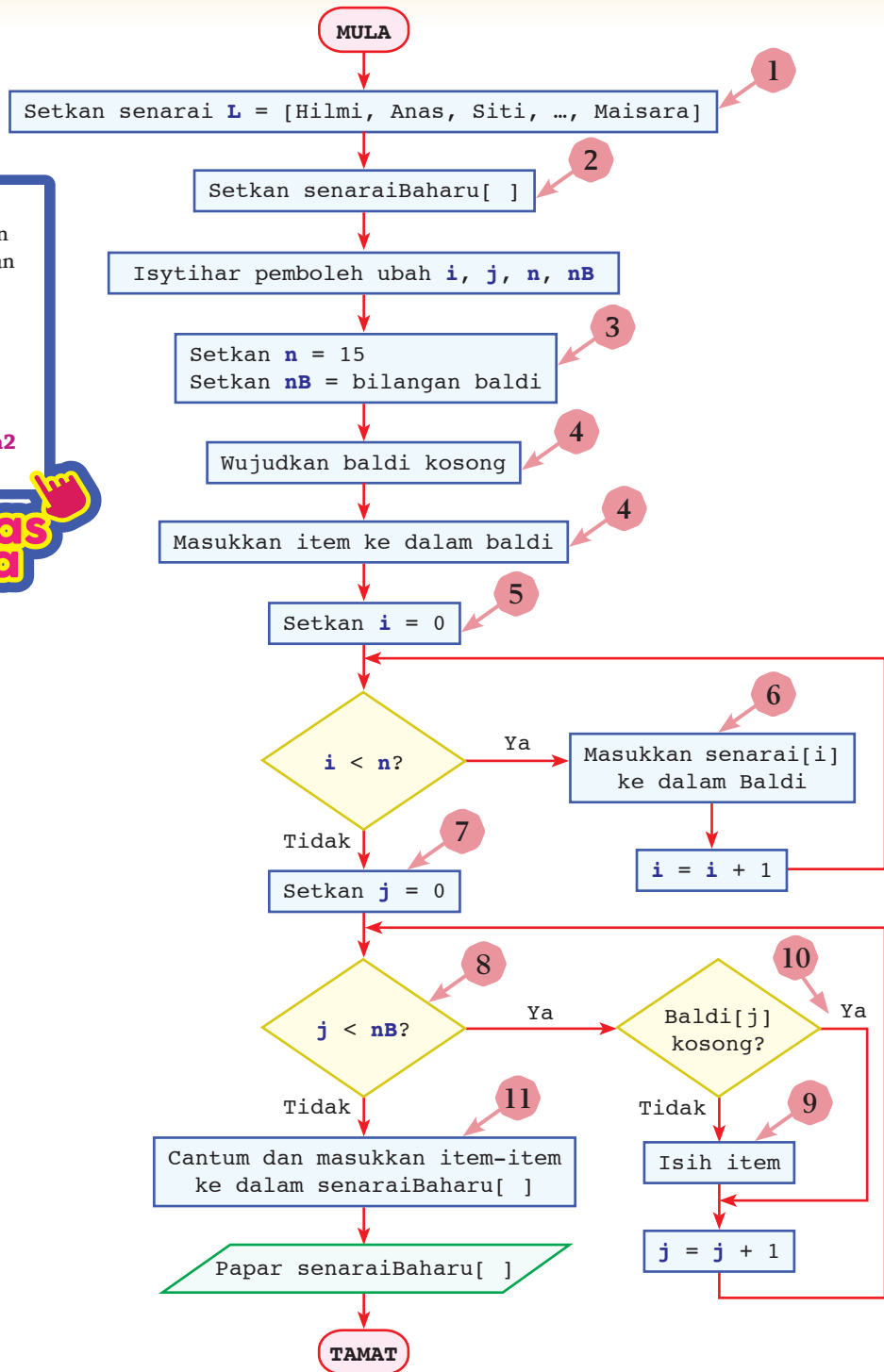
## 2 Melukis carta alir

Imbas QR code untuk menonton video penerangan carta alir ini.



[goo.gl/bE1rn2](https://goo.gl/bE1rn2)

**Imbas Saya**



Rajah 3.14 Carta alir bagi *bucket sort*

Simbol	Penerangan
1	Senarai <b>L</b> diumpukkan kepada item-item dalam senarai [Hilmi, Anas, Siti, ..., Maisara]
2	senaraiBaharu[ ] diumpukkan dengan item-item dalam senarai yang telah diisih.
3	Pemboleh ubah <b>n</b> diumpukkan dengan nilai 15 yang mewakili bilangan item dalam senarai. Pemboleh ubah <b>nB</b> diumpukkan dengan bilangan baldi yang diperlukan.
4	Wujudkan baldi kosong supaya item-item dapat dimasukkan ke dalam baldi.
5	Pemboleh ubah <b>i</b> diumpukkan dengan nilai 0, yang mewakili kedudukan item pertama dalam senarai.
6	Untuk <b>i</b> kurang daripada <b>n</b> , item-item akan dimasukkan ke dalam baldi.
7	Jika <b>i</b> lebih daripada <b>n</b> , pemboleh ubah <b>j</b> diumpukkan dengan nilai 0, yang mewakili kedudukan baldi pertama.
8	Untuk <b>j</b> kurang daripada, <b>nB</b> , baldi pada kedudukan <b>j</b> , Baldi[j] akan disemak sama ada kosong atau tidak.
9	Jika Baldi[j] tidak kosong, item-item di dalam baldi akan diisih hingga baldi terakhir.
10	Jika Baldi[j] kosong, item-item di dalam baldi seterusnya akan disemak dan diisih.
11	Jika <b>j</b> lebih daripada <b>nB</b> , item-item dalam setiap baldi akan dicantumkan dan dimasukkan ke dalam suatu senaraiBaharu[ ]. Senarai tersebut akan dipaparkan dan atur cara tamat.



## Aktiviti 3.7

### Aktiviti Kumpulan

Menghasilkan Pseudokod dan Carta Alir bagi *Bucket Sort*



Kaedah  
Gallery Walk

1. Guru memberikan setiap kumpulan senarai berikut.



2. Bincangkan bersama-sama ahli kumpulan tentang langkah-langkah untuk mengisih senarai di atas mengikut urutan menurun menggunakan *bucket sort* dan tuliskan senarai item yang telah diisih.
3. Tuliskan pseudokod dan lukiskan carta alir bagi isihan yang telah dilakukan.
4. Setiap kumpulan bergerak ke kumpulan lain untuk membuat penilaian serta memberikan komen tentang hasil kerja yang disediakan.

## Standard Pembelajaran

Murid boleh:

- 3.1.4** Mengesan dan membaiki ralat dalam pseudokod dan melukis carta alir bagi penyelesaian masalah yang melibatkan:
- search*
  - sort*

3.1.4

## Mengesan dan Membaiki Ralat dalam Pseudokod dan Carta Alir

Apabila terdapat ralat semasa menulis pseudokod dan melukis carta alir, sesuatu atur cara yang dibangunkan tidak dapat menghasilkan output yang dikehendaki. Sebelum atur cara diuji, pseudokod dan carta alir yang telah dibina perlu disemak terlebih dahulu supaya semua ralat dapat dikesan dan dibaiki. Jenis-jenis teknik semakan, iaitu semakan meja (*desk checking*) dan semakan langkah demi langkah (*stepping through*) boleh digunakan untuk mengesan ralat.

### (i) Mengesan dan Membaiki Ralat bagi Penyelesaian Masalah yang Melibatkan *Search*

Penerangan tentang teknik semakan meja.



[goo.gl/XPRUs9](http://goo.gl/XPRUs9)

Penerangan tentang teknik langkah demi langkah.



[goo.gl/wJg7U4](http://goo.gl/wJg7U4)

Imbas Saya



### Uji Minda 3.11

Pada pendapat anda, jenis ralat yang manakah sukar dibaiki? Berikan alasan anda.

### Contoh 3.11

Pengesanan dan membaiki ralat dalam satu algoritma *search*.

Melanie diberikan satu tugas oleh gurunya untuk membina satu atur cara yang mencari dan memaparkan nilai 8 dalam senarai berikut.

3	7	2	9	6	4	1	5	8	10	11	13
---	---	---	---	---	---	---	---	---	----	----	----

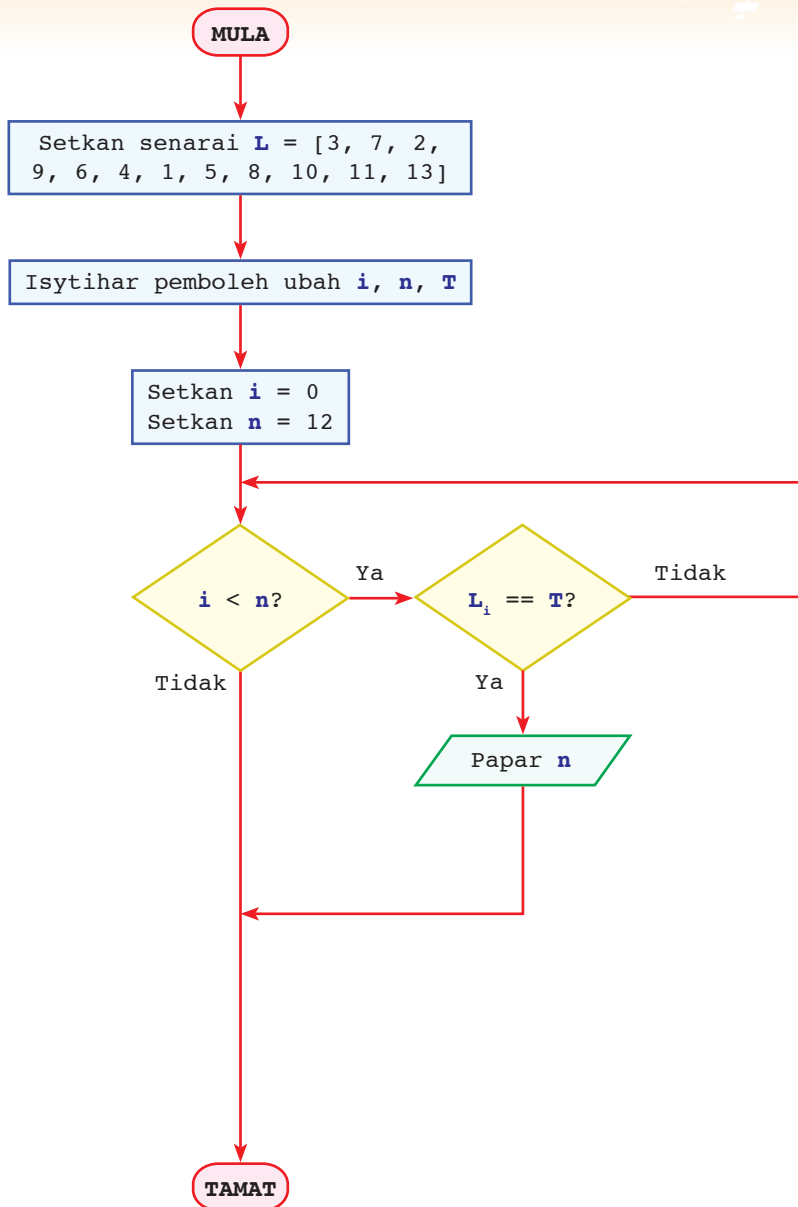
Rajah 3.15 dan Rajah 3.16 menunjukkan pseudokod dan carta alir yang dihasilkan oleh Melanie. Apabila atur cara disemak, Melanie mendapati bahawa atur cara hanya menyemak nilai pertama dalam senarai dan seterusnya semakan berulang tanpa memaparkan sebarang hasil.

Pseudokod yang dihasilkan oleh Melanie

```
1 Mula
2 Setkan senarai L = [3, 7, 2, 9, 6, 4, 1, 5, 8, 10, 11, 13]
3 Isytihar pemboleh ubah i, n, T
4 Setkan i = 0
5 Setkan n = 12
6 for i < n
  6.1 Jika Li == T
    6.1.1 Papar n
    6.1.2 Keluar gelung
  6.2 Jika tidak
    6.2.1 Ulang Langkah 6
  6.3 Tamat jika
7 Tamat
```

Rajah 3.15 Pseudokod yang dihasilkan oleh Melanie

Carta alir yang dihasilkan oleh Melanie



Rajah 3.16 Carta alir yang dihasilkan oleh Melanie

Apabila teknik langkah demi langkah dilakukan, terdapat tiga ralat yang dijumpai dalam pseudokod dan carta alir. Rajah 3.17 dan Rajah 3.18 menunjukkan ralat-ralat yang terdapat pada pseudokod dan carta alir yang dibina oleh Melanie dan Rajah 3.19 dan Rajah 3.20 menunjukkan pseudokod dan carta alir yang telah dibaiki.

Imbas QR code untuk menonton video penerangan carta alir ini.



[goo.gl/is6mqU](https://goo.gl/is6mqU)



**Uji Minda 3.12**

Apakah jenis ralat yang dilakukan oleh Melanie dalam pseudokod dan carta alir ini?

## Pengesanan ralat pada pseudokod

```

1 Mula
2 Setkan senarai L = [3, 7, 2, 9, 6, 4, 1, 5, 8, 10, 11, 13]
3 Isytihar pemboleh ubah i, n, T
4 Setkan i = 0
5 Setkan n = 12
6 for i < n
6.1 Jika Li == T
6.1.1 Papar n
6.1.2 Keluar gelung
6.2 Jika tidak
6.2.1 Ulang Langkah 6
6.3 Tamat jika
7 Tamat
    
```

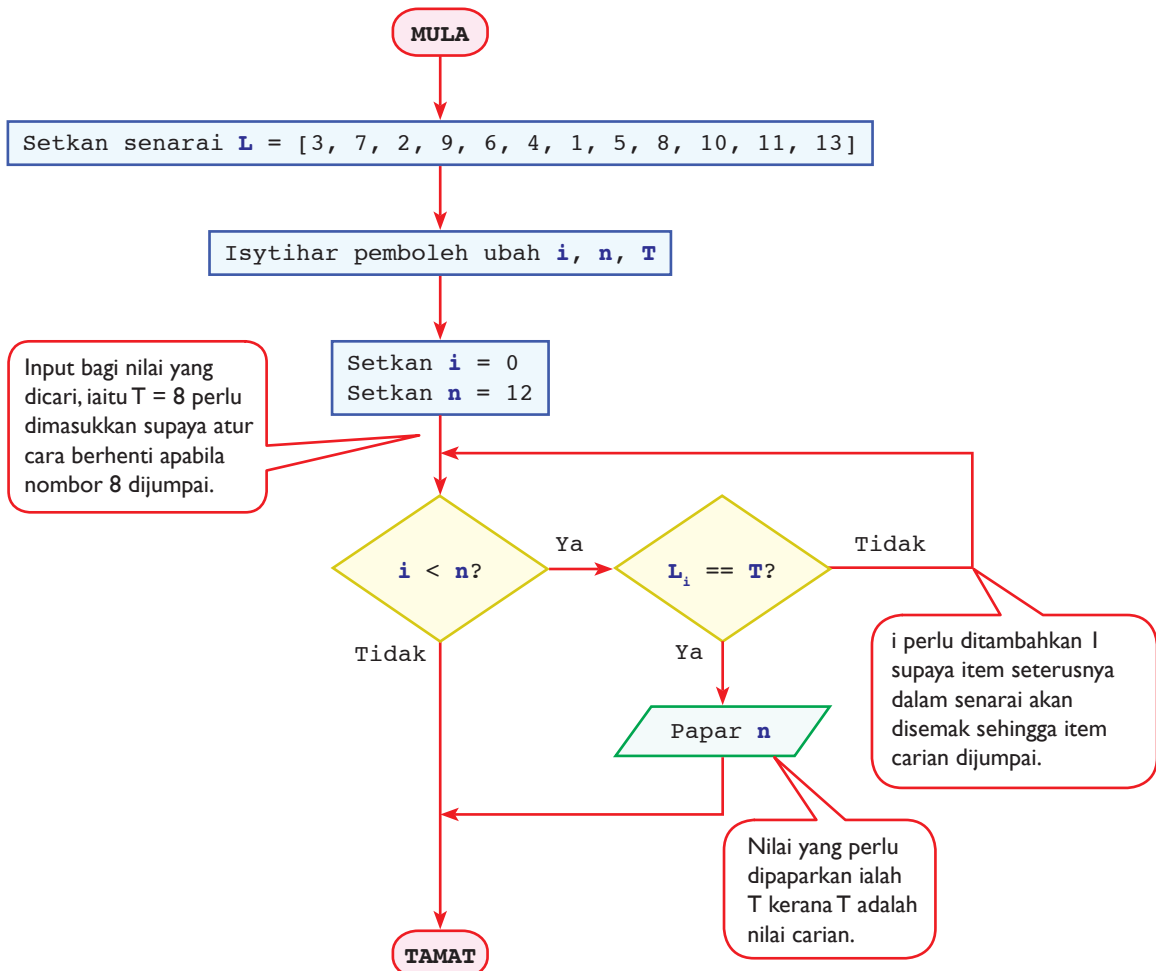
Input bagi nilai yang dicari, iaitu T = 8 perlu dimasukkan supaya atur cara berhenti apabila nombor 8 dijumpai.

Nilai yang perlu dipaparkan ialah T kerana T adalah nilai carian.

i perlu ditambahkan 1 supaya item seterusnya dalam senarai akan disemak sehingga item carian dijumpai.

Rajah 3.17 Pengesanan ralat pada pseudokod

## Pengesanan ralat pada carta alir



Rajah 3.18 Pengesanan ralat pada carta alir

## Pseudokod yang telah dibaiki

```

1 Mula
2 Setkan senarai  $L = [3, 7, 2, 9, 6, 4, 1, 5, 8, 10, 11, 13]$ 
3 Isytihar pemboleh ubah  $i, n, T$ 
4 Setkan  $i = 0$ 
5 Setkan  $n = 12$ 
6 Masukkan satu nilai carian  $T$ 
7 for  $i < n$ 
  7.1 Jika  $L_i == T$ 
    7.1.1 Papar  $T$ 
    7.1.2 Keluar gelung
  7.2 Jika tidak
    7.2.1 Kira  $i = i + 1$ 
  7.3 Tamat jika
8 Tamat

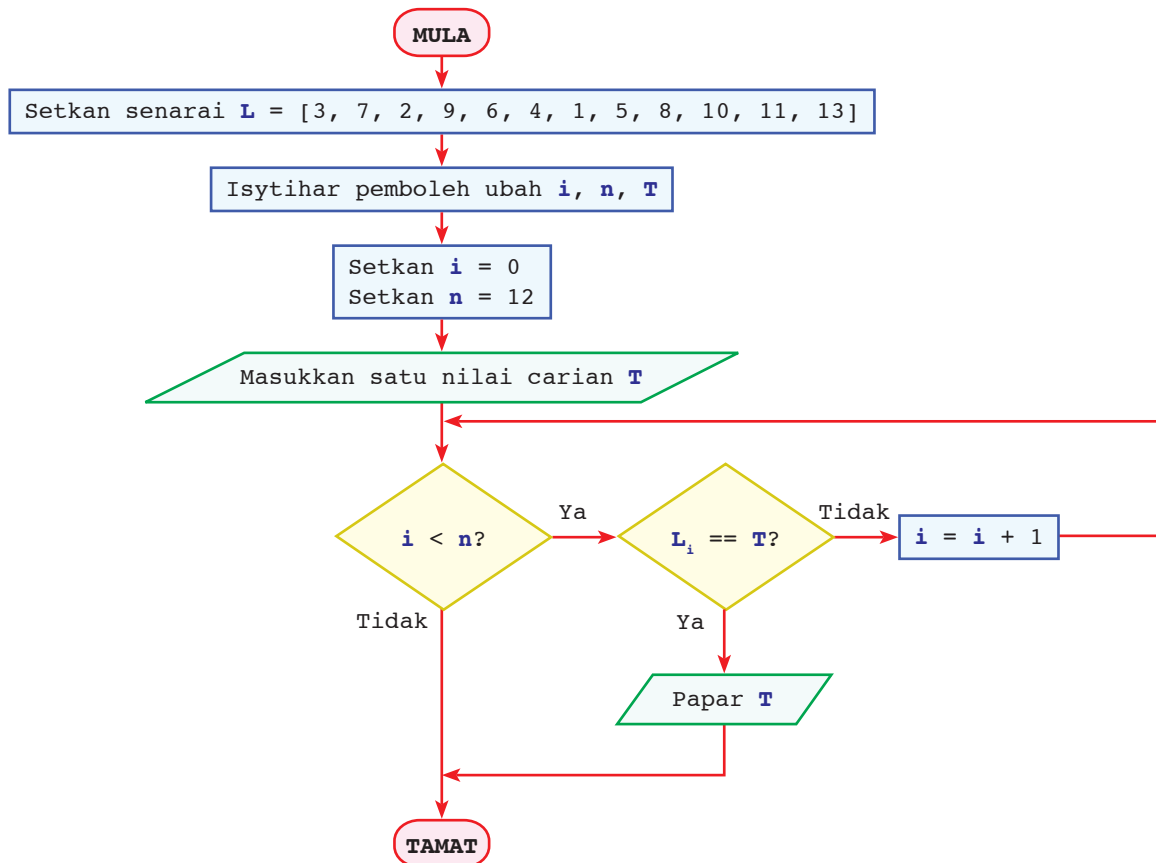
```


**Uji Minda 3.13**

Adakah teknik semakan meja sesuai digunakan untuk mengesan ralat dalam Contoh 3.11? Jelaskan.

Rajah 3.19 Pseudokod Melanie yang telah dibaiki

## Carta alir yang telah dibaiki



Rajah 3.20 Carta alir Melanie yang telah dibaiki

Dalam Contoh 3.11, apakah yang akan berlaku sekiranya syarat pada langkah 7 dalam pseudokod yang telah dibaiki ditukar kepada “for i <= n”? Jelaskan.

## **Aktiviti 3.8**

**Aktiviti Berpasangan**

### Mengesan dan Membaiki Ralat



Kaedah  
*Think-Pair-Share*

1. Kaji pernyataan masalah di bawah.

Dayang Arni ingin mencari nilai 58 dalam senarai nombor [1, 6, 10, 18, 20, 40, 58, 67, 74, 87, 99]. Dayang Arni menulis satu pseudokod seperti di bawah.

```

1  Mula
2  Setkan senarai L = [1, 6, 10, 18, 20, 40, 58, 67, 74, 87, 99]
3  Isytihar pemboleh ubah indeksAwal, indeksAkhir, nilaiTengah
4  Setkan indeksAwal = 0
5  Setkan indeksAkhir = 10
6  while indeksAwal <= indeksAkhir
6.1 indeksTengah = (indeksAwal + indeksAkhir)//2
6.2 Jika nilaiTengah = nilaiCari
6.2.1 Papar nilaiCari
6.2.2 Keluar gelung
6.3 Jika nilaiTengah < nilaiCari
6.3.1 indeksAwal = indeksTengah + 1
6.4 Jika nilaiTengah > nilaiCari
6.4.1 indeksAkhir = indeksTengah - 1
6.5 Tamat jika
7  Tamat
    
```

2. Pseudokod yang dihasilkan oleh Dayang Arni mempunyai ralat.
3. Gunakan teknik semakan yang sesuai untuk mengesan dan membaiki ralat yang terdapat dalam pseudokod. Tuliskan hasil carian anda pada kertas.
4. Kumpulkan hasil carian dan kongsikan bersama-sama rakan sepasangan anda. Lakukan perbincangan untuk menghasilkan satu pseudokod yang telah dibaiki.
5. Lukiskan satu carta alir berdasarkan pseudokod yang telah dibaiki.



**(ii) Mengesan dan Membaiki Ralat bagi Penyelesaian Masalah yang Melibatkan Sort**

**Contoh 3.12** Pengesanan dan membaiki ralat dalam satu algoritma *sort*.

Soon Huat ingin membina satu algoritma untuk mengisih satu senarai integer dalam urutan menaik menggunakan *bubble sort*. Senarai integer yang ada serta output yang dikehendaki oleh Soon Huat adalah seperti berikut:

**Senarai integer**

11, 5, 77, 8, 44, 7

**Output yang dikehendaki**

5, 7, 8, 11, 44, 77

Apabila algoritma yang dibina oleh Soon Huat disemak, didapati output yang diperolehi adalah seperti berikut:

**Output yang diperolehi**

77, 44, 11, 8, 7, 5

Bantu Soon Huat mengesan ralat pada pseudokod dan carta alir yang dihasilkannya.

**Pseudokod yang dihasilkan oleh Soon Huat**

```

1  Mula
2  Setkan senarai  $L = [11, 5, 77, 8, 44, 7]$ 
3  Isytihar pemboleh ubah  $n, i, j, temp$ 
4  Setkan  $n = 6$ 
5  Setkan  $i = 0$ 
6  Setkan  $temp = 0$ 
7  for  $i < n - 1$ 
    7.1 Setkan  $j = 0$ 
    7.2 for  $j < n - i - 1$ 
        7.2.1 Jika  $L_j < L_{j+1}$ 
            7.2.1.1  $temp = L_j$ 
            7.2.1.2  $L_j = L_{j+1}$ 
            7.2.1.3  $L_{j+1} = temp$ 
        7.2.2 Tamat jika
        7.2.3 Kira  $j = j + 1$ 
    7.3 Kira  $i = i + 1$ 
8  Papar senarai
9  Tamat

```

**Rajah 3.21** Pseudokod bagi mengisih senarai integer dalam urutan menaik

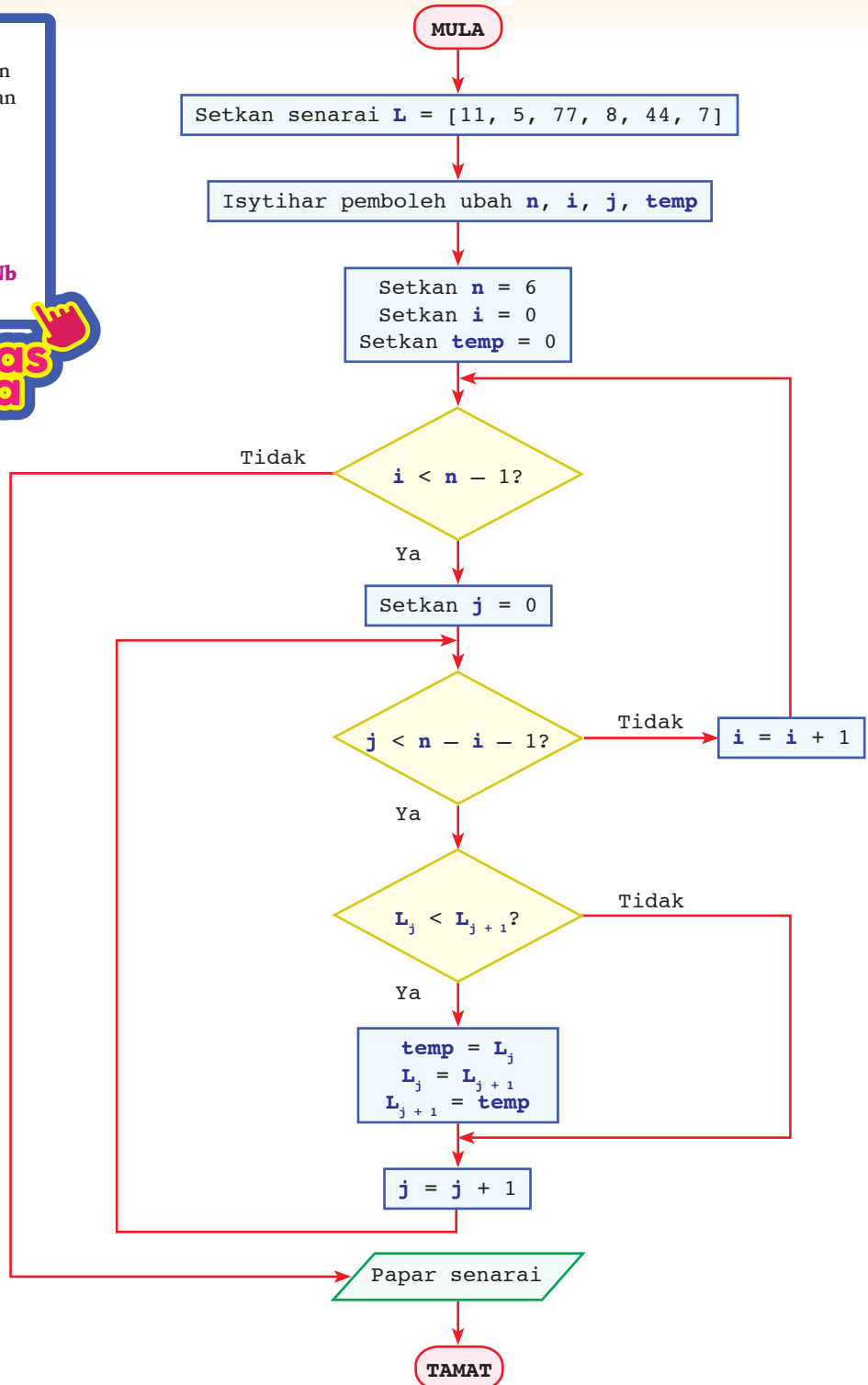
Carta alir yang dihasilkan oleh Soon Huat

Imbas QR code untuk menonton video penerangan carta alir ini.



[goo.gl/ftXxNb](https://goo.gl/ftXxNb)

**Imbas Saya**



Rajah 3.22 Carta alir bagi mengisih senarai integer dalam urutan menaik

Rajah 3.23 dan Rajah 3.24 menunjukkan ralat yang dikesan pada pseudokod dan carta alir yang dihasilkan oleh Soon Huat.

### Pengesanan ralat dalam pseudokod

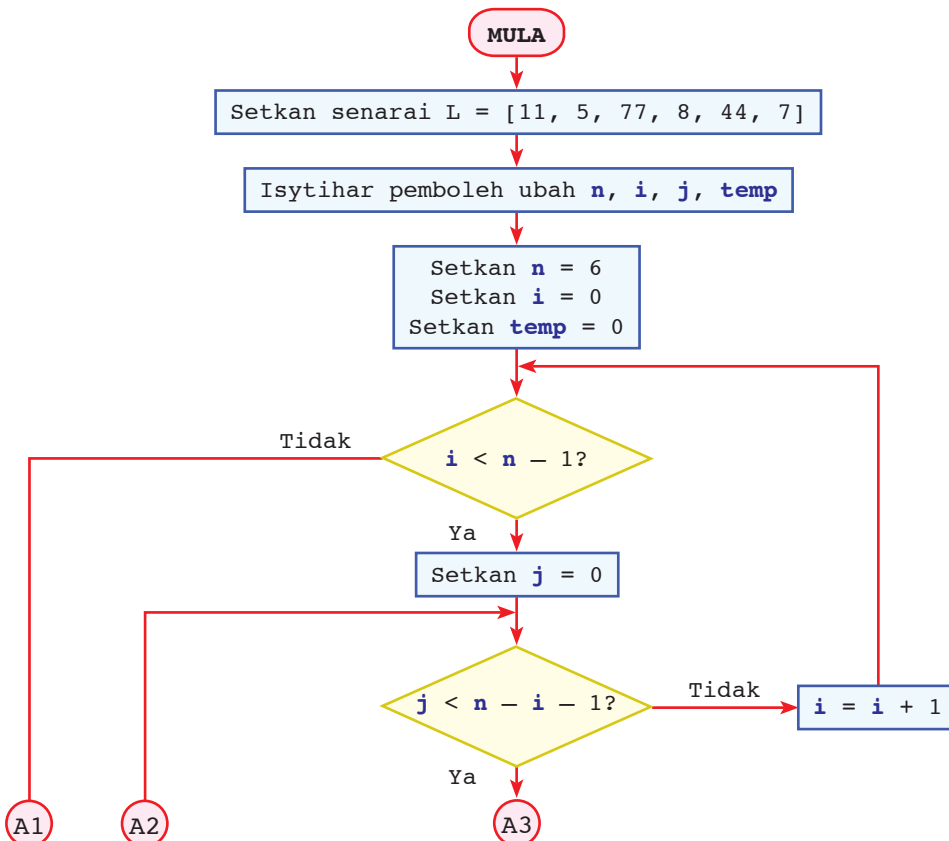
```

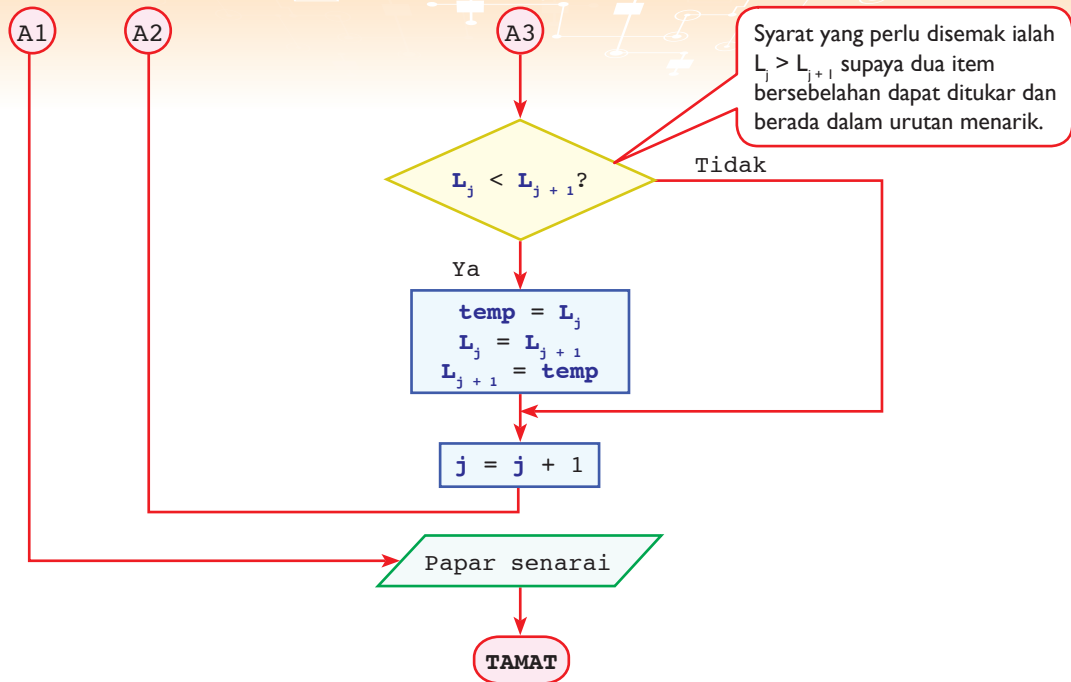
1 Mula
2 Setkan senarai  $L = [11, 5, 77, 8, 44, 7]$ 
3 Isytihar pemboleh ubah  $n, i, j, temp$ 
4 Setkan  $n = 6$ 
5 Setkan  $i = 0$ 
6 Setkan  $temp = 0$ 
7 for  $i < n - 1$ 
  7.1 Setkan  $j = 0$ 
  7.2 for  $j < n - i - 1$ 
    7.2.1 Jika  $L_j < L_{j+1}$ 
      7.2.1.1  $temp = L_j$ 
      7.2.1.2  $L_j = L_{j+1}$ 
      7.2.1.3  $L_{j+1} = temp$ 
    7.2.2 Tamat jika
    7.2.3 Kira  $j = j + 1$ 
  7.3 Kira  $i = i + 1$ 
8 Papar senarai
9 Tamat
  
```

Syarat yang perlu disemak ialah  $L_j > L_{j+1}$  supaya dua item bersebelahan dapat ditukar dan berada dalam urutan menaik.

Rajah 3.23 Pengesanan ralat pada pseudokod

### Pengesanan ralat pada carta alir





Rajah 3.24 Pengesanan ralat pada carta alir

Rajah 3.25 dan Rajah 3.26 menunjukkan pseudokod dan carta alir yang telah dibaiki oleh Soon Huat.

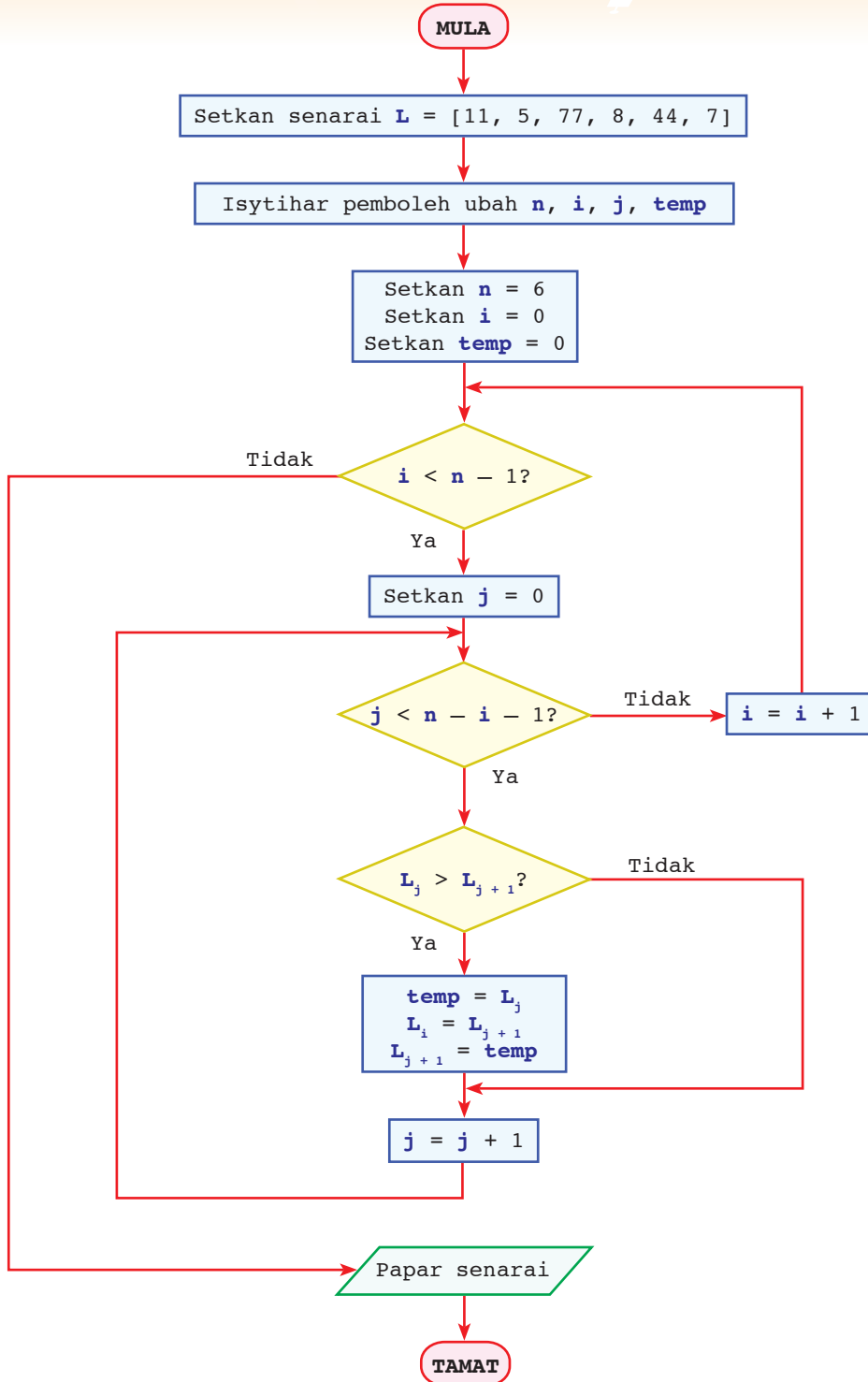
Pseudokod yang telah dibaiki ralat

```

1 Mula
2 Setkan senarai L = [11, 5, 77, 8, 44, 7]
3 Isytihar pemboleh ubah n, i, j, temp
4 Setkan n = 6
5 Setkan i = 0
6 Setkan temp = 0
7 for i < n - 1
  7.1 Setkan j = 0
  7.2 for j < n - i - 1
    7.2.1 Jika L_j > L_{j+1}
      7.2.1.1 temp = L_j
      7.2.1.2 L_j = L_{j+1}
      7.2.1.3 L_{j+1} = temp
    7.2.2 Tamat jika
    7.2.3 Kira j = j + 1
  7.3 Kira i = i + 1
8 Papar senarai
9 Tamat
  
```

Rajah 3.25 Penambahbaikan pseudokod

Carta alir yang telah dibaiki ralat

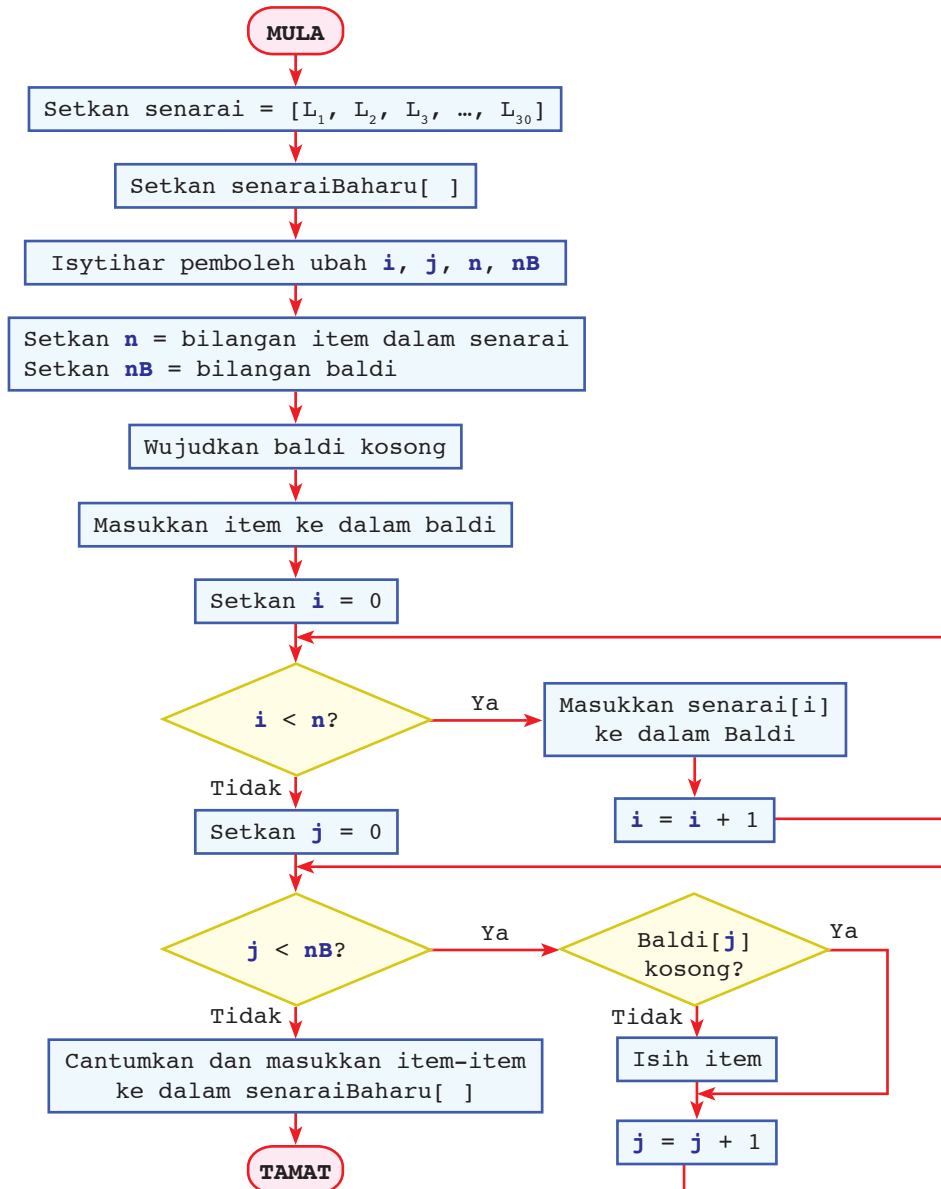


Rajah 3.26 Penambahbaikan carta alir



### Mengesan dan Membaiki Ralat dalam Carta Alir *Bucket Sort*

Nurul mengeluarkan semua wang simpanannya dari sebuah tabung. Wang simpanan Nurul terdiri daripada 30 keping wang syiling yang bernilai 5 sen, 10 sen, 20 sen dan 50 sen. Nurul ingin mengisih kesemua wang syiling tersebut mengikut nilai menaik menggunakan *bucket sort*. Nurul melukis carta alir seperti yang berikut. Apabila carta alir diuji, Nurul mendapati bahawa carta alir tidak memaparkan senarai yang telah diisih. Bantu Nurul mengesan dan membaiki ralat yang terdapat dalam carta alir yang dilukisnya. Kemudian, lukis semula carta alir yang telah dibaiki.



## 3.1.5

## Pengecaman Corak dalam Algoritma Search dan Sort

Pengecaman corak amat penting dalam pembangunan algoritma kerana pengatur cara boleh mengenal pasti dan memahami corak masalah yang perlu diselesaikan. Kefahaman ini akan memudahkan pengatur cara membina algoritma yang cekap dan berkesan. Di samping itu, pengatur cara boleh membandingkan masalah yang hendak diselesaikan dengan masalah-masalah lain yang serupa dan telah memperoleh penyelesaian. Jika didapati corak masalah yang hendak diselesaikan sama dengan masalah-masalah yang lain, pengatur cara boleh menggunakan penyelesaian yang sedia ada atau menggunakannya sebagai asas untuk membina algoritma yang lebih baik.

Pengecaman corak ialah satu teknik untuk mengenal pasti ciri-ciri, struktur, bentuk input serta output dan sebagainya yang terdapat dalam sesuatu masalah. Pengecaman corak sering melibatkan perbandingan corak antara masalah yang serupa atau sama untuk mencari persamaan dan perbezaan dalam masalah-masalah tersebut. Dalam topik ini, anda akan mengecam dan membandingkan corak bagi algoritma *search* dan algoritma *sort* supaya pengertian yang lebih mendalam dapat diketahui. Rajah 3.27 menunjukkan perbandingan antara ciri-ciri algoritma *search* dan *sort*.

### Standard Pembelajaran

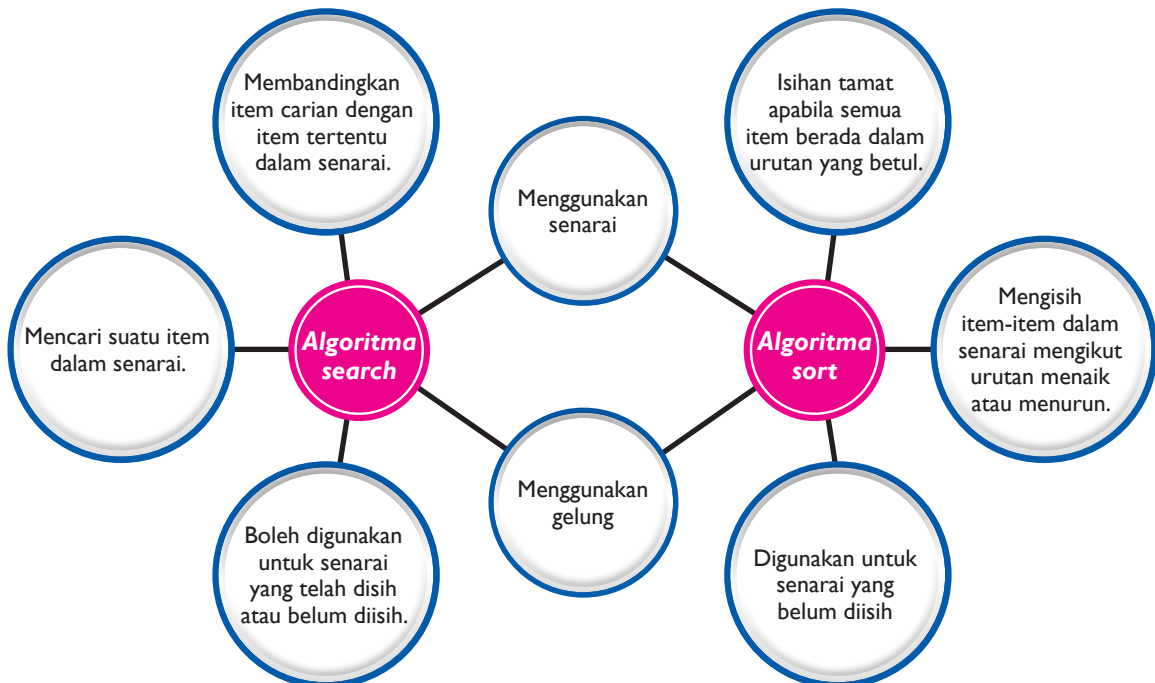
Murid boleh:

- 3.1.5** Membandingkan algoritma *search* dan *sort* melalui pengecaman corak.

Pengecaman corak dalam pengkomputeran



[goo.gl/EbZPW8](http://goo.gl/EbZPW8)



Rajah 3.27 Perbandingan ciri-ciri algoritma *search* dan *sort*



### Pengecaman Corak dalam Algoritma



Kaedah  
Peta i-Think

1. Teliti dua algoritma berikut.

#### Algoritma 1

- 1 Wujudkan beberapa buah baldi untuk menyimpan item-item daripada senarai secara sementara.
- 2 Tentukan julat bagi setiap baldi supaya item-item dapat dimasukkan ke dalam baldi.
- 3 Masukkan setiap item secara sementara ke dalam baldi berdasarkan julat yang ditetapkan.
- 4 Isih item-item di dalam setiap baldi.
- 5 Setelah item-item dalam kesemua baldi diisih, cantumkan item-item tersebut dan masukkan semula ke dalam senarai.

#### Algoritma 2

- 1 Pastikan item-item dalam senarai telah diisih mengikut urutan menaik.
- 2 Lihat item yang berada di tengah senarai.
- 3 Bandingkan item carian dengan item yang berada di tengah senarai.
- 4 Jika nilai bagi item carian sama dengan nilai bagi item yang berada di tengah senarai, carian akan dihentikan.
- 5 Jika nilai item carian kurang daripada nilai item yang berada di tengah senarai, abaikan item di tengah senarai dan item-item selepasnya. Kemudian, lihat pada senarai yang tinggal.
- 6 Jika nilai item carian lebih daripada nilai item di tengah senarai, abaikan item di tengah senarai dan item-item sebelumnya. Kemudian, lihat pada senarai yang tinggal.
- 7 Ulang Langkah 2 hingga Langkah 6 sehingga item carian dijumpai atau apabila carian selesai tanpa menjumpai item carian.

2. Lakukan perbandingan antara kedua-dua algoritma ini dan tentukan algoritma yang manakah mewakili *search* dan *sort*?
3. Beri justifikasi terhadap jawapan anda.
4. Huraikan persamaan dan perbezaan antara kedua-dua algoritma ini dalam bentuk peta i-Think yang menarik.

### Standard Pembelajaran

Murid boleh:

- 3.1.6** Menghasilkan algoritma melibatkan gabungan teknik *search* dan *sort*.

3.1.6

### Algoritma bagi Gabungan *Search* dan *Sort*

Dalam kehidupan seharian, adakalanya aktiviti mencari dan mengisih memerlukan gabungan algoritma *search* dan *sort* untuk memudahkan penyelesaian masalah. Begitu juga dalam proses membina atur cara untuk menyelesaikan sesuatu masalah, kadangkala gabungan teknik *search* dan *sort* perlu digunakan. Bagi masalah atau situasi yang mengaplikasikan teknik *search* dan *sort*, pengatur cara perlu memilih gabungan teknik yang sesuai supaya output yang diperoleh adalah seperti yang diinginkan. Gabungan kedua-dua teknik ini akan mempercepat dan memudahkan proses penyelesaian sesuatu masalah.



**Contoh 3.13** Pembinaan algoritma bagi gabungan teknik *search* dan *sort*.

Semasa aktiviti penangkapan ikan di laut, apabila pukat dinaikkan ke atas kapal, nelayan akan mencari ikan-ikan berkualiti yang mempunyai nilai dalam pasaran. Ikan-ikan ini akan diasingkan dan dimasukkan ke dalam bakul manakala ikan-ikan yang lain akan dilepaskan semula ke laut. Ikan-ikan yang terdapat di dalam bakul akan diisih mengikut jenisnya.

Algoritma bagi carian kualiti ikan dan isihan ikan-ikan mengikut jenis ditunjukkan dalam Rajah 3.28

1. Naikkan pukat ke atas kapal.
2. Cari ikan-ikan berkualiti yang mempunyai nilai dalam pasaran.
3. Jika ikan tidak berkualiti yang tidak mempunyai nilai dalam pasaran, lepaskan semula ke laut.
4. Isih ikan-ikan mengikut jenis.

**Rajah 3.28** Algoritma carian kualiti ikan dan isihan ikan mengikut jenis

Berdasarkan algoritma dalam Rajah 3.28, proses carian dilakukan terlebih dahulu dan diikuti dengan proses isihan.

**Tahukah Anda?**

Dalam perisian aplikasi pemprosesan perkataan, carian boleh dilakukan dengan lebih pantas, iaitu dengan menekan "Ctrl+F" pada papan kekunci. Anda hanya perlu menaip perkataan yang ingin dicari pada kotak yang disediakan dan komputer akan membuat carian perkataan tersebut.

**Uji Minda 3.15**

Adakah aktiviti dalam Contoh 3.13 boleh dimulakan dengan mengisih ikan mengikut jenis dan kemudian mencari ikan-ikan yang berkualiti?



## Aktiviti 3.11

**Aktiviti Kumpulan**

Algoritma Gabungan *Search* dan *Sort*



Kaedah *Home Group* dan *Colour Group*

1. Setiap kumpulan diberikan satu senarai seperti di bawah. Isihkan senarai tersebut mengikut urutan dan seterusnya lakukan carian nombor 13.

16	5	27	13	9	1	6
----	---	----	----	---	---	---

2. Bincangkan teknik *sort* yang sesuai digunakan untuk mengisih senarai di atas bersama-sama ahli kumpulan.
3. Selepas melakukan perbincangan, setiap ahli daripada setiap kumpulan akan membentuk kumpulan yang baharu.
4. Kemudian, ahli kumpulan baharu akan melakukan perbincangan mengenai pemilihan teknik *search* yang sesuai digunakan untuk mencari nombor 13.

## Standard Pembelajaran

Murid boleh:

- 3.1.7** Menghasilkan algoritma melibatkan gabungan teknik *search* atau *sort*.

3.1.7

## Algoritma bagi Gabungan Search atau Sort

Selain menggabungkan teknik *search* dengan teknik *sort*, gabungan satu teknik carian dengan teknik carian yang lain dan gabungan satu teknik isihan dengan teknik isihan yang lain sering dilakukan untuk menyelesaikan pelbagai jenis masalah dalam kehidupan seharian.

### Contoh 3.14

Pembinaan algoritma bagi gabungan teknik *search*.

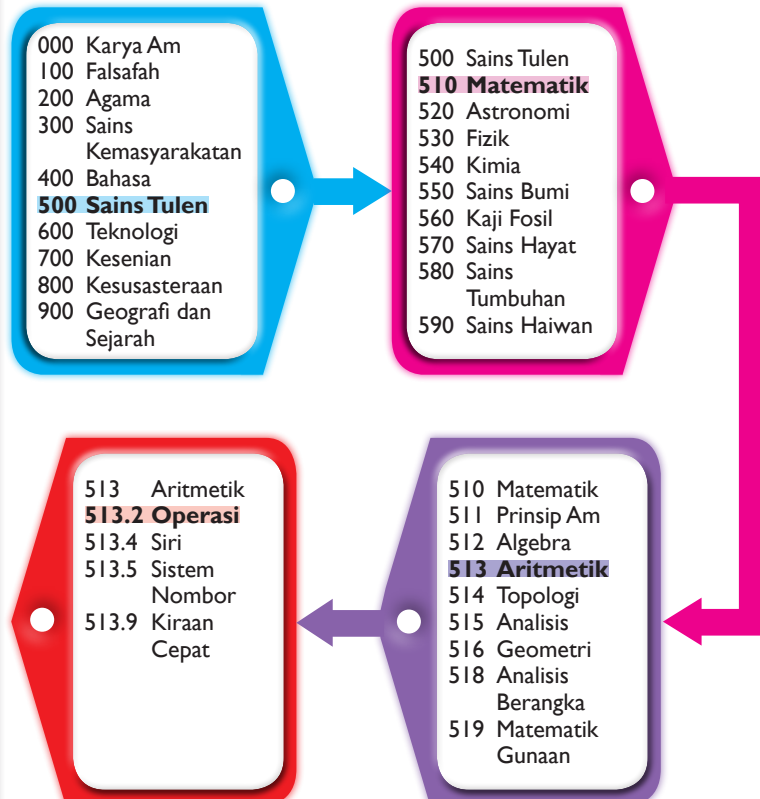


### Tahukah Anda?

DDC adalah sistem klasifikasi yang digunakan secara meluas oleh perpustakaan di seluruh dunia. Sistem ini telah digunakan oleh lebih daripada 138 negara. Sejak 1983, DDC telah diterjemahkan ke dalam 30 bahasa. Oleh sebab DDC ialah sistem klasifikasi yang berbentuk hierarki, maka bukan semua klasifikasi telah ditentukan. Ini bergantung kepada perpustakaan yang menggunakan klasifikasi tersebut.

Naresh ingin mencari senaskhah buku rujukan dalam kategori operasi aritmetik. Perpustakaan sekolah Naresh menggunakan Sistem Perpuluhan Dewey (DDC) untuk mengklasifikasikan buku-buku yang disusun pada rak.

Rajah 3.29 menunjukkan sebahagian daripada Penggunaan DDC bagi mencari buku berkategori Operasi.



Rajah 3.29 Penggunaan DDC bagi mencari buku-buku berkategori Operasi

Rajah 3.30 menunjukkan algoritma bagi mencari buku berkategori Operasi.

1. Masuk ke perpustakaan.
2. Lihat carta DDC.
3. Cari nombor rak berlabel 500. SainsTulen.
4. Pergi ke rak berlabel 500. SainsTulen.
5. Cari bahagian berlabel 510. Matematik.
6. Cari bahagian berlabel 513. Aritmetik.
7. Cari bahagian berlabel 513.2 Operasi.
8. Cari buku yang dikehendaki.
9. Tamat.

**Rajah 3.30** Algoritma carian buku berkategori Operasi

Berdasarkan algoritma di atas, dapat diperhatikan bahawa gabungan dua jenis teknik carian digunakan. Naresh menggunakan *binary search* untuk mencari rak berlabel 500. SainsTulen kerana terdapat lebih 900 label pada carta DDC. Kemudian, Naresh menggunakan *linear search* untuk mencari bahagian berlabel 510. Matematik kerana bahagian ini berada di rak pertama bagi label 500. Seterusnya, untuk mencari bahagian berlabel 513. Aritmetik, Naresh menggunakan *binary search* kerana bahagian ini berada di tengah-tengah label antara 510 hingga 519. Naresh menggunakan *linear search* untuk mencari bahagian berlabel 513.2. Operasi dan seterusnya mencari buku yang dikehendaki.

**Contoh 3.15** Pembinaan algoritma bagi gabungan teknik *sort*.

Adila ialah seorang pembantu klinik. Dia dikehendaki menyusun fail-fail pesakit yang terdapat di klinik itu bagi memudahkan fail pesakit dicari apabila pesakit datang untuk mendapatkan rawatan. Adila memulakan tugasnya dengan mengisih fail-fail mengikut tahun kelahiran pesakit. Bagi setiap tahun kelahiran, Adila menyusun fail-fail itu mengikut abjad nama pesakit.

Rajah 3.31 menunjukkan algoritma bagi isihan fail-fail yang dilakukan oleh Adila.

1. Semak tahun kelahiran yang terdapat pada fail pesakit.
2. Isih fail-fail pesakit dengan memasukkan fail-fail ke dalam beberapa buah kotak mengikut tahun kelahiran.
3. Isih fail-fail yang terdapat di dalam setiap kotak mengikut abjad nama pesakit.

**Rajah 3.31** Algoritma isihan fail-fail di klinik

Berdasarkan algoritma dalam Rajah 3.31, terdapat gabungan dua isihan yang berlaku dalam proses Adila menyusun fail-fail pesakit. Adila menggunakan *bucket sort* untuk mengisih fail-fail pesakit mengikut tahun kelahiran kerana pesakit terdiri daripada pelbagai peringkat umur. Kemudian, Adila menggunakan *bubble sort* untuk mengisih fail-fail pesakit yang berada di dalam setiap kotak mengikut abjad nama pesakit.



## Aktiviti 3.12

Aktiviti Kumpulan

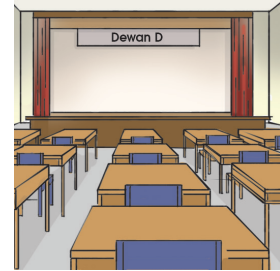
### Membina Algoritma Gabungan Teknik *Search*



Kaedah Pembentangan Hasil Sendiri

1. Teliti situasi berikut.

Amira perlu menduduki peperiksaan memasuki sekolah berasrama penuh yang diadakan di dewan D. Terdapat 6 buah dewan yang digunakan untuk peperiksaan. Apabila tiba di dewan D, Amira mencari meja bernombor 38. Di dalam dewan tersebut, terdapat 60 buah meja dan setiap meja dilabelkan dengan nombor 1 hingga 60 dalam urutan menaik.



2. Bersama-sama ahli kumpulan, bincangkan cara-cara untuk membantu Amira mencari dewan D dan seterusnya mencari meja bernombor 38.
3. Bina algoritma bagi gabungan teknik *search* yang digunakan.
4. Bentangkan hasil dapatan kumpulan di hadapan kelas mengikut giliran.



## Aktiviti 3.13

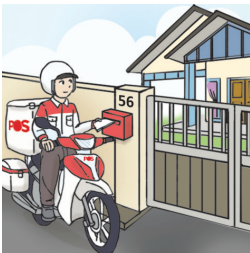
Aktiviti Kumpulan

### Membina Algoritma Gabungan Teknik *Sort*



Kaedah *Gallery Walk*

1. Teliti situasi berikut.



Matthew Lucas ialah seorang posmen. Dia dikehendaki menghantar sejumlah surat ke sebuah taman perumahan. Sebelum menghantar surat, tugas Matthew adalah mengasingkan surat-surat itu mengikut rumah yang bernombor ganjil dan genap. Kemudian, Matthew perlu mengisih surat-surat tersebut mengikut urutan menaik

2. Bersama-sama ahli kumpulan, bina algoritma bagi membantu Matthew mengasingkan dan mengisih surat-surat itu menggunakan gabungan teknik *sort* yang sesuai.
3. Tampilkan hasil kerja kumpulan di atas meja untuk dilihat oleh kumpulan lain.
4. Setiap kumpulan bergerak ke kumpulan lain untuk melihat dan memberikan komen atau pengukuhan berkaitan hasil kerja mereka.
5. Kembali semula kepada kumpulan masing-masing. Kemudian, ceritakan atau laporkan pembelajaran serta pengalaman yang diperolehi.

## Uji Kendiri 3.1

1. Terangkan ciri-ciri bagi teknik-teknik yang berikut.
  - (a) *Linear search*
  - (b) *Binary search*
  - (c) *Bubble sort*
  - (d) *Bucket sort*

2. Diberikan satu senarai seperti berikut.



Cari nombor 14 dalam senarai di atas menggunakan teknik *binary search*. Tuliskan pseudokod atau lukiskan carta alir bagi teknik *binary search* yang digunakan.

3. Cikgu Indra merupakan penyelarar Tingkatan 3. Beliau ditugaskan oleh pengetua untuk menyenaraikan nama murid mengikut peratus markah peperiksaan yang diperoleh. Murid-murid Tingkatan 3 di sekolah beliau adalah seramai 150 orang. Apakah teknik yang paling sesuai yang boleh digunakan oleh beliau?

4. Baca dan teliti pernyataan di bawah.

Pencarian suatu item dalam satu senarai yang mempunyai bilangan item yang banyak dan item-item tersebut berada dalam urutan menaik, teknik *binary search* lebih sesuai digunakan.

Berikan justifikasi anda bagi pernyataan di atas.

5. Teliti senarai abjad berikut.



- (a) Tuliskan pseudokod bagi mencari huruf E menggunakan teknik *linear search* dan teknik *binary search*.
- (b) Berdasarkan pseudokod yang ditulis, gunakan teknik pengecaman corak untuk membandingkan persamaan dan perbezaan antara kedua-dua teknik carian tersebut.

6. Jessica diberikan kad-kad yang bernombor 1 hingga 100 secara rawak. Jessica diminta untuk menyediakan atur cara bagi mengenal pasti nombor-nombor yang mengandungi digit 5. Kesemua kad nombor dengan digit 5 yang diperoleh perlu diisih mengikut urutan menurun.

- (a) Gabungan teknik carian dan isihan yang manakah sesuai digunakan oleh Jessica untuk mendapatkan kad-kad bernombor dengan digit 5 dan seterusnya mengisih kad-kad tersebut? Berikan alasan anda bagi gabungan teknik yang dipilih.
- (b) Tuliskan algoritma bagi gabungan teknik yang dipilih untuk menyelesaikan masalah ini.

# RUMUSAN GRAFIK 3



## SEMAK SEMULA 3

Pada akhir bab ini, saya dapat...

- |  |  |
|--|--|
| mengenal pasti ciri-ciri <i>linear search</i> dan <i>binary search</i> .   |  |
| mengenal pasti ciri-ciri <i>bubble sort</i> and <i>bucket sort</i> .   |  |
| menulis pseudokod dan melukis carta alir yang menunjukkan <i>linear search</i> .   |  |
| menulis pseudokod dan melukis carta alir yang menunjukkan <i>binary search</i> .   |  |
| menulis pseudokod dan melukis carta alir yang menunjukkan <i>bubble sort</i> .   |  |
| menulis pseudokod dan melukis carta alir yang menunjukkan <i>bucket sort</i> .   |  |
| menganalisis dan membaiki ralat dalam pseudokod dan melukis carta alir bagi penyelesaian masalah yang melibatkan <i>search</i> . |  |
| menganalisis dan membaiki ralat dalam pseudokod dan melukis carta alir bagi penyelesaian masalah yang melibatkan <i>sort</i> .   |  |
| membandingkan algoritma <i>search</i> dan <i>sort</i> melalui pengecaman corak.  |  |
| menghasilkan algoritma melibatkan teknik <i>search</i> dan <i>sort</i> .   |  |
| menghasilkan algoritma melibatkan teknik <i>search</i> atau <i>sort</i> .  |  |

# PENTAKSIRAN 3

## Soalan Objektif

1. Teknik ini akan mencari dan meletakkan item-item daripada suatu senarai ke dalam kumpulan-kumpulan tertentu sebelum item-item dalam setiap kumpulan tersebut disusun mengikut tertib.

A *Linear search*  
 B *Binary search*  
 C *Bubble sort*  
 D *Bucket sort*

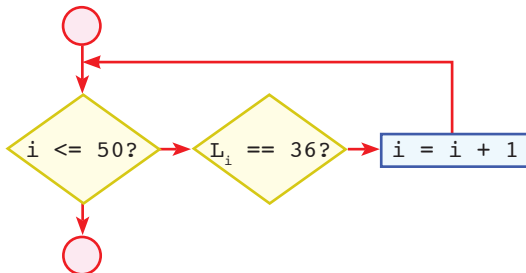
2. Teliti maklumat berikut.

- Senarai mengandungi item yang banyak.
- Item-item dalam senarai telah diisih.

Apakah teknik yang sesuai digunakan untuk mencari suatu item dalam senarai tersebut?

A *Linear search*  
 B *Binary search*  
 C *Bubble sort*  
 D *Bucket sort*

3. Teliti segmen carta alir di bawah.



Apakah teknik yang menggunakan jenis gelung di atas?

A *Linear search*  
 B *Binary search*  
 C *Bubble sort*  
 D *Bucket sort*

4. Hanisah diberikan satu senarai nombor [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Nombor-nombor yang manakah menjadi nilai tengah semasa proses *binary search* bagi nombor 22?

A 19, 20, 21      C 17, 21, 23  
 B 21, 22, 23      D 17, 14, 21

5. Juliana telah membangunkan satu aplikasi *search* berdasarkan empat angka akhir nombor kad pengenalan. Antara berikut, yang manakah aplikasi yang mungkin telah dibangunkan oleh Juliana?

I Aplikasi mencari judul buku dalam perpustakaan.  
 II Aplikasi mencari biodata murid dalam pangkalan data.  
 III Aplikasi mencari markah peperiksaan murid.  
 IV Aplikasi mencari inventori barangan di dalam kelas.

A I dan II      C III dan IV  
 B II dan III      D I, II dan III

6. Antara berikut, aktiviti yang manakah menggunakan teknik *search* dan *sort*?

I Seorang suri rumah mengasingkan pakaian mengikut warna sebelum dimasukkan ke dalam mesin basuh.  
 II Seorang guru merekod gred markah ujian murid.  
 III Seorang wanita membeli ikan di pasar.  
 IV Seorang posmen mengasingkan surat di pejabat pos sebelum diposkan kepada penerima.

A I dan II      C I, II dan IV  
 B I, II dan III      D II, III dan IV

7. Manakah antara berikut menunjukkan isihan menurun senarai selepas lelaran pertama?

- A [3, 10, 8, 9, 4] → [3, 8, 9, 10, 4]
- B [3, 10, 8, 9, 4] → [3, 8, 4, 9, 10]
- C [3, 10, 8, 9, 4] → [10, 8, 9, 3, 4]
- D [3, 10, 8, 9, 4] → [3, 8, 9, 4, 10]

8. Dalam industri penternakan, telur-telur ayam akan diproses mengikut gred atau saiz.



Aktiviti ini menggambarkan

- A *linear search*
- B *binary search*
- C *bubble sort*
- D *bucket sort*

9. Cikgu Khor mempunyai satu senarai yang mengandungi markah ujian Asas Sains Komputer bagi 250 orang murid Tingkatan 3. Senarai markah tersebut tidak diisih. Cikgu Khor ingin mencari markah yang terendah dan tertinggi dalam senarai tersebut. Apakah teknik *sort* dan *search* yang akan digunakan oleh Cikgu Khor dalam menyelesaikan masalah ini?



- A *Bucket sort* dan *binary search*
- B *Bubble sort* dan *linear search*
- C *Bucket sort* dan *linear search*
- D *Bubble sort* dan *binary search*

10. Pseudokod berikut digunakan untuk menyelesaikan suatu masalah. Apabila semakan dilakukan ke atas pseudokod itu, output yang dihasilkan tidak seperti yang dikehendaki.

```
1 Mula
2 Setkan senarai = [3, 10, 8, 9, 4]
3 Isytihar pemboleh ubah i, j, n, temp
4 Setkan i = 0
5 Setkan n = bilangan item dalam senarai
6 Setkan temp = 0
7 for i < n - 1
  7.1 Setkan j = 0
  7.2 for j < n - i - 1
    7.2.1 Jika  $L_j > L_{j+1}$ 
      7.2.1.1 temp =  $L_j$ 
      7.2.1.2  $L_j = L_{j+1}$ 
      7.2.1.3  $L_{j+1} = temp$ 
    7.2.2 Tamat jika
    7.2.3 Kira  $j = j + 1$ 
8 Papar senarai
9 Tamat
```

Terdapat ralat dikesan dalam pseudokod yang diberikan. Apakah ralat tersebut?

- A i disetkan kepada 0 sebelum gelung *for* bermula
- B i tidak ditambah pada 1 dalam gelung *for*
- C  $j = 0$  dalam gelung *for*
- D Syarat diuji dari  $i = 0$  hingga  $n - 1$

### Soalan Berstruktur

1. Natasya ingin mencari kedudukan suatu item yang terdapat dalam satu senarai.



Dia menggunakan teknik *binary search*. Mengapakah Natasya menggunakan teknik tersebut? Berikan pendapat anda berdasarkan ciri-ciri teknik tersebut.

2. Anda diberikan satu senarai [epal, limau, anggur, duku, ciku]. Anda diminta untuk



mengisih item-item dalam senarai tersebut mengikut urutan menaik berdasarkan abjad. Apakah teknik isihan yang akan anda gunakan? Berikan justifikasi anda.

3. Huraikan persamaan dan perbezaan bagi *linear search* dan *binary search*.

### E-PRAKTIS



[goo.gl/jDWYiV](https://goo.gl/jDWYiV)



4. Nurin menerima satu pseudokod bagi isihan item-item dalam senarai mengikut urutan menaik menggunakan *bubble sort* yang tidak lengkap daripada Alias. Nurin telah mengenal pasti segmen kod yang tidak lengkap berada pada baris ke-7. Bantu Nurin melengkapkan pseudokod itu.

```

1 Mula
2 Setkan senarai L = [70, 60, 32, 65.5, 25, 44, 30.7, 85]
3 Isytihar pemboleh ubah i, j, n, temp
4 Setkan n = bilangan item dalam senarai L
5 Setkan i = 0
6 Setkan temp = 0
7 for i < n - 1
  7.1 Setkan j = 0
  7.2 for j < n - i - 1
      [Empty box]
8 Papar senarai
9 Tamat

```

### Soalan Penyelesaian Masalah

1. Teliti jadual berikut.



Indeks	0	1	2	3	4
Item	30	22	33	55	42

- (a) Tentukan teknik isihan yang sesuai untuk mengisih item-item dalam senarai di atas mengikut urutan menaik dan tuliskan algoritma bagi isihan tersebut.
- (b) Tuliskan pseudokod atau lukiskan carta alir untuk menunjukkan proses pengisihan tersebut.
2. Batrisyia menggunakan satu teknik carian untuk mencari nilai 24 dalam senarai di bawah.



3	4	7	10	15	24	36	46	72	90
---	---	---	----	----	----	----	----	----	----

Langkah-langkah yang digunakan adalah seperti berikut.

**Langkah 1**

3	4	7	10	15	24	36	46	72	90
---	---	---	----	----	----	----	----	----	----

**Langkah 2**

					24	36	46	72	90
--	--	--	--	--	----	----	----	----	----

**Langkah 3**

					24	36			
--	--	--	--	--	----	----	--	--	--

Berdasarkan langkah-langkah di atas, tuliskan pseudokod atau lukiskan carta alir untuk mewakili teknik carian tersebut.

3. Tulis satu aplikasi yang boleh anda hasilkan menggunakan teknik *search* dan *sort*.



Terangkan secara ringkas tentang aplikasi tersebut serta tulis algoritma bagi aplikasi itu.

# BAB 4

## KOD ARAHAN

Penggunaan pangkalan data dan kod arahan dalam sistem komputer dan peranti elektronik pintar telah menjadikan urusan dalam pelbagai bidang dapat dilaksanakan dengan pantas dan efisien. Selain itu, banyak masa boleh diijamatkan malah produktiviti juga dapat ditingkatkan. Contohnya, mesin-mesin automatik dan *mobile apps* membantu menyenangkan tugas dan menyelesaikan masalah manusia dengan mudah.

### Standard Kandungan

- 4.1 Pangkalan Data dan SQL
- 4.2 Struktur Kod Arahan

#### Pasar Raya

Harga barangan boleh disemak dengan mudah dan cepat dengan hanya mengimbas kod bar pada mesin penyemak harga.



#### Carian Istilah

Carian Istilah ialah aplikasi yang dibangunkan untuk memberikan maklumat istilah dengan pantas. Sumber kepada hasil carian adalah berasaskan <https://prpm.dbp.gov.my>



#### Hospital / Bank / Pejabat Pos

Penggunaan sistem giliran menjadi popular di tempat-tempat awam seperti hospital, bank dan pejabat pos membolehkan pesakit atau pelanggan menunggu giliran masing-masing dengan tenang.

Tonton video ini untuk mengetahui tentang gambaran Bab 4.

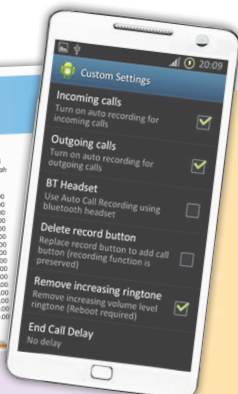


[goo.gl/dq7tqV](https://goo.gl/dq7tqV)



Statement date : 24/12/2017  
Phone number : (016) 2388588

Date	Time	From	To	Number called	Duration	Period	Gross	Discount	Total
Tarikh	Masa	Dial	Kep	No Panggilan	Tempoh	Kadar	Amount	Penjat	Jumlah
<b>Domestic Voice Calls/ Panggilan Suara Domestik</b>									
27/11/17	19:22:41	Central (SG)	Central	881922330145	00:00:35	OP	0.00		0.00
27/11/17	19:22:41	Central (SG)	Central	881922330146	00:00:35	OP	0.00		0.00
28/11/17	18:46:00	Central (SG)	North	881928279159	00:00:05	P	0.00		0.00
30/11/17	17:06:00	Central (SG)	South	881928279159	00:00:13	OP	0.00		0.00
04/12/17	19:43:46	Central (SG)	North	881928279159	00:00:59	P	0.00		0.00
07/12/17	15:38:09	Central (SG)	North	881928279159	00:00:54	P	0.00		0.00
12/12/17	19:29:07	Central (SG)	North	881928279159	00:00:37	P	0.00		0.00
13/12/17	23:01:14	Central (SG)	North	881928279159	00:00:38	OP	0.00		0.00
14/12/17	21:07:09	Central (SG)	North	881928279159	00:01:14	P	0.00		0.00
16/12/17	13:03:09	Central (SG)	Central	881928279159	00:01:14	P	0.00		0.00
16/12/17	15:53:33	Central (SG)	North	881928279159	00:01:56	P	0.00		0.00
17/12/17	17:08:36	Central (SG)	South	881928279159	00:00:56	P	0.00		0.00
18/12/17	17:00:32	Central (SG)	South	881928279159	00:00:56	P	0.00		0.00
22/12/17	13:34:47	Central (SG)	North	881928279159	00:00:13	OP	0.00		0.00



### Perkhidmatan Telefon

Setiap penerimaan atau panggilan telefon yang anda lakukan akan direkodkan untuk kegunaan pada masa hadapan.



### Pejabat

Sistem kehadiran dengan pengesanan cap jari telah digunakan secara umum di pejabat bagi memastikan hanya individu yang terlibat sahaja boleh memasuki pejabat.

## Penggunaan Pangkalan Data dan Kod Arah Dalam Kehidupan Sehari-hari

### Juruwang Berautomatik (Automated teller machine)

Mesin berkomputer yang berfungsi sebagai juruwang dalam urusan perbankan membenarkan pelanggan menikmati perkhidmatan-perkhidmatan perbankan pada bila-bila masa.



## Kata Kunci

- Atribut
- Entiti
- Function
- Kekardinalan
- Kekunci asing
- Kekunci primer
- Pangkalan data
- Procedure
- Query
- SQL

# 4.1 PANGKALAN DATA DAN SQL

## Imbas Kembali

Semasa TMK Tahun 5, anda telah mempelajari data dalam pangkalan data terdiri daripada beberapa jenis seperti teks, nombor dan tarikh.



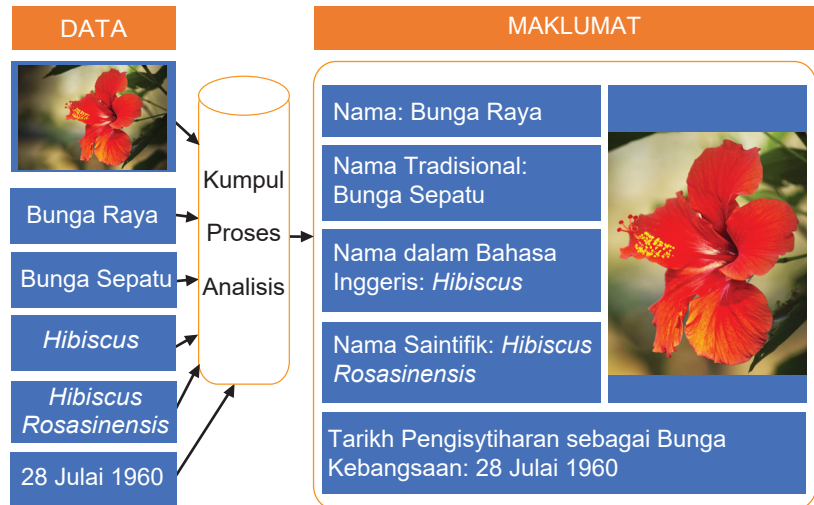
## Kerjaya Bidang Sains Komputer

Pentadbir pangkalan data bertanggungjawab untuk memastikan pangkalan data sentiasa berfungsi dan terkini.

Penganalisis pangkalan data bertanggungjawab untuk menganalisis dan membuat laporan bagi data-data dalam pangkalan data.

Data memainkan peranan yang penting dalam proses membuat keputusan. Data digunakan secara meluas untuk menyelesaikan sesuatu masalah. Data dianggap sebagai aset penting bagi sesebuah organisasi. Kejayaan sesebuah organisasi bergantung kepada kebolehannya untuk memperoleh data yang tepat, mengurus data dengan cekap dan menggunakan data untuk membuat analisis dengan tepat. Walau bagaimanapun, data yang belum diproses kepada maklumat adalah tidak bermakna.

Maklumat adalah hasil daripada **pengumpulan**, **pemprosesan** dan **penganalisan** data yang boleh digunakan untuk membuat sesuatu keputusan. Maklumat yang dihasilkan akan lebih bermakna dan mudah difahami oleh setiap individu. Rajah 4.1 menunjukkan hubungan antara data dan maklumat.



Rajah 4.1 Hubungan antara data dan maklumat

## Standard Pembelajaran

Murid boleh:

- 4.1.1** Menerangkan maksud dan penggunaan:
- (i) Pangkalan Data
  - (ii) *Structured Query Language* (SQL)

### 4.1.1

## Maksud dan Penggunaan Pangkalan Data dan *Structured Query Language* (SQL)

### (i) Pangkalan data

**Pangkalan data** ialah tempat pengumpulan dan penyimpanan data secara berpusat bagi sistem maklumat yang berasaskan komputer. Pangkalan data juga merupakan koleksi data yang saling berhubung antara satu sama lain yang disimpan tanpa berlaku pengulangan data yang tidak dikehendaki. Pangkalan data dapat memudahkan data dan maklumat untuk disimpan, dicapai atau dikemas kini.

Pengurusan data dalam sesebuah organisasi terbahagi kepada dua cara, iaitu dengan menggunakan sistem fail atau sistem pangkalan data.

### >> Sistem fail

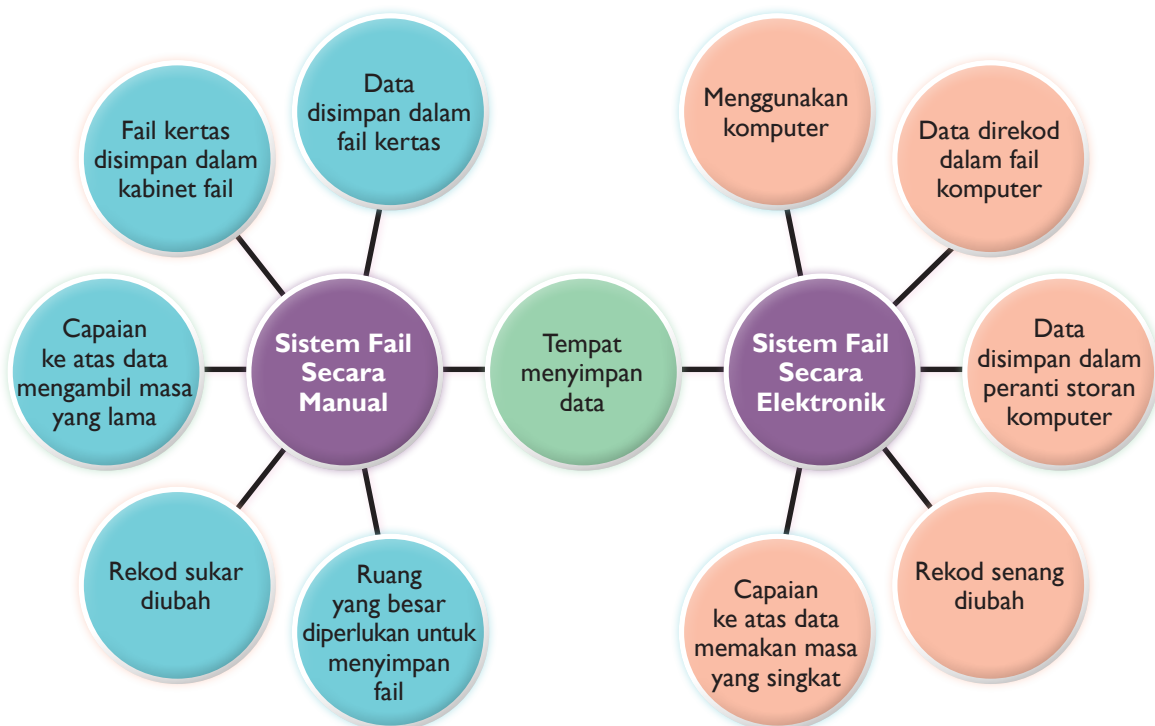
Terdapat dua jenis sistem fail yang digunakan, iaitu sistem fail secara manual dan sistem fail elektronik. Rajah 4.2 menunjukkan ciri-ciri sistem fail secara manual dan elektronik dalam bentuk peta pemikiran i-THINK.

### >> Sistem pangkalan data

Konsep pangkalan data muncul hasil daripada keperluan untuk membina sistem pemprosesan data yang cekap bagi menyokong keperluan sesebuah organisasi. Berbanding dengan sistem fail, pangkalan data bukan sekadar sistem pemprosesan fail elektronik. Pangkalan data menyatupadukan koleksi data yang terkandung dalamnya untuk memudahkan penyimpanan, pemprosesan, capaian dan keselamatan data. Rajah 4.3 menunjukkan fungsi, kebaikan dan contoh kegunaan sistem pangkalan data.

## Tahukah Anda?

Pangkalan data yang pertama telah diperkenalkan pada tahun 1960. Sistem pangkalan data ini dikenali sebagai *SABRE System* yang digunakan oleh *International Business Machines (IBM)* untuk membantu *American Airlines* bagi mengatasi masalah berkenaan data penempahan penerbangan.



Rajah 4.2 Ciri-ciri sistem fail secara manual dan elektronik

## Tahukah Anda?

Projek pendaratan Apollo di bulan menggunakan pangkalan data dikenali sebagai *Information Management System (IMS)* yang diperkenalkan pada tahun 1968.

Pengurusan pangkalan data aplikasi media sosial.



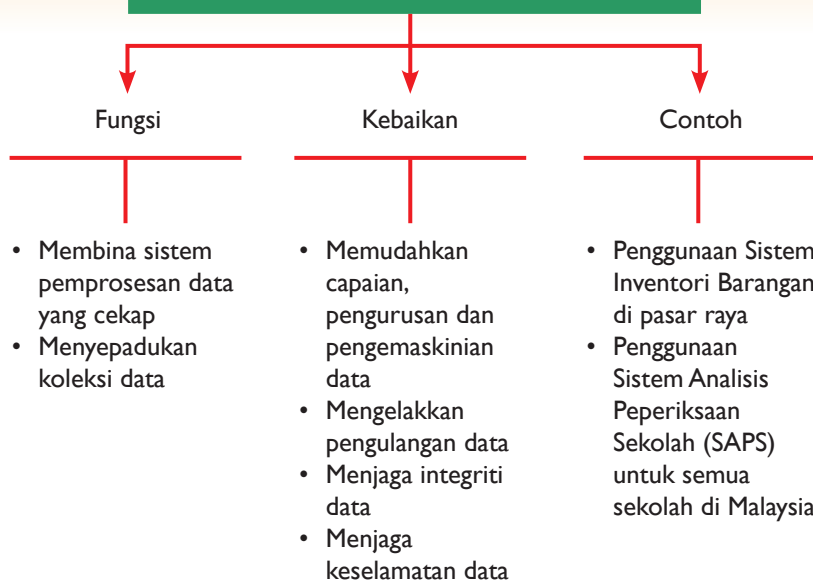
[goo.gl/rpffSf](https://goo.gl/rpffSf)

Imbas Saya

## Tahukah Anda?

SAPS Semakan Ibu Bapa merupakan pangkalan data interaktif yang merangkumi lebih 10,000 sekolah rendah dan sekolah menengah di Malaysia. Sistem ini memudahkan perkongsian maklumat dan data antara pihak sekolah dan ibu bapa atau penjaga murid-murid.

## SISTEM PANGKALAN DATA



Rajah 4.3 Fungsi, kebaikan dan contoh kegunaan sistem pangkalan data

Rajah 4.4 menunjukkan paparan laman sesawang Sistem Analisis Peperiksaan Sekolah (SAPS) bagi Menu Semakan Ibu Bapa. Bagaimanakah data-data yang disimpan di dalam pangkalan data ini dapat dicapai?

**KEMENTERIAN PENDIDIKAN MALAYSIA**  
Pendidikan Berkualiti Insan Terdidik Negara Sejahtera

SISTEM ANALISA PEPERIKSAAN SEKOLAH (SAPS)

**SEMAM SLIP PEPERIKSAAN PELAJAR**

SILA HUBUNGI PIHAK SEKOLAH UNTUK MENGETAHUI TARIKH SEMAKAN KEPUTUSAN PEPERIKSAAN

**LANGKAH 1: CARIAN MURID**  
SILA MASUKKAN NOMBOR SUJIL LAHIR/ KAD PENGENALAN PELAJAR (TANPA "-" ATAU "SPACE")

No KP / Sijil Lahir :  Cari

**LANGKAH 2: CARIAN SEKOLAH**  
SILA MASUKKAN MAKLUMAT SEKOLAH PELAJAR, SEKOLAH TERKINI ATAU SEBELUM INI

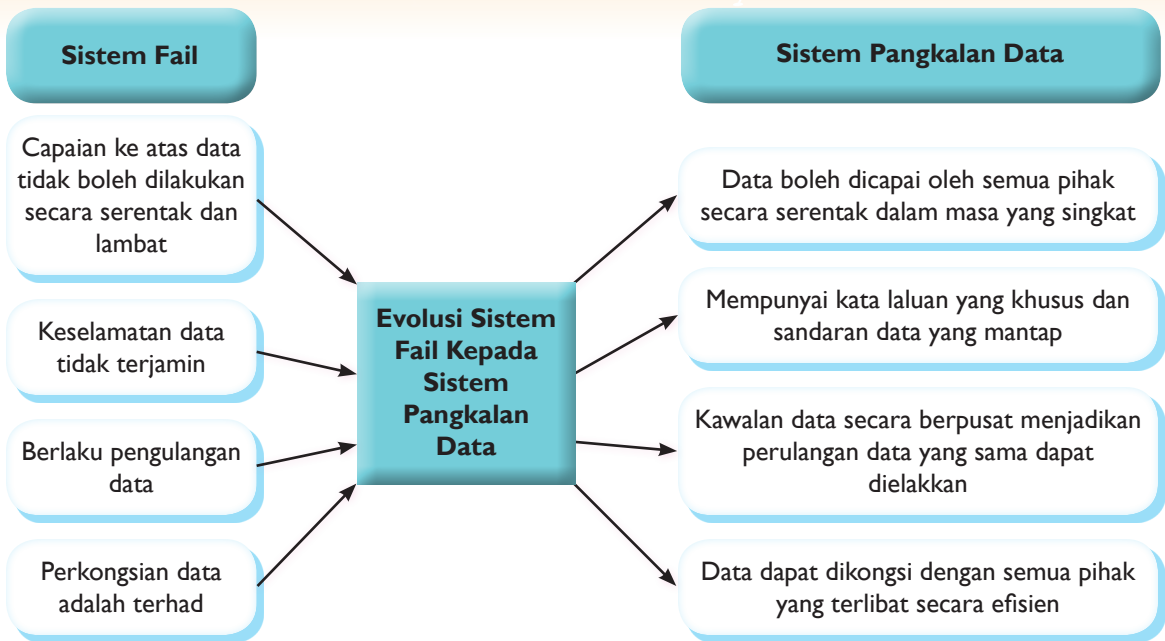
Negeri :

Nama Sekolah : Cari di Sini

Semak

Rajah 4.4 Sistem Analisis Peperiksaan Sekolah (SAPS) merupakan satu pangkalan data yang mampu menyimpan data murid-murid sekolah di Malaysia

Rajah 4.5 menunjukkan kebaikan evolusi dari sistem fail kepada sistem pangkalan data.



Rajah 4.5 Evolusi sistem fail kepada sistem pangkalan data

## (ii) Structured Query Language (SQL)

Capaian data dan maklumat dalam pangkalan data dilakukan melalui penggunaan bahasa pengaturcaraan seperti *Structured Query Language (SQL)*. SQL ialah bahasa pengaturcaraan aras tinggi generasi keempat yang ringkas dan mudah untuk dibaca serta mempunyai arahan-arahan yang terhad. SQL digunakan untuk menyimpan, memanipulasi dan mendapatkan data daripada pangkalan data.

SQL dapat berinteraksi dengan pangkalan data secara mudah. Oleh itu, SQL digunakan secara meluas dalam aplikasi yang melibatkan penggunaan pangkalan data. Bahasa pengaturcaraan mempunyai sintaks tertentu begitu juga dengan SQL. Antara pernyataan SQL yang akan anda pelajari ialah SELECT... FROM, SELECT...WHERE dan SELECT...ORDER BY.

### Tahukah Anda?

SQL tidak sensitif huruf. Kata kunci SQL boleh ditulis dalam huruf besar atau huruf kecil. Maksudnya, SELECT \* FROM abc sama kesannya dengan select \* from abc

## Uji Minda 4.1

Nyatakan perbezaan-perbezaan antara sistem fail dan sistem pangkalan data.

## Standard Pembelajaran

Murid boleh:

- 4.1.2** Menyenaraikan entiti dan atribut berdasarkan situasi dalam suatu masalah.

Entiti dan atribut sebuah pangkalan data



[goo.gl/PPpsgg](http://goo.gl/PPpsgg)



**Uji Minda 4.2**

Nyatakan entiti dan atribut-atribut yang wujud dalam persekitaran pusat sumber sekolah.

## Tahukah Anda?

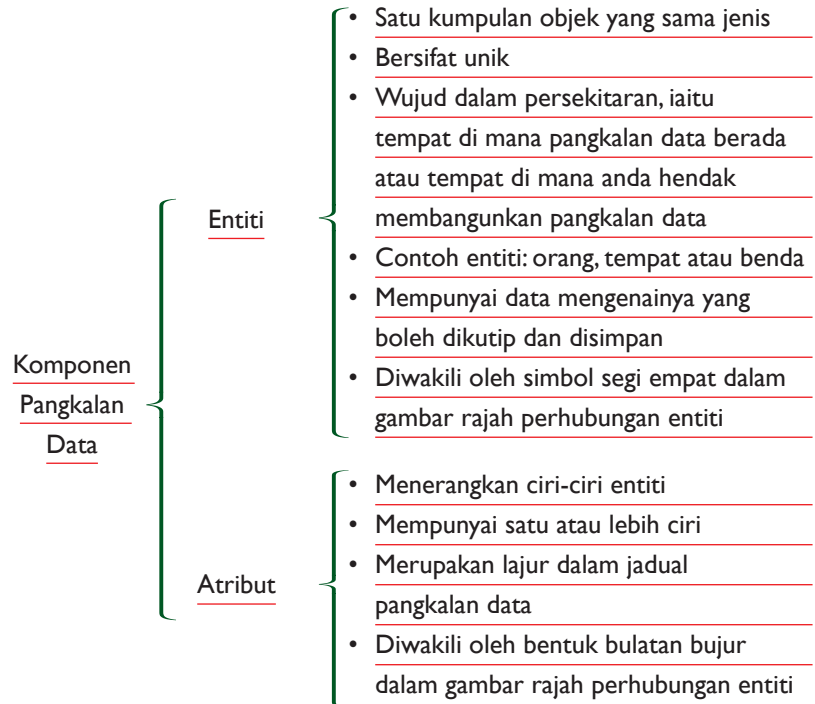
Entiti dan atribut diwakili oleh simbol piawai seperti berikut:

<Nama Entiti>

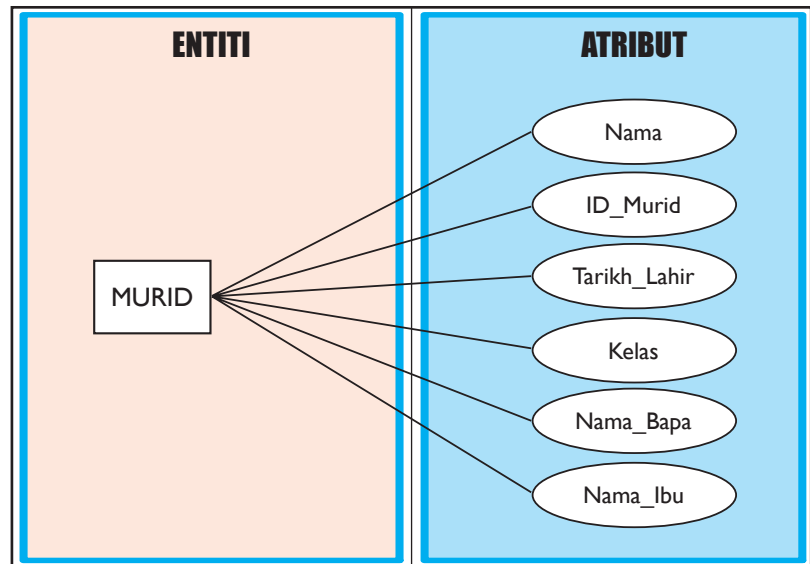
<Nama Atribut>

## 4.1.2 Entiti dan Atribut

Pangkalan data boleh wujud dalam pelbagai jenis organisasi. Namun begitu, reka bentuk pangkalan data mungkin berbeza antara organisasi yang lain. Hal ini disebabkan setiap organisasi mempunyai entiti-entiti dengan atribut-atribut yang berlainan. Rajah 4.6 menunjukkan komponen dalam pangkalan data.



Rajah 4.6 Komponen dalam pangkalan data



Rajah 4.7 Hubungan antara entiti dan atribut

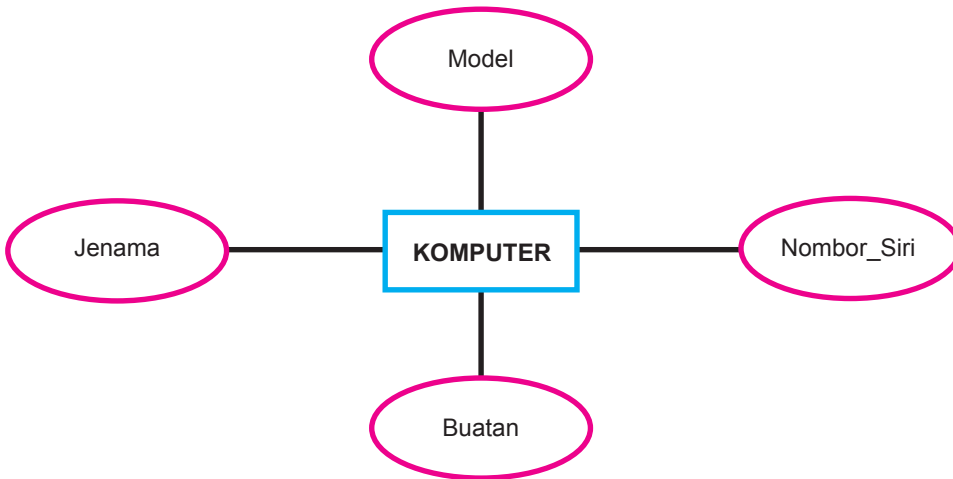


**Contoh 4.1** Rajah 4.8 menunjukkan lakaran entiti-entiti bagi persekitaran bilik darjah dengan menggunakan simbol yang betul.



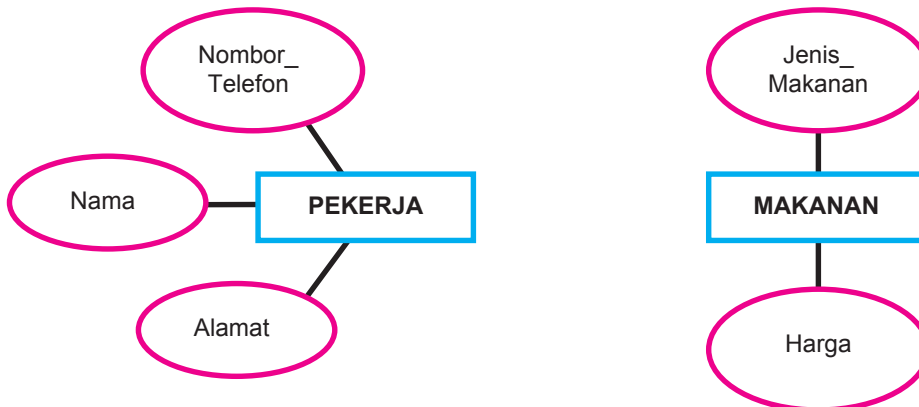
**Rajah 4.8** Lakaran entiti-entiti bagi persekitaran bilik darjah

**Contoh 4.2** Rajah 4.9 menunjukkan lakaran entiti dan atribut-atribut bagi persekitaran makmal komputer dengan menggunakan simbol yang betul.



**Rajah 4.9** Lakaran entiti dan atribut-atribut bagi persekitaran makmal komputer

**Contoh 4.3** Rajah 4.10 menunjukkan lakaran dua entiti dan atribut-atribut bagi persekitaran kantin sekolah dengan menggunakan simbol yang betul.



**Rajah 4.10** Lakaran entiti dan atribut-atribut bagi persekitaran kantin sekolah

Dalam pangkalan data hubungan, setiap entiti diwakili oleh satu jadual (juga disebut jadual hubungan). Setiap jadual hubungan terdiri daripada beberapa lajur yang dikenali sebagai medan. Setiap lajur atau medan pula mewakili satu atribut. Setiap baris dalam entiti mewakili satu rekod bagi seorang murid. Satu rekod mengandungi data bagi setiap atribut yang ada dalam entiti. Jadual 4.1 menunjukkan perwakilan entiti murid dan atributnya dalam pangkalan data hubungan.

**Jadual 4.1** Perwakilan entiti dan atribut dalam pangkalan data hubungan

		Atribut (lajur atau medan)					
		ID_Murid	Nama	Tarikh_Lahir	Jantina	Bangsa	Alamat
Rekod	→	2016020	Lee Yong Yong	4.10.01	Lelaki	Cina	12, Jln. Labu Jaya, Seremban
	→	2016059	Maniam a/l Manikam	8.3.01	Lelaki	India	30, Jln. Manis, Seremban
	→	2016321	Zulkifli bin Zamani	12.2.01	Lelaki	Melayu	7, Jln. Ampang, KL
	→	2016288	Elizabeth Paul	30.4.01	Perempuan	Dusun	82, Lrg Belimbing 2, Tmn ABC



## Aktiviti 4.1



**Aktiviti Berpasangan**

### Kenal Pasti Entiti dan Atribut-atribut



Kaedah  
*Think-Pair-Share*

1. Kaji jadual di bawah. Kenal pasti dan lakarkan entiti dan atribut-atribut berdasarkan data dalam jadual.

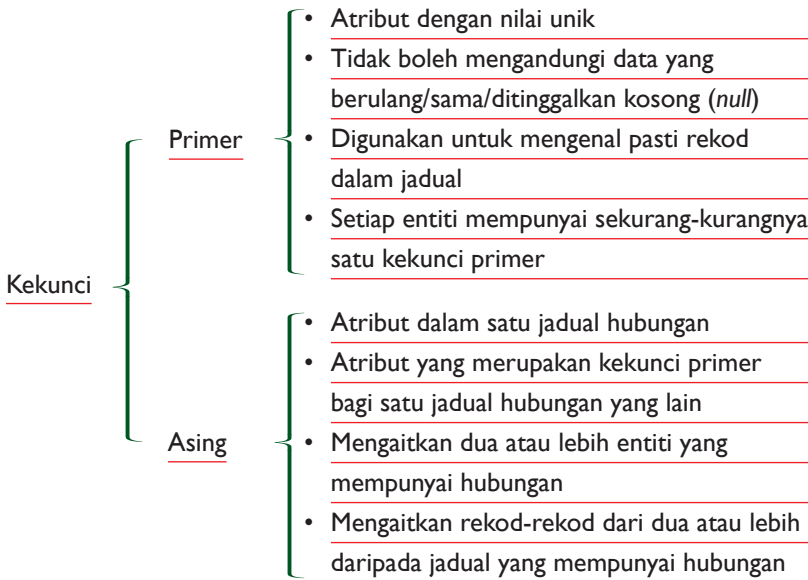
MURID				
No_Daftar	Nama_Murid	Kategori	Rumah_Sukan	
1111	AKASHAH BIN ZAINAL	LELAKI 1	GARUDA	
1112	KIRTHANA A/P MUTHU	PEREMPUAN 1	HELANG	
1113	LEE CHONG ZHE	LELAKI 2	RAJAWALI	
1114	YAP MEI LING	PEREMPUAN 2	KASAWARI	

2. Kongsikan idea lakaran-lakaran yang telah dibuat bersama-sama rakan anda.
3. Bentangkan hasil lakaran kepada semua murid di dalam kelas.
4. Berdasarkan perbentangan, murid diminta menghasilkan lakaran yang mempunyai dua entiti dalam persekitaran kedai runcit.



### 4.1.3 Kekunci Primer dan Kekunci Asing

Entiti mengandungi rekod-rekod. Setiap baris dalam satu entiti merupakan satu rekod. Rekod merupakan himpunan atribut yang berhubungan antara satu sama lain. Rajah 4.11 menunjukkan dua jenis kekunci dalam sistem pangkalan data.



Rajah 4.11 Dua jenis kekunci dalam sistem pangkalan data

Rajah 4.12 menunjukkan kepentingan kekunci primer dalam satu sistem pangkalan data.



Rajah 4.12 Kepentingan kekunci primer

### Standard Pembelajaran

Murid boleh:

- 4.1.3** Mengetahui dan menerangkan kekunci primer dan kekunci asing.

Kekunci primer dan kekunci asing sebuah pangkalan data



[goo.gl/jQvzhc](https://goo.gl/jQvzhc)



### Tahukah Anda?

Kekunci primer kebiasaannya terletak pada medan pertama sebelah kiri jadual manakala kekunci asing terletak pada medan kedua selepas kekunci primer.

### Uji Minda 4.3

Bolehkah wajah anda dijadikan sebagai kekunci primer? Berikan alasan kepada jawapan anda.

## Contoh 4.4

Pengenalpastian kekunci primer dan kekunci asing bagi entiti-entiti BUTIR\_MURID dan BUTIR\_PENJAGA.

ID_Murid	Nama	Jantina	Kaum	Agama
T17-124	DENISH RAIMI BIN MALIK	LELAKI	MELAYU	ISLAM
T17-125	TAN CHEN HONG	LELAKI	CINA	BUDDHA
T17-182	NURUL AIN BINTI AHMAD	PEREMPUAN	MELAYU	ISLAM
T17-201	PRIYA DAVID	PEREMPUAN	INDIA	KRISTIAN
T17-202	ARUL A/L PARAM	LELAKI	INDIA	HINDU
T17-225	RACHEL LIM YEE LING	PEREMPUAN	CINA	KRISTIAN

Rajah 4.13 Entiti BUTIR\_MURID

No_Daftar	ID_Murid	Nama_Ibu	Nama_Bapa	No_Telefon
1111	T17-125	YEOH MIN LI	TAN TZE SOOK	009-1234567
1112	T17-201	DIYVA A/P SAMY	DAVID THOMAS	009-7654321
1113	T17-225	GAN PUI MUI	RICHARD LIM KAH PEW	009-1122334
1114	T17-124	SITI NOOR BINTI ALI	MALIK BIN RASHID	009-2211334
1115	T17-182	SARAH BINTI HASSAN	AHMAD BIN RAHMAN	009-3322115
1116	T17-202	KARISHMA A/P SIVA	PARAM A/L GOPAL	009-4433115

Rajah 4.14 Entiti BUTIR\_PENJAGA

### Penyelesaian:

Entiti BUTIR\_MURID : Kekunci primer ialah ID\_Murid

Entiti BUTIR\_PENJAGA : Kekunci primer ialah No\_Daftar  
Kekunci asing ialah ID\_Murid

### Uji Minda 4.4

Anda telah diberikan satu borang yang mengandungi maklumat seperti nama, jantina, kaum, alamat rumah dan nombor telefon. Pada pendapat anda, manakah yang paling sesuai untuk dijadikan kekunci primer?



## Aktiviti 4.2



### Aktiviti Kumpulan

#### Eksplorasi Kekunci Primer dan Kekunci Asing



Kaedah *Three Stray One Stay*

1. Teliti pernyataan di bawah.

Puan Ainie menghadapi masalah untuk menyediakan senarai nama murid yang mengambil bahagian dalam acara sukan tahunan berdasarkan rumah sukan dan kategori. Dua entiti yang akan digunakan ialah MURID dan RUMAH\_SUKAN.

2. Kenal pasti atribut-atribut yang sesuai bagi entiti MURID dan RUMAH\_SUKAN.
3. Tulis atribut yang boleh dijadikan kekunci primer dan kekunci asing bagi kedua-dua entiti.
4. Catatkan dapatan anda dalam bentuk jadual hubungan pada kertas putih bersaiz A3.
5. Seorang murid daripada setiap kumpulan dilantik sebagai ketua kumpulan.
6. Ketua kumpulan akan membuat pembentangan kepada semua murid manakala ahli kumpulan yang lain perlu bergerak untuk melihat hasil kerja kumpulan-kumpulan yang lain.
7. Ulang langkah 6 sehingga semua ahli kumpulan telah mendengar pembentangan daripada ketua kumpulan yang lain.

### Tahukah Anda

Terdapat perbezaan antara kekunci primer dan kekunci asing.

Aspek	Kekunci primer	Kekunci asing
Input	Tidak boleh menerima nilai kosong ( <i>null</i> )	Boleh menerima beberapa nilai kosong ( <i>null</i> )
Bilangan kekunci dalam jadual	Satu sahaja	Boleh lebih daripada satu

## Standard Pembelajaran

Murid boleh:

- 4.1.4** Mengenal pasti dan menjelaskan kekardinalan (*cardinality*) antara entiti dalam hubungan:
- one to one* (1:1)
  - one to many* (1:M)

4.1.4

## Kekardinalan (*Cardinality*) antara Entiti dalam Hubungan

**Kekardinalan** merujuk kepada perhubungan antara entiti-entiti dalam satu pangkalan data. Terdapat beberapa jenis kekardinalan yang digunakan untuk menghubungkan satu entiti dengan satu entiti yang lain. Ini termasuklah kekardinalan *one to one* (1:1) dan *one to many* (1:M).

Perhubungan antara dua entiti lazimnya diwakili dengan menggunakan kata kerja. Dalam gambar rajah perhubungan entiti (*Entity Relationship Diagram – ERD*), hubungan dilukis dengan simbol rombus. Jadual 4.2 menunjukkan penerangan untuk jenis-jenis kekardinalan berserta contoh.

**Jadual 4.2** Penerangan untuk jenis-jenis kekardinalan berserta contoh

### Kekardinalan *one to one* (1:1) atau dikenali hubungan satu-ke-satu

Satu entiti yang berhubung dengan hanya satu entiti yang lain dan sebaliknya.

#### Contoh:

Seorang pengetua akan menguruskan sebuah sekolah dan sebuah sekolah akan hanya mempunyai seorang pengetua. Rajah berikut menunjukkan gambar rajah perhubungan entiti (ERD) antara satu entiti dengan satu entiti.



Dalam pangkalan data hubungan, ini bermaksud setiap rekod dalam satu jadual mempunyai hubungan dengan satu rekod sahaja dalam jadual yang lain, dan sebaliknya.

PENGETUA		
No_Pegawai	Nama	Kod_Sekolah
A12340	NORIAH BINTI ZAKI	AAA5678
A12341	JOHARI BIN ABD SAMAD	AAA5679
A12342	THANABALAN A/L SIVAM	AAA5680
A12343	ZAHRAA BINTI ZAINAL	AAA5681
A12344	LIM BENG HENG	AAA5682

SEKOLAH				
Kod_Sekolah	No_Pegawai	Nama	Bilangan_Guru	Bilangan_Murid
AAA5678	A12340	SMK BUNGA ORKID	46	594
AAA5679	A12341	SMK SERI ANGKASA	90	1188
AAA5680	A12342	SMK SETIA	60	767
AAA5681	A12343	SMK PERDANA	57	713
AAA5682	A12344	SMK BITARA	82	1069

Satu rekod daripada entiti PENGETUA hanya akan mempunyai kekardinalan (1:1) dengan satu rekod daripada entiti SEKOLAH.

## Kekardinalan *one to many* (1:M) atau dikenali hubungan satu-ke-banyak

Satu entiti yang mempunyai hubungan dengan satu atau lebih entiti yang lain.

### Contoh:

Seorang murid dapat meminjam banyak buku tetapi sebuah buku hanya boleh dipinjam oleh seorang murid sahaja. Rajah berikut menunjukkan perhubungan antara satu entiti dengan lebih daripada satu entiti yang lain.



Dalam pangkalan data hubungan, ini bermaksud setiap rekod dalam sesuatu jadual mempunyai hubungan dengan satu atau lebih daripada satu rekod dalam jadual yang lain, dan sebaliknya.

MURID				
ID_Murid	Nama	Jantina	Nama_Kelas	
PSS001	NUR ATHIRAH BINTI AZMI	PEREMPUAN	3 ANGGERIK	
PSS002	ADAM AQIL BIN ISMAIL	LELAKI	3 ANGGERIK	
PSS003	WONG SOO LEE	PEREMPUAN	3 ANGGERIK	
PSS004	NITHIAN A/L KOLANTHAI	LELAKI	3 BAKAWALI	
PSS005	DHIA ARISSA BINTI RAHIM	PEREMPUAN	3 BAKAWALI	

BUKU					
Kod_Buku	Judul	ID_Murid	ISBN	Tarikh_Pinjaman	
A101	Buku Teks KSSM Bahasa Melayu Tingkatan 2	PSS001	9789834916466	15/1/2018	
A102	Buku Teks KSSM Pendidikan Moral Tingkatan 2	PSS003	9789834916497	15/1/2018	
BF101	Buku Teks KSSM Sains Tingkatan 2	PSS005	9789671447260	15/1/2018	
BF102	Buku Teks KSSM Matematik Tingkatan 2	PSS004	9789672031055	15/1/2018	
F101	Buku Teks KSSM Sejarah Tingkatan 2	PSS002	9789834916473	15/1/2018	
F102	Buku Teks KSSM Ekonomi Tingkatan 4	PSS002	9789833485109	15/1/2018	

Satu rekod daripada entiti MURID hanya akan mempunyai hubungan (1:M) dengan satu atau lebih rekod daripada entiti BUKU.

## Tahukah Anda ?

Selain dua jenis kekardinalan yang telah anda pelajari, masih terdapat satu jenis lagi kekardinalan, iaitu *many to many* (M:N).

### Contoh 4.5

Lakaran jenis kekardinalan dalam lukisan gambar rajah perhubungan entiti bagi situasi-situasi berikut.

- (a) Seorang guru menjadi guru kelas bagi satu kelas sahaja.
- (b) Seorang murid boleh mendaftar lebih daripada satu acara sukan.

#### Penyelesaian:

- (a) Jenis kekardinalan ialah *one to one* (1:1)



- (b) Jenis kekardinalan ialah *one to many* (1:M)



## Aktiviti 4.3



### Aktiviti Kumpulan

#### Jejaki Kekardinalan



Kaedah  
Gallery Walk

1. Teliti pernyataan-pernyataan di bawah. Berdasarkan syarat pada setiap pernyataan, lakukan aktiviti yang seterusnya.

Murid yang mengambil mata pelajaran Asas Sains Komputer telah diminta oleh guru mereka untuk menyediakan folio digital tentang pangkalan data. Semua murid perlu mematuhi syarat-syarat berikut:

- (a) Setiap murid perlu menyiapkan satu tugasan.
  - (b) Setiap murid boleh merujuk kepada pelbagai jenis sumber untuk mendapatkan maklumat.
  - (c) Guru menyemak folio yang dihantar oleh semua murid.
2. Kenal pasti jenis kekardinalan bagi setiap pernyataan.
  3. Sediakan satu peta pemikiran i-THINK untuk:
    - (a) Mengkategorikan jenis kekardinalan.
    - (b) Melukis rajah perhubungan entiti (ERD) mengikut kategori di 3 (a).
  4. Seorang ahli daripada setiap kumpulan dilantik sebagai pembentang.
  5. Setiap pembentang harus berada berhampiran meja masing-masing untuk membuat pembentangan manakala ahli-ahli kumpulan yang lain akan bergerak untuk melihat hasil kerja kumpulan yang lain.
  6. Ulang langkah 5 sehingga semua kumpulan telah mendengar pembentangan daripada kumpulan yang lain.



## 4.1.5

## Membina Pangkalan Data

Pangkalan data dibina dengan menggunakan perisian sistem pengurusan pangkalan data atau lebih dikenali sebagai DBMS (*Database Management System*). Satu contoh perisian DBMS ialah *Microsoft Access*. Perisian ini dilengkapi dengan berbagai-bagai alat perisian (*software tools*) bagi memudahkan pembangunan, pengurusan dan penyelenggaraan pangkalan data hubungan (*relational database*).

Jadual 4.3 menunjukkan objek-objek pangkalan data *Microsoft Access* yang wujud dan penerangan tentang kegunaan objek-objek ini.

## Standard Pembelajaran

Murid boleh:

- 4.1.5** Membina pangkalan data yang terdiri daripada:
- (i) entiti (*table*)
  - (ii) atribut (*field*)
  - (iii) hubungan

## Tahukah Anda ?

*Microsoft's SQL Server, Oracle, Informix, DB2, Sybase, Postgress* dan *MySQL* boleh digunakan untuk membangunkan pangkalan data.



## Uji Minda 4.5

Anda telah mengetahui bahawa *Microsoft Access* merupakan salah satu aplikasi DBMS. Berikan contoh aplikasi DBMS selain *Microsoft Access* dengan menyatakan kelebihan aplikasi tersebut.

**Jadual 4.3** Objek-objek pangkalan data dalam perisian aplikasi *Microsoft Access*

Jenis objek	Penerangan tentang kegunaan objek
Jadual ( <i>Table</i> )	<ul style="list-style-type: none"> <li>• Satu entiti diwakili oleh satu jadual. Atribut-atribut entiti yang berkenaan disimpan dalam lajur (atau medan) pada jadual.</li> <li>• Menyimpan data mengikut baris. Setiap baris data dipanggil satu rekod.</li> </ul>
Pertanyaan ( <i>Query</i> )	<ul style="list-style-type: none"> <li>• Meminta data yang tertentu daripada satu atau lebih daripada satu jadual.</li> <li>• Data yang diperolehi boleh dicetak atau dipaparkan.</li> <li>• Data yang dipaparkan boleh dibuat perubahan.</li> </ul>
Borang ( <i>Form</i> )	<ul style="list-style-type: none"> <li>• Merupakan antara muka antara pengguna dan komputer.</li> <li>• Memudahkan kemasukan data.</li> <li>• Memaparkan data daripada satu atau lebih daripada satu jadual dalam bentuk selain daripada baris dan lajur.</li> <li>• Memaparkan data daripada pertanyaan.</li> <li>• Mengubah data yang dipaparkan melalui satu borang.</li> <li>• Mengubah, melihat atau memasukkan data.</li> </ul>
Laporan ( <i>Report</i> )	<ul style="list-style-type: none"> <li>• Menyediakan laporan berdasarkan data dalam satu atau lebih daripada satu jadual entiti atau berdasarkan satu pertanyaan.</li> <li>• Laporan boleh dicetak atau menganalisis data.</li> </ul>

Pangkalan data dibina berdasarkan beberapa langkah. Rajah 4.15 menunjukkan langkah-langkah untuk membina pangkalan data berdasarkan situasi berikut:

SMK Sinar Gemilang mengalami masalah untuk mendapatkan analisis keputusan peperiksaan murid-murid di sekolah tersebut dengan cepat kerana markah murid-murid perlu dikira secara manual. Bagi menyelesaikan masalah ini, anda bercadang untuk menghasilkan Sistem Maklumat Peperiksaan atau dikenali sebagai SisMaP.

**Langkah 1: Tentukan tujuan pembinaan pangkalan data**

- untuk menghasilkan analisis keputusan peperiksaan dengan cepat

**Langkah 2: Menentukan entiti yang diperlukan oleh SisMaP**

- entiti MURID
- entiti MARKAH

**Langkah 3: Menentukan atribut yang diperlukan bagi SisMaP**

- atribut-atribut bagi entiti MURID ialah ID\_Murid, Nama\_Murid, Jantina, Kaum, Nama\_Kelas, Nama\_Guru\_Kelas
- atribut-atribut bagi entiti MARKAH ialah ID\_Markah, ID\_Murid, BM, BI, Sn, Mate, Sej, Geo, ASK

**Langkah 4: Menentukan hubungan dan kekardinalan yang diperlukan bagi SisMaP**

- Entiti MURID  
Kekunci primer: ID\_Murid
- Entiti MARKAH  
Kekunci primer: ID\_Markah  
Kekunci asing: ID\_Murid
- Kekardinalan  
Hubungan *one to many* (1:M)

**Langkah 5 : Menghasilkan pangkalan data SisMaP**

- Jadual  
Membina dua jadual berdasarkan entiti MURID dan entiti MARKAH
- Borang  
Membina dua borang berdasarkan jadual bagi entiti MURID dan entiti MARKAH
- Pertanyaan (*Query*)  
Membina beberapa pertanyaan (*Query*) berdasarkan entiti MURID dan entiti MARKAH menggunakan SQL
- Laporan  
Menghasilkan laporan berdasarkan hasil pertanyaan (*Query*)

**Rajah 4.15** Langkah-langkah pembinaan pangkalan data



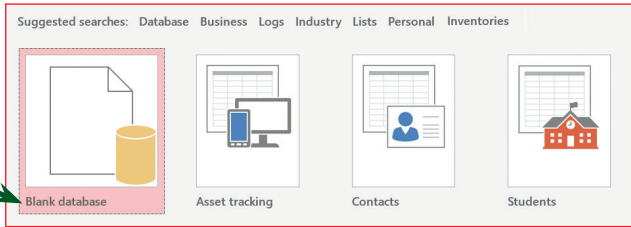
**Uji Minda 4.6**


Pangkalan data yang menghasilkan sistem maklumat yang efisien direka bentuk melalui beberapa fasa. Nyatakan fasa-fasa dalam pembangunan sistem maklumat.

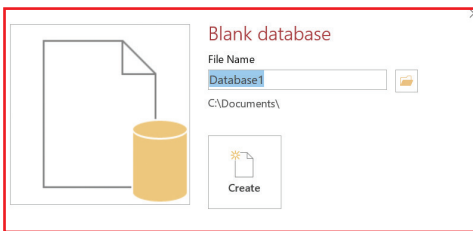
Capaian kepada perisian aplikasi *Microsoft Access*

**Langkah 1** Lancarkan perisian aplikasi *Microsoft Access* dan pilih pangkalan data yang kosong.

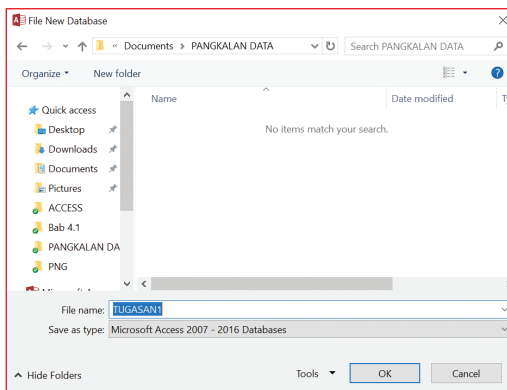
Klik **Blank database**



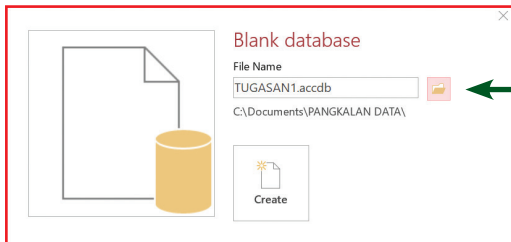
**Langkah 2** Klik ikon  untuk pilih lokasi simpanan pangkalan data anda.



**Langkah 3** Bina satu folder baharu bernama **PANGKALAN DATA** dalam **Documents** komputer anda. Namakan fail anda **TUGASAN1** dan klik **OK**.



**Langkah 4** Paparan berikut akan dipaparkan. Klik **Create** untuk membina pangkalan data **TUGASAN1**.



Klik ini untuk pilih lokasi simpanan pangkalan data.

Pengenalan kepada perisian aplikasi *Microsoft Access*



[goo.gl/kx8Y7b](http://goo.gl/kx8Y7b)

**Imbas Saya**

**Tahukah Anda?**

Perisian aplikasi *Microsoft Access* dengan versi yang berbeza akan mempunyai fitur-fitur yang berbeza. Kebiasaannya, versi yang terkini lebih mesra pengguna berbanding dengan versi yang lama.

**Uji Minda 4.7**

Nyatakan versi-versi *Microsoft Access* yang wujud sehingga kini.

**Rajah 4.16** Langkah-langkah pencapaian kepada perisian aplikasi *Microsoft Access*

Membina jadual dengan menggunakan perisian aplikasi *Microsoft Access*.



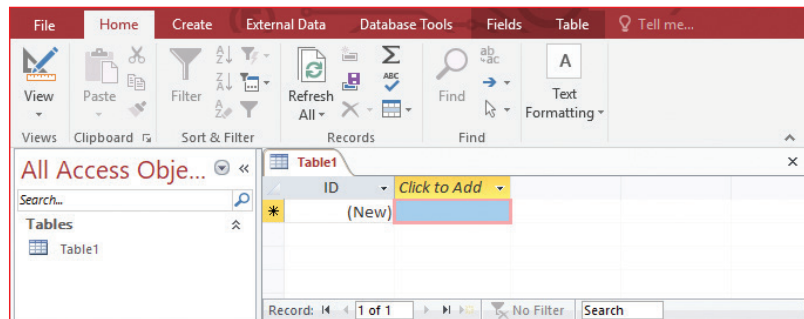
[goo.gl/4aj9D3](http://goo.gl/4aj9D3)




### (i) Langkah-langkah membina pangkalan data yang terdiri daripada entiti (*table*)

Dalam satu pangkalan data, jadual (*table*) digunakan untuk menyimpan data bagi sesuatu entiti. Rajah 4.17 menunjukkan langkah-langkah untuk membina jadual.

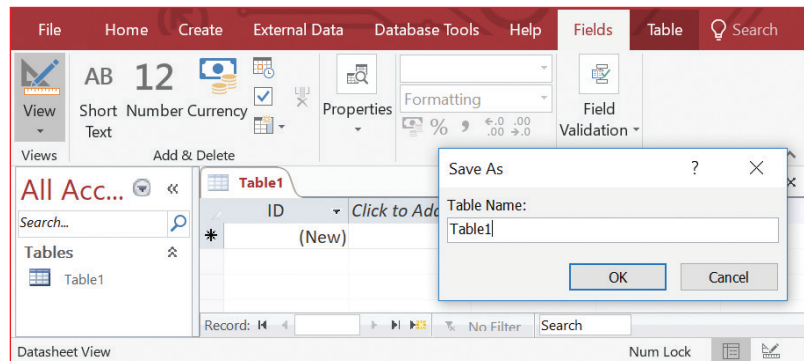
**Langkah 1** Setelah melengkapkan langkah-langkah dalam Rajah 4.16, paparan di bawah diperhatikan.



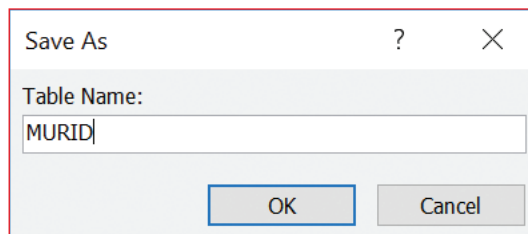
**Langkah 2** Klik ikon  untuk mendapatkan paparan seperti berikut.

**Tahukah Anda?**

nama entiti sama atribut  
lain table seperti field



**Langkah 3** Ubahkan nama Table1 kepada MURID dan klik OK.



**Rajah 4.17** Langkah-langkah membina jadual

**(ii)** Langkah-langkah membina pangkalan data yang terdiri daripada atribut (*field*)

Pada peringkat ini, medan-medan (lajur-lajur jadual) dicipta untuk mewakili atribut-atribut bagi entiti. Pada masa yang sama, jenis data yang bersesuaian bagi setiap medan (atribut) perlu ditetapkan. Jadual 4.4 menunjukkan beberapa jenis data yang boleh ditetapkan dalam perisian aplikasi *Microsoft Access*.

**Jadual 4.4** Jenis-jenis data

Jenis data	Penerangan jenis data
<i>AutoNumber</i>	Menjana pembilang secara automatik dalam susunan menaik.
<i>Text/Short Text</i>	Panjang yang terhad kepada maksimum 255 aksara sahaja.
<i>Memo/Long Text</i>	Boleh menyimpan teks sehingga satu Gigabait.
<i>Number</i>	Hanya menyimpan nilai angka.
<i>Date/Time</i>	Nilai tarikh dan masa bagi tahun bermula daripada 100 hingga 9999.

### Tahukah Anda?

Himpunan atribut dikenali sebagai rekod manakala himpunan rekod dikenali sebagai entiti.

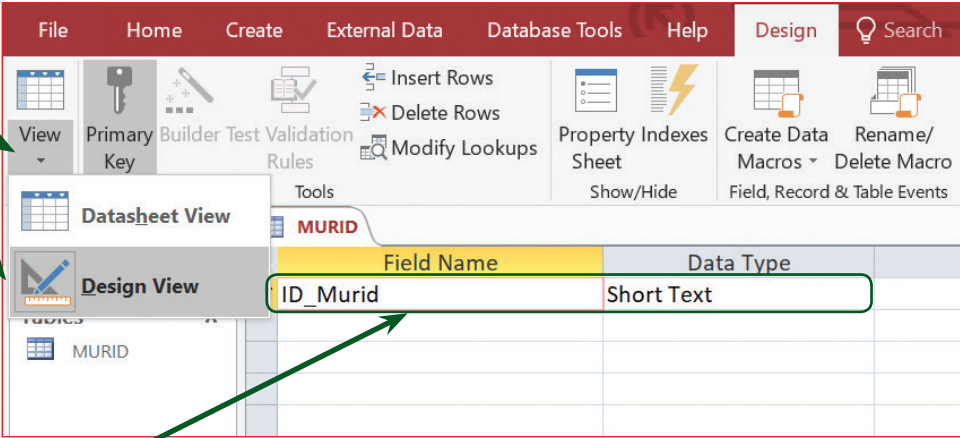
### Tahukah Anda?

Perisian aplikasi *Microsoft Access* tidak membenarkan nama atribut mengandungi simbol-simbol seperti titik (.), tanda seru (!) dan tanda sama dengan (=).

Rajah 4.18 menunjukkan langkah-langkah mencipta medan-medan bagi jadual (entiti) yang dibina.

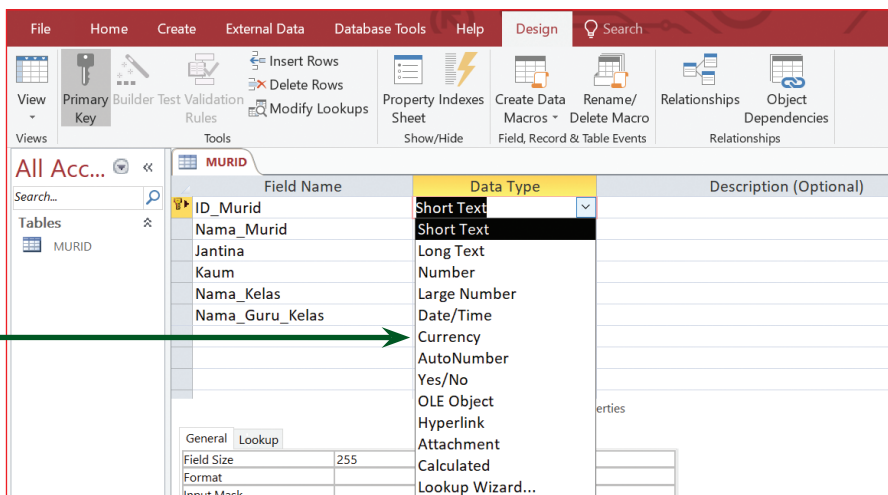
**Langkah 1** Klik ikon **View** dan pilih **Design View**. Kemudian, masukkan nama medan dan jenis data.

**1. Klik View dan pilih Design View.**



**2. Masukkan nama medan dan jenis data.**

**Langkah 2** Medan-medan (lajur-lajur) bagi atribut-atribut yang seterusnya boleh dicipta dengan menaip nama atribut yang dikehendaki pada ruangan **Field Name** pada baris seterusnya. Bagi jenis data pula, anda hanya perlu memilih daripada senarai yang telah disediakan.



Senarai jenis data yang sedia ada.

**Rajah 4.18** Langkah-langkah mencipta medan-medan bagi jadual (entiti) yang dibina

### Tahukah Anda?

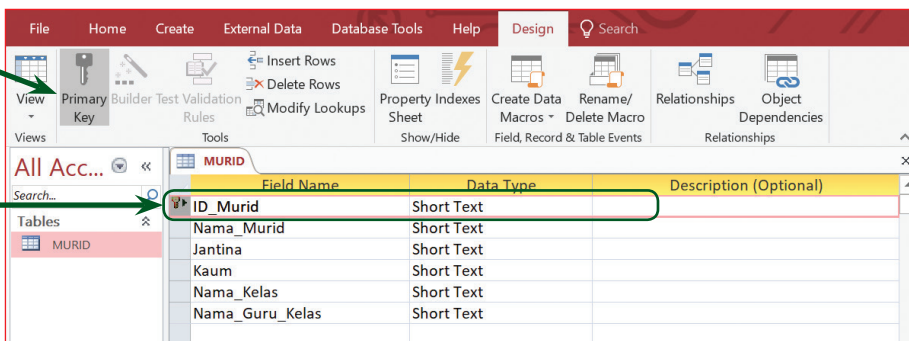
Dalam perisian aplikasi *Microsoft Access*, nama bagi atribut yang lebih daripada satu perkataan boleh dijarakkan tetapi dalam penulisan SQL tidak boleh dijarakkan.

### Langkah-langkah menetapkan kekunci primer

Perisian aplikasi *Microsoft Access* secara automatik akan menjadikan atribut pertama sebagai kekunci primer. Walau bagaimanapun, anda boleh menetapkan sendiri medan lain untuk dijadikan sebagai kekunci primer. Rajah 4.19 menunjukkan langkah-langkah untuk menetapkan kekunci primer.

**Langkah 1** Pilih atribut yang mempunyai nilai-nilai unik sebagai kekunci primer. Klik pada nama medan yang berpadanan dengan atribut ini.

**Langkah 2** Klik **Primary Key**.



2. Klik **Primary Key**.

1. Pilih atribut unik sebagai kekunci primer.

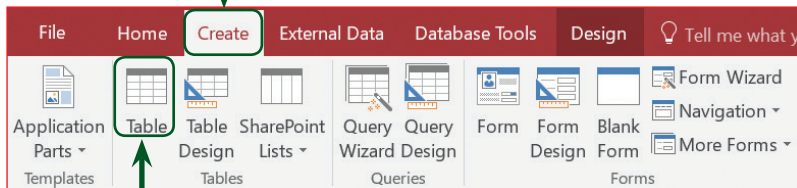
**Rajah 4.19** Langkah-langkah menetapkan kekunci primer

## Langkah-langkah membina jadual (*table*) yang seterusnya

Jadual untuk entiti yang kedua dan seterusnya boleh dibina dengan menggunakan ikon yang terdapat pada bar menu (*menu bar*). Pemberian nama kepada suatu jadual perlu bersesuaian dengan entiti berkenaan serta atribut-atribut yang terdapat di dalamnya. Rajah 4.20 menunjukkan langkah-langkah untuk membina dan menamakan jadual bagi entiti yang kedua.

**Langkah 1** Bina satu jadual baharu dengan klik **Create** di bar menu. Kemudian, klik ikon **Table**.

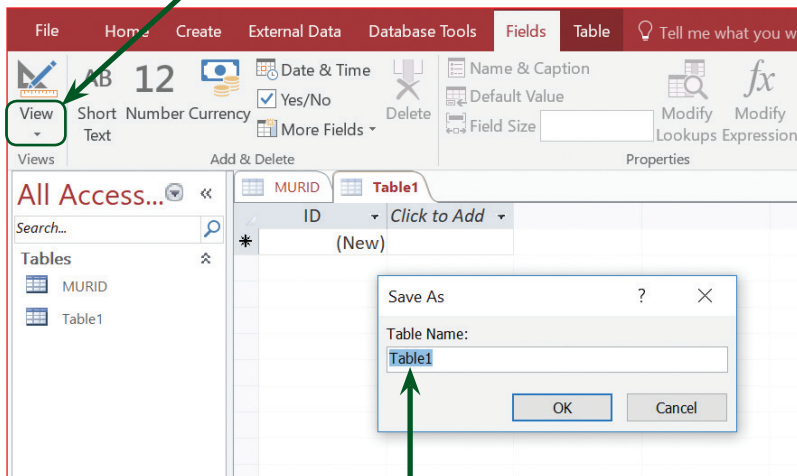
1. Klik **Create** di bar menu.



2. Klik ikon **Table**

**Langkah 2** Klik **View** dan pilih **Design View**. Tetingkap **Save As** akan dipaparkan. Namakan jadual ini sebagai **MARKAH** dan klik **OK**.

1. Klik **View** dan pilih **Design View**.



2. Namakan jadual sebagai **MARKAH**.

**Rajah 4.20** Langkah-langkah membina dan menamakan jadual bagi entiti yang kedua

## Tahukah Anda?

Tiada had maksimum bagi entiti yang boleh wujud dalam satu pangkalan data kerana bilangan entiti bergantung kepada keperluan pangkalan data tersebut.

Kekunci primer



[goo.gl/utN4my](http://goo.gl/utN4my)

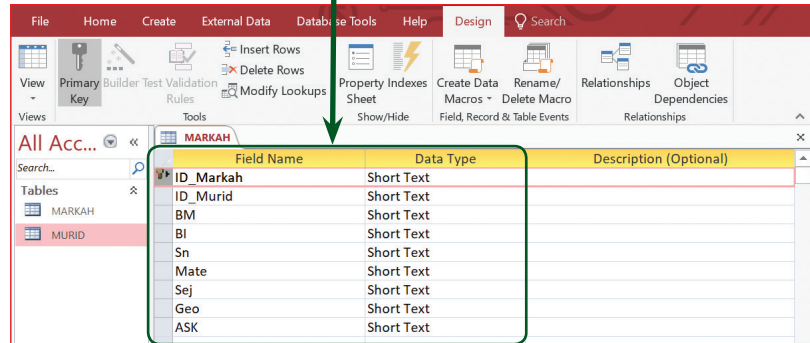


## Tahukah Anda?

Semua medan perlu mempunyai data dan tidak patut dibiarkan kosong (*null*).

Rajah 4.21 menunjukkan medan-medan yang ditambahkan kepada jadual MARKAH dan jenis data yang sesuai ditetapkan bagi medan-medan itu.

Nama medan dan jenis data dimasukkan.



Field Name	Data Type	Description (Optional)
ID_Markah	Short Text	
ID_Murid	Short Text	
BM	Short Text	
BI	Short Text	
Sn	Short Text	
Mate	Short Text	
Sej	Short Text	
Geo	Short Text	
ASK	Short Text	

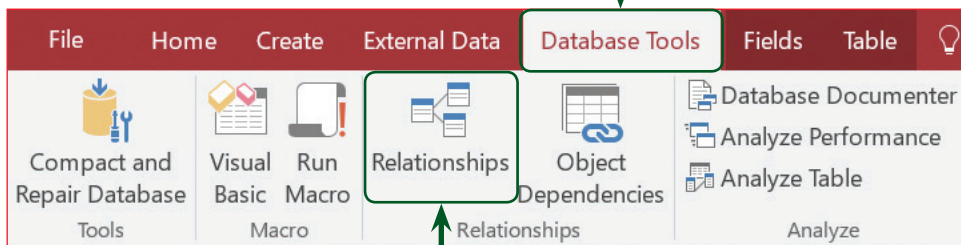
Rajah 4.21 Medan-medan untuk jadual MARKAH

### (iii) Langkah-langkah mewujudkan hubungan antara jadual (entiti)

Selepas semua jadual dibina dan data telah lengkap diisi, perhubungan kedua-dua jadual perlu dibina. Perhubungan antara entiti MURID dan entiti MARKAH ialah kekardinalan *one to one* (1:1) kerana setiap murid akan mempunyai satu set markah mengikut mata pelajaran masing-masing. Rajah 4.22 menunjukkan langkah-langkah untuk membina hubungan (*relationships*) bagi jadual MURID dan jadual MARKAH.

**Langkah 1** Klik **Database Tools** di bar menu. Kemudian, klik **Relationships**.

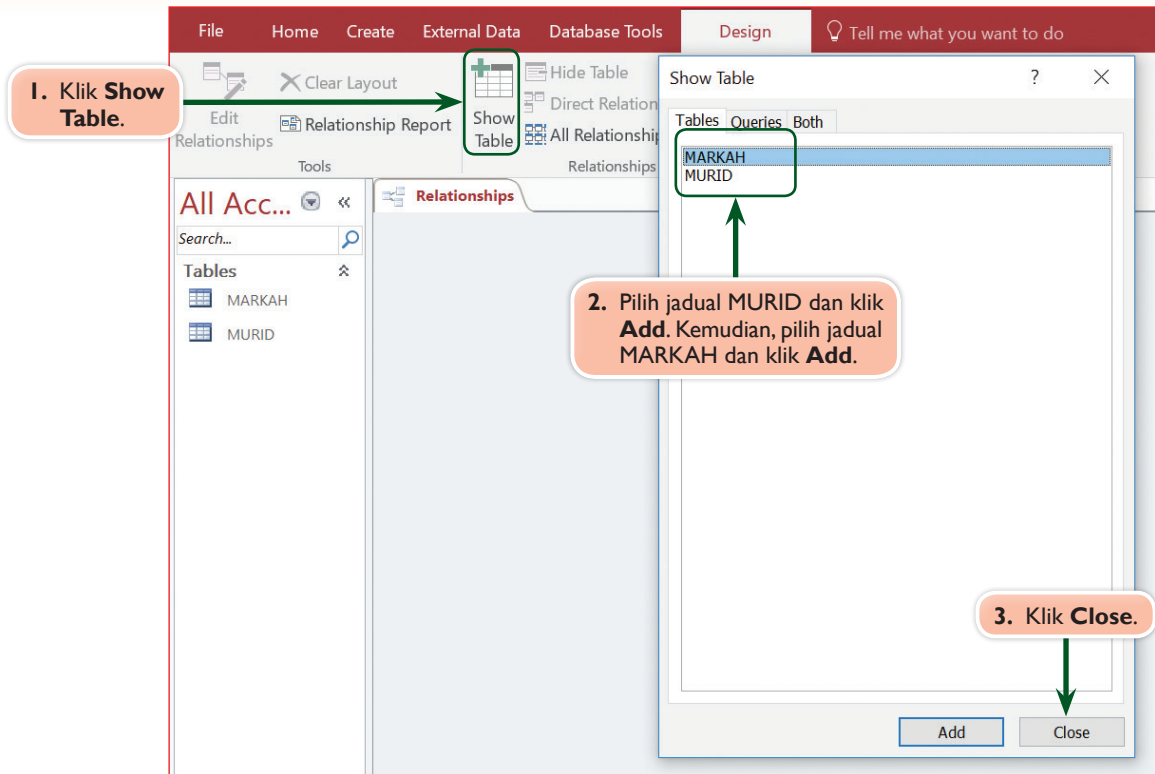
1. Klik **Database Tools** di bar menu.



2. Klik **Relationships**.



**Langkah 2** Klik **Show Table** dan paparan berikut akan diperhatikan. Pilih jadual **MURID** dan klik **Add**. Kemudian, pilih jadual **MARKAH** dan klik **Add**.



**Rajah 4.22** Langkah-langkah untuk membina hubungan (*relationships*) bagi jadual **MURID** dan jadual **MARKAH**

Perhatikan kedua-dua entiti **MARKAH** dan entiti **MURID** mempunyai atribut **ID\_Murid** serta menggunakan atribut ini sebagai kekunci primer. Oleh itu, atribut **ID\_Murid** sesuai digunakan untuk menghubungkan kedua-dua entiti ini dalam satu hubungan *one to many* (1:M). Rajah 4.23 menunjukkan langkah-langkah untuk menghubungkaitkan kedua-dua jadual **MURID** dan **MARKAH**.



**Langkah 1** Klik dan seret tetikus daripada atribut ID\_Murid pada entiti MURID sehingga menyentuh atribut ID\_Murid pada entiti MARKAH. Tetingkap Edit Relationships akan dipaparkan.

**Langkah 2** Dalam tetingkap Edit Relationships, klik petak Enforce Referential Integrity bagi membolehkan hubungan kekardinalan *one to many* (1:M) dipaparkan. Kemudian, klik Create.

**2. Klik ketiga-tiga kotak.**

- Pilihan kotak *Enforce Referential Integrity* memastikan bahawa kekunci asing pada suatu jadual adalah kekunci primer dalam jadual yang lain.
- Pilihan kotak *Cascade Update Related Fields* akan mengemaskini kekunci asing secara automatik pada suatu jadual jika kekunci primer dalam jadual yang lain berubah.
- Pilihan kotak *Cascade Delete Related Records* akan menghapuskan kekunci asing pada suatu jadual jika kekunci primer dalam jadual yang lain dihapus.

**1. Klik dan seret tetikus** daripada atribut ID\_Murid entiti MURID sehingga menyentuh atribut ID\_Murid pada atribut ID\_Murid entiti MARKAH.

**3. Klik Create.**

Hubungan kekardinalan *one to many* (1:M) dipaparkan.

**Langkah 3** Penetapan hubungan kekardinalan *one to many* (1:M) bagi kedua-dua jadual telah siap dibina. Paparan berikut yang menunjukkan jadual MURID telah berjaya dihubungkan dengan jadual MARKAH melalui kekunci primer ID\_Murid.

Kekardinalan *one to many* (1:M).

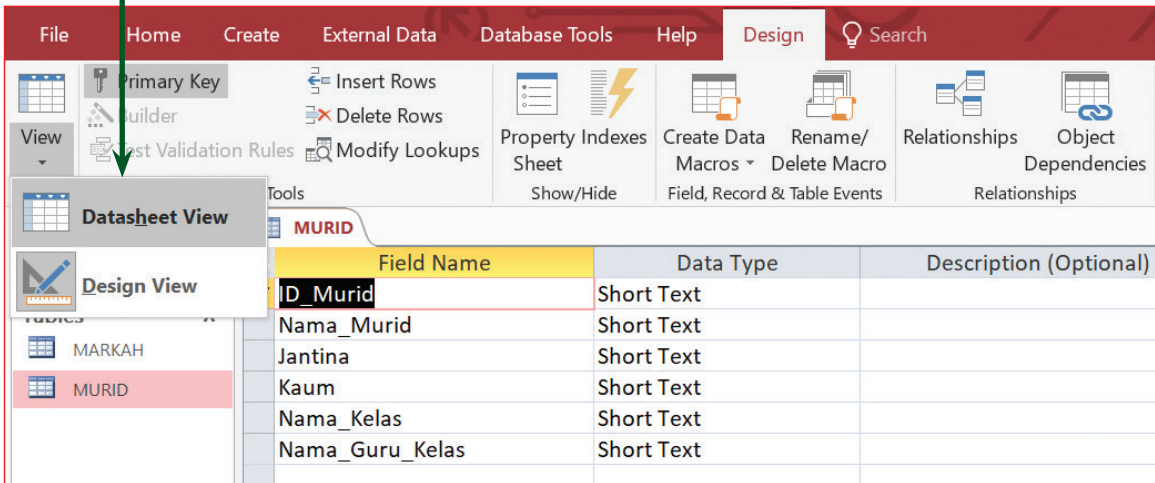
**Rajah 4.23** Langkah-langkah untuk menghubungkan kedua-dua jadual MURID dan MARKAH

## Langkah-langkah memasukkan data ke dalam jadual

Rajah 4.24 menunjukkan langkah-langkah untuk memasukkan data-data ke dalam jadual MURID dan jadual MARKAH.

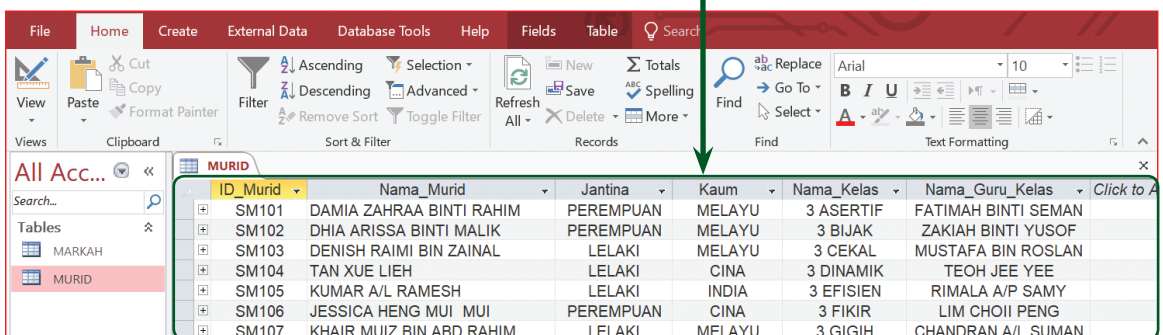
**Langkah 1** Klik View dan pilih Datasheet View.

Klik **View** dan pilih **Datasheet View**.



**Langkah 2** Setelah Datasheet View dibuka, anda boleh memasukkan data-data bagi semua medan untuk jadual MURID dan jadual MARKAH.

Data bagi semua medan untuk jadual MURID perlu dimasukkan.



Data bagi semua medan untuk jadual MARKAH perlu dimasukkan.

ID_Markah	ID_Murid	BM	BI	Sn	Mate	Sej	Geo	ASK
PT_SM101	SM101	45	50	TH	75	45	78	78
PT_SM102	SM102	56	85	87	68	84	90	79
PT_SM103	SM103	TH	58	78	65	89	85	79
PT_SM104	SM104	67	68	84	TH	77	TH	59
PT_SM105	SM105	77	67	65	87	46	58	58
PT_SM106	SM106	80	TH	54	76	85	81	85
PT_SM107	SM107	89	67	80	85	76	68	85

Rajah 4.24 Langkah-langkah untuk memasukkan data bagi dua jadual

Hubungan  
kekardinalan



[goo.gl/dkcnHU](http://goo.gl/dkcnHU)



### Uji Minda 4.8

Cadangkan perubahan untuk atribut entiti MARKAH supaya kekardinalan hubungan antara MURID dengan MARKAH menjadi *one to one* (1:1).



## Praktis Amali 4.1

Menghasilkan Pangkalan Data TUGASANI

1. Lancarkan perisian aplikasi *Microsoft Access*.
2. Bina Pangkalan Data TUGASANI.
3. Ikuti langkah-langkah dalam Rajah 4.16 hingga Rajah 4.24.
4. Bandingkan hasil Pangkalan Data anda bersama-sama rakan anda.
5. Setelah berjaya membina pangkalan data, tunjukkan kepada guru anda untuk disemak.



**Aktiviti 4.4****Aktiviti Berpasangan****Membina Satu Pangkalan Data**Kaedah  
*Think-Pair-Share*

Selaku Pengerusi dan Naib Pengerusi Kelab Komputer, anda diminta oleh guru penasihat untuk membangunkan sebuah **Pangkalan Data Kehadiran**.

1. Berdasarkan pernyataan di atas, anda diminta untuk mengenal pasti perkara-perkara yang berikut:
  - (a) Dua entiti untuk Pangkalan Data Kehadiran.
  - (b) Atribut-atribut bagi kedua-dua entiti.
  - (c) Hubungan kekardinalan bagi kedua-dua entiti.
2. Kemudian, lukiskan hasil kerja masing-masing dalam bentuk peta pemikiran i-THINK.
3. Secara berpasangan, murid-murid diminta berbincang dan berkongsi hasil kerja masing-masing. Murid digalakkan bertanya dan membaiki hasil kerja masing-masing dengan menulis di atas kertas yang sama.
4. Setelah masa yang diperuntukkan untuk perbincangan tamat, ketua kumpulan perlu membentangkan hasil kerja masing-masing.

**Praktis Amali 4.2****Menghasilkan Pangkalan Data Kehadiran yang Dibina Dalam Aktiviti 4.4**

1. Lancarkan perisian aplikasi *Microsoft Access*.
2. Berdasarkan hasil kerja dalam Aktiviti 4.4, bina Pangkalan Data Kehadiran.
3. Anda boleh merujuk Rajah 4.17 hingga Rajah 4.24 untuk langkah-langkah yang telah anda pelajari bagi membina pangkalan data.
4. Setelah berjaya membina pangkalan data, tunjuk kepada guru anda untuk disemak.

## Standard Pembelajaran

Murid boleh:

- 4.1.6** Menghasilkan borang (*form*) dan memasukkan data dalam jadual melalui borang bagi pangkalan data yang telah dibina.

4.1.6

## Menghasilkan Borang dan Memasukkan Data Melalui Borang

Anda boleh menggunakan borang (*form*) untuk menyusun dan mengurus rekod-rekod dalam pangkalan data dengan lebih efisien. Penggunaan borang membolehkan pengguna memasukkan dan mengubah suai data dengan mudah kerana borang berfungsi sebagai antara muka pengguna yang interaktif.

Perisian aplikasi *Microsoft Access* membolehkan anda menghasilkan borang dengan mudah. Ini kerana borang-borang dapat dijana secara automatik menggunakan **Form Wizard**. Borang yang dihasilkan digunakan untuk memasukkan, mengedit atau memaparkan data dan rekod dalam jadual hubungan sesuatu entiti. Rajah 4.25 menunjukkan langkah-langkah untuk menghasilkan BORANG MURID dengan menggunakan pangkalan data TUGASAN 1.

Pengenalan kepada borang

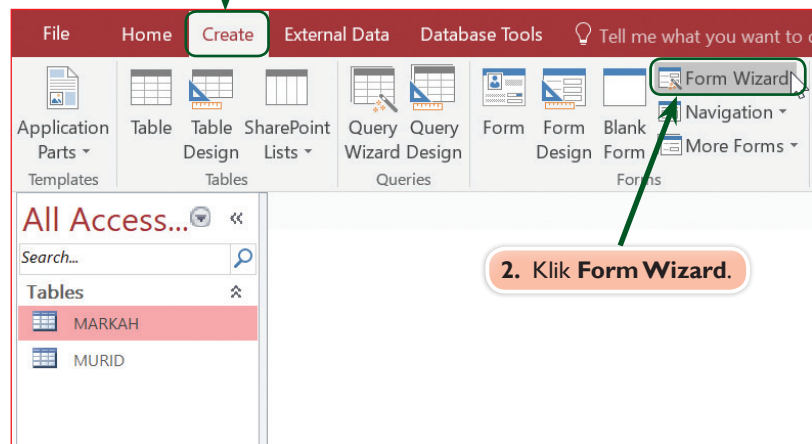


[goo.gl/t24Vqf](https://goo.gl/t24Vqf)

Imbas Saya

- Langkah 1** Buka pangkalan data TUGASAN1. Pada menu bar, klik pada tab **Create**. Kemudian, klik **Form Wizard**.

1. Klik **Create** di menu bar.

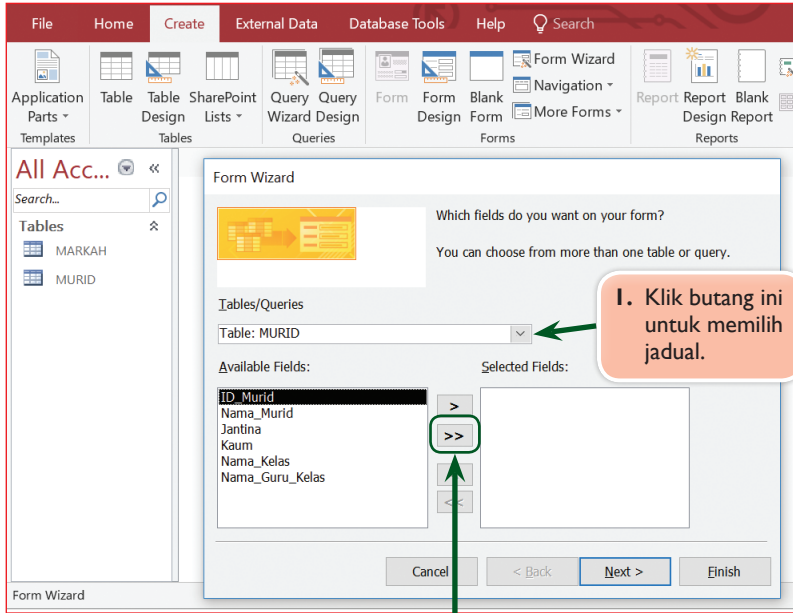


2. Klik **Form Wizard**.

Tahukah Anda?

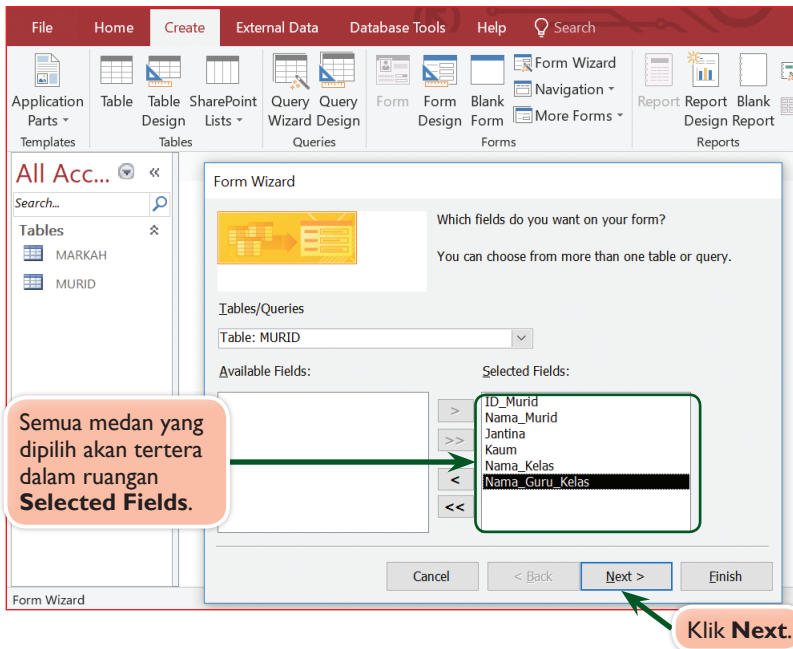
Kebiasaannya, semua medan dipilih untuk menghasilkan borang bagi tujuan kemasukan data ke dalam jadual.

**Langkah 2** Tetingkap Form Wizard akan dipaparkan. Pilih jadual MURID dan klik butang >> untuk memilih semua medan yang ada dalam jadual.



2. Klik butang ini untuk memilih semua medan yang ada dalam jadual. Ini bermaksud, semua atribut dalam entiti akan dipaparkan di dalam borang.

**Langkah 3** Paparan berikut akan diperhatikan. Klik Next untuk langkah yang seterusnya.



Semua medan yang dipilih akan tertera dalam ruangan **Selected Fields**.

### Tahukah Anda?

Simbol dan maksud

Simbol	Maksud
>	Pindah medan yang diklik kepada ruangan medan-medan terpilih.
>>	Pindah semua medan yang sedia ada dalam jadual kepada ruangan medan-medan terpilih.
<	Pindah balik medan yang diklik kepada ruangan medan-medan telah sedia ada.
<<	Pindah balik semua medan kepada ruangan medan-medan telah sedia ada.

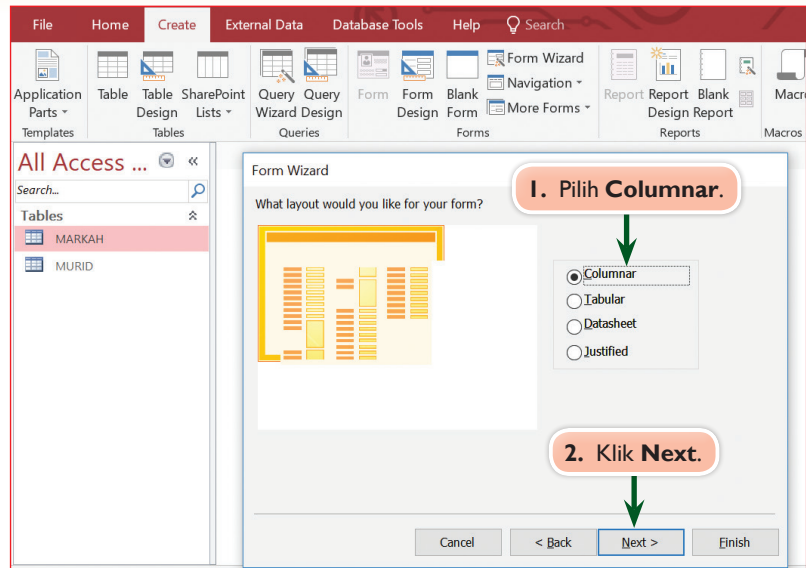
### Uji Minda 4.9

Pada pendapat anda, mengapakah adakala medan-medan daripada dua jadual yang berlainan tidak digabungkan untuk menghasilkan borang?

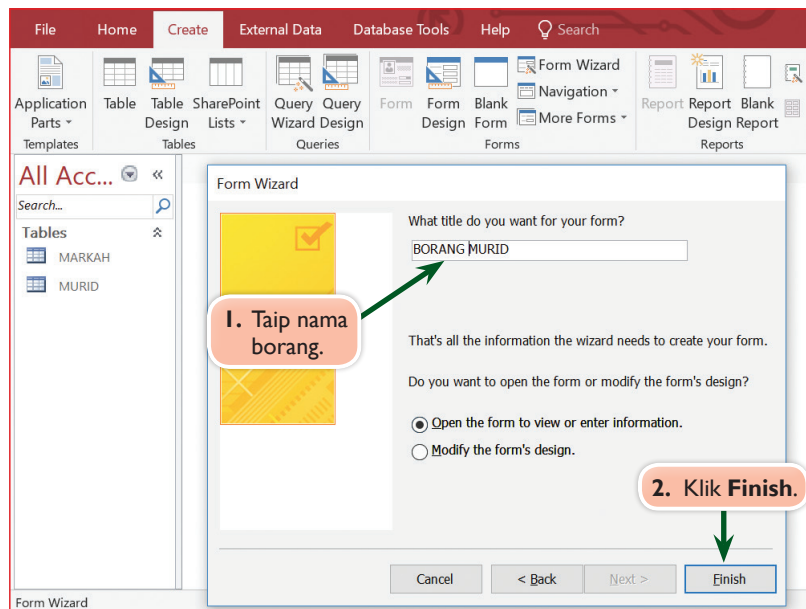
## Tahukah Anda?

Jenis susun atur	Maksud
<b>Columnar</b>	Rekod dipaparkan satu demi satu.
<b>Tabular</b>	Semua rekod dipaparkan dalam bentuk jadual yang menarik.
<b>Datasheet</b>	Semua rekod dipaparkan dalam bentuk helaian data yang agak menarik.
<b>Justified</b>	Semua medan ( <i>fields</i> ) disusun pada baris-baris blok yang mempunyai margin kiri dan kanan yang tetap.

**Langkah 4** Klik butang radio *Columnar* untuk memilih susun atur *Columnar*. Kemudian, klik *Next*.



**Langkah 5** Paparan berikut akan diperhatikan. Namakan borang anda sebagai BORANG MURID dengan menaip pada ruangan yang disediakan. Kemudian, klik *Finish*.



**Rajah 4.25** Langkah-langkah membina BORANG MURID daripada jadual MURID

Susun atur *columnar*



[goo.gl/ZTf1mk](http://goo.gl/ZTf1mk)

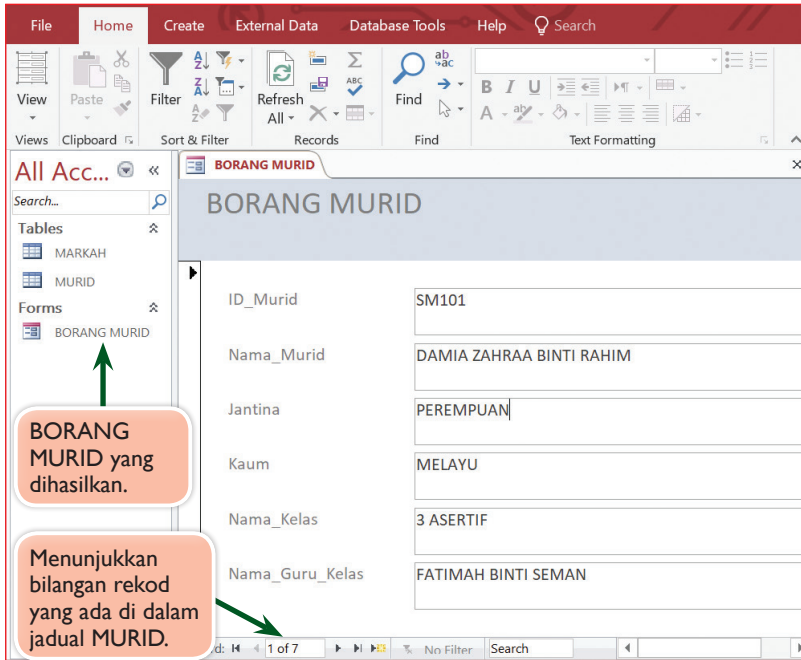
Imbas  
saya



BORANG MURID yang telah berjaya dihasilkan akan dipaparkan pada ruangan *All Access Objects*. Bilangan rekod yang ditunjukkan adalah sama dengan bilangan rekod yang ada dalam jadual MURID. Rajah 4.26 menunjukkan paparan BORANG MURID.

**Uji Minda 4.10**

Berikan kelebihan-kelebihan susun atur *columnar* berbanding susun atur yang lain.



Rajah 4.26 Paparan BORANG MURID

Jadual 4.5 menunjukkan penerangan kepada butang-butang yang terdapat pada bahagian bawah borang.

Jadual 4.5 Penerangan butang-butang yang terdapat pada bahagian bawah borang

Butang	Maksud
	Alih kepada rekod yang pertama.
	Alih kepada rekod yang sebelumnya.
	Alih kepada rekod yang berikutnya.
	Alih kepada rekod yang terakhir.
	Wujudkan satu rekod yang baharu.
<input type="text" value="Search"/>	Carian rekod.

**Tahukah Anda?**

Nama borang tidak semestinya sama dengan nama jadual. Anda boleh meletakkan sebarang nama yang sesuai.

Mengubah suai reka bentuk borang

[goo.gl/DgbrRp](http://goo.gl/DgbrRp)

**Imbas Saya**

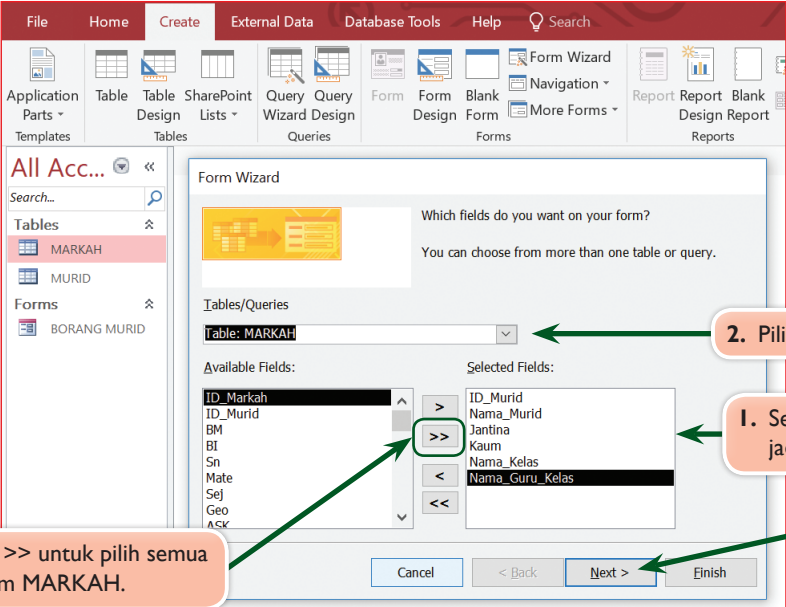
**Uji Minda 4.11**

Bagaimanakah cara untuk menghapuskan rekod yang tidak diperlukan lagi?

Selain membina borang daripada satu jadual, borang juga boleh dibina dengan memilih atribut-atribut daripada dua jadual yang berhubungan. Rajah 4.27 menunjukkan langkah-langkah membina borang daripada jadual MURID dan jadual MARKAH.

**Langkah 1** Buka pangkalan data TUGASAN1. Pada menu bar, klik pada tab **Create**. Kemudian, klik **Form Wizard**.

**Langkah 2** Tetingkap **Form Wizard** akan dipaparkan. Pilih jadual MURID dan klik butang >> untuk memilih semua medan yang ada dalam jadual. Kemudian, pilih jadual MARKAH dan klik butang >>.



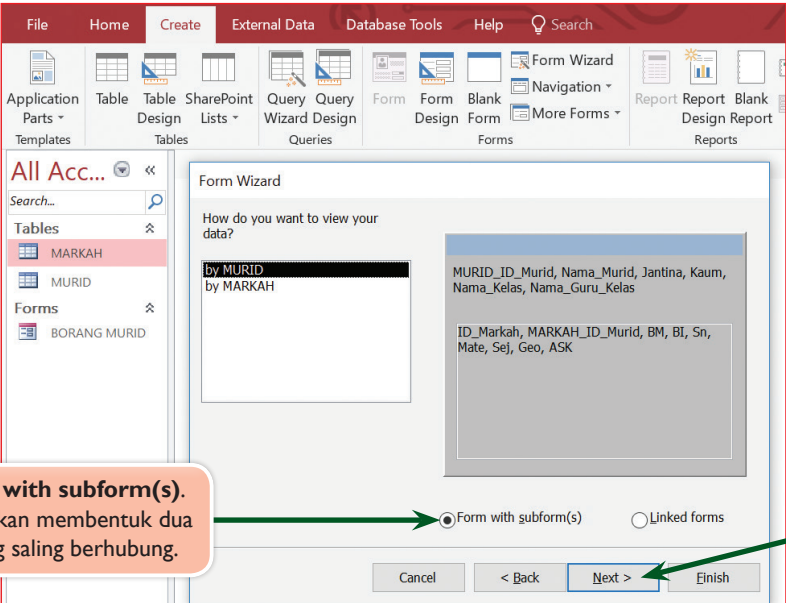
1. Semua medan dalam jadual MURID dipilih.

2. Pilih jadual MARKAH.

3. Klik butang >> untuk pilih semua medan dalam MARKAH.

4. Klik Next.

**Langkah 3** Paparan berikut akan diperhatikan. Pilih **Form with subform(s)**. Kemudian, klik **Next**.

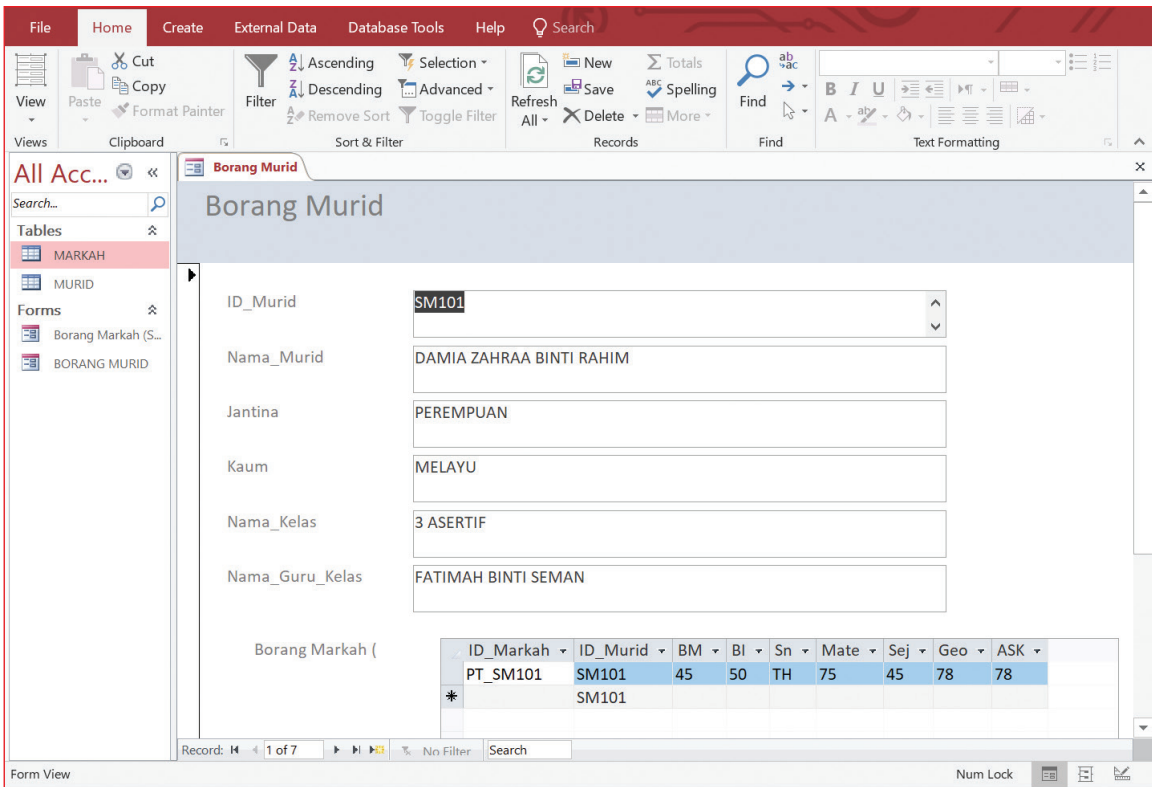


1. Pilih **Form with subform(s)**. Pilihan ini akan membentuk dua borang yang saling berhubungan.

2. Klik Next.



**Langkah 6** Borang Murid dan Borang Markah (*Subform*) telah berjaya dihasilkan. Perhatikan borang ini memudahkan pengguna memasukkan data untuk dua jadual tanpa perlu memasukkan ID\_Murid iaitu kekunci primer secara berulang.




**Rajah 4.27** Langkah-langkah membina borang daripada dua jadual

### Tahukah Anda?

Jika jenis data (*data type*) yang ditetapkan untuk satu medan ialah nombor, maka nilai lalai yang diberikan kepada medan itu ialah sifar.


Anda berjaya membina borang untuk jadual MURID dan jadual MARKAH. Kini, anda akan mempelajari langkah-langkah untuk memasukkan data ke dalam jadual-jadual melalui borang-borang dengan menggunakan fungsi penambahan rekod. Rajah 4.28 menunjukkan langkah-langkah menambah rekod baharu ke dalam jadual MURID dan jadual MARKAH dengan menggunakan borang yang dibina.

**Langkah 1** Klik ikon  di bahagian bawah borang seperti yang ditunjukkan di bawah.

### Uji Minda 4.12

Namakan objek-objek pangkalan data yang telah dihasilkan setakat ini.



Klik ikon  ini. Paparan **New (blank) record** akan muncul apabila kursor ditunjukkan padanya.

**Langkah 2** Paparan berikut akan diperhatikan. Kini anda boleh memasukkan data dalam ruangan kosong. Anda juga boleh memasukkan data untuk dua jadual dalam satu Borang Murid dan Borang Markah (*Subform*).

Ruangan kosong untuk kemasukan data bagi dua jadual.

Bilangan rekod bertambah.

**Langkah 3** Mula masukkan data murid dan data markah ke dalam ruangan kosong. Ruangan ID\_Murid bagi Borang Markah tidak perlu dimasukkan kerana data akan dimasukkan secara automatik apabila ID\_Murid diisi dalam Borang Murid. Tekan kekunci **Enter** pada papan kekunci bagi tujuan memasukkan data yang seterusnya.

**Rajah 4.28** Langkah-langkah untuk menambah rekod baharu ke dalam borang yang dibina

Terdapat dua cara untuk memasukkan data ke dalam pangkalan data, iaitu secara langsung ke dalam jadual atau melalui borang. Anda telah mempelajari kedua-dua cara tersebut. Rajah 4.29 menunjukkan jadual MURID dan jadual MARKAH yang menunjukkan bilangan rekod di dalamnya turut bertambah.

ID_Murid	Nama_Murid	Jantina	Kaum	Nama_Kelas	Nama_Guru_Kelas
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU	3 ASERTIF	FATIMAH BINTI SEMAN
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU	3 BIJAK	ZAKIAH BINTI YUSOF
SM103	DENISH RAIMI BIN ZAINAL	LELAKI	MELAYU	3 CEKAL	MUSTAFA BIN ROSLAN
SM104	TAN XUE LIEH	LELAKI	CINA	3 DINAMIK	TEOH JEE YEE
SM105	KUMAR A/L RAMESH	LELAKI	INDIA	3 EFISIEN	RIMALA A/P SAMY
SM106	JESSICA HENG MUI MUI	PEREMPUAN	CINA	3 FIKIR	LIM CHOII PENG
SM107	KHAIR MUIZ BIN ABD RAHIM	LELAKI	MELAYU	3 GIGIH	CHANDRAN A/L SUMAN
SM108	DEVI A/P MUTHU	PEREMPUAN	INDIA	3 EFISIEN	RIMALA A/P SAMY

ID_Markah	ID_Murid	BM	Bl	Sn	Mate	Sej	Geo	ASK
PT_SM101	SM101	45	50	TH	75	45	78	78
PT_SM102	SM102	56	85	87	68	84	90	79
PT_SM103	SM103	TH	58	78	65	89	85	79
PT_SM104	SM104	67	68	84	TH	77	TH	59
PT_SM105	SM105	77	67	65	87	46	58	58
PT_SM106	SM106	80	TH	54	76	85	81	85
PT_SM107	SM107	89	67	80	85	76	68	85
PT_SM108	SM108	89	67	80	85	76	68	85

Rajah 4.29 Paparan jadual MURID dan jadual MARKAH yang menunjukkan penambahan rekod dengan menggunakan borang

Penambahan rekod ke dalam jadual dengan menggunakan borang



[goo.gl/YyJgNa](http://goo.gl/YyJgNa)



### Uji Minda 4.13

1. Adakah jenis data *AutoNumber* yang disetkan untuk ID\_Murid merupakan satu pilihan yang tepat? Berikan alasan kepada jawapan anda.
2. Jika *AutoNumber* tidak digunakan, apakah jenis data lain yang sesuai? Jelaskan.
3. Jika anda menukarkan jenis data untuk ID\_Murid dalam jadual MARKAH, bagaimana dengan ID\_Murid dalam jadual MURID? Jelaskan.
4. Apakah cara lain untuk menghasilkan borang selain menggunakan **Form Wizard**?

## Praktis Amali 4.3

1. Lancarkan perisian aplikasi *Microsoft Access*.
2. Ikuti langkah-langkah dalam Rajah 4.27 untuk membina borang daripada dua jadual.
3. Kemudian, masukkan data ke dalam Borang yang dihasilkan.



## Aktiviti 4.5



### Aktiviti Kumpulan


#### Membina Borang



Kaedah Pembentangan  
Hasil Sendiri

- Jadual berikut menunjukkan satu rekod pesakit.

ID Pesakit	: SA00174180	Kewarganegaraan	: Malaysia
Nama Pesakit	: Norain binti Ramli	No. Fail	: MR00158050
Jantina	: Perempuan	No. Wad	: 2
Tarikh Lahir	: 02/07/1975	No. Katil	: 14
Umur	: 43	Penyakit	: Darah Tinggi
Bangsa	: Melayu	Nama Doktor	: Dr. Zainuddin bin Ismail
Alamat	: No. 3, Jalan Payung, 47800 Petaling Jaya		

- Anda diminta membentuk kumpulan.
- Berdasarkan maklumat dalam jadual di atas, anda diminta untuk mengenal pasti dua entiti dan atribut-atribut yang sesuai. Hasil dapatan anda perlu ditulis pada kertas. 
- Setelah masa yang diperuntukkan tamat, ahli-ahli dalam kumpulan perlu berkumpul dan membentangkan hasil dapatan masing-masing.
- Dalam kumpulan, semua murid diminta menunjukkan jawapan masing-masing.



## Praktis Amali 4.4

#### Membina Pangkalan Data bagi Hasil Dapatan Dalam Aktiviti 4.5



- Lancarkan perisian aplikasi *Microsoft Access*.
- Ikuti langkah-langkah untuk membina pangkalan data yang telah anda pelajari dalam subtopik 4.1.5 untuk membina pangkalan data bagi hasil dapatan anda dalam **Aktiviti 4.5**.
- Setelah berjaya membina pangkalan data, anda perlu menghasilkan borang dan memasukkan data ke dalam jadual melalui borang yang dibina.
- Anda boleh mengkaji semula Rajah 4.25, Rajah 4.27 dan Rajah 4.28 untuk mengetahui langkah-langkah menghasilkan borang dan memasukkan data baharu ke dalam jadual melalui borang.

## Standard Pembelajaran

Murid boleh:

- 4.1.7** Menggunakan arahan SQL yang melibatkan:
- (i) *SELECT...*
  - (ii) *SELECT...WHERE*
  - (iii) *SELECT...ORDER BY*

4.1.7

## Penggunaan Pernyataan-pernyataan SQL

Perisian aplikasi Sistem Pengurusan Pangkalan Data (DBMS) menggunakan arahan SQL untuk mencapai data dari rekod yang disimpan dalam pangkalan. SQL mempunyai sintaks tertentu yang mesti diikuti semasa digunakan. Sintaks merupakan set peraturan yang menetapkan set perintah yang boleh digunakan dan cara-cara menulis perintah-perintah dalam bentuk ayat agar boleh difahami dan dilaksanakan oleh perisian aplikasi Sistem Pengurusan Pangkalan Data (DBMS). Sintaks bagi SQL adalah berasaskan sintaks dalam bahasa Inggeris.

Semua contoh pernyataan SELECT yang akan dibincangkan akan menggunakan entiti MURID seperti Rajah 4.30.

ID_Murid	Nama_Murid	Jantina	Kaum	Nama_Kelas	Nama_Guru_Kelas
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU	3 ASERTIF	FATIMAH BINTI SEMAN
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU	3 BIJAK	ZAKIAH BINTI YUSOF
SM103	DENISH RAIMI BIN ZAINAL	LELAKI	MELAYU	3 CEKAL	MUSTAFA BIN ROSLAN
SM104	TAN XUE LIEH	LELAKI	CINA	3 DINAMIK	TEOH JEE YEE
SM105	KUMAR A/L RAMESH	LELAKI	INDIA	3 EFISIEN	RIMALA A/P SAMY
SM106	JESSICA HENG MUI MUI	PEREMPUAN	CINA	3 FIKIR	LIM CHOI PENG
SM107	KHAIR MUIZ BIN ABD RAHIM	LELAKI	MELAYU	3 GIGIH	CHANDRAN A/L SUMAN
SM108	DEVI A/P MUTHU	PEREMPUAN	INDIA	3 EFISIEN	RIMALA A/P SAMY

Rajah 4.30 Entiti MURID

Jadual 4.6, 4.7 dan 4.8 menunjukkan keterangan dan contoh masing-masing untuk pernyataan SELECT..., pernyataan SELECT...WHERE dan pernyataan SELECT...ORDER BY.

Jadual 4.6 Keterangan dan contoh untuk pernyataan SELECT

(i) Pernyataan SELECT	
Keterangan	Contoh
<p><b>Pernyataan SELECT</b> membenarkan anda untuk memilih atribut-atribut yang dikehendaki dari pangkalan data.</p> <p><b>Pernyataan FROM</b> membenarkan anda memilih entiti bagi atribut-atribut yang telah dipilih dalam pernyataan SELECT. Pernyataan FROM secara spesifik menyatakan nama entiti yang digunakan untuk capaian data.</p>	<p><b>Sintaks Umum</b></p> <pre>SELECT atribut_1, atribut_2, atribut_3, atribut_N FROM nama_jadual;</pre>



**Pernyataan SELECT...FROM**

memulangkan nilai-nilai daripada atribut yang telah dipilih berdasarkan entiti yang telah ditetapkan.

**Pernyataan SELECT \* FROM** pula akan mencapai data bagi semua atribut yang terdapat dalam entiti yang telah ditetapkan.

**Tahukah Anda?**

Baris akhir sintaks SQL tidak perlu berakhir dengan simbol semicolon (;) tetapi ianya merupakan praktis yang baik dalam pengaturcaraan.

**Contoh 1**

Pemilihan dua atribut sahaja daripada entiti MURID.

```
SELECT ID_Murid, Nama_Murid
FROM MURID;
```

**Output**

Paparan output jelas menunjukkan bahawa hanya data bagi dua atribut sahaja, iaitu ID\_Murid dan Nama\_Murid telah dicapai daripada entiti MURID. Output yang terhasil adalah seperti Rajah 4.31.

ID_Murid	Nama_Murid
SM101	DAMIA ZAHRAA BINTI RAHIM
SM102	DHIA ARISSA BINTI MALIK
SM103	DENISH RAIMI BIN ZAINAL
SM104	TAN XUE LIEH
SM105	KUMAR A/L RAMESH
SM106	JESSICA HENG MUI MUI
SM107	KHAIR MUIZ BIN ABD RAHIM
SM108	DEVI A/P MUTHU

**Rajah 4.31** Output Pertanyaan 1

**Contoh 2**

Pemilihan semua atribut daripada entiti MURID.

```
SELECT *
FROM MURID;
```

**Output**

Paparan output jelas menunjukkan bahawa data bagi semua atribut, iaitu ID\_Murid, Nama\_Murid, Jantina, Kaum, Nama\_Kelas dan Nama\_Guru\_Kelas telah dicapai daripada entiti MURID. Output yang terhasil adalah seperti Rajah 4.32.

ID_Murid	Nama_Murid	Jantina	Kaum	Nama_Kelas	Nama_Guru_Kelas
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU	3 ASERTIF	FATIMAH BINTI SEMAN
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU	3 BIJAK	ZAKIAH BINTI YUSOF
SM103	DENISH RAIMI BIN ZAINAL	LELAKI	MELAYU	3 CEKAL	MUSTAFA BIN ROSLAN
SM104	TAN XUE LIEH	LELAKI	CINA	3 DINAMIK	TEOH JEE YEE
SM105	KUMAR A/L RAMESH	LELAKI	INDIA	3 EFISIEN	RIMALA A/P SAMY
SM106	JESSICA HENG MUI MUI	PEREMPUAN	CINA	3 FIKIR	LIM CHOI PENG
SM107	KHAIR MUIZ BIN ABD RAHIM	LELAKI	MELAYU	3 GIGIH	CHANDRAN A/L SUMAN
SM108	DEVI A/P MUTHU	PEREMPUAN	INDIA	3 EFISIEN	RIMALA A/P SAMY

**Rajah 4.32** Output Pertanyaan 2

**Jadual 4.7** Keterangan dan contoh untuk pernyataan SELECT...WHERE

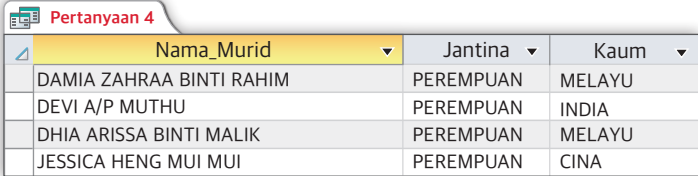
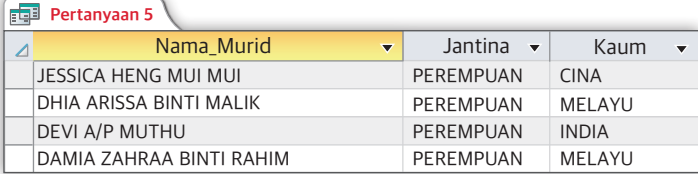
(ii) Pernyataan SELECT...WHERE																
Keterangan	Contoh															
<p>Pernyataan SELECT... WHERE hampir serupa dengan pernyataan SELECT... FROM. Perbezaan utama ialah pernyataan SELECT... WHERE mencapai data-data yang tertentu yang memenuhi sesuatu kriteria sahaja. Oleh itu, data yang dicapai lebih khusus.</p>	<p><b>Sintaks Umum</b>            SELECT atribut_1, atribut_2,...atribut_N            FROM nama_jadual            WHERE kriteria;</p> <p><b>Contoh</b>            Pemilihan atribut yang memenuhi sesuatu kriteria.            Katakan anda ingin memilih beberapa atribut yang memenuhi kriteria Jantina = "PEREMPUAN" daripada entiti MURID. Maka anda perlu menulis sintaks seperti berikut:</p> <div style="border: 1px solid orange; border-radius: 15px; padding: 10px; margin: 10px 0;"> <pre>SELECT Nama_Murid, Jantina, Kaum FROM MURID WHERE Jantina = "PEREMPUAN";</pre> </div> <p>Kriteria yang telah ditetapkan ditulis dalam klausa WHERE, iaitu Jantina = "PEREMPUAN". Oleh itu, carian bagi atribut Nama_Murid, Jantina dan Kaum adalah berdasarkan kriteria PEREMPUAN.</p> <p><b>Output</b>            Paparan output jelas menunjukkan bahawa data bagi atribut-atribut seperti Nama_Murid, Jantina dan Kaum adalah berdasarkan kriteria PEREMPUAN telah dicapai daripada entiti MURID. Output yang terhasil seperti Rajah 4.33.</p> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p><b>Pertanyaan 3</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Nama_Murid</th> <th style="text-align: left;">Jantina</th> <th style="text-align: left;">Kaum</th> </tr> </thead> <tbody> <tr> <td>DAMIA ZAHRAA BINTI RAHIM</td> <td>PEREMPUAN</td> <td>MELAYU</td> </tr> <tr> <td>DHIA ARISSA BINTI MALIK</td> <td>PEREMPUAN</td> <td>MELAYU</td> </tr> <tr> <td>JESSICA HENG MUI MUI</td> <td>PEREMPUAN</td> <td>CINA</td> </tr> <tr> <td>DEVI A/P MUTHU</td> <td>PEREMPUAN</td> <td>INDIA</td> </tr> </tbody> </table> </div>	Nama_Murid	Jantina	Kaum	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU	JESSICA HENG MUI MUI	PEREMPUAN	CINA	DEVI A/P MUTHU	PEREMPUAN	INDIA
Nama_Murid	Jantina	Kaum														
DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU														
DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU														
JESSICA HENG MUI MUI	PEREMPUAN	CINA														
DEVI A/P MUTHU	PEREMPUAN	INDIA														

**Tahukah Anda?**

Penulisan kriteria perlu mempunyai simbol pembuka dan penutup kata (""). Jika anda tidak berbuat demikian, maka akan berlaku ralat sintaks.

**Rajah 4.33** Output Pertanyaan 3

**Jadual 4.8** Keterangan dan contoh untuk pernyataan SELECT...ORDER BY

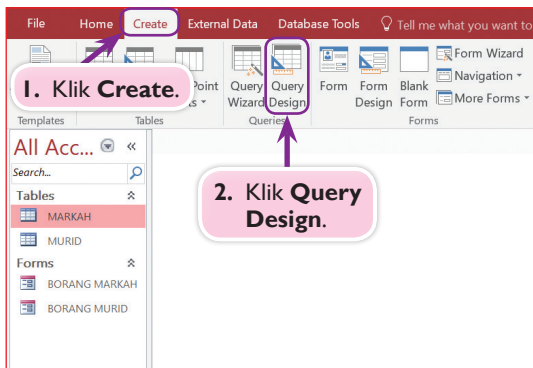
(iii) Pernyataan SELECT...ORDER BY	
Keterangan	Contoh
<p>Pernyataan SELECT... ORDER BY membolehkan anda mengisih data yang dicapai mengikut susunan menaik (ASC) atau susunan menurun (DESC) berdasarkan satu atau lebih daripada satu medan isihan.</p> <p>Sekiranya ASC ditaip, rekod-rekod yang dicapai akan diisih secara menaik mengikut susunan abjad A-Z berdasarkan medan isihan.</p> <p>Jika DESC ditaip, maka rekod-rekod akan diisih mengikut susunan abjad Z-A berdasarkan medan isihan.</p>	<p><b>Sintaks Umum</b>  SELECT atribut_1, atribut_2,...atribut_N  FROM nama_jadual  WHERE kriteria  ORDER BY atribut_1 ASC/DESC;</p> <p><b>Contoh 1</b>  Penggunaan ASC untuk mengisih secara menaik.</p> <pre>SELECT Nama_Murid, Jantina, Kaum FROM MURID WHERE Jantina = "PEREMPUAN" ORDER BY Nama_Murid ASC;</pre> <p><b>Output</b>  Paparan output jelas menunjukkan bahawa data bagi atribut Nama_Murid telah diisih secara menaik.</p>  <p><b>Rajah 4.34</b> Output Pertanyaan 4</p> <p><b>Contoh 2</b>  Penggunaan DESC untuk mengisih secara menurun.</p> <pre>SELECT Nama_Murid, Jantina, Kaum FROM MURID WHERE Jantina = "PEREMPUAN" ORDER BY Nama_Murid DESC;</pre> <p><b>Output</b>  Paparan output jelas menunjukkan bahawa data bagi atribut Nama_Murid telah diisih secara menurun.</p>  <p><b>Rajah 4.35</b> Output Pertanyaan 5</p>

## Langkah-langkah menghasilkan pertanyaan (*query*) dengan menggunakan SQL

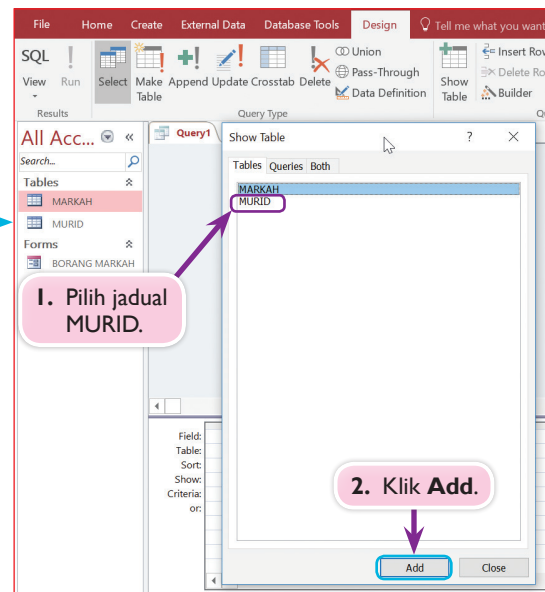
Pertanyaan (*query*) bermaksud pertanyaan kepada pangkalan data hubungan dalam bahasa *query* untuk mencapai data dan maklumat yang disimpan dalamnya. Pernyataan SQL yang telah anda pelajari ialah pertanyaan (*query*).

Capaian data dan maklumat daripada pangkalan data yang dibina dengan perisian aplikasi *Microsoft Access* dapat dilakukan dengan menggunakan arahan SQL. Pertanyaan (*query*) dapat dihasilkan dengan menggunakan pernyataan-pernyataan SQL yang telah anda pelajari. Rajah 4.36 menunjukkan langkah-langkah menghasilkan pertanyaan (*query*) dengan menggunakan arahan SQL.

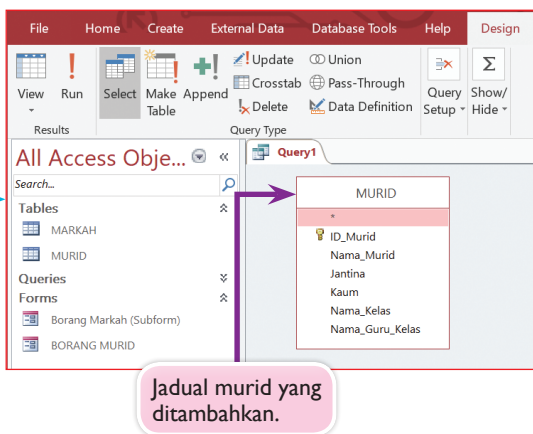
**Langkah 1** Klik **Create** di bar Menu, kemudian klik **Query Design**.



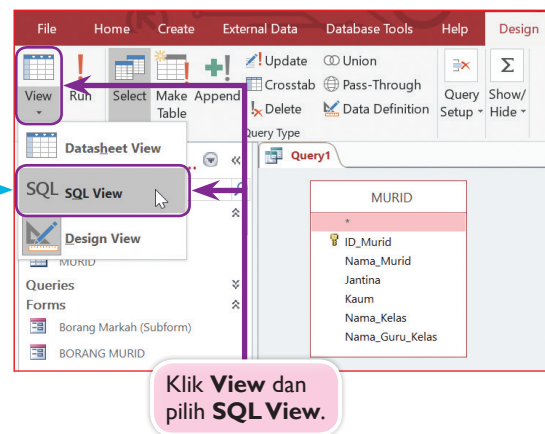
**Langkah 2** Pilih entiti MURID dan klik **Add**.



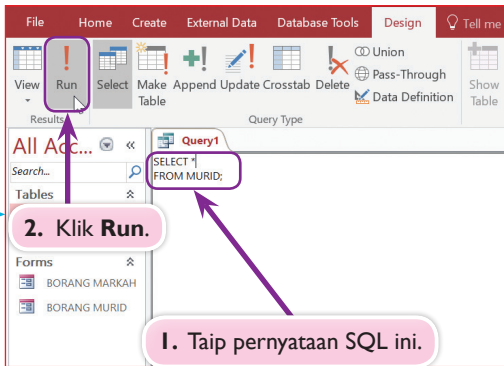
**Langkah 3** Jadual MURID telah ditambah.



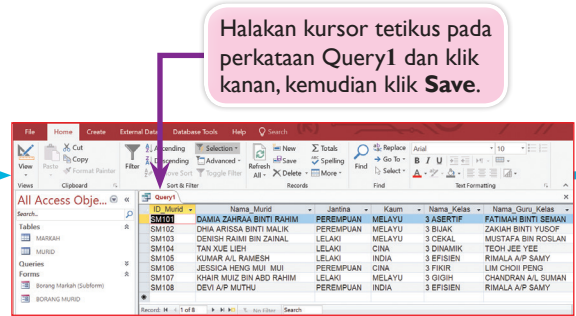
**Langkah 4** Klik **View** dan pilih **SQL View**.



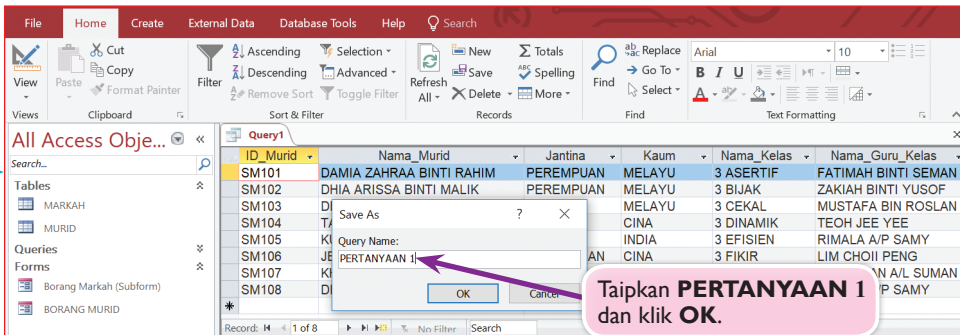
**Langkah 5** Taip `SELECT * FROM MURID` dalam **Query1**. Kemudian klik **Run**.



**Langkah 6** Data yang dicapai akan terpapar dalam **Query1**. Halakan kursor tetikus pada perkataan **Query1** dan klik kanan, kemudian klik **Save**.



**Langkah 7** Tetingkap **Save As** terpapar. Tukarkan nama **Query1** kepada **PERTANYAAN 1** dan klik **OK**.



**Langkah 8** Paparan berikut menunjukkan **query SELECT \*** yang telah disimpan dan paparan data yang dicapai.



Rajah 4.36 Langkah-langkah menghasilkan pertanyaan (*query*) dengan menggunakan SQL

## Tahukah Anda ?

Anda digalakkan memasukkan nama yang bersesuaian dengan kehendak pertanyaan (*query*) yang dibina.



### Pertanyaan(Query) Menggunakan Arahan SQL



Kaedah Gallery Walk

Andaikan anda seorang pembangun sistem pangkalan data. Anda telah ditugaskan untuk membangunkan Sistem Maklumat Kokurikulum. Sebagai pembangun sistem maklumat ini, anda telah berjaya menghasilkan entiti MURID dan entiti GURU PENASIHAT.

ID_Murid	Nama_Murid	Nama_Kelas
M001	SOFIA BINTI JAMAL	3 JATI
M002	ANAS BIN SUFIAN	3 JATI
M003	WONG MUI LING	3 MERBAU
M004	ANANTH A/L SAVOO	3 MERBAU
M005	MELISSA YAP LI LING	3 CENGAL

ID_Guru	ID_Murid	Nama_Guru	Persatuan	Unit_Beruniform	Sukan
GP01	M001	AZLINA BINTI AZMI	BAHASA MELAYU	KADET POLIS	BOLA JARING
GP02	M002	ARSHAD BIN YUSOFF	BAHASA MELAYU	KADET POLIS	BOLA KERANJANG
GP03	M003	TANG SIANG HENG	MATEMATIK	KADET POLIS	BOLA KERANJANG
GP04	M004	ANGELINA KHOO XING XUE	MATEMATIK	PANDU PUTERI	BADMINTON
GP05	M005	THULASI A/P MARIAPPAN	SAINS	PANDU PUTERI	BADMINTON

Lakukan aktiviti dan jawab soalan-soalan berikut berdasarkan jadual-jadual bagi entiti MURID dan entiti GURU\_PENASIHAT.

1. Anda diminta membentuk kumpulan.
2. Setiap murid diminta menulis pertanyaan SQL yang sesuai untuk mendapatkan paparan PERTANYAAN 1, PERTANYAAN 2, PERTANYAAN 3 DAN PERTANYAAN 4.

ID_Guru	ID_Murid	Nama_Guru	Persatuan	Unit_Beruniform	Sukan
GP01	M001	AZLINA BINTI AZMI	BAHASA MELAYU	KADET POLIS	BOLA JARING
GP02	M002	ARSHAD BIN YUSOFF	BAHASA MELAYU	KADET POLIS	BOLA KERANJANG
GP03	M003	TANG SIANG HENG	MATEMATIK	KADET POLIS	BOLA KERANJANG
GP04	M004	ANGELINA KHOO XING XUE	MATEMATIK	PANDU PUTERI	BADMINTON
GP05	M005	THULASI A/P MARIAPPAN	SAINS	PANDU PUTERI	BADMINTON

ID_Guru	ID_Murid	Nama_Guru	Unit_Beruniform
GP01	M001	AZLINA BINTI AZMI	KADET POLIS
GP02	M002	ARSHAD BIN YUSOFF	KADET POLIS
GP03	M003	TANG SIANG HENG	KADET POLIS

ID_Guru	Nama_Guru	Persatuan	Sukan
GP02	ARSHAD BIN YUSOFF	BAHASA MELAYU	BOLA KERANJANG
GP01	AZLINA BINTI AZMI	BAHASA MELAYU	BOLA JARING

ID_Guru	ID_Murid	Nama_Guru
GP05	M005	THULASI A/P MARIAPPAN
GP04	M004	ANGELINA KHOO XING XUE
GP03	M003	TANG SIANG HENG
GP02	M002	ARSHAD BIN YUSOFF
GP01	M001	AZLINA BINTI AZMI

3. Anda perlu berbincang dalam kumpulan dan jawapan terakhir perlu ditulis pada kertas.
4. Setelah masa perbincangan tamat, setiap kumpulan diminta untuk menampalkan jawapan masing-masing pada dinding yang berdekatan dengan tempat mereka berada.
5. Setiap kumpulan akan bergerak dari satu kumpulan ke kumpulan yang lain sehingga semua kumpulan telah habis dilawati bagi tujuan membuat penilaian terhadap hasil kerja kumpulan yang lain.
6. Setiap ahli kumpulan perlu menulis komen terhadap hasil kerja kumpulan yang lain dengan menggunakan pen yang berlainan warna.



## Praktis Amali 4.5

### Menghasilkan Pangkalan Data dan Pertanyaan (*Query*)

1. Lancarkan perisian aplikasi *Microsoft Access*.
2. Hasilkan entiti MURID dan entiti GURU\_PENASIHAT yang ditunjukkan dalam Aktiviti 4.6.
3. Berdasarkan pertanyaan-pertanyaan SQL yang anda peroleh daripada Aktiviti 4.6, hasilkan PERTANYAAN 1, PERTANYAAN 2, PERTANYAAN 3 dan PERTANYAAN 4.
4. Rujuk semula Rajah 4.36 untuk langkah-langkah menghasilkan pertanyaan (*query*).
5. Setelah berjaya, tunjukkan kepada guru anda untuk disemak.



4.1.8

### Penggunaan Pernyataan-pernyataan SQL yang Melibatkan Ungkapan Boolean

Semasa membuat pertanyaan (*query*), anda mungkin pernah berada dalam situasi yang memerlukan anda menggunakan operator logik. Operator logik juga dikenali sebagai operator *boolean*. Operator logik yang digunakan dalam arahan SQL ialah OR, AND, EQV, NOT dan XOR. Akan tetapi, operator OR dan operator AND ialah dua operator logik yang paling kerap digunakan.

Operator logik digunakan untuk menggabungkan dua atau lebih kriteria bagi menghasilkan satu ungkapan *boolean*. Ungkapan *boolean* akan memberi nilai BENAR atau PALSU. Penggunaan operator logik dalam arahan SQL membenarkan pelbagai kriteria atau syarat digabungkan untuk menghasilkan pertanyaan yang lebih kompleks untuk mencapai data yang dikehendaki.

#### (i) Operator OR

Apabila operator OR digunakan untuk menggabungkan dua kriteria P dan Q, ungkapan P OR Q akan dihasilkan. Nilai *boolean* BENAR akan dipulangkan jika salah satu kriteria P atau Q adalah benar. Jadual 4.9 menunjukkan jadual kebenaran OR.

Jadual 4.9 Jadual kebenaran OR

P	Q	P OR Q
BENAR	BENAR	BENAR
BENAR	PALSU	BENAR
PALSU	BENAR	BENAR
PALSU	PALSU	PALSU

### Standard Pembelajaran

Murid boleh:

- 4.1.8 Menggunakan arahan SQL melibatkan ungkapan *Boolean*:
- (i) Operator OR
  - (ii) Operator AND

Video tentang operator OR dan operator AND dalam SQL



[goo.gl/5fLhrc](https://goo.gl/5fLhrc)



## Tahukah Anda?

Nilai *boolean* BENAR dan PALSU juga kadangkala diwakili oleh nombor 1 dan 0, di mana 1 mewakili BENAR dan 0 mewakili PALSU. Operator OR boleh diwakili dengan simbol (v). Jadual kebenaran OR berikut berdasarkan nombor 0, 1 dan simbol v.

P	Q	P v Q
1	1	1
1	0	1
0	1	1
0	0	0

Operator OR digunakan untuk menggabungkan pelbagai kriteria atau syarat yang ditetapkan dalam klausa WHERE. Pernyataan SQL yang berikut menunjukkan sintaks umum penggunaan operator OR dalam klausa WHERE.

```
SELECT atribut_1, atribut_2, ..., atribut_N
FROM nama_jadual
WHERE kriteria_1 OR kriteria_2, ... OR kriteria_N;
```

Rajah 4.37 menunjukkan jadual bernama MURID. Pernyataan SQL dalam Contoh 4.10 dan Contoh 4.11 adalah berdasarkan jadual ini.

ID_Murid	Nama_Murid	Jantina	Kaum	Nama_Kelas	Nama_Guru_Kelas
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU	3 ASERTIF	FATIMAH BINTI SEMAN
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU	3 BIJAK	ZAKIAH BINTI YUSOF
SM103	DENISH RAIMI BIN ZAINAL	LELAKI	MELAYU	3 CEKAL	MUSTAFA BIN ROSLAN
SM104	TAN XUE LIEH	LELAKI	CINA	3 DINAMIK	TEOH JEE YEE
SM105	KUMAR A/L RAMESH	LELAKI	INDIA	3 EFISIEN	RIMALA A/P SAMY
SM106	JESSICA HENG MUI MUI	PEREMPUAN	CINA	3 FIKIR	LIM CHOI PENG
SM107	KHAIR MUIZ BIN ABD RAHIM	LELAKI	MELAYU	3 GIGIH	CHANDRAN A/L SUMAN
SM108	DEVI A/P MUTHU	PEREMPUAN	INDIA	3 EFISIEN	RIMALA A/P SAMY

Rajah 4.37 Jadual bagi entiti MURID

## Contoh 4.6 Penggunaan operator OR dalam pernyataan SQL.

### Uji Minda 4.14

Apakah yang dimaksudkan dengan kriteria\_N?

Pernyataan SQL berikut menunjukkan penggunaan operator OR dalam klausa WHERE untuk mendapatkan data yang memenuhi salah satu daripada dua kriteria yang dinyatakan, iaitu kriteria jantina ialah perempuan atau kriteria kaum ialah Melayu.

```
SELECT ID_Murid, Nama_Murid, Jantina, Kaum
FROM MURID
WHERE Jantina = "PEREMPUAN" OR Kaum = "MELAYU";
```

Rajah 4.38 menunjukkan output yang terhasil daripada pernyataan SQL di atas.

ID_Murid	Nama_Murid	Jantina	Kaum
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU
SM103	DENISH RAIMI BIN ZAINAL	LELAKI	MELAYU
SM106	JESSICA HENG MUI MUI	PEREMPUAN	CINA
SM107	KHAIR MUIZ BIN ABD RAHIM	LELAKI	MELAYU
SM108	DEVI A/P MUTHU	PEREMPUAN	INDIA

Rajah 4.38 Output pertanyaan OPERATOR OR



**(ii) Operator AND**

Apabila operator AND digunakan untuk menggabungkan dua kriteria P dengan Q, ungkapan P AND Q akan dihasilkan. Nilai *boolean* BENAR akan dipulangkan hanya apabila kedua-dua kriteria P dan Q adalah BENAR. Jadual 4.10 menunjukkan jadual kebenaran AND.

**Jadual 4.10** Jadual kebenaran AND

P	Q	P AND Q
BENAR	BENAR	BENAR
BENAR	PALSU	PALSU
PALSU	BENAR	PALSU
PALSU	PALSU	PALSU

Pernyataan SQL yang berikut menunjukkan sintaks umum penggunaan operator AND dalam klausa WHERE.

```
SELECT atribut_1, atribut_2, ..., atribut_N
FROM nama_jadual
WHERE kriteria_1 AND kriteria_2 AND ...kriteria_N;
```

Penerangan operator OR dan operator AND dalam SQL



[goo.gl/z4Ehc9](https://goo.gl/z4Ehc9)

**Tahukah Anda?**

Jadual kebenaran ungkapan *boolean* yang menggunakan operator AND (^) adalah seperti berikut:

P	Q	P ^ Q
1	1	1
1	0	0
0	1	0
0	0	0

**Contoh 4.7** Penggunaan operator AND dalam pernyataan SQL.

Pernyataan SQL berikut menunjukkan penggunaan operator AND dalam klausa WHERE untuk mendapatkan data yang memenuhi kedua-dua kriteria yang dinyatakan, iaitu kriteria jantina ialah perempuan atau kriteria kaum ialah Melayu.

```
SELECT ID_Murid, Nama_Murid, Jantina, Kaum
FROM MURID
WHERE Jantina = "PEREMPUAN" AND Kaum = "MELAYU";
```

Rajah 4.39 menunjukkan output yang terhasil daripada pernyataan SQL di atas.

ID_Murid	Nama_Murid	Jantina	Kaum
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU

**Rajah 4.39** Output pertanyaan OPERATOR AND



### Arahan SQL yang Melibatkan Ungkapan Boolean

Rajah-rajah di bawah menunjukkan jadual entiti MURID dan entiti GURU\_PENASIHAT.

ID_Murid	Nama_Murid	Nama_Kelas
M001	SOFIA BINTI JAMAL	3 JATI
M002	ANAS BIN SUFIAN	3 JATI
M003	WONG MUI LING	3 MERBAU
M004	ANANTH A/L SAVOO	3 MERBAU
M005	MELISSA YAP LI LING	3 CENGAL

ID_Guru	ID_Murid	Nama_Guru	Persatuan	Unit_Beruniform	Sukan
GP01	M001	AZLINA BINTI AZMI	BAHASA MELAYU	KADET POLIS	BOLA JARING
GP02	M002	ARSHAD BIN YUSOFF	BAHASA MELAYU	KADET POLIS	BOLA KERANJANG
GP03	M003	TANG SIANG HENG	MATEMATIK	KADET POLIS	BOLA KERANJANG
GP04	M004	ANGELINA KHOO XING XUE	MATEMATIK	PANDU PUTERI	BADMINTON
GP05	M005	THULASI A/P MARIAPPAN	SAINS	PANDU PUTERI	BADMINTON

Berdasarkan kedua-dua jadual ini, lakukan aktiviti dan jawab soalan-soalan berikut:

1. Anda diminta membentuk kumpulan.
2. Anda diminta menulis dua pernyataan SQL yang menggunakan operator OR dan operator AND.
3. Setelah masa yang diberi untuk menulis pernyataan SQL tamat, guru memberikan isyarat pertama supaya anda mula bergerak secara individu untuk melihat hasil perbincangan rakan-rakan.
4. Apabila isyarat kedua dibunyikan, anda perlu kembali ke tempat asal dan berkongsi maklumat yang diperolehi daripada kumpulan lain.
5. Anda diminta memaparkan output daripada pertanyaan dengan menggunakan perisian aplikasi persembahan yang sesuai sebagai hasil perbincangan.

## 4.1.9

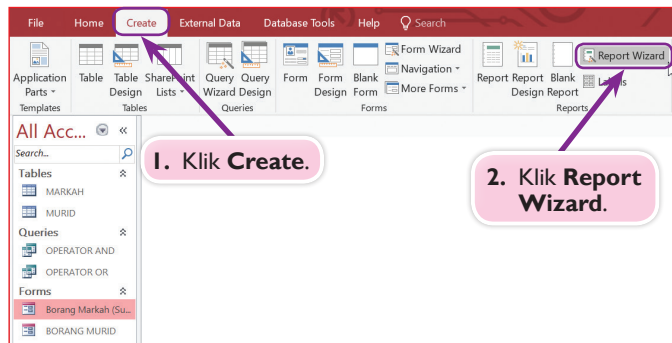
## Menjana Laporan Berdasarkan Hasil Pertanyaan (Query)

Penggunaan laporan untuk mengurus dan mempersembahkan data daripada pangkalan data merupakan satu cara yang sistematik dan terbaik. Laporan membolehkan anda menetapkan format data dalam bentuk yang menarik dan bermaklumat bagi tujuan cetakan dan paparan pada skrin.

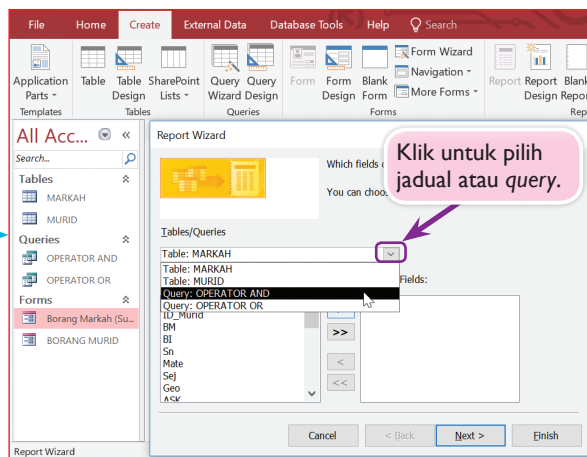
Laporan boleh dihasilkan berdasarkan maklumat daripada satu jadual atau daripada beberapa jadual. Sekiranya maklumat tersebut adalah daripada beberapa jadual, anda perlu menghasilkan pertanyaan (*query*) untuk mencapai maklumat yang dikehendaki terlebih dahulu. Laporan boleh dijana berdasarkan pertanyaan yang dilakukan.

Dalam perisian aplikasi *Microsoft Access*, cara yang paling mudah untuk menghasilkan laporan adalah dengan menggunakan **Report Wizard**. Rajah 4.40 menunjukkan langkah-langkah menjana laporan dengan menggunakan **Report Wizard**.

**Langkah 1** Pada menu bar, klik **Create**. Kemudian, klik **Report Wizard**.



**Langkah 2** Paparan tettingkap **Report Wizard** akan muncul. Pilih jadual atau *query* yang anda ingin gunakan untuk menghasilkan laporan.



### Standard Pembelajaran

Murid boleh:

- 4.1.9** Menjana laporan berdasarkan hasil pertanyaan (*query*)

Video tentang  
*Report Wizard*



[goo.gl/PMCaVG](http://goo.gl/PMCaVG)



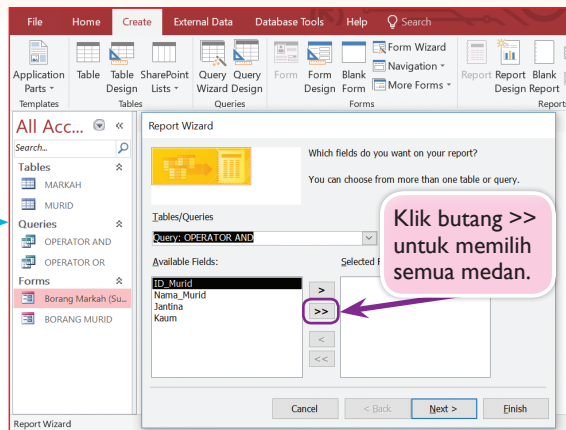
### Tahukah Anda?

Selain menggunakan *Report Wizard*, laporan juga boleh dijana menggunakan *Report* dengan hanya satu klik. Anda boleh mencubanya nanti!

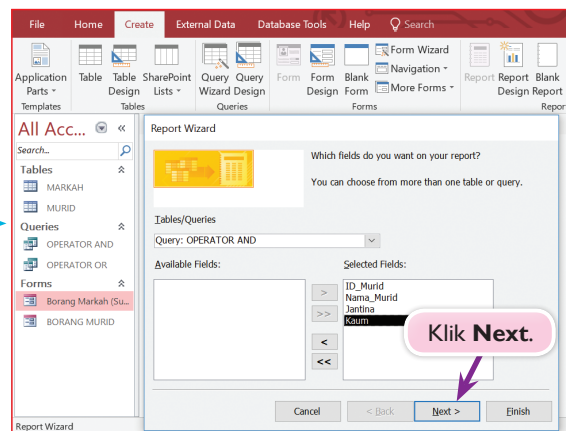
### Uji Minda 4.15

Apakah objek-objek pangkalan data yang boleh digunakan untuk menghasilkan laporan?

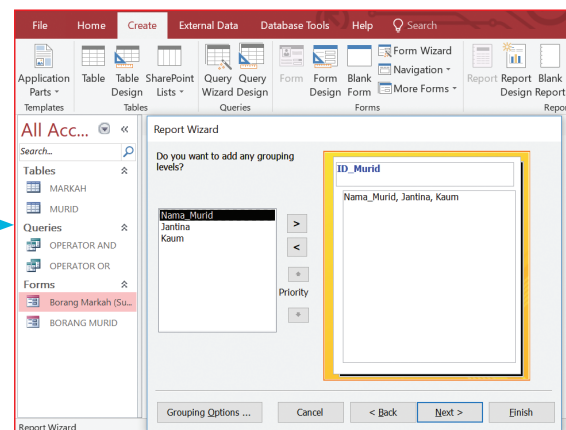
**Langkah 3** Setelah jadwal atau pertanyaan dipilih, tettingkap Available Fields menunjukkan semua medan yang ada di dalam jadwal. Klik ikon >> untuk memilih semua medan.



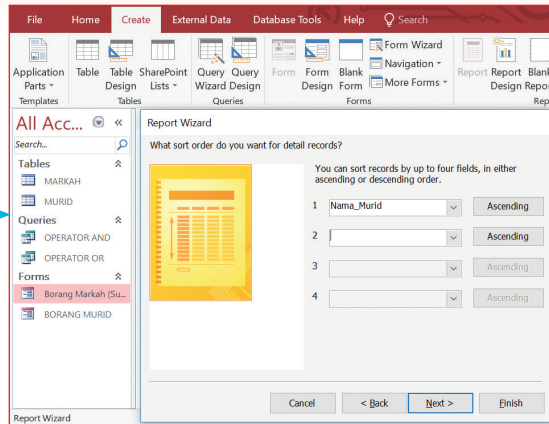
**Langkah 4** Medan yang dipilih akan muncul dalam tettingkap Selected Fields. Klik Next.



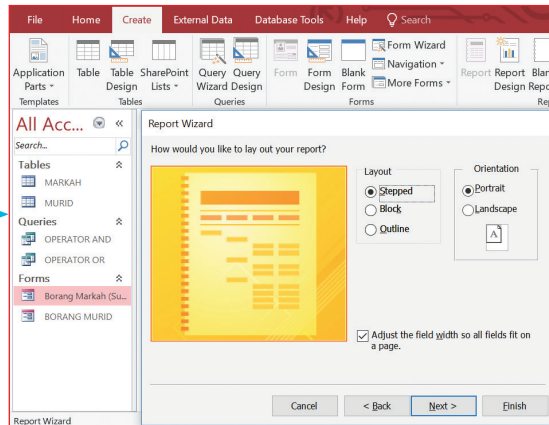
**Langkah 5** Anda boleh klik ikon > atau < untuk menentukan susunan medan yang dipaparkan dalam laporan. Klik Next.



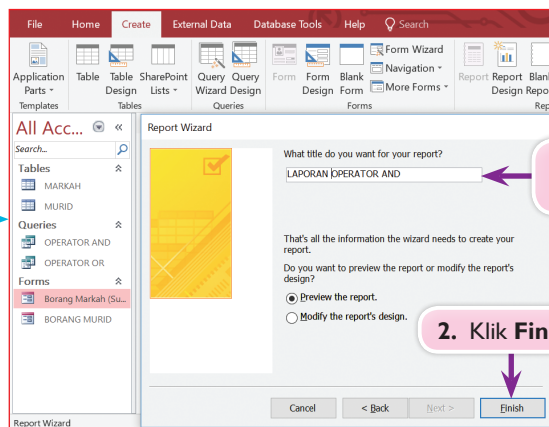
**Langkah 6** Anda boleh memilih medan untuk tujuan isihan dan klik **Ascending** atau **Descending** untuk menentukan isihan secara menaik atau menurun. Klik **Next**.



**Langkah 7** Seterusnya, pilih cara susun atur (*Layout*) dan orientasi (*Orientation*) untuk menetapkan format laporan anda. Klik **Next**.



**Langkah 8** Masukkan nama laporan anda. Klik **Finish**.



**Rajah 4.40** Langkah-langkah menjana laporan dengan menggunakan *Report Wizard*

Langkah-langkah menghasilkan laporan



[goo.gl/AuqRNx](http://goo.gl/AuqRNx)



**Tahukah Anda?**

Bilangan maksimum atribut yang boleh dimasukkan dalam laporan adalah sebanyak 255.

**Uji Minda 4.16**

Bagaimanakah cara untuk menamakan semula laporan yang telah disimpan?

Rajah 4.41 menunjukkan contoh laporan yang dipaparkan dalam susun atur (*layout*) yang berbeza, iaitu *Stepped Layout*, *Block Layout* dan *Outline Layout*.

LAPORAN OPERATOR AND\_Stepped Layout

ID_Murid	Nama_Murid	Jantina	Kaum
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU

(a) *Stepped Layout*

LAPORAN OPERATOR AND\_Block Layout

ID_Murid	Nama_Murid	Jantina	Kaum
SM101	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU
SM102	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU

(b) *Block Layout*

LAPORAN OPERATOR AND\_Outline Layout

ID_Murid	SM101	Jantina	Kaum
Nama_Murid	DAMIA ZAHRAA BINTI RAHIM	PEREMPUAN	MELAYU
ID_Murid	SM102	Jantina	Kaum
Nama_Murid	DHIA ARISSA BINTI MALIK	PEREMPUAN	MELAYU

(c) *Outline Layout*

**Rajah 4.41** Paparan laporan dalam susun atur (*layout*) yang berbeza

## Praktis Amali 4.6

### Penjanaan Laporan Berdasarkan Hasil Pertanyaan (*Query*)

1. Rujuk kepada Contoh 4.6 dan 4.7, hasilkan pertanyaan bagi OPERATOR AND dan OPERATOR OR.
2. Janakan laporan-laporan berdasarkan pertanyaan-pertanyaan yang telah dibuat.
3. Setelah berjaya, tunjukkan kepada guru anda untuk disemak.
4. Cetak satu set laporan yang telah dijana dan tampal di dalam kelas.

## Uji Kendiri 4.1

1. \_\_\_\_\_ merupakan fakta yang belum diproses.
2. Pangkalan data ialah tempat pengumpulan dan penyimpanan data secara \_\_\_\_\_ bagi sistem maklumat yang berasaskan komputer.
3. Entiti merujuk kepada suatu objek yang \_\_\_\_\_ dan boleh dikenal pasti dalam persekitaran tertentu.
4. Atribut merupakan ciri-ciri yang menerangkan sesuatu \_\_\_\_\_.
5. Nyatakan **dua** kepentingan kekunci primer.
6. (a) Nyatakan maksud kekardinalan dalam pangkalan data.  
(b) Berdasarkan situasi-situasi berikut, lakarkan jenis kekardinalan dalam lukisan gambar rajah perhubungan entiti.
  - (i) Seorang murid meminjam buku dari perpustakaan sekolah.
  - (ii) Seorang warganegara Malaysia boleh memiliki satu Pasport Malaysia Antarabangsa.
7. Nyatakan objek-objek pangkalan data yang terdapat dalam perisian aplikasi *Microsoft Access*.
8. Nyatakan **dua** kegunaan utama borang (*form*) dalam pangkalan data.
9. (a) Apakah bahasa pertanyaan (*query*) yang digunakan untuk memanipulasi data-data dalam pangkalan data?  
(b) Nyatakan pernyataan-pernyataan yang telah anda pelajari.
10. Berdasarkan pertanyaan (*query*) di bawah, nyatakan semua jadual (*table*) dan atribut (*fields*) yang terlibat.

```
SELECT ID_Murid, Nama_Murid
FROM MURID;
```

11. Lengkapkan jadual kebenaran bagi ungkapan *Boolean* yang menggunakan operator OR dan operator AND.

Operator OR

P	Q	$P \vee Q$
1	1	
1	0	
0	1	
0	0	

Operator AND

P	Q	$P \wedge Q$
1	1	
1	0	
0	1	
0	0	

12. Nyatakan cara untuk menghasilkan laporan dengan mudah yang telah anda pelajari.

13. Berikut menunjukkan dua jadual bagi entiti BUTIR\_MURID dan entiti BUTIR\_PENJAGA.



BUTIR_MURID					
ID_Murid	Nama	Jantina	Kaum	Agama	
T17-124	DENISH RAIMI BIN MALIK	LELAKI	MELAYU	ISLAM	
T17-125	TAN CHEN HONG	LELAKI	CINA	BUDDHA	
T17-182	NURUL AIN BINTI AHMAD	PEREMPUAN	MELAYU	ISLAM	
T17-201	PRIYA DAVID	PEREMPUAN	INDIA	KRISTIAN	
T17-202	ARUL A/L PARAM	LELAKI	INDIA	HINDU	
T17-225	RACHEL LIM YEE LING	PEREMPUAN	CINA	KRISTIAN	

BUTIR_PENJAGA					
No_Daftar	ID_Murid	Nama_Ibu	Nama_Bapa	No_Telefon	
1111	T17-125	YEOH MIN LI	TAN TZE SOOK	009-1234567	
1112	T17-201	DIYVA A/P SAMY	DAVID THOMAS	009-7654321	
1113	T17-225	GAN PUI MUI	RICHARD LIM KAH PEW	009-1122334	
1114	T17-124	SITI NOOR BINTI ALI	MALIK BIN RASHID	009-2211334	
1115	T17-182	SARAH BINTI HASSAN	AHMAD BIN RAHMAN	009-3322115	
1116	T17-202	KARISHMA A/P SIVA	PARAM A/L GOPAL	009-4433115	

Tuliskan pertanyaan-pertanyaan (*queries*) yang sesuai untuk memperoleh paparan yang berikut.

(a)

PERTANYAAN A		
Nama_Ibu	Nama_Bapa	
YEOH MIN LI	TAN TZE SOOK	
DIYVA A/P SAMY	DAVID THOMAS	
GAN PUI MUI	RICHARD LIM KAH PEW	
SITI NOOR BINTI ALI	MALIK BIN RASHID	
SARAH BINTI HASSAN	AHMAD BIN RAHMAN	
KARISHMA A/P SIVA	PARAM A/L GOPAL	

(c)

PERTANYAAN C		
ID_Murid	Nama	Kaum
T17-125	TAN CHEN HONG	CINA
T17-225	RACHEL LIM YEE LING	CINA

(b)

PERTANYAAN B		
ID_Murid	Nama	Kaum
T17-202	ARUL A/L PARAM	INDIA
T17-124	DENISH RAIMI BIN MALIK	MELAYU
T17-182	NURUL AIN BINTI AHMAD	MELAYU
T17-201	PRIYA DAVID	INDIA
T17-225	RACHEL LIM YEE LING	CINA
T17-125	TAN CHEN HONG	CINA

(d)

PERTANYAAN D		
Nama	Jantina	Kaum
ARUL A/L PARAM	LELAKI	INDIA
DENISH RAIMI BIN MALIK	LELAKI	MELAYU
TAN CHEN HONG	LELAKI	CINA



## 4.2 STRUKTUR KOD ARAHAN

Dalam kehidupan seharian, suatu masalah harus dianalisis, dikenal pasti dan dibahagikan kepada beberapa submasalah melalui proses leraian supaya masalah tersebut bukan sahaja kelihatan kurang rumit tetapi lebih mudah untuk diselesaikan. Bagi setiap submasalah, suatu algoritma khusus boleh dihasilkan. Gabungan kesemua algoritma yang dihasilkan akan mampu menyelesaikan masalah asal tersebut.

Pembangunan atur cara juga dilakukan dengan cara yang sama. Masalah pengaturcaraan dibahagikan kepada tugas utama dan beberapa tugas kecil (subtugas). Bagi setiap tugas kecil, satu algoritma dapat dibina.

Semasa fasa pengekodan, algoritma bagi tugas utama diterjemahkan kepada atur cara utama dan algoritma setiap tugas kecil diterjemahkan kepada subatur cara yang dipanggil *function* atau *procedure*. *Function* dan *procedure* akan diguna pakai dalam atur cara utama bagi menyelesaikan masalah tersebut.

### Tahukah Anda?

Python ialah bahasa pengaturcaraan sumber terbuka. Python telah mengalami banyak penambahbaikan dengan keluaran versi-versi baharu sejak mula digunakan pada tahun 1991.

### 4.2.1 Fungsi *Function* dan *Procedure* dalam Atur Cara

#### (i) *Function*

*Function* sesuai digunakan bagi subtugas yang perlu memulangkan satu nilai selepas tugas itu diselesaikan. Sebagai contoh, dalam permainan yang melibatkan pembelian, situasi pengiraan dana yang sedia ada dan baki dana selepas pembelian diperlukan. *Function* sesuai digunakan bagi proses pengiraan kerana kod yang sama tidak perlu ditulis semula setiap kali pemain ingin melakukan pembelian. Rajah 4.42 dan Rajah 4.43 menunjukkan paparan simulasi aplikasi android MyStemVille bagi peringkat pembelian.

Apabila pemilihan dibuat dan pembelian disahkan, *function* yang berada di belakang item-item yang dipilih itu akan terlaksana. Pengiraan berlaku dan baki terakhir dikemaskinikan di ruang penjurong sebelah kiri. Perhatikan bahawa sebelum pembelian, baki sedia ada ialah 220 dan selepas pembelian disahkan, baki terbaharu ialah 20 iaitu proses pengiraan dilakukan oleh *function* dan nilai baki terakhir dipulangkan untuk dipaparkan.

### Standard Pembelajaran

Murid boleh:

- 4.2.1 Menerangkan fungsi struktur berikut dalam atur cara:
- Function*
  - Procedure*

*Function* dan *procedure*



[goo.gl/LGsjnK](http://goo.gl/LGsjnK)





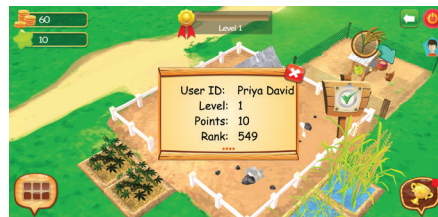
Rajah 4.42 Paparan simulasi permainan MyStemVille

(ii) Procedure

Permainan android ini boleh dimuat turun dari Google Play Store atau scan QR code ini.

[goo.gl/82yD3g](http://goo.gl/82yD3g)

*Procedure* sesuai digunakan dalam situasi di mana satu tugas perlu dilaksanakan berulang kali apabila ia diperlukan tanpa pemulangan nilai. Perhatikan bahawa *procedure* yang disimpan dalam butang profil akan terlaksana setiap kali butang ini diklik. *Procedure* ini akan memaparkan tetitangkap yang mengandungi profil pemain seperti yang ditunjukkan dalam Rajah 4.43.



Rajah 4.43 Paparan profil pemain

Penggunaan subatur cara seperti *function* dan *procedure* sesuai untuk pelaksanaan tugas yang khusus atau berulang kali dalam satu atur cara yang sama atau dalam atur cara yang berlainan. Ini menjadikan kod arahan seluruh atur cara menjadi lebih kemas, teratur, sistematik, bersifat modular dan lebih mudah dinyahpejijat serta diselenggara.

Penggunaan *function* dan *procedure* dalam menghasilkan suatu atur cara merupakan satu pendekatan yang harus dikuasai oleh setiap murid. Kemahiran ini akan membolehkan murid memainkan peranannya sebagai *team player* dalam menghasilkan atur cara yang bertaraf komersial.

Dalam kebanyakan bahasa pengaturcaraan, *function* dan *procedure* mempunyai persamaan dan perbezaan yang jelas dari segi takrifan dan kegunaan. Dalam bahasa pengaturcaraan Python, *procedure* tidak wujud kerana *procedure* dianggap sebagai *implicit function*. *Procedure* Python tetap memulangkan nilai *None* walaupun memainkan peranan

yang sama seperti *procedure* dalam bahasa pengaturcaraan yang lain, iaitu tidak melakukan sebarang pengiraan dan hanya memaparkan maklumat. *None* ialah nilai lalai dalam Python. Takrifan dan kegunaan *function* serta *procedure* dalam buku ini akan dibuat berdasarkan bahasa pengaturcaraan Python. Rajah 4.44 menunjukkan persamaan dan perbezaan antara *function* dan *procedure* Python.

Perbezaan antara *function* dan *procedure* dalam Python.



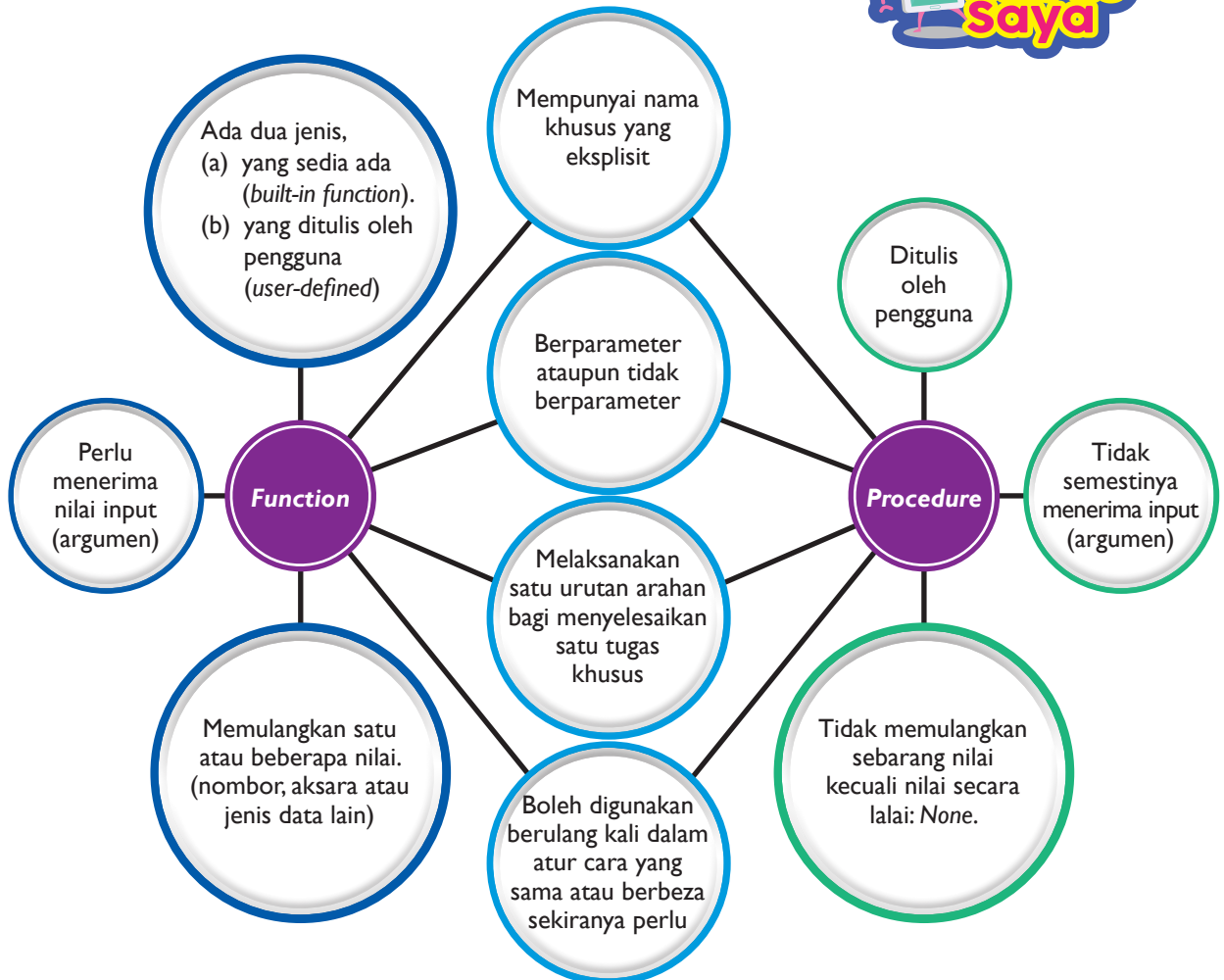
[goo.gl/5jxz9z](https://goo.gl/5jxz9z)

(Dipetik dari: <http://index-of.es/Python/Core.Python.Programming.2nd.Edition.Wesley.Chun.2006.pdf>, muka surat 448)

### Tahukah Anda ?

**Parameter** ialah pemboleh ubah yang digunakan oleh sesuatu *function* atau *procedure*.

**Argumen** ialah data atau nilai sebenar dalam pemboleh ubah tersebut.



Rajah 4.44 Persamaan dan perbezaan antara *function* dan *procedure* Python

## Standard Pembelajaran

Murid boleh:

- 4.2.2 Memberi contoh penggunaan pernyataan *function*:
- dalam (built-in)
  - dihasilkan sendiri (user-defined)

Function dalam bahasa pengaturcaraan Python



[goo.gl/DWBx41](http://goo.gl/DWBx41)

Imbas Saya

4.2.2

## Penggunaan Pernyataan Function

*Function* boleh dibahagikan kepada *built-in function* (fungsi dalaman) dan *user-defined function* (fungsi dihasilkan sendiri). Kedua-duanya mempunyai fungsi yang sama, iaitu melakukan suatu tugas yang khusus.

### (i) Function dalaman (*built-in*)

*Built-in function* ialah fungsi yang sedia ada dan disimpan dalam *library* bahasa pengaturcaraan. Kod sumber untuk setiap *built-in function* tidak boleh dilihat oleh pengatur cara. Setiap *built-in function* mempunyai nama yang deskriptif, iaitu nama yang dapat mencerminkan tugas *built-in function* berkenaan. Nama *built-in function* tidak boleh ditukar. Dalam pengaturcaraan Python, fungsi input dan fungsi output ialah *built-in function* yang sering digunakan. Tugas yang dilakukan oleh *built-in function* yang dibekalkan adalah terhad dan ringkas sahaja.

Bagi kebanyakan bahasa pengaturcaraan, sebelum sesuatu *built-in function* boleh digunakan, nama fail *library* di mana *built-in function* berkenaan disimpan hendaklah dinyatakan terlebih dahulu di bahagian atas atur cara. Dalam pengaturcaraan Python, terdapat sekumpulan *built-in function* yang boleh digunakan secara terus tanpa menyatakan terlebih dahulu nama fail *library* di mana fungsi berkenaan disimpan.

## Contoh 4.8 Penggunaan *built-in function* dalam bahasa pengaturcaraan Python.

Contoh ini menunjukkan penggunaan *built-in function* `sum()`, `max()`, `min()` dan `index()` bagi mendapatkan nama murid yang memperoleh jumlah markah tertinggi dan terendah. Jadual 4.11 menunjukkan markah yang diperoleh lima murid bagi empat mata pelajaran.

Jadual 4.11

ID murid	Nama murid	BM	BI	Mate	Sains
0001	Ali bin Azmi	87	70	80	78
0002	Aminah binti Yusof	95	80	65	75
0003	Chong Yee Ling	74	85	90	85
0004	Dayang Minsu	55	85	71	68
0005	Denish A/L Kathigasu	65	78	68	65

```

Markah Min dan Mak.py - C:\Documents\Tutorial Python\Markah Min dan Mak.py (3.6.5)
File Edit Format Run Options Window Help

#Menyimpan markah setiap murid dalam senarai
murid_1 = [87, 70, 80, 78]
murid_2 = [95, 80, 65, 75]
murid_3 = [74, 85, 90, 85]
murid_4 = [55, 85, 71, 68]
murid_5 = [65, 78, 68, 65]

#Menyimpan nama setiap murid dalam urutan senarai nama
nama = ["Ali bin Azmi", "Aminah binti Yusof", "Chong Yee Ling", "Dayang Minsu",
        "Denish A/L Kathigasu"]

#Menjumlahkan markah setiap murid
jumlah_1 = sum(murid_1)
jumlah_2 = sum(murid_2)
jumlah_3 = sum(murid_3)
jumlah_4 = sum(murid_4)
jumlah_5 = sum(murid_5)

#Menyimpan jumlah markah setiap murid dalam senarai jumlah
jumlah = [jumlah_1, jumlah_2, jumlah_3, jumlah_4, jumlah_5]

#Mencari markah tertinggi dan terendah dalam senarai jumlah
tertinggi = max(jumlah)
terendah = min(jumlah)

#Menggunakan sub-function untuk mencari indeks
indeks_tertinggi = jumlah.index(tertinggi)
indeks_terendah = jumlah.index(terendah)

#Menggunakan sub-function untuk mencari nama murid
murid_markah_tertinggi = nama[indeks_tertinggi]
murid_markah_terendah = nama[indeks_terendah]

#Mencetak nama berserta markah tertinggi dan terendah
print(murid_markah_tertinggi, "memperoleh jumlah markah tertinggi, iaitu ", tertinggi)
print(murid_markah_terendah, "memperoleh jumlah markah terendah, iaitu ", terendah)

```

Semua markah bagi setiap murid mesti disimpan dalam senarai masing-masing.

Nama semua murid harus disimpan dalam satu senarai mengikut tertib senarai markah.

Senarai markah setiap murid dihantar ke *built-in function sum()* dan nilai yang dipulangkan, iaitu jumlah markah disimpan dalam pemboleh ubah yang sepadan.

Jumlah markah setiap murid disimpan dalam satu senarai bernama jumlah. Kedudukan jumlah markah adalah mengikut tertib.

Senarai jumlah dihantar sebagai argumen kepada *function max()* dan *min()*. *Function max()* akan memulangkan markah tertinggi dan *function min()* akan memulangkan markah terendah dalam senarai jumlah.

Jumlah markah tertinggi dan terendah dihantar sebagai argumen kepada *sub-function index()*. Nombor indeks yang dipulangkan disimpan dalam pemboleh ubah masing-masing.

Nilai indeks yang disimpan dalam pemboleh ubah digunakan untuk mengesan nama murid yang mendapat jumlah markah tertinggi dan terendah.

Ln: 39 Col:0

Rajah 4.45 Kod atur cara Contoh 4.8

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Documents\Tutorial Python\Markah Min dan Mak.py =====
Chong Yee Ling memperoleh jumlah markah tertinggi, iaitu 334
Denish A/L Kathigasu memperoleh jumlah markah terendah, iaitu 276
>>>
Ln: 7 Col:4

```

Rajah 4.46 Output bagi kod atur cara Contoh 4.8



(ii) **Function dihasilkan sendiri (user-defined)**

*User-defined function* ialah satu set arahan yang ditulis oleh pengatur cara untuk melaksanakan suatu tugas khas yang akan berulang dalam atur cara utama. *User-defined function* yang ditulis juga perlu mempunyai nama yang deskriptif, iaitu nama yang dapat mencerminkan tugasnya. Lazimnya, *user-defined function* ditulis kerana ketiadaan *built-in function* yang dapat memenuhi kehendak pengatur cara. Seperti *built-in function*, *user-defined function* mampu menerima argumen dan memulangkan nilai kepada atur cara yang memanggilnya.

**Contoh 4.9**

Penggunaan *user-defined function* yang mengira kuasa dua dalam bahasa pengaturcaraan Python.

```

kuasa_dua.py - C:\Documents\Tutorial Python\kuasa_dua.py (3.6.5)
File Edit Format Run Options Window Help
# Fungsi yang mengira kuasa dua
def kuasadua(x):
    return x*x

# Bahagian utama atur cara
# Minta pengguna memasukkan satu nombor
nom = int(input("Masukkan satu nombor integer: "))

# panggilan fungsi
nom_kuasa = kuasadua(nom)
print("Kuasa dua bagi", nom, "ialah", nom_kuasa)
Ln: 1 Col:0

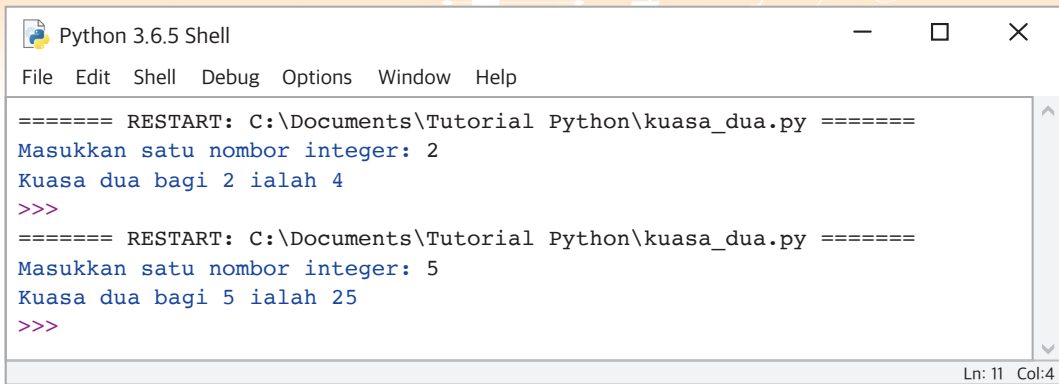
```

*User-defined function* (kuasadua) ini:

- Menerima nilai argumen yang dihantar dan menyimpannya dalam parameter x
- Mengira nilai  $x^2$
- Memulangkan hasil kiraan kepada pernyataan kod yang meminta kiraan ini

- Pernyataan kod ini memanggil *function* kuasadua dan menghantar nilai dalam pemboleh ubah nom kepadanya.
- Terima nilai yang dipulangkan oleh *function* kuasadua dan umpukkan kepada pemboleh ubah nom\_kuasa.

Rajah 4.47 Kod atur cara Contoh 4.9



```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help

===== RESTART: C:\Documents\Tutorial Python\kuasa_dua.py =====
Masukkan satu nombor integer: 2
Kuasa dua bagi 2 ialah 4
>>>

===== RESTART: C:\Documents\Tutorial Python\kuasa_dua.py =====
Masukkan satu nombor integer: 5
Kuasa dua bagi 5 ialah 25
>>>

Ln: 11 Col:4

```

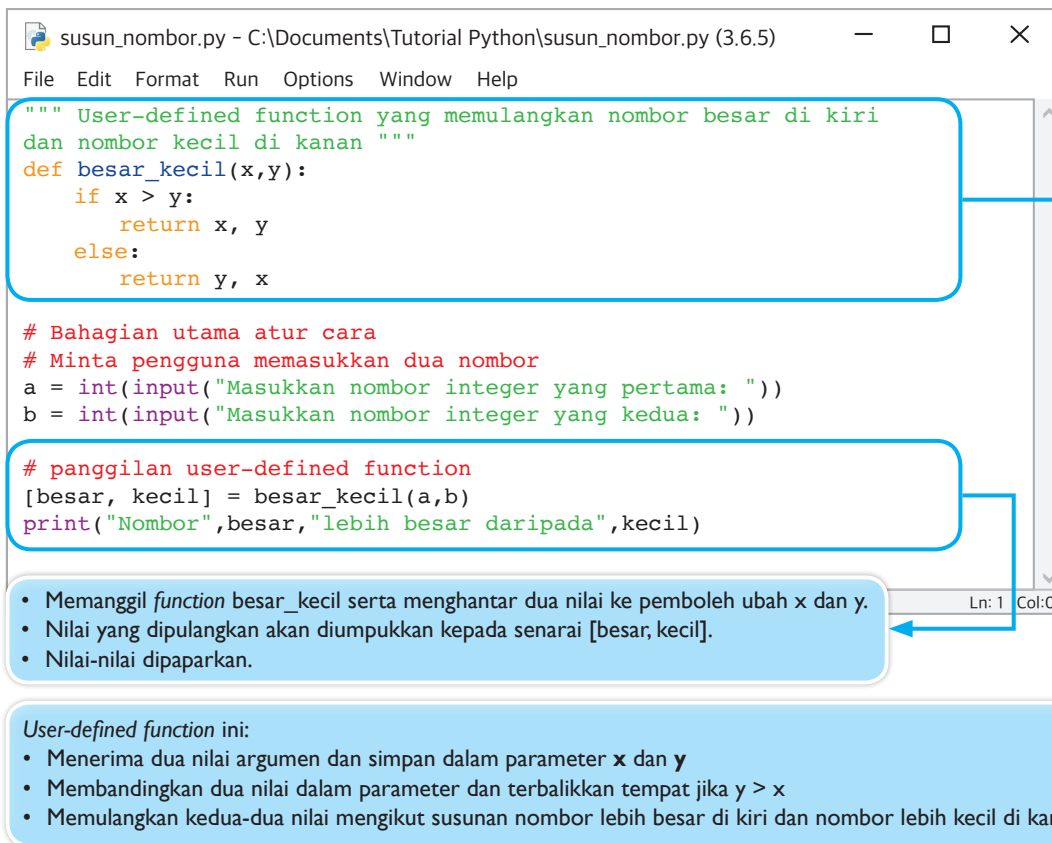
Rajah 4.48 Output bagi kod atur cara Contoh 4.9

### Uji Minda 4.17

Nyatakan *built-in function* yang digunakan dalam bahagian utama atur cara dalam Contoh 4.9 yang membolehkan pengguna memasukkan nombor.

### Contoh 4.10

Penggunaan *user-defined function* yang membandingkan dua nombor dalam bahasa pengaturcaraan Python.



```

susun_nombor.py - C:\Documents\Tutorial Python\susun_nombor.py (3.6.5)
File Edit Format Run Options Window Help

""" User-defined function yang memulangkan nombor besar di kiri
dan nombor kecil di kanan """
def besar_kecil(x,y):
    if x > y:
        return x, y
    else:
        return y, x

# Bahagian utama atur cara
# Minta pengguna memasukkan dua nombor
a = int(input("Masukkan nombor integer yang pertama: "))
b = int(input("Masukkan nombor integer yang kedua: "))

# panggilan user-defined function
[besar, kecil] = besar_kecil(a,b)
print("Nombor",besar,"lebih besar daripada",kecil)

Ln: 1 Col:0

```

- Memanggil *function* `besar_kecil` serta menghantar dua nilai ke pemboleh ubah `x` dan `y`.
- Nilai yang dipulangkan akan diumpukkan kepada senarai `[besar, kecil]`.
- Nilai-nilai dipaparkan.

*User-defined function* ini:

- Menerima dua nilai argumen dan simpan dalam parameter `x` dan `y`
- Membandingkan dua nilai dalam parameter dan terbalikkan tempat jika  $y > x$
- Memulangkan kedua-dua nilai mengikut susunan nombor lebih besar di kiri dan nombor lebih kecil di kanan

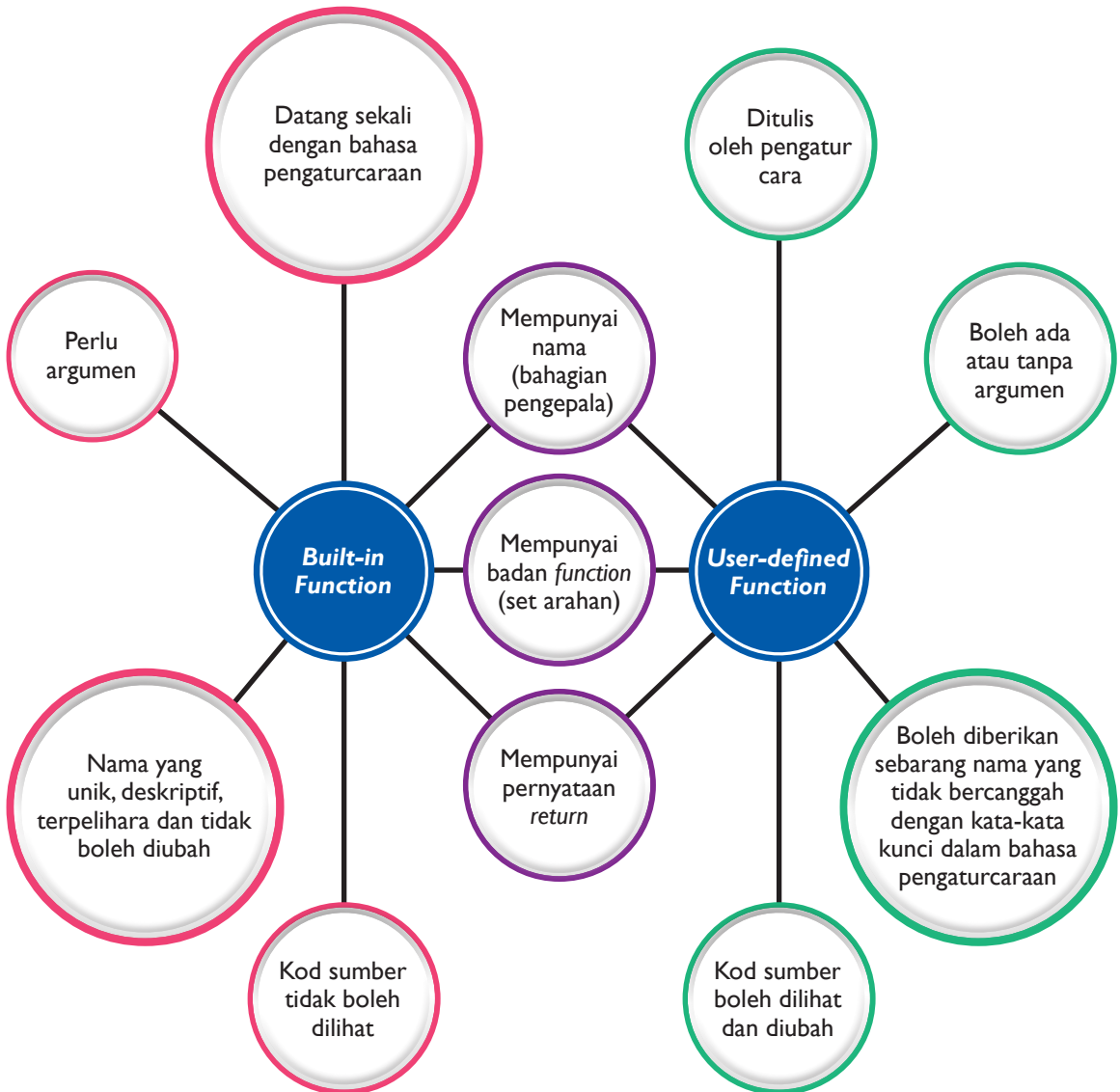
Rajah 4.49 Kod atur cara Contoh 4.10

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Documents\Tutorial Python\susun_nombor.py =====
Masukkan nombor integer yang pertama: 6
Masukkan nombor integer yang kedua: 9
Nombor 9 lebih besar daripada 6
>>>
Ln: 8 Col:4

```

Rajah 4.50 Output bagi kod atur cara Contoh 4.10



Rajah 4.51 Persamaan dan perbezaan antara *built-in function* dan *user-defined function*





## Aktiviti 4.8



### Aktiviti Kumpulan

#### Mengenal Pasti *Function* dan *Procedure*



Kaedah  
*Think-Pair-Share*

1. Kaji kod atur cara berikut, kenal pasti semua *function* dan *procedure*.

```

Harga Tempahan Kek.py - C:\Documents\Tutorial Python\Harga Temp...
File Edit Format Run Options Window Help

jenis_kek = ["keju", "mentega", "pelangi", "kopi"]
harga_kek = [40, 35, 35, 30]
jumlah = [0, 1, 2, 3]

a = int(input("Masukkan tempahan untuk kek keju: "))
b = int(input("Masukkan tempahan untuk kek mentega: "))
c = int(input("Masukkan tempahan untuk kek pelangi: "))
d = int(input("Masukkan tempahan untuk kek kopi: "))

tempahan = [a, b, c, d]

def jumlah_harga():
    for i in range(4):
        jumlah[i] = harga_kek[i] * tempahan[i]
    return (jumlah)

def cetak():
    print("\n\nTempahan anda ialah:")
    print(a, "kek", jenis_kek[0])
    print(b, "kek", jenis_kek[1])
    print(c, "kek", jenis_kek[2])
    print(d, "kek", jenis_kek[3])
    print("\nJumlah harga untuk tempahan ialah RM", sum(jumlah))

jumlah_harga()
cetak()
Ln: 1 Col: 0
  
```



[goo.gl/wMqamQ](http://goo.gl/wMqamQ)

2. Kenal pasti perkara berikut dan tulis pada sehelai kertas.
  - (a) Jenis dan fungsi *built-in function*
  - (b) *User-defined function* dan *procedure* serta tugasnya
  - (c) Output atur cara
3. Kongsikan idea bersama-sama ahli kumpulan anda. Bincangkan cara-cara untuk menambah baik hasil dapatan anda.
4. Sediakan satu persembahan dan bentangkan hasil dapatan kumpulan anda di dalam kelas.

## Standard Pembelajaran

Murid boleh:

4.2.3 Menulis pernyataan *function* dan *procedure*

4.2.3

## Menulis Pernyataan *Function* dan *Procedure*

### >> *Function*

Penulisan *user-defined function* bermula dengan mentakrifkan *function* tersebut di permulaan atur cara. Selepas itu, pengguna boleh memanggil fungsi tersebut setiap kali tugas khas *function* perlu dilaksanakan dalam atur cara. Berikut menunjukkan sintaks pentakrifan suatu *user-defined function*.

```
def nama_function (parameter):  
    badan_function  
    return (nilai)
```

### Tahukah Anda?

Parameter digunakan untuk:

1. Menerima dan menyimpan data yang dihantar kepadanya semasa dipanggil
2. Memulangkan data kepada pernyataan arahan, *function*, *procedure* atau atur cara lain yang memanggilnya

Kata kunci **def** digunakan untuk mentakrifkan *function*. Nama sesuatu *function* harus mencerminkan tugas khas yang akan dilakukannya. Parameter ialah pemboleh ubah untuk menerima argumen (nilai) yang dihantar kepada *function* semasa dipanggil. Parameter dimasukkan dalam tanda kurungan ( ). Pentakrifan nama *function* dan parameter mesti diakhiri dengan tanda titik bertindih.

Badan *function* dimulakan sebaris selepas nama *function* dan perlu inden dari jidar kiri. Badan *function* terdiri daripada set arahan (*instructions*) bagi tugasnya. Perkataan **return** digunakan untuk memulangkan nilai atau output yang dihasilkan oleh *function* tersebut.

Dalam kebanyakan bahasa pengaturcaraan, pernyataan **return** dalam *function* memulangkan satu nilai sahaja. Nilai ini boleh merupakan nilai tunggal atau satu ungkapan yang menghasilkan satu nilai tunggal. Walau bagaimanapun, dalam bahasa pengaturcaraan Python, nilai yang dipulangkan dalam pernyataan **return** *function* boleh berupa satu nilai tunggal atau sekumpulan nilai yang diasingkan dengan tanda koma. Sebagai contoh,

#### Situasi 1:

```
def hasiltambah (x,y):  
    jumlah = x + y  
    return (jumlah)
```

Pernyataan **return** memulangkan satu nilai tunggal.

#### Situasi 2:

```
def hasiltambah (x,y):  
    return (x + y)
```

Pernyataan **return** memulangkan satu nilai tunggal dari ungkapan yang digunakan.

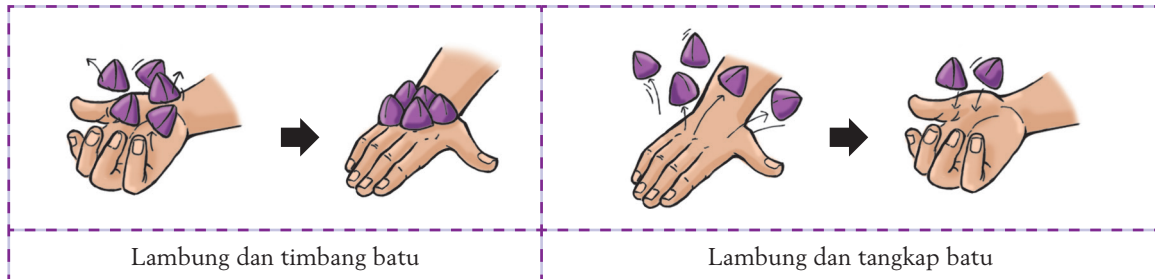
#### Situasi 3:

```
def susun_naik (x, y):  
    if x > y:  
        return (y, x)  
    else:  
        return (x, y)
```

Pernyataan **return** memulangkan lebih daripada satu nilai. Pernyataan **return** ini memulangkan dua nilai dalam susunan menaik.

### Contoh 4.11 Penggunaan *user-defined function* bagi simulasi permainan batu seremban.

Batu seremban ialah sejenis permainan tradisional yang dimainkan di Malaysia. Dalam permainan ini, seorang pemain bermula dengan peringkat yang mudah sehingga ke peringkat terakhir yang disebut buah lapan. Kemudian, barulah pemain melakukan aktiviti timbang dan tangkap untuk mendapatkan mata.



Rajah 4.52 Cara lambung dan timbang dan lambung dan tangkap batu

Dalam Contoh 4.11, simulasi aktiviti timbang dan tangkap dalam permainan boleh diwujudkan dan nama *user-defined function* yang akan dibina dinamakan `bilangan_batu()`. *User-defined function* ini akan menerima satu nilai yang mewakili bilangan batu yang akan digunakan dalam permainan, iaitu lima. Nilai ini akan dihantar ke *sub-function* bagi *built-in function* `random()`, iaitu `random.randint()`. Nilai yang dipulangkan kali pertama adalah bilangan batu yang berjaya ditimbang. Selepas itu, nilai ini dihantar ke *function* `random.randint()` semula dan nilai yang dipulangkan kali kedua mewakili bilangan batu yang berjaya ditangkap.

```

Batu Seremban.py - C:\Documents\Tutorial Python\Batu Seremban.py (3.6.5)
File Edit Format Run Options Window Help

import random

# User-defined function untuk simulasi menimbang batu
def bilangan_batu(x):
    hasil =(random.randint(0,x))
    return hasil

# Membuat panggilan function.
# Hantar nilai bilangan batu yang digunakan ke dalam bilangan_batu()
terus = "y"
while terus == "y":
    hasil_timbang = bilangan_batu(5)
    print("Hasil timbangan anda ialah " + str(hasil_timbang) + " batu.")

# Menangkap batu. Membuat panggilan fungsi.
# Hantar hasil timbangan ke dalam bilangan_batu()
if hasil_timbang > 0:
    tangkap = bilangan_batu(hasil_timbang)
    print("Hasil tangkapan anda ialah " + str(tangkap))
    terus = input("\nTeruskan [Y] atau Berhenti [T]? Tekan [Y/T] ")
    print("")

```

Gelung *while* digunakan supaya aktiviti timbang dan tangkap dapat diteruskan sehingga pemain menamatkan permainan.

Sekiranya batu berjaya ditimbang, barulah *user-defined function* `bilangan_batu()` dipanggil semula. Kali ini argumen yang dihantar ialah `hasil_timbang` dan nilai yang dipulangkan ialah bilangan batu yang berjaya ditangkap.

Rajah 4.53 Kod atur cara Contoh 4.11

Panggilan *function* `bilangan_batu` diletakkan dalam gelung ulangan *while* supaya aktiviti timbang dan tangkap batu dapat dilakukan secara berulang.

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Documents\Tutorial Python\Batu Seremban.py =====
Hasil timbangan anda ialah 1 batu.
Hasil tangkapan anda ialah 1

Teruskan [Y] atau Berhenti [T]? Tekan [Y|T] Y

Hasil timbangan anda ialah 2 batu.
Hasil tangkapan anda ialah 1

Teruskan [Y] atau Berhenti [T]? Tekan [Y|T] T

>>>
Ln: 25 Col:4

```

Rajah 4.54 Output bagi kod atur cara Contoh 4.11



## Aktiviti 4.9



Aktiviti Kumpulan

### Menulis Pernyataan *Function*



Kaedah *Brainstorming Activity*

Corak perbelanjaan murid sekolah di pesta buku:

Nama murid	RM		
	Buku cerita	Buku latihan	Buku rujukan
Aini	12	15	33
Aru	10	12	15
Lee	15	18	0

Anda ialah ahli jawatankuasa penganjur pesta buku. Anda ditugaskan untuk mendapatkan data-data mengenai:

- Jumlah bilangan buku yang dibeli oleh murid sekolah;
  - secara keseluruhan.
  - mengikut jenis bahan.
- Jenis bahan bacaan yang paling disukai oleh murid sekolah.

- Anda diberi masa untuk:
  - Mengkaji situasi di atas dan meleraikan masalah
  - Menyatakan tugas khas yang telah dikenal pasti
  - Menulis pernyataan *function* bagi melakukan tugas khas yang telah dikenal pasti
- Selepas tamat tempoh yang diberikan, anda dikehendaki berkumpul dalam kumpulan. Kongsikan dan bincangkan hasil dapatan masing-masing bersama-sama rakan sekumpulan.
- Gunakan bahasa pengaturcaraan Python untuk mengkod dan menguji pernyataan *function* yang dihasilkan.



## ➤➤ Procedure

Sintaks untuk menulis *procedure* hampir serupa dengan sintaks pentakrifan *user-defined function*, cuma *procedure* tidak memulangkan sebarang nilai (output) kembali kepada atur cara atau pernyataan kod yang memanggilnya. Badan *procedure* hanya terdiri daripada set arahan bagi tugas khas.

```
def nama_procedure (parameter):
    badan_procedure
```

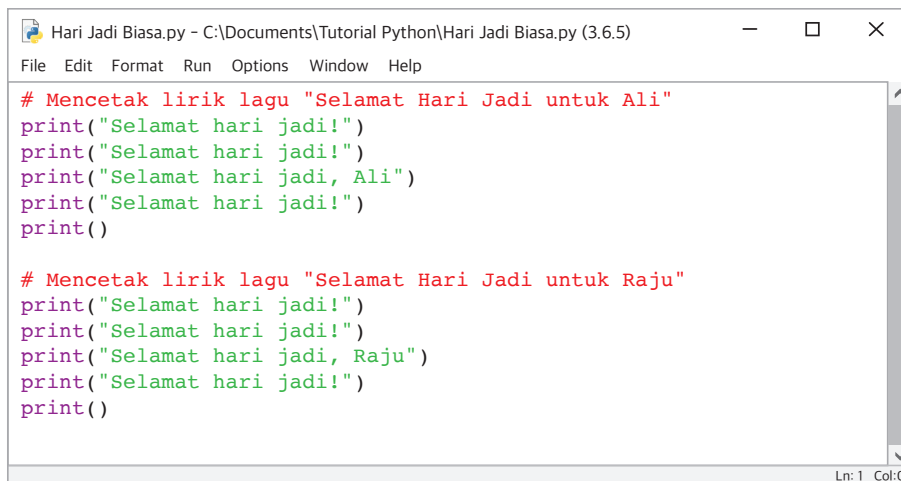
Kata kunci **def** digunakan untuk mentakrifkan *procedure*. Nama sesuatu *procedure* harus mencerminkan tugas khas yang akan dilakukannya. Jika mempunyai input atau parameter, maka parameter perlu dinyatakan dalam tanda kurungan ( ). Jika tidak, tanda kurungan dibiarkan kosong. Pentakrifan nama *procedure* dengan atau tanpa parameter mesti diakhiri dengan tanda titik bertindih.

Badan *procedure* dimulakan sebaris selepas nama *procedure* dan perlu inden dari jidar kiri. Badan *procedure* terdiri daripada set arahan bagi melaksanakan tugasnya.

Setelah *procedure* siap dilaksanakan, pelaksanaan atur cara akan kembali semula ke baris selepas pemanggilan *procedure*.

**Contoh 4.12** Penggunaan *procedure* untuk mencetak lirik lagu menggunakan bahasa pengaturcaraan Python.

Aimah ialah seorang guru tadika dan kelasnya mempunyai 15 orang murid. Setiap bulan, Aimah akan menyediakan lirik lagu “Selamat Hari Jadi” untuk murid-muridnya. Katakan dua orang murid menyambut hari jadi pada bulan ini. Berikut menunjukkan satu segmen kod atur cara yang akan mencetak lirik lagu “Selamat Hari Jadi” yang bernama tanpa menggunakan *procedure*.



```
Hari Jadi Biasa.py - C:\Documents\Tutorial Python\Hari Jadi Biasa.py (3.6.5)
File Edit Format Run Options Window Help

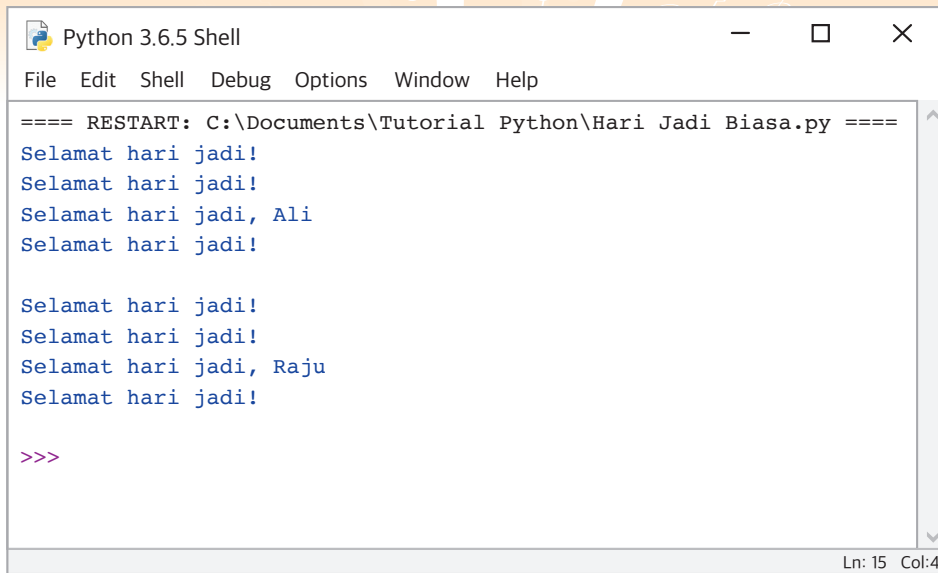
# Mencetak lirik lagu "Selamat Hari Jadi untuk Ali"
print("Selamat hari jadi!")
print("Selamat hari jadi!")
print("Selamat hari jadi, Ali")
print("Selamat hari jadi!")
print()

# Mencetak lirik lagu "Selamat Hari Jadi untuk Raju"
print("Selamat hari jadi!")
print("Selamat hari jadi!")
print("Selamat hari jadi, Raju")
print("Selamat hari jadi!")
print()

Ln: 1 Col:0
```

Rajah 4.55 Kod atur cara Contoh 4.12





```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
==== RESTART: C:\Documents\Tutorial Python\Hari Jadi Biasa.py ====
Selamat hari jadi!
Selamat hari jadi!
Selamat hari jadi, Ali
Selamat hari jadi!

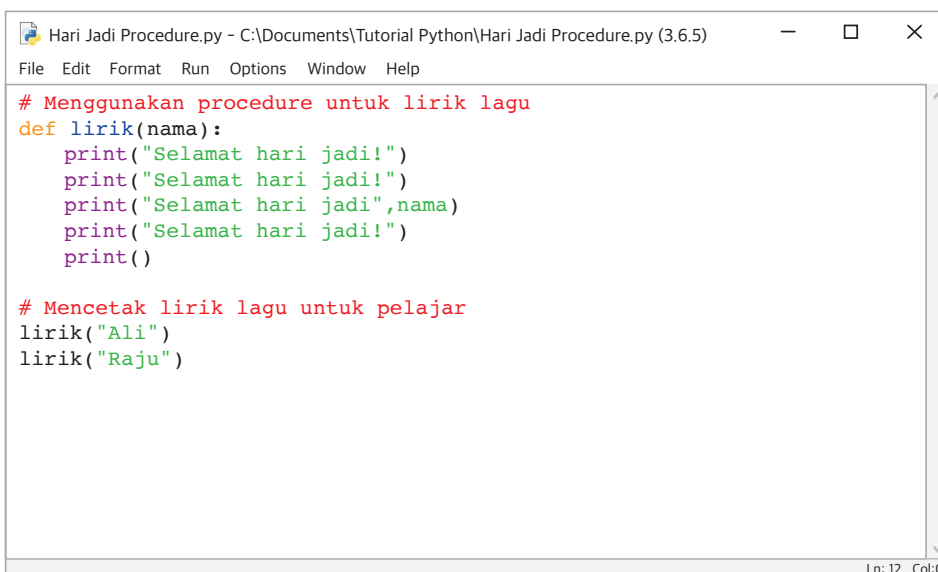
Selamat hari jadi!
Selamat hari jadi!
Selamat hari jadi, Raju
Selamat hari jadi!

>>>
```

**Rajah 4.56** Output bagi kod atur cara Contoh 4.12

Aimah mendapati lirik lagu tersebut adalah sama dan berulang. Bayangkan jika Aimah ingin mencetak lirik lagu untuk semua muridnya.

Penggunaan *procedure* lirik() dengan parameter boleh digunakan supaya seluruh lirik tidak perlu ditaip berulang kali. Ini telah menjimatkan masa menaip, kejadian ralat berkurangan dan segmen kod kelihatan lebih kemas. Berikut menunjukkan segmen kod yang telah ditambah baik dengan penggunaan *procedure* untuk mencetak lirik lagu “Selamat Hari Jadi”.



```
Hari Jadi Procedure.py - C:\Documents\Tutorial Python\Hari Jadi Procedure.py (3.6.5)
File Edit Format Run Options Window Help
# Menggunakan procedure untuk lirik lagu
def lirik(nama):
    print("Selamat hari jadi!")
    print("Selamat hari jadi!")
    print("Selamat hari jadi",nama)
    print("Selamat hari jadi!")
    print()

# Mencetak lirik lagu untuk pelajar
lirik("Ali")
lirik("Raju")
```

**Rajah 4.57** Kod atur cara Contoh 4.12 yang menggunakan *procedure*

Setelah penggunaan *procedure*, hanya dua baris kod sahaja diperlukan oleh Aimah bagi mencetak lirik lagu untuk murid-muridnya. Ini memudahkan pencetakan lirik lagu yang seterusnya untuk murid-murid yang lain.



## Aktiviti 4.10



### Aktiviti Kumpulan

#### Menulis Pernyataan *Procedure*



Kaedah  
Gallery Walk

1. Kaji segmen kod berikut. Tulis cadangan anda untuk meringkaskan kod di bawah dengan menggunakan *procedure*.



[goo.gl/DzbwE3](https://goo.gl/DzbwE3)

```
Lagu Chan Mali Chan.py - C:\Documents\Tutorial Python\Lagu Chan Mali...
File Edit Format Run Options Window Help

# Lirik Lagu Chan Mali Chan

print("Lirik Lagu Chan Mali Chan\n")

print("Di mana dia anak kambing saya?")
print("Anak kambing saya yang makan daun talas")
print("Di mana dia buah hati saya?")
print("Buah hati saya bagai telur dikupas\n")

print("Chan mali chan, chan mali chan")
print("Chan mali chan, ketipung payung")
print("Chan mali chan, chan mali chan")
print("Chan mali chan, ketipung payung\n")

print("Di mana dia anak kambing saya?")
print("Anak kambing saya yang makan daun talas")
print("Di mana dia buah hati saya?")
print("Buah hati saya bagai telur dikupas\n")

print("Chan mali chan, chan mali chan")
print("Chan mali chan, ketipung payung")
print("Chan mali chan, chan mali chan")
print("Chan mali chan, ketipung payung\n")

Ln: 27 Col:0
```

2. Pamerkan hasil kerja anda pada dinding kelas untuk dilihat oleh kumpulan lain.
3. Murid-murid digalakkan untuk menulis komen tentang hasil kerja kumpulan lain dan menampalkannya di atas hasil kerja tersebut.
4. Guru akan memilih hasil kerja yang terbaik.
5. Gunakan *procedure* yang serupa, sediakan satu segmen kod Python untuk lagu negeri anda. Anda digalakkan untuk menguji segmen kod anda menggunakan perisian aplikasi Python.



## Standard Pembelajaran

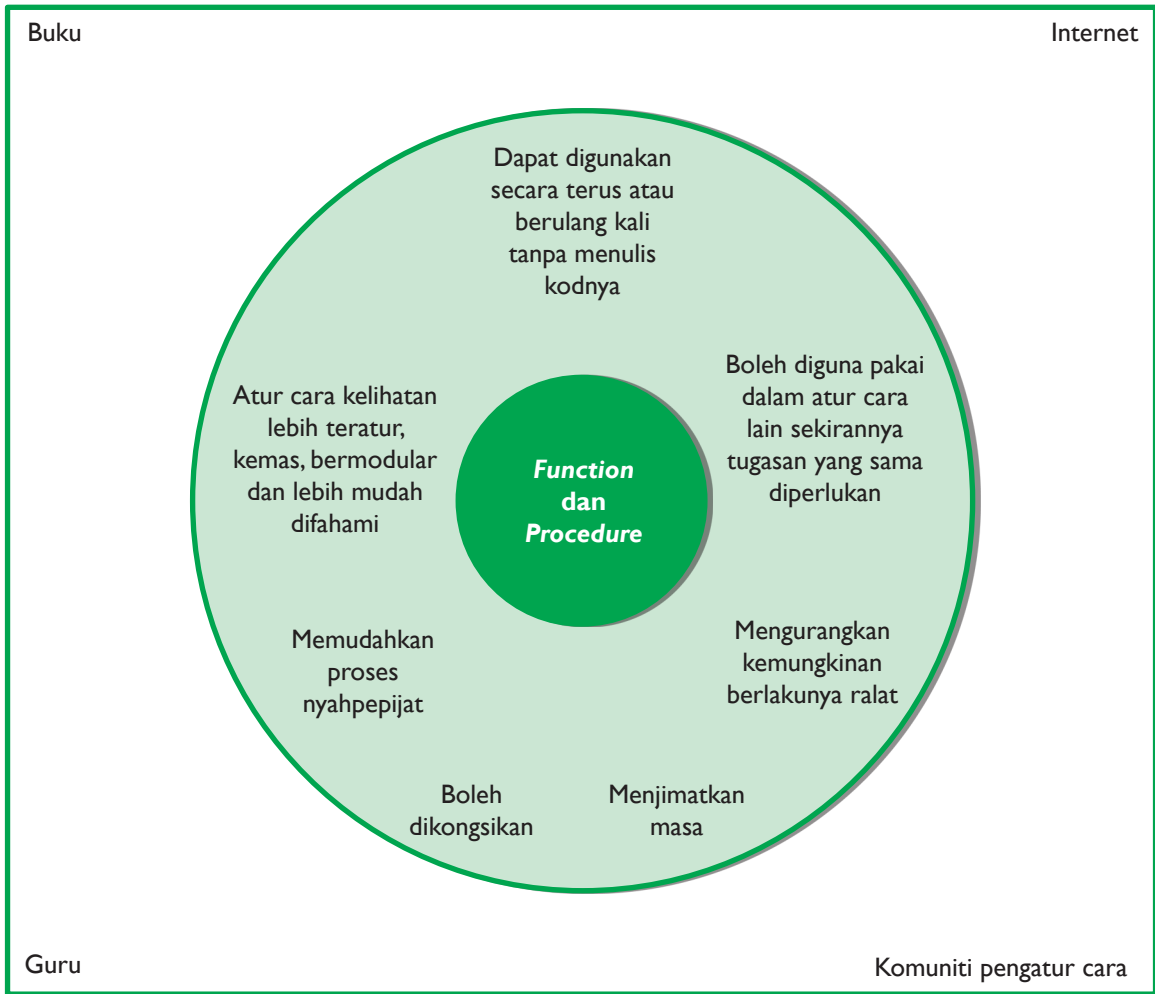
Murid boleh:

- 4.2.4 Menghasilkan atur cara yang melibatkan:
- function*
  - procedure*

4.2.4

## Menulis Atur Cara yang Melibatkan *Function* dan *Procedure*

Anda telah mempelajari cara menulis pernyataan *function* dan *procedure* di subtopik 4.2.3. Penggunaan *function* dan *procedure* semasa penulisan kod untuk satu atur cara amat penting dan wajib. Rajah 4.58 menunjukkan kepentingan penggunaan *function* dan *procedure*.



Rajah 4.58 Kepentingan *function* dan *procedure* dalam penghasilan atur cara



**Contoh 4.13** Menghasilkan satu atur cara kalkulator yang melibatkan gabungan *function* dan *procedure*.

Guru meminta anda menghasilkan satu atur cara kalkulator yang dapat melakukan empat operasi asas, iaitu tambah, tolak, darab dan bahagi terhadap dua nombor integer yang dimasukkan oleh pengguna atur cara.

Bagi menyelesaikan masalah ini, anda perlu mengaplikasikan konsep pemikiran komputasional dalam fasa-fasa pembangunan atur cara. Output yang dikehendaki adalah seperti berikut:

Menu Kalkulator

1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat

Pilihan anda [1 hingga 5]: 3

Masukkan nombor pertama : 4

Masukkan nombor kedua : 5

Output:  $4 \times 5 = 20$

Terima kasih kerana menggunakan saya.

### Fasa Analisis Masalah

1. Mengenal pasti masalah: Membina atur cara yang boleh melaksanakan empat operasi.
2. Mengenal pasti input, proses dan output:
  - (a) Input: Pilihan operasi dari menu kalkulator dan memasukkan dua nombor.
  - (b) Proses: Lakukan operasi yang dipilih.
  - (c) Output: Paparkan jawapan bagi dua nombor yang dimasukkan berdasarkan operasi yang dipilih.

### Fasa Reka Bentuk Atur Cara

1. Membangunkan algoritma bagi tugas-tugas kecil (*sub-task*).
2. Menulis pseudokod.
3. Melakar carta alir.
4. Mereka bentuk antara muka pengguna untuk input dan output data.

### Imbas Kembali

Fasa-fasa pembangunan atur cara.

Analisis Masalah

Reka Bentuk  
Atur Cara

Pengekodan

Pengujian dan  
Penyahpejijatan

Dokumentasi

### Uji Minda 4.18

Apakah teknik-teknik pemikiran komputasional yang digunakan semasa fasa analisis masalah?

## Pseudokod

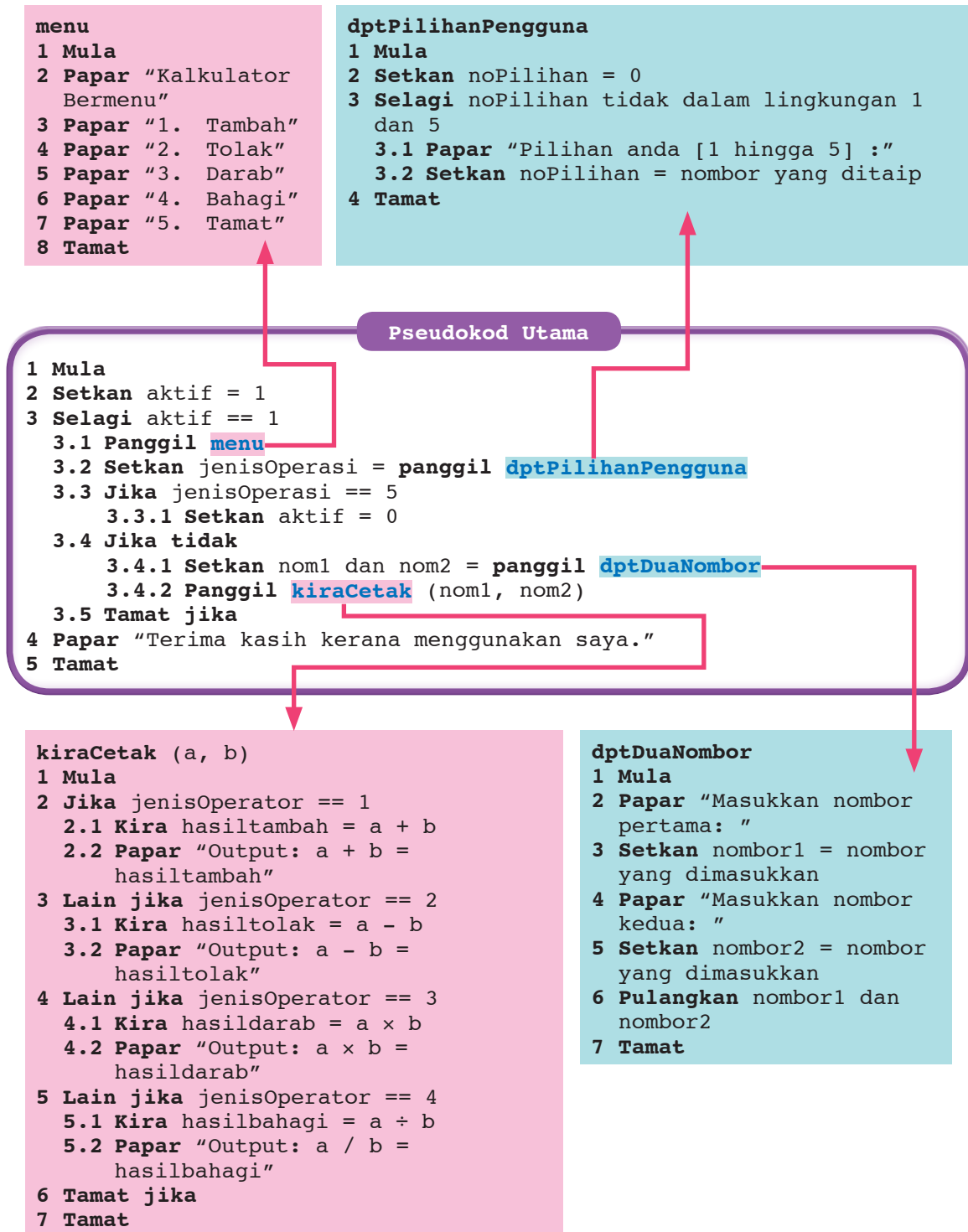
```
1 Mula
2 Isytihar pemboleh ubah aktif, pilihan, nombor1, nombor2, hasil
3 Setkan aktif = 1
4 Selagi aktif == 1
    4.1 Papar menu berikut:
        Kalkulator Bermenu
        1. Tambah
        2. Tolak
        3. Darab
        4. Bahagi
        5. Tamat
    4.2 Papar mesej "Pilihan anda [1 hingga 5]:"
    4.3 Setkan pilihan = nombor yang dimasukkan
    4.4 Jika pilihan == 1
        4.4.1 Papar mesej "Masukkan nombor pertama:"
        4.4.2 Setkan pertama = nombor yang dimasukkan
        4.4.3 Papar mesej "Masukkan nombor kedua:"
        4.4.4 Setkan kedua = nombor yang dimasukkan
        4.4.5 Kira hasil = pertama + kedua
        4.4.6 Papar "Output: nilai pertama + nilai kedua = hasil"
    4.5 Jika pilihan == 2
        4.5.1 Papar mesej "Masukkan nombor pertama:"
        4.5.2 Setkan pertama = nombor yang dimasukkan
        4.5.3 Papar mesej "Masukkan nombor kedua:"
        4.5.4 Setkan kedua = nombor yang dimasukkan
        4.5.5 Kira hasil = pertama - kedua
        4.5.6 Papar "Output: nilai pertama - nilai kedua = hasil"
    4.6 Jika pilihan == 3
        4.6.1 Papar mesej "Masukkan nombor pertama:"
        4.6.2 Setkan pertama = nombor yang dimasukkan
        4.6.3 Papar mesej "Masukkan nombor kedua:"
        4.6.4 Setkan kedua = nombor yang dimasukkan
        4.6.5 Kira hasil = pertama * kedua
        4.6.6 Papar "Output: nilai pertama x nilai kedua = hasil"
    4.7 Jika pilihan == 4
        4.7.1 Papar mesej "Masukkan nombor pertama:"
        4.7.2 Setkan pertama = nombor yang dimasukkan
        4.7.3 Papar mesej "Masukkan nombor kedua:"
        4.7.4 Setkan kedua = nombor yang dimasukkan
        4.7.5 Kira hasil = pertama / kedua
        4.7.6 Papar "Output: nilai pertama / nilai kedua = hasil"
    4.8 Jika pilihan == 5
        4.8.1 Setkan aktif = 0
5 Papar mesej "Terima kasih kerana menggunakan saya."
6 Tamat
```

### Petunjuk:

- Biru** – Set pernyataan ini diasingkan dan dijadikan satu subpseudokod `dptDuaNombor`.
- Ungu** – Set pernyataan ini diasingkan dan dijadikan satu subpseudokod `kiraCetak`.
- Hijau** – Set pernyataan ini diasingkan dan dijadikan satu subpseudokod `menu`.
- Kelabu** – Set pernyataan ini diasingkan dan dijadikan satu subpseudokod `dptPilihanPengguna`.

## Pseudokod yang Dimurnikan Melalui Pengaplikasian Pemikiran Komputasional

Pseudokod berikut adalah lebih tersusun, padat, kemas dan bermodular.





## Uji Minda 4.19

Apakah teknik pemikiran komputasional yang digunakan semasa fasa pengekodan?

## Fasa Pengekodan

Semasa fasa pengekodan, anda perlu mengenal pasti penulisan sintaks-sintaks bagi *function*, *procedure*, input, proses dan output. Selain itu, anda juga perlu mengenal pasti jenis data input yang harus digunakan.



[goo.gl/onbYbT](http://goo.gl/onbYbT)

```

Kalkulator Bermenu_2.py - C:\Documents\Tutorial Python\Kalkulator Bermenu_2.py (3.6.5)
File Edit Format Run Options Window Help

# Procedure menu()
def menu():
    print("Kalkulator Bermenu")
    print("1. Tambah")
    print("2. Tolak")
    print("3. Darab")
    print("4. Bahagi")
    print("5. Tamat")

# Function dptPilihanPengguna()
def dptPilihanPengguna():
    noPilihan = 0
    while (noPilihan < 1) or (noPilihan > 5):
        noPilihan = int(input("Pilihan anda [1 hingga 5]: "))
    return noPilihan

# Function dptDuaNombor()
def dptDuaNombor():
    nombor1 = int(input("Masukkan nombor pertama : "))
    nombor2 = int(input("Masukkan nombor kedua : "))
    return nombor1, nombor2

# Procedure kiraCetak()
def kiraCetak(jenisOperator, a, b):
    if jenisOperator == 1:
        print("Output: " + str(a) + " + " + str(b) + " = " + str(a + b) + "\n")
    elif jenisOperator == 2:
        print("Output: " + str(a) + " - " + str(b) + " = " + str(a - b) + "\n")
    elif jenisOperator == 3:
        print("Output: " + str(a) + " * " + str(b) + " = " + str(a * b) + "\n")
    elif jenisOperator == 4:
        print("Output: " + str(a) + " / " + str(b) + " = " + str(a / b) + "\n")

# main -----
aktif = 1
while aktif == 1:
    menu()
    jenisOperasi = dptPilihanPengguna()
    if jenisOperasi == 5:
        aktif = 0
    else:
        [nom1, nom2] = dptDuaNombor()
        kiraCetak(jenisOperasi, nom1, nom2)
print("Terima kasih kerana menggunakan saya.")

# -----
Ln: 47 Col:0

```

Rajah 4.59 Kod atur cara Contoh 4.13

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help

== RESTART: C:\Documents\Tutorial Python\Kalkulator
Bermenu_2.py ==
Kalkulator Bermenu
1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat
Pilihan anda [1 hingga 5]: 1
Masukkan nombor pertama : 4
Masukkan nombor kedua : 7
Output: 4 + 7 = 11

Kalkulator Bermenu
1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat
Pilihan anda [1 hingga 5]: 2
Masukkan nombor pertama : 6
Masukkan nombor kedua : 2
Output: 6 - 2 = 4

Kalkulator Bermenu
1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat
Pilihan anda [1 hingga 5]: 3
Masukkan nombor pertama : 3
Masukkan nombor kedua : 9
Output: 3 * 9 = 27

Kalkulator Bermenu
1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat
Pilihan anda [1 hingga 5]: 4
Masukkan nombor pertama : 3
Masukkan nombor kedua : 5
Output: 3 / 5 = 0.6

Kalkulator Bermenu
1. Tambah
2. Tolak
3. Darab
4. Bahagi
5. Tamat
Pilihan anda [1 hingga 5]: 5
Terima kasih kerana menggunakan saya.
>>>
Ln: 59 Col:4

```

Rajah 4.60 Output kod atur cara Contoh 4.13

Imbas QR code ini untuk menonton video carta alir bagi atur cara Kalkulator Bermenu.



[goo.gl/otCmBc](https://goo.gl/otCmBc)



### Uji Minda 4.20

1. Apakah teknik pemikiran komputasional yang digunakan untuk mengenal pasti corak yang berulang di dalam pseudokod?
2. Apakah teknik pemikiran komputasional yang digunakan untuk mengenal pasti persamaan antara empat pernyataan dalam subpseudokod **KiraCetak**?



## Aktiviti 4.11



Aktiviti Kumpulan

### Menambahkan Operasi Kuasa dan Punca Kuasa untuk Kalkulator Bermenu

1. Perhatikan Contoh 4.13.
2. Anda perlu menambah operasi kuasa dan punca kuasa  $n$  bagi suatu nombor bulat di mana  $n$  ialah suatu integer positif.
3. Berdasarkan pseudokod di halaman 184, tambahkan operasi kuasa dan punca kuasa. Tuliskan pseudokod baharu anda di atas kertas.
4. Dalam kumpulan, kongsi hasil pseudokod masing-masing dan bincangkan bersama-sama untuk menghasilkan satu pseudokod yang terbaik.
5. Muat turun fail [Kalkulator Bermenu\\_2.py](https://www.goo.gl/onbYbT) dari [goo.gl/onbYbT](https://www.goo.gl/onbYbT). Kemudian, lancarkan perisian aplikasi Python untuk menambahkan kod yang diperlukan bagi menambahkan operasi kuasa dan punca kuasa. Simpan fail baharu ini dengan nama [Kalkulator Bermenu Baharu.py](#).



## Praktis Amali 4.7

### Membangunkan Satu Atur Cara yang Mengandungi *Function* dan *Procedure*



```

*****
Menu Mengira Isi padu
*****
1. Kuboid
2. Silinder
3. Kon
4. Sfera
*****
Masukkan pilihan anda: [1 - 4] : __

```

#### Imbas Kembali

Isi padu kuboid = panjang  $\times$  lebar  $\times$  tinggi  
 Isi padu silinder = luas tapak  $\times$  tinggi;  
 luas tapak =  $\pi \times$  jejari  $\times$  jejari  
 Isi padu kon =  $1/3 \times$  luas tapak  $\times$  tinggi;  
 luas tapak =  $\pi \times$  jejari  $\times$  jejari  
 Isi padu sfera =  $4/3 \times \pi \times$  jejari  $\times$  jejari  $\times$  jejari;  
 $\pi$  ialah 22/7 atau 3.142

Sekiranya pilihan ialah 2, maka paparan berikut akan muncul:

Masukkan jejari : 3  
 Masukkan tinggi : 21  
 Isi padu ialah 593.838

Sekiranya pilihan ialah 4, maka paparan berikut akan muncul:

Masukkan jejari : 3  
 Isi padu ialah 113.112

Bangunkan satu atur cara yang mengira isi padu kuboid, silinder, kon dan sfera dengan menggunakan langkah-langkah di bawah:

1. Tuliskan satu *procedure* untuk mencetak Menu Mengira Isi padu.
2. Bagi setiap bentuk geometri, tuliskan satu *function* khusus untuk mengira isi padunya.
3. Tuliskan satu atur cara utama yang menggabungkan *procedure* dan semua *function* yang telah dibina untuk membolehkan pengguna mengira isi padu bentuk geometri pilihannya.
4. Cadangkan cara bagaimana anda memurnikan atur cara yang dibina dalam langkah 3 agar atur cara itu boleh digunakan berkali-kali sehingga pengguna memilih untuk menamatkannya.

4.2.5

## Menguji Atur Cara dan Membaiki Ralat

Menguji dan membaiki ralat ialah satu fasa penting dalam pembangunan atur cara. Atur cara yang baik harus bebas ralat. Ralat pengaturcaraan dinamakan pepijat (*bugs*) dan proses menjejaki ralat ini disebut sebagai nyahpepijat (*debugging*). Ralat dalam pengaturcaraan boleh dibahagikan kepada tiga kategori, iaitu ralat sintaks (*syntax error*), ralat masa larian (*runtime error*) dan ralat logik (*logic error*).

### Standard Pembelajaran

Murid boleh:

**4.2.5** Menguji atur cara dan membaiki ralat pada atur cara yang dihasilkan.

### Uji Minda 4.21

Nyatakan kaedah-kaedah yang telah anda gunakan semasa di Tingkatan 1 dan Tingkatan 2 untuk menguji atur cara.



## Ralat sintaks

Setiap bahasa pengaturcaraan mempunyai set tatabahasa, hukum dan peraturan yang mengawal cara penggunaan bahasa tersebut. Kesemua ini dikenali sebagai sintaks kepada bahasa pengaturcaraan. Sintaks yang difahami oleh penterjemah C dan Python adalah berlainan.

Jadual 4.12 menunjukkan format penulisan sintaks bagi struktur pilihan kedua-dua bahasa C dan Python yang berlainan. Struktur pilihan bahasa Python mempunyai kata kunci *elif* yang hanya boleh difahami oleh penterjemah Python dan setiap kata kuncinya diakhiri dengan tanda ':'. Selain itu, bahasa Python tidak menggunakan simbol { } walaupun blok arahan mengandungi lebih daripada satu pernyataan dan setiap arahan tidak diakhiri dengan tanda ';'. Hanya setiap baris yang bermula dengan kata kunci diakhiri dengan tanda ': '.

**Jadual 4.12** Sintaks bagi keratan atur cara dalam bahasa C dan Python

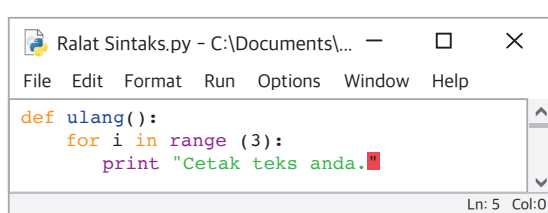
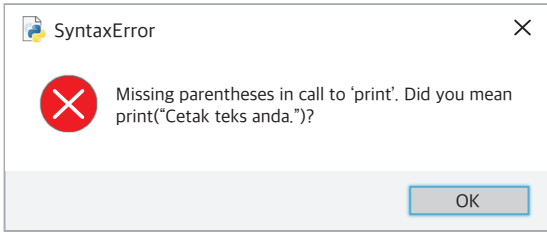
Bahasa C	Bahasa Python
<pre>if (a== b) {     pernyataan1;     pernyataan2; } else if (a &gt; b)     pernyataan3; else     pernyataan4;</pre>	<pre>if (a== b) :     pernyataan1     pernyataan2 elif (a &gt; b) :     pernyataan3 else :     pernyataan4</pre>

Sekiranya satu baris kod arahan yang ditulis itu tidak menepati laras bahasa pengaturcaraan yang digunakan, maka baris kod arahan itu bukan sahaja tidak dapat difahami oleh penterjemah bahasa tersebut bahkan tidak dapat dilaksanakan. Ralat ini dikenali sebagai **ralat sintaks**. Ralat ini sering berlaku kerana kecuaiian pengatur cara semasa menaip kod arahan. Jujukan pelaksanaan (*execution*) arahan dalam *function* atau *procedure* akan terhenti pada baris kod arahan yang mempunyai ralat dan mesej ralat akan dipaparkan.

**Contoh 4.14** Pengesanan ralat sintaks dalam *procedure* oleh penterjemah Python.

Jadual 4.13 menunjukkan kod *procedure* ulang() yang mempunyai ralat sintaks. Pelaksanaan arahan dalam *procedure* ini akan terhenti pada penghujung baris yang ditandakan dengan **█** oleh penterjemah Python. Tetingkap *SyntaxError* akan dipaparkan dengan penjelasan ralat yang berlaku dan cadangan membaikinya.

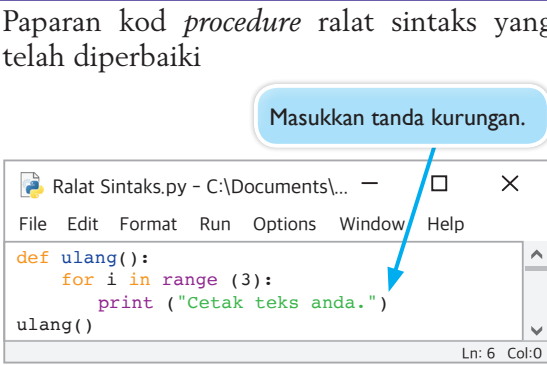
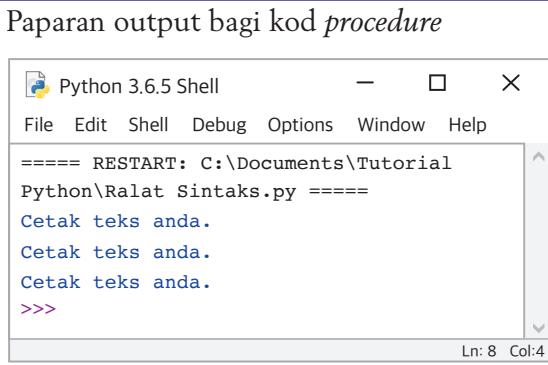
**Jadual 4.13**

<p>Paparan kod <i>procedure</i> yang mempunyai ralat sintaks</p>  <pre>def ulang():     for i in range(3):         print "Cetak teks anda.█"</pre> <p>Ln: 5 Col: 0</p>	<p>Paparan mesej ralat sintaks</p>  <p>SyntaxError</p> <p>Missing parentheses in call to 'print'. Did you mean print("Cetak teks anda.")?</p> <p>OK</p>
---	---



Selepas ralat sintaks ini dibaiki, pelaksanaan arahan dalam *procedure* ini dapat disempurnakan oleh penterjemah Python. Kod arahan yang telah dibaiki dan outputnya seperti yang ditunjukkan dalam Jadual 4.14.

Jadual 4.14

Paparan kod <i>procedure</i> ralat sintaks yang telah diperbaiki	Paparan output bagi kod <i>procedure</i>
 <pre>def ulang():     for i in range(3):         print("Cetak teks anda.") ulang()</pre>	 <pre>==== RESTART: C:\Documents\Tutorial Python\Ralat Sintaks.py ==== Cetak teks anda. Cetak teks anda. Cetak teks anda. &gt;&gt;&gt;</pre>

### Contoh 4.15

Cara menguji suatu *function*.

Ramli dan Kien Tien telah ditugaskan untuk menguji satu *function*. Jadual 4.15 menunjukkan cara yang digunakan oleh mereka. Terdapat beberapa cara untuk menguji *function*. Cara yang diguna pakai ini bergantung kepada tahap penguasaan bahasa pengaturcaraan yang dipelajari dan kreativiti pengatur cara. Dalam buku ini, dua cara yang asas akan dibincangkan.

Jadual 4.15

Ramli	Kien Tien	Output
<pre>def kuasdua(x):     return x**2 hasil=kuasdua(4) print(hasil)</pre>	<pre>def kuasdua(x):     return x**2 print(kuasdua(4))</pre>	16

Ramli menggunakan cara yang mengumpulkan nilai yang dipulangkan oleh *function* *kuasdua(x)* kepada satu pemboleh ubah. Selepas itu, Ramli menggunakan *built-in function* *print()* untuk mencetak nilai pemboleh ubah tersebut.

Kien Tien pula menggunakan *built-in function* *print()* untuk terus mencetak nilai yang dipulangkan oleh *function* *kuasdua(x)*.

Kedua-dua kaedah yang digunakan memberikan output yang betul.

### Uji Minda 4.22

Berdasarkan contoh 4.15, jawab soalan-soalan berikut:

1. Dapatkah anda mengenal pasti cara penulisan murid yang manakah lebih bagus?
2. Apakah kekurangan bagi cara yang bukan pilihan anda?
3. Bincangkan kelebihan dan kekurangan cara yang digunakan oleh Ramli dan Kien Tien untuk menguji *function*.

## Contoh 4.16 Menyemak kod arahan dan membaiki ralat.

Atur cara berikut ditulis oleh seorang murid untuk mengira isi padu sebuah piramid bertapak segi empat sama. Pelaksanaan aturan ini tidak menghasilkan output yang sepatutnya. Jadual 4.16 menunjukkan kod aturan dan paparan output yang diperolehi.

Jadual 4.16 Kod aturan dan paparan output yang diperolehi

Kod aturan
<pre># Mengira isi padu piramid bertapak segi empat sama def kira_Isipadu_piramid(a,b):     isipadu_piramid = (1/3) * (sisi * sisi) * tinggi     return(isipadu_piramid)  # Atur cara utama print("Kira Isi Padu Piramid") sisi = int(input("Masukkan ukuran sisi tapak piramid:")) tinggi = int(input("Masukkan tinggi piramid:"))  # Pemanggilan function dan pemulangan nilai print("Isi Padu Piramid =",kira_Isipadu_piramid(a,b))</pre>
Paparan output dan mesej ralat
<pre>Kira Isi Padu Piramid Masukkan ukuran sisi tapak piramid: 4 Masukkan tinggi piramid: 3 Traceback (most recent call last):   File "C:/Documents/Tutorial Python/kira_Isipadu_piramid.py", line 12, in &lt;module&gt;     print("Isi Padu Piramid =",kira_Isipadu_piramid(a,b)) NameError: name 'a' is not defined &gt;&gt;&gt;</pre>

Semakan ke atas kod aturan harus dibuat secara tertib.

## Tahukah Anda

Apabila *function* dan *procedure* digunakan dalam pembangunan sesuatu aturan, pendekatan yang diguna pakai dalam proses nyahpepijatan (*debugging*) adalah berlainan sedikit. Teknik *step through* tidak digunakan sepenuhnya kerana *function* dan *procedure* adalah subaturan yang bersifat modular yang tidak bersandaran kepada aturan utama. Oleh demikian, setiap subaturan ini perlu diuji dan dibaiki ralatnya sehingga bebas ralat sebelum subaturan ini boleh diguna pakai dalam aturan utama.

## Langkah 1

Pengujian kod *function* kira\_Isipadu\_piramid()

*Function* kira\_Isipadu\_piramid( ) harus diuji dahulu dengan menggunakan *function* print( ). Panggilan *function* kira\_Isipadu\_piramid( ) boleh dibuat dalam *function* print( ) dengan meletakkan dua argumen (nilai) semasa panggilan *function* dilakukan. Kod *function* diuji dan didapati mengandungi ralat seperti yang ditunjukkan dalam Jadual 4.17.

**Jadual 4.17** Kod *function* yang mempunyai ralat dan paparan output yang diperoleh

Pengujian Kod <i>function</i>	
<pre># Mengira isi padu piramid bertapak segi empat sama def kira_Isipadu_piramid(a,b):     isipadu_piramid = (1/3) * (sisi * sisi) * tinggi     return(isipadu_piramid)  # Atur cara utama print("Kira Isi Padu Piramid") sisi = int(input("Masukkan ukuran sisi tapak piramid:")) tinggi = int(input("Masukkan tinggi piramid:"))  # Pemanggilan function dan pemulangan nilai print("Isi Padu Piramid =",kira_Isipadu_piramid(a,b))</pre>	
<p><b>Uji <i>function</i> kira_Isipadu_piramid( ) dengan <i>function</i> print( )</b></p> <pre>print(kira_Isipadu_piramid(2,3))</pre>	
<p><b>Mesej Ralat</b></p> <pre>Traceback (most recent call last):   File "C:/Documents/Tutorial Python/function kira_Isipadu_piramid.py", line 6, in &lt;module&gt;     print(kira_Isipadu_piramid(2,3))   File "C:/Documents/Tutorial Python/function kira_Isipadu_piramid.py", line 3, in kira_Isipadu_piramid     isipadu_piramid = (1/3) * (sisi * sisi) * tinggi NameError: name 'sisi' is not defined</pre>	

Merujuk kepada mesej ralat di atas, didapati pemboleh ubah bernama sisi tidak ditakrifkan.

### Uji Minda 4.23

1. Berdasarkan penelitian anda, mengapakah hanya dua argumen (nilai) digunakan semasa panggilan *function* kira\_Isipadu\_piramid dibuat?
2. Pada pendapat anda, mengapakah nilai 2 dan 3 telah digunakan sebagai argumen?

## Uji Minda 4.24

Bagaimanakah anda dapat mengesahkan bahawa ketiga-tiga pernyataan dalam atur cara utama ini bebas ralat?

### Langkah 2

### Baiki ralat yang dijumpai

Gantikan pemboleh ubah a dan b dalam parameter *function* masing-masing dengan pemboleh ubah sisi dan tinggi. Uji semula *function* `kira_Isipadu_piramid()`.

Jadual 4.18 Baiki ralat yang dijumpai

Baiki ralat yang dijumpai dalam kod <i>function</i>	
<pre># Mengira isi padu piramid bertapak segi empat sama def kira_Isipadu_piramid(sisi,tinggi):     isipadu_piramid = (1/3) * (sisi * sisi) * tinggi     return(isipadu_piramid)  # Atur cara utama print("Kira Isi Padu Piramid") sisi = int(input("Masukkan ukuran sisi tapak piramid:")) tinggi = int(input("Masukkan tinggi piramid:"))  # Pemanggilan function dan pemulangan nilai print("Isi Padu Piramid =",kira_Isipadu_piramid(a,b))</pre>	<p>Gantikan pemboleh ubah a dan b dalam parameter <i>function</i> masing-masing kepada sisi dan tinggi.</p> <p>Uji semula <i>function</i> <code>kira_Isipadu_piramid()</code> dengan <i>function</i> <code>print()</code> <code>print(kira_Isipadu_piramid(2,3))</code></p> <p><b>Output</b> 4.0 &gt;&gt;&gt;</p>

Hasil pengujian menunjukkan *function* `kira_Isipadu_piramid()` kini telah bebas ralat.

### Langkah 3

### Uji atur cara utama

Tiga pernyataan dalam atur cara utama diuji untuk mencari ralat lain.

```
# Atur cara utama
print("Kira Isi Padu Piramid")
sisi = int(input("Masukkan ukuran sisi tapak piramid: "))
tinggi = int(input("Masukkan tinggi piramid: "))
```

Tiada ralat dijumpai.

### Langkah 4

### Uji pernyataan yang memanggil *function*

Semasa panggilan *function* `kira_Isipadu_piramid()`, dua argumen diperlukan. Dua argumen ini ialah nilai yang disimpan dalam pemboleh ubah sisi dan tinggi. Maka a dan b masing-masing harus digantikan dengan pemboleh ubah sisi dan tinggi.

```
print("Isi Padu Piramid =",kira_Isipadu_piramid(sisi,tinggi))
```

Masukkan argumen (nilai sebenar) yang betul semasa memanggil *function*.

## Ralat masa larian

Ralat masa larian akan timbul atau kelihatan apabila *user-defined function*, *procedure* atau atur cara sedang dilaksanakan (*executed*). Ralat jenis ini akan menyebabkan pelaksanaan kod arahan terhenti secara tiba-tiba dan mesej ralat akan dipaparkan.



Kesilapan umum ialah punca utama kejadian ralat masa larian.

### Contoh 4.17 Pengesanan ralat masa larian dalam *function*.

Contoh ini menunjukkan cara pengesanan ralat masa larian dalam *user-defined function* *kira\_purata*. Semak kod atur cara yang berikut. Penulisan sintaks adalah betul. Akan tetapi, semasa pelaksanaan kod arahan tersebut atur cara terhenti secara tiba-tiba apabila input pertama yang dimasukkan bukan satu angka.

Jadual 4.19 Kod atur cara dan paparan output yang diperolehi

Kod atur cara	
<pre># Function mengira purata def kira_purata(x,y):     purata = jumlah / bilangan     return(purata)  # Atur cara utama senaraiNo = [] cont = True while cont:     try:         nom = float(input("Masukkan nombor [X utk berhenti]: "))         senaraiNo.append(nom)     except ValueError:         cont = False  bilangan = len(senaraiNo) jumlah = sum(senaraiNo) print("Purata = ",str(kira_purata(jumlah,bilangan))) print("Tamat.")</pre>	<p>Gelung <i>while</i> ini akan meminta pengguna memasukkan nombor secara berulang kali. Nombor yang diterima oleh atur cara akan dimasukkan ke dalam <b>senaraiNo</b>. Sekiranya nilai bukan angka dimasukkan, gelung akan berhenti.</p>
Mesej ralat terpapar secara tiba-tiba semasa pelaksanaan	
<pre>RESTART: C:\Documents\Tutorial Python\Ralat Masa Larian.py Masukkan nombor [X utk berhenti]: 0 Masukkan nombor [X utk berhenti]: X Purata = 0.0 Tamat. &gt;&gt;&gt; RESTART: C:\Documents\Tutorial Python\Ralat Masa Larian.py Masukkan nombor [X utk berhenti]: X Traceback (most recent call last):   File "C:\Documents\Tutorial Python\Ralat Masa Larian.py", line 18, in &lt;module&gt;     print("Purata = ",str(kira_purata(jumlah,bilangan)))   File "C:\Documents\Tutorial Python\Ralat Masa Larian.py", line 3, in kira_purata     purata = jumlah / bilangan ZeroDivisionError: division by zero &gt;&gt;&gt;</pre>	<p>Dalam mesej ralat, baris yang menyebabkan ralat dan punca kejadian ralat akan ditunjukkan.</p>

Punca atur cara terhenti kerana atur cara diminta melakukan pembahagian dengan sifar. Kejadian ini berlaku pada baris kod dalam *function* `kira_purata(x,y)`.

```
purata = jumlah / bilangan
```

Ini disebabkan nilai bagi pemboleh ubah bilangan adalah sifar. Nilai ini datang dari baris kod dalam atur cara utama yang berikut.

```
bilangan = len(senaraiNo)
```

Jadual 4.20 menunjukkan kod atur cara yang telah dibaiki dan paparan output yang diperolehi.

**Jadual 4.20** Kod atur cara dan paparan output yang diperolehi

Kod atur cara yang dibaiki ralat	
<pre># Function mengira purata def kira_purata(x,y):     purata = jumlah / bilangan     return(purata)  # Atur cara utama senaraiNo = [] cont = True while cont:     try:         nom = float(input("Masukkan nombor [X utk berhenti]: "))         senaraiNo.append(nom)     except ValueError:         cont = False  bilangan = len(senaraiNo)  if bilangan &gt; 0:     jumlah = sum(senaraiNo)     print("Purata = ",str(kira_purata(jumlah,bilangan)))  print("Tamat.")</pre>	<p>Gunakan struktur kawalan pilihan untuk membenarkan pengiraan diteruskan hanya apabila bilangan lebih besar daripada sifar.</p>
Paparan output	
<pre>RESTART: C:\Documents\Tutorial Python\Ralat Masa Larian 2.py Masukkan nombor [X utk berhenti]: 0 Masukkan nombor [X utk berhenti]: X Purata = 0.0 Tamat. &gt;&gt;&gt; RESTART: C:\Documents\Tutorial Python\Ralat Masa Larian 2.py Masukkan nombor [X utk berhenti]: X Tamat. &gt;&gt;&gt;</pre>	

## Ralat logik

Ralat logik tidak akan menyebabkan pelaksanaan sesuatu atur cara terhenti secara tiba-tiba dengan atau tanpa memaparkan mesej ralat. Ralat logik akan menyebabkan atur cara melakukan sesuatu tindakan yang tidak dijangka ataupun memberikan output yang tidak sepatutnya. Ralat logik sukar dikesan.

**Contoh 4.18** Pengesanan ralat logik dalam *user-defined function*.

Contoh ini membincangkan cara mengesan ralat logik dalam atur cara yang menggunakan *user-defined function* tambah() dan semak() untuk mencari hasil tambah dua nombor bulat yang dijanakan secara rawak oleh komputer serta menyemak jawapan pengguna.

Atur cara di bawah telah dihasilkan. Apabila dilaksanakan, pelaksanaannya berjaya.

**Jadual 4.21** Kod atur cara dan paparan output yang diperolehi

Kod atur cara yang dibaiki ralat	
<pre># Atur cara ini menggunakan built-in function random import random  # User-defined function tambah() def tambah(a,b):     return(a+b)</pre>	<p><b>Langkah 1 Uji function tambah()</b></p> <pre>def tambah(a, b):     return(a + b)  print(tambah(2,3))</pre> <p>Output: 5 Keputusan: Function ini tiada ralat.</p>
<pre># User-defined function semak() def semak(jawapan, jumlah):     if jawapan == jumlah:         print("Betul!")     else:         print("Salah! Jawapannya ialah", jumlah)</pre>	<p><b>Langkah 2 Uji function semak()</b></p> <pre>def semak(jawapan, jumlah):     if jawapan == jumlah:         print("Betul!")     else:         print("Salah!         Jawapannya ialah",         jumlah)  print(semak(2,3))</pre> <p>Output: Betul! Keputusan: Function ini tiada ralat.</p>
<pre># Atur cara utama nombor1 = random.randint(1, 10) nombor2 = random.randint(1, 10) print("Selesaikan", nombor1, "+", nombor2, ".") jawapan = input()</pre>	<p><b>Langkah 3 Semak jenis data</b></p> <pre>jawapan = input("Jawapan anda:") print(type(jawapan))</pre> <p>Output: Jawapan anda: 5 &lt;class 'str'&gt; Keputusan: Input mempunyai jenis data str</p>
<pre># Pemanggilan function tambah() dan semak() jumlah = tambah(nombor1, nombor2) semak(jawapan, jumlah)</pre>	

Ralat hanya kelihatan dalam situasi tertentu seperti Jadual 4.22.

**Jadual 4.22** Atur cara menghasilkan output yang tidak tepat

Katakan pelaksanaan atur cara kali pertama menghasilkan soalan: Selesaikan 1 + 8?  Sekiranya jawapan yang dimasukkan ialah 10, maka respons atur cara adalah seperti yang dijangkakan.	<b>Output</b>
	Selesaikan 1 + 8. 10 Salah! Jawapannya ialah 9
Katakan pelaksanaan atur cara kali kedua menghasilkan soalan: Selesaikan 2 + 3?  Sekiranya jawapan yang dimasukkan ialah 5, maka respons atur cara adalah tidak seperti yang dijangkakan.	<b>Output</b>
	Selesaikan 2 + 3. 5 Salah! Jawapannya ialah 5

Didapati respons yang diberikan oleh atur cara adalah tidak seperti yang dijangkakan. Oleh itu, setiap *function* harus diuji. Berdasarkan Jadual 4.21, Langkah 1 dan 2 telah mengesahkan kedua-dua *function* tambah() dan semak() adalah bebas ralat. Maka, tinggal kod dalam atur cara utama.

Pernyataan-pernyataan ini tidak melibatkan manipulasi data, maka tidak mungkin menyebabkan ralat.

```
nombor1 = random.randint(1, 10)
nombor2 = random.randint(1, 10)
print("Selesaikan", nombor1, "+", nombor2)
jawapan = input()
```

Berdasarkan keputusan Langkah 3, jenis data bagi pemboleh ubah jawapan ialah jenis str. Oleh sebab argumen (nilai) ini perlu dibandingkan dengan nilai jumlah yang berjenis int melalui operator ==, maka jenis data bagi pemboleh ubah jawapan harus diubah kepada int supaya pernyataan **if jawapan == jumlah:** dalam *function* semak() dapat dilakukan dengan betul, iaitu data integer berbanding dengan data integer.

Berikut cara memperbaiki ralat yang ditemui dalam Langkah 3.

```
jawapan = int(input())
```

Nilai yang dimasukkan oleh pengguna ditukarkan kepada jenis data integer.

### Uji Minda 4.25

Pada pendapat anda, mengapakah hanya set data ujian {2, 3} dipilih untuk digunakan dalam pengujian ini?





## Aktiviti 4.12

### Aktiviti Pasangan

Menguji Atur Cara dan Membaiki Ralat yang Dijumpai



Kaedah  
Think-Pair-Share

Jadual 1 Kod atur cara 1 dan 2

Kod atur cara 1	Kod atur cara 2
<pre>def bilangturun(simbol):     bil_baris = 1     while bil_baris &lt; 10:         print(simbol * bil_baris)         bil_baris = bil_baris + 1  #Atur cara utama aksara=input("Masukkan simbol: ") bilangturun("aksara")</pre>	<pre>def banding(x,y):     if x &gt; y:         besar = x     else:         kecil = x     return (besar,kecil)  #Atur cara utama [besar, kecil] = banding(6,9) print("Nombor ",besar,"lebih besar daripada ",kecil)</pre>

Jadual 2 Contoh output atur cara 1 dan 2

Contoh output atur cara 1	Contoh output atur cara 2
<pre>Masukkan simbol: &amp; &amp; &amp;&amp; &amp;&amp;&amp; &amp;&amp;&amp;&amp; &amp;&amp;&amp;&amp;&amp; &amp;&amp;&amp;&amp;&amp;&amp; &amp;&amp;&amp;&amp;&amp;&amp;&amp; &amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp; &amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;</pre>	<pre>Nombor 9 lebih besar daripada 6</pre>

- Perhatikan Jadual 1 dan Jadual 2, kemudian jawab soalan-soalan berikut dalam masa yang diperuntukkan.
  - Uji kod-kod atur cara dalam Jadual 1 dengan kaedah-kaedah yang telah anda pelajari. Tuliskan cara kerja anda pada sehelai kertas.
  - Kenal pasti jenis ralat yang dikesan.
  - Baiki ralat yang dikesan supaya output yang diperolehi adalah serupa dengan contoh output yang ditunjukkan.
- Selepas tempoh masa yang diperuntukkan tamat, kongsi idea bersama-sama rakan anda. Lakukan perbincangan untuk menambah baik hasil dapatan anda.
- Sediakan satu persembahan dan bentangkan hasil dapatan kumpulan anda di dalam kelas.

## Standard Pembelajaran

Murid boleh:

**4.2.6** Menghasilkan atur cara yang melibatkan gabungan struktur kod arahan bagi menyelesaikan masalah dalam kehidupan seharian

4.2.6

## Menghasilkan Atur Cara yang Melibatkan Gabungan Struktur Kod Arahan bagi Menyelesaikan Masalah

Atur cara yang mempunyai susunan struktur kod arahan yang sistematik, logik dan bermodular dapat memudahkan pengatur cara dan penyenggara atur cara mengenal pasti tugasan setiap struktur atau modul. Atur cara yang bermodular dapat dihasilkan dengan penggunaan *function* dan *procedure*.

### Contoh 4.19

Penghasilan atur cara yang menggunakan *function* dan *procedure* bagi mengira peratus keuntungan dan kerugian.

Tarmini ingin membina satu atur cara bagi mengira peratus keuntungan dan kerugian barangan yang dijual oleh abangnya. Dalam mata pelajaran Matematik, Tarmini telah mempelajari rumus seperti dalam Jadual 4.23.

Jadual 4.23

Untung = Harga Jualan – Harga Kos	Rugi = Harga Kos – Harga Jualan
$\% \text{ keuntungan} = \frac{\text{Untung}}{\text{Harga Kos}} \times 100$	$\% \text{ kerugian} = \frac{\text{Rugi}}{\text{Harga Kos}} \times 100$

Tarmini akan menggunakan rumus di atas dan mengaplikasikan fasa-fasa pembangunan atur cara yang telah dipelajari dalam Bab 1.

### Fasa Menganalisis Masalah

Gunakan teknik leraian untuk memecahkan masalah mengira peratus keuntungan dan kerugian kepada beberapa bahagian kecil seperti mendapatkan input bagi harga kos dan harga jualan, proses pengiraan bagi peratus keuntungan dan kerugian serta cara memaparkan output yang sepadan. Bahagian-bahagian kecil ini lebih mudah diuruskan.



### Fasa Reka Bentuk Atur Cara

Corak dan prinsip yang menghasilkan corak ini harus dikenal pasti. Analisis rumus matematik bagi pengiraan peratus keuntungan dan peratus kerugian diperlihatkan. Kedua-dua pengiraan ini menggunakan **item input** yang sama, iaitu **harga kos** dan **harga jualan**. Perbezaan antara harga jualan dan harga kos akan menentukan peratus keuntungan, peratus kerugian atau tiada kerugian.



Jadual 4.24

Untung = Harga Jualan – Harga Kos	Rugi = Harga Kos – Harga Jualan
$\% \text{ keuntungan} = \frac{\text{Untung}}{\text{Harga Kos}} \times 100$	$\% \text{ kerugian} = \frac{\text{Rugi}}{\text{Harga Kos}} \times 100$

Berdasarkan maklumat yang diperolehi, kini Tarmini perlu menyediakan algoritma bagi atur cara mengira peratus keuntungan dan kerugian. Kemudian, algoritma ini diterjemahkan ke dalam bentuk pseudokod atau carta alir.



### Algoritma

1. Dapatkan harga kos (**h1**) dan harga jualan (**h2**) daripada pengguna.
2. Bandingkan nilai harga kos dengan harga jualan.
3. Jika harga jualan sama dengan harga kos, paparkan mesej "Tiada keuntungan".
4. Jika harga jualan lebih tinggi daripada harga kos, kira dan paparkan peratus keuntungan.
5. Jika harga kos lebih tinggi daripada harga jualan, kira dan paparkan peratus kerugian.

### Pseudokod

```

1  Mula
2  Papar "Masukkan harga kos RM: "
3  Setkan h1 = harga kos yang dimasukkan
4  Papar "Masukkan harga jualan RM: "
5  Setkan h2 = harga jualan yang dimasukkan
6  Jika h1 == h2
   6.1 Papar "Tiada keuntungan."
7  Jika h2 > h1
   7.1 Kira peratus keuntungan =  $\frac{h2 - h1}{h1} \times 100$ 
   7.2 Papar peratus keuntungan
8  Jika tidak
   8.1 Kira peratus kerugian =  $\frac{h1 - h2}{h1} \times 100$ 
   8.2 Papar peratus kerugian
9  Tamat jika
10 Tamat

```

Jika algoritma yang dihasilkan dikaji semula, anda akan mendapati bahawa ada proses yang berulang kerana mempunyai corak yang sama. Sebagai contoh, proses memasukkan data mempunyai corak yang sama, iaitu meminta pengguna memasukkan data sebanyak dua kali:



**Papar** "Masukkan harga kos RM: "  
**h1** = nilai harga kos  
**Papar** "Masukkan harga jualan RM: "  
**h2** = nilai harga jualan

Carta alir bagi mengira peratus keuntungan dan kerugian.



[goo.gl/9QCKDy](https://goo.gl/9QCKDy)



Video carta alir bagi mengira peratus keuntungan dan kerugian dengan menggunakan *function* dan *procedure*.



[goo.gl/J4YoSM](https://goo.gl/J4YoSM)



Begitu juga dengan proses pengiraan, di mana pengiraan peratus keuntungan dan peratus kerugian mempunyai formula yang serupa dan kedua-dua formula tersebut menggunakan data yang sama.

$$\begin{aligned} \text{Kira peratus keuntungan} &= \frac{h2 - h1}{h1} \times 100 \\ \text{Papar peratus keuntungan} & \\ \text{Kira peratus kerugian} &= \frac{h1 - h2}{h1} \times 100 \\ \text{Papar peratus kerugian} & \end{aligned}$$

Bagi menjadikan penyelesaian masalah lebih mudah dan bermodular, proses-proses yang berulang ini, iaitu proses memasukkan data dan proses pengiraan serta pemaparan peratus keuntungan dan kerugian, perlu dijadikan *function* ataupun *procedure*. Berikut menunjukkan pseudokod setelah diubah suai.

### Pseudokod *function* dan *procedure*

*Function* **inputPegguna** untuk pengguna menginput harga kos dan harga jualan:



#### **Function inputPegguna(mesejInput)**

- 1 Mula
- 2 **Papar** mesejInput
- 3 **Setkan** harga = input pengguna
- 4 **Pulangkan** harga
- 5 **Tamat**

*Procedure* **kiraPeratus** untuk mengira dan memaparkan peratus keuntungan atau peratus kerugian:

#### **Procedure kiraPeratus(h1, h2)**

- 1 Mula
- 2 **Kira** peratus =  $\frac{h2 - h1}{h1} \times 100$
- 3 **Jika** peratus > 0
  - 3.1 **Papar** "Peratus keuntungan =" dan peratus
- 4 **Jika tidak**
  - 4.1 **Papar** "Peratus kerugian =" dan peratus (nilai mutlak)
- 5 **Tamat jika**
- 6 **Tamat**

**Pseudokod atur cara utama**

```

1 Mula
2 h1 = panggil inputPegguna("Masukkan harga kos RM: ")
3 h2 = panggil inputPegguna ("Masukkan harga jualan RM:")
4 Jika h1 == h2
    4.1 Papar "Tiada keuntungan."
5 Jika tidak
    5.2 Panggil kiraPeratus(h1, h2)
6 Tamat jika
7 Tamat

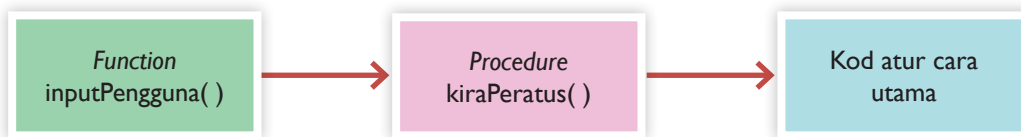
```

**Fasa Pengkodan**

Semasa pengkodan, lazimnya *procedure* atau *function* dihasilkan dahulu dan diuji supaya bebas ralat sebelum proses penulisan kod atur cara utama dimulakan. Dalam fasa ini, algoritma diterjemahkan kepada kod arahan bahasa pengaturcaraan yang terpilih. Teknik-teknik pemikiran komputasional masih boleh diaplikasikan mengikut kesesuaian dengan sintaks bahasa pengaturcaraan yang digunakan.



Dalam Contoh 4.19, kod arahan *function* `inputPegguna()` dihasilkan dan diuji supaya bebas ralat sebelum *procedure* `kiraPeratus()` dihasilkan kerana output daripada *function* `inputPegguna()` akan menjadi input kepada *procedure* `kiraPeratus()`. Apabila kod *procedure* `kiraPeratus()` diuji dan didapati bebas ralat, baharulah penulisan kod atur cara utama dimulakan. Proses ini ditunjukkan dalam peta alir di bawah:



**Rajah 4.61** Proses penghasilan atur cara Contoh 4.19

**Langkah 1****Mengekod *function* `inputPegguna`**

Anda harus menentukan jenis data input yang diterima oleh parameter, proses dan output bagi *function* ini sebelum mengekod.

```

inputPegguna.py - C:\Documents\Tutorial Python\inputPegguna.py (3.6.5)
File Edit Format Run Options Window Help
# Function inputPegguna
def inputPegguna(mesejInput):
    print(mesejInput)
    harga = float(input())
    return harga
Ln: 6 Col: 0

```

**Rajah 4.62** Kod *function* `inputPegguna`

Selepas siap mengekod *function* `inputPegguna`, kod arahan harus diuji supaya bebas ralat.

**Jadual 4.25** Pengujian `inputPegguna()`

Pengujian <code>inputPegguna()</code>	Output
<pre>def inputPegguna(mesejInput):     print(mesejInput)     harga = float(input())     return harga  h1 = inputPegguna("Masukkan harga kos RM ") print(h1)</pre>	<p>Masukkan harga kos RM</p> <p>5.45 ←</p> <p>5.45</p> <p>Input pengguna</p>

**Langkah 2**

**Mengekod *procedure* kiraPeratus**

Argumen yang diterima oleh *procedure* ini ialah data jenis *float*. Hasil kiraan peratus mesti sentiasa suatu nilai yang positif. *Procedure* `kiraPeratus` juga harus diuji.

```
kira Peratus.py - C:\Documents\Tutorial Python\kira Peratus.py (3.6.5)
File Edit Format Run Options Window Help

# Procedure kiraPeratus
def kiraPeratus(h1,h2):
    peratus =(h2-h1)/h1*100
    peratus = round(peratus, 2) # dua tempat perpuluhan
    if peratus > 0:
        print("Keuntungan ialah", peratus, "%")
    else:
        print("Kerugian ialah", abs(peratus), "%") # positifkan nilai

Ln: 5 Col:19
```

**Rajah 4.63** Kod *procedure* `kiraPeratus`

**Jadual 4.26** Pengujian `kiraPeratus()`

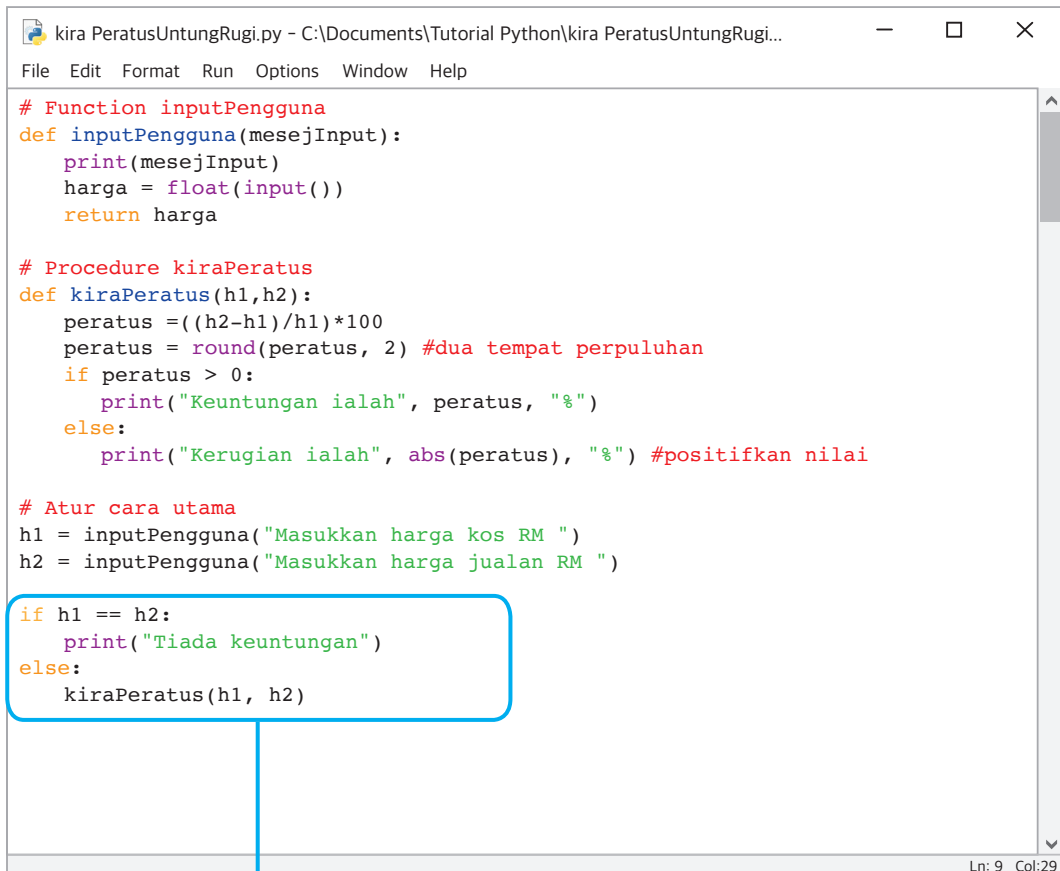
Pengujian <code>kiraPeratus()</code>	Output
<pre># Procedure kiraPeratus def kiraPeratus(h1,h2):     peratus =(h2-h1)/h1*100     peratus = round(peratus, 2)     if peratus &gt; 0:         print("Keuntungan ialah", peratus, "%")     else:         print("Kerugian ialah", abs(peratus), "%")  kiraPeratus(2.50, 3.00) kiraPeratus(1.50, 2.80) kiraPeratus(2.75, 2.55) kiraPeratus(2.50, 2.50)</pre>	<p>Keuntungan ialah 20.0 %</p> <p>Keuntungan ialah 86.67 %</p> <p>Kerugian ialah 7.27 %</p> <p>Kerugian ialah 0.0 %</p> <p>Jika h1 dan h2 adalah sama, output yang diperoleh tidak tepat.</p>

Anda perlu menguji *procedure* kiraPeratus( ) dengan pelbagai situasi yang berlainan supaya pengujian yang dilakukan adalah lengkap. Didapati kod *procedure* kiraPeratus( ) tidak dapat menghasilkan output yang sepatutnya apabila nilai h1 sama dengan nilai h2. Maka, atur cara utama perlu menentukan bahawa argumen (nilai h1 dan h2) yang diberikan kepada *procedure* kiraPeratus( ) tidak mempunyai nilai yang sama.

### Langkah 3

### Mengekod atur cara utama

Dalam pengekodan atur cara utama, *function* dan *procedure* yang bebas ralat itu harus dimasukkan dan panggilan *function* dan *procedure* boleh dibuat apabila diperlukan.



```
# Function inputPegguna
def inputPegguna(mesejInput):
    print(mesejInput)
    harga = float(input())
    return harga

# Procedure kiraPeratus
def kiraPeratus(h1,h2):
    peratus = ((h2-h1)/h1)*100
    peratus = round(peratus, 2) #dua tempat perpuluhan
    if peratus > 0:
        print("Keuntungan ialah", peratus, "%")
    else:
        print("Kerugian ialah", abs(peratus), "%") #positifkan nilai

# Atur cara utama
h1 = inputPegguna("Masukkan harga kos RM ")
h2 = inputPegguna("Masukkan harga jualan RM ")

if h1 == h2:
    print("Tiada keuntungan")
else:
    kiraPeratus(h1, h2)
```

Struktur kawalan ini memastikan nilai-nilai h1 dan h2 yang dihantar kepada *procedure* kiraPeratus( ) adalah berlainan.

Rajah 4.64 Kod atur cara utama Contoh 4.19

Selepas atur cara yang lengkap siap dikodkan, kod perlu diuji semula. Rajah 4.65 menunjukkan tiga set data ujian berserta output sepadannya. Output setiap set data ujian ini adalah seperti yang dijangkakan.

```

Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
=== RESTART: C:\Documents\Tutorial Python\kira PeratusUntungRugi.py ===
Masukkan harga kos RM
3.50
Masukkan harga jualan RM
3.50
Tiada keuntungan
>>>
=== RESTART: C:\Documents\Tutorial Python\kira PeratusUntungRugi.py ===
Masukkan harga kos RM
2.00
Masukkan harga jualan RM
3.00
Keuntungan ialah 50.0 %
>>>
=== RESTART: C:\Documents\Tutorial Python\kira PeratusUntungRugi.py ===
Masukkan harga kos RM
2.00
Masukkan harga jualan RM
1.50
Kerugian ialah 25.0 %
>>>
Ln: 24 Col:4

```

Ujian data (h1 = 3.50; h2 = 3.50)  
Output yang benar

Ujian data (h1 = 2.00; h2 = 3.00)  
Output yang benar

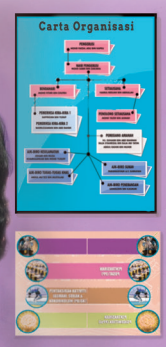
Ujian data (h1 = 2.00; h2 = 1.50)  
Output yang benar

Rajah 4.65



Semasa melakukan pengujian atur cara dan membaiki ralat, apakah masalah yang anda hadapi? Apakah cadangan anda?

Setiap kali menguji, atur cara itu terpaksa dilaksanakan. Ini memakan masa. Untuk membolehkan atur cara itu digunakan berulang kali tanpa perlu dilaksanakan setiap kali, saya mencadangkan agar keseluruhan kod atur cara itu diletakkan dalam gelung *while*. Ini akan menjadikan atur cara itu lebih efisien.







## Praktis Amali 4.8

### Mengesan dan Membaiki Ralat serta Mengenal Pasti *Function* dan *Procedure*



Atur cara berikut telah dibangunkan oleh seorang pelajar Tingkatan 3 untuk menghantar dan menerima mesej berahsia.

```
1 def corak(aksara, bilangan):
2     print(aksara * bilangan)
3 "def menu()"
4     print("1. Penyulitan(Encryption)")
5     print("2. Nyahsulit(Decryption)")
6 def mesej():
7     teks = input("Masukkan nama anda:")
8     return teks
9
10 def songsang(ayat):
11     str = " "
12     for i in ayat:
13         str = i + str
14     print("")
15     corak("#", len(str))
16     print(str)
17     corak("#", len(str))
18
19 corak("***", 32)
20 print("Penyulitan / Penyahsulitan Mesej")
21 print(corak("***", 32))
22 menu()
23 print("")
24 pilihan = (input("Masukkan pilihan anda: "))
25
26 if pilihan == 1:
27     mesej_asal = input("Masukkan mesej anda: ")
28     songsang(mesej_asal)
29 else:
30     mesej_sulit = input("Masukkan mesej sulit anda: ")
31     songsang(mesej_sulit)
32
33
```

Kaji atur cara di atas, kemudian jawab soalan-soalan yang seterusnya.

1. Dalam kumpulan, surih dan kenal pasti semua ralat dalam atur cara.

No. Baris	Komen
4	"def menu( )" sepatutnya def menu( ):

2. Dalam kumpulan, kenal pasti semua *built-in function*, *user-defined function* dan *procedure*.

<i>Built-in function</i>	<i>User-defined function</i>	<i>Procedure</i>

3. Taipkan semula atur cara ini. Baiki semua ralat yang ditemui dalam Langkah 1 dan sebarang ralat yang masih wujud. Uji atur cara anda.
4. Masukkan komen-komen berkesan dalam atur cara anda.
5. Nyahsulitkan mesej di bawah:

ayajreb halet adnA !hainhaT



## Aktiviti 4.13

### Aktiviti Kumpulan

#### Membina Atur Cara



Kaedah  
Round Table

Berikut menunjukkan satu pernyataan masalah. Anda dikehendaki mengaplikasikan fasa-fasa pembinaan atur cara serta penggunaan *function* dan *procedure* untuk menyelesaikan masalah ini.

Pak Cik Sairin seorang pemilik kedai runcit di Kampung Paris, Kinabatangan. Selama ini, urusan pembayaran di kedai runcitnya diurus dan dicatat secara manual. Anda ingin membantu Pak Cik Sairin dengan membina satu atur cara supaya urusan pembayaran di kedai runcitnya lebih mudah. Atur cara yang dibina perlu mempunyai beberapa *function* dan berguna. Berikut menunjukkan paparan output yang anda perlu hasilkan:

```

                Kedai Runcit Pakcik Sairin
Teruskan atau Masukkan T untuk mendapatkan JUMLAH:
Masukkan amaun: 7.7
Teruskan atau Masukkan T untuk mendapatkan JUMLAH:
Masukkan amaun: 8.9
Teruskan atau Masukkan T untuk mendapatkan JUMLAH:
Masukkan amaun: t

Jumlah belian: RM 16.60
Masukkan amaun diterima: RM 50
*****
Terima (RM): 50.00
Jumlah Belian (RM): 16.60
Baki (RM): 33.40

Pulangkan.....
*****
RM20: 1
RM10: 1
RM1: 3
20sen: 2
*****

```

Berdasarkan petikan yang ditunjukkan di atas, ikuti langkah-langkah berikut untuk menyelesaikan masalah.

1. Kaji masalah dan tuliskan penyelesaian terhadap masalah tersebut di atas kertas.
2. Dalam kumpulan, murid-murid perlu mengedarkan catatan tentang penyelesaian masing-masing mengikut arah pusingan jam. Setiap ahli kumpulan perlu mengkaji catatan ahli kumpulan yang lain. Setiap ahli kumpulan boleh menambah atau membetulkan catatan penyelesaian yang mereka baca.
3. Guru memberikan rumusan pembelajaran kepada kelas.
4. Murid-murid dikehendaki menghasilkan atur cara dengan menggunakan bahasa pengaturcaraan Python.

**STEM**

## Uji Kendiri 4.2

1. Nyatakan **tiga** kebaikan menggunakan *function* dan *procedure* dalam penulisan satu atur cara.
2. Salah satu *built-in function* yang terdapat dalam *module Math Python* ialah **pow (x, y)**. *Built-in function* ini memulangkan nilai kuasa  $n$  bagi suatu nombor bulat,  $x^n$ . Berikut menunjukkan contoh kod Python penggunaan *built-in function* ini dan contoh output yang diperoleh:

Kod Python:

```
import math
a = pow(2,8)
print(a)
```

Contoh output:

```
256
>>>
```

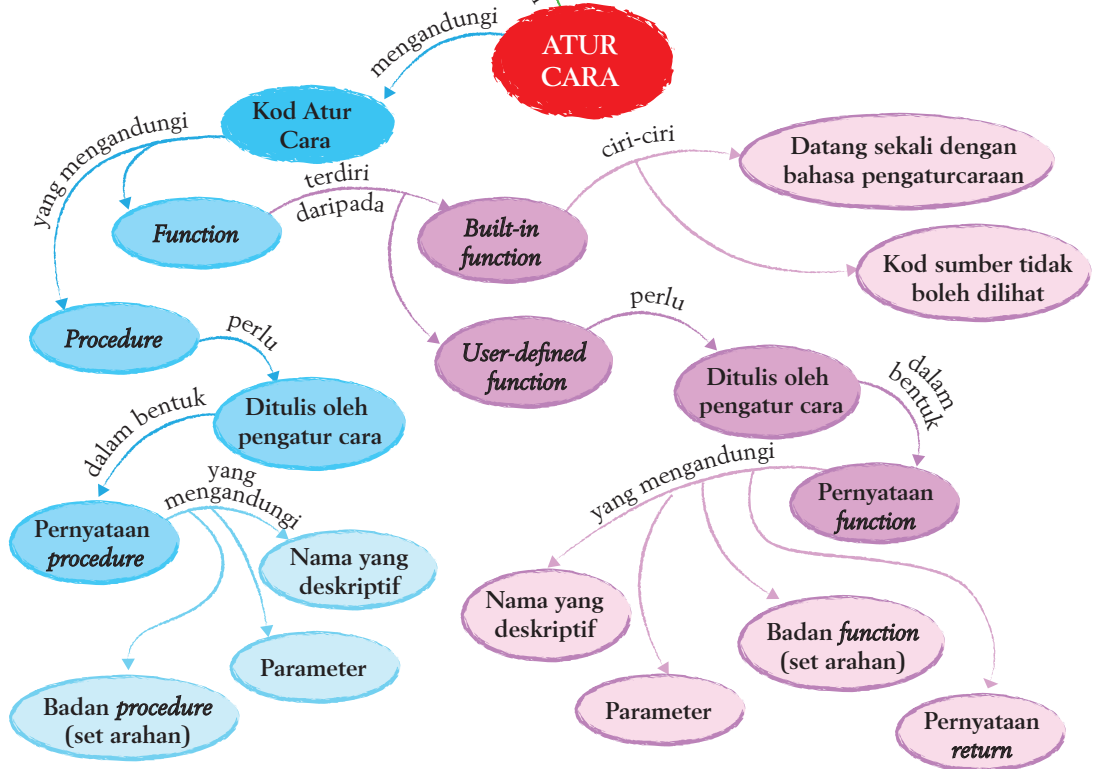
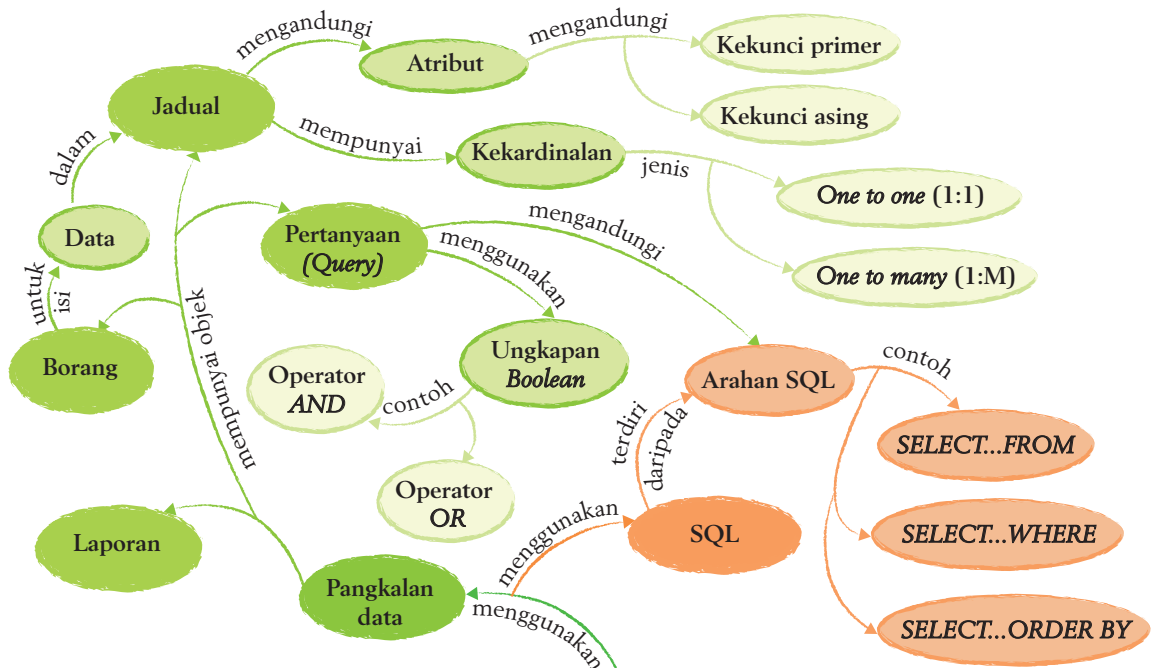
Hasilkan **dua** *user-defined function* yang akan memberikan output yang serupa dengan *built-in function pow (x, y)*.

3. Input( ) adalah suatu *built-in function* yang sering digunakan dalam pembinaan program yang berinteraktif. Bagaimanakah anda menentukan jenis data yang dikembalikan oleh *function* berkenaan?
4. Lihat paparan output berikut dan jawab soalan-soalan di bawah.

```
*
**
***
****
*****
>>>
```

- (a) Tuliskan satu *function* untuk menghasilkan output seperti di atas.
- (b) Tambah baik *function* yang dihasilkan di (a) supaya *function* baharu ini dapat menghasilkan corak yang sama tetapi dengan aksara yang berlainan.
- (c) Tambah baik *function* di (b) supaya *function* baharu ini bukan sahaja dapat menghasilkan corak dengan aksara yang berlainan tetapi dengan bilangan baris yang berlainan.
- (d) Tambah baik *function* di (c) supaya *function* baharu ini dapat melakukan tugas yang sama tetapi secara interaktif, iaitu meminta aksara dan bilangan baris daripada pengguna.

# RUMUSAN GRAFIK 4



# SEMAK SEMULA 4

Pada akhir bab ini, saya dapat...

menerangkan maksud dan penggunaan pangkalan data.	
menerangkan maksud dan penggunaan <i>Structured Query Language</i> (SQL).	
menyenaraikan entiti dan atribut berdasarkan situasi dalam suatu masalah.	
mengenal pasti dan menerangkan kekunci primer dan kekunci asing.	
mengenal pasti dan menjelaskan kekardinalan ( <i>cardinality</i> ) antara entiti dalam hubungan <i>one to one</i> (1:1).	
mengenal pasti dan menjelaskan kekardinalan ( <i>cardinality</i> ) antara entiti dalam hubungan <i>one to many</i> (1:M).	
membina pangkalan data yang terdiri daripada entiti ( <i>table</i> ).	
membina pangkalan data yang terdiri daripada atribut ( <i>field</i> ).	
membina pangkalan data yang terdiri daripada hubungan.	
menghasilkan borang ( <i>form</i> ) dan memasukkan data dalam jadual melalui borang bagi pangkalan data yang telah dibina.	
menggunakan arahan SQL yang melibatkan <i>SELECT</i> ...	
menggunakan arahan SQL yang melibatkan <i>SELECT...WHERE</i>	
menggunakan arahan SQL yang melibatkan <i>SELECT...ORDER BY</i>	
menggunakan arahan SQL melibatkan ungkapan <i>Boolean</i> bagi operator OR.	
menggunakan arahan SQL melibatkan ungkapan <i>Boolean</i> bagi operator AND.	
menjana laporan berdasarkan hasil pertanyaan ( <i>query</i> ).	
menerangkan fungsi struktur <i>function</i> dalam atur cara.	
menerangkan fungsi struktur <i>procedure</i> dalam atur cara.	
memberi contoh penggunaan pernyataan <i>function</i> dalaman ( <i>built-in function</i> ).	
memberi contoh penggunaan pernyataan <i>function</i> yang dihasilkan sendiri ( <i>user-defined function</i> ).	
menulis pernyataan <i>function</i> dan <i>procedure</i> .	
menghasilkan atur cara yang melibatkan <i>function</i> .	
menghasilkan atur cara yang melibatkan <i>procedure</i> .	
menguji atur cara dan memperbaiki ralat pada atur cara yang dihasilkan.	
menghasilkan atur cara yang melibatkan gabungan struktur kod arahan bagi menyelesaikan masalah dalam kehidupan seharian.	

# PENTAKSIRAN 4

## Soalan Objektif

- Pangkalan data memudahkan data atau maklumat dalam semua perkara berikut, **kecuali** \_\_\_\_\_.  
A dicapai  
B diurus  
C dikemas kini  
D dikenal pasti
- Reka bentuk pangkalan data mempunyai komponen-komponen seperti \_\_\_\_\_.  
I entiti  
II atribut  
III kekardinalan  
IV persekitaran  
A I, II dan III  
B I, II dan IV  
C I, III dan IV  
D II, III dan IV
- Kekunci \_\_\_\_\_ merupakan kekunci dengan atribut unik.  
A pertama  
B primer  
C sekunder  
D asing
- Kekunci primer bagi satu entiti boleh dijadikan kekunci \_\_\_\_\_ bagi entiti yang lain untuk mewujudkan hubungan antara dua entiti itu.  
A pertama  
B tertiar  
C sekunder  
D asing
- Lebih rendah kadar kekardinalan, maka semakin \_\_\_\_\_ potensi untuk berlaku pengulangan data.  
A rendah  
B sederhana  
C tinggi  
D menurun
- Apakah jenis kekardinalan bagi situasi seorang murid dapat meminjam banyak buku di perpustakaan?  
A *one to one* (1:1)  
B *one to many* (1:M)  
C *many to one* (M:1)  
D *many to many* (M:M)
- Dalam sesuatu sistem pengurusan pangkalan data hubungan, data boleh dimasukkan dengan menggunakan jadual dan \_\_\_\_\_.  
A borang  
B laporan  
C hasil pertanyaan  
D hubungan
- Dalam sesuatu sistem pengurusan pangkalan data hubungan, pengguna boleh membuat capaian ke atas data melalui \_\_\_\_\_.  
A jadual  
B borang  
C laporan  
D hasil pertanyaan
- Apakah nilai lalai yang dipulangkan oleh sesuatu Python *function* di mana tiada nilai pemulangan dinyatakan secara eksplisit?  
A *Null*                      C *Void*  
B *None*                      D *Public*
- Function* manakah yang tidak memulangkan sebarang nilai?  
A *Function* yang mencetak integer 1 hingga 100.  
B *Function* yang menukarkan huruf besar kepada huruf kecil.  
C *Function* yang memulangkan secara rawak integer 1 hingga 100.  
D *Function* yang menyemak sama ada suatu integer itu berada dalam julat yang dibenarkan atau tidak.

11. Manakah antara pernyataan mengenai *function* dan *procedure* berikut adalah benar?
- A *Function* dan *procedure* perlu ditulis oleh pengguna.
  - B *Function* akan memulangkan suatu nilai tetapi *procedure* tidak memulangkan sebarang nilai kepada atur cara utama.
  - C *Function* dan *procedure* mesti menerima input daripada atur cara utama.
  - D *Function* boleh memanggil *function* lain sahaja dalam kod arahnya.

12. Berikut ialah kod arahan *function* yang mengira kuasa tiga.

```
def f(nombor):
    pernyataan function
    # -----Atur cara utama-----
    noInput = input("Masukkan satu nombor :")
    print ("Kuasa tiga bagi "+noInput+" ialah "+str(f(int(noInput))))
```

Manakah antara kod berikut merupakan kod arahan yang betul untuk pernyataan *function*?

- A return (nombor)
  - B print (nombor)
  - C return (nombor \*\* 3)
  - D print (nombor\*\*3)
13. Kod arahan berikut menunjukkan penggunaan beberapa *built-in function*.

```
>>> x = "Sains Komputer"
>>> len(x)
>>> I
>>> print(x*2)
>>> II
>>> print("x")
>>> III
```

Manakah antara berikut menunjukkan output yang betul?

- I 14
- II Sains KomputerSains Komputer
- III Sains Komputer
- A I sahaja
- B I dan II
- C I dan III
- D I, II dan III

14. Berikut menunjukkan kod arahan Python.

```
def ujian1(bilangan):
    return str(bilangan)

def ujian2(bilangan):
    return str(2 * bilangan)

# -----Atur cara utama-----
jumlah = ujian1(1) + ujian2(2)
print (jumlah)
```

Apakah outputnya apabila kod arahan di atas dilaksanakan?

- A 3                      C 12
- B 5                      D 14

15. Berikut ialah satu *user-defined function* yang dihasilkan oleh Ali.

```
def cetak(a, b=5, c=10):
    print("a ialah", a," b
    ialah", b,"dan c
    ialah", c)
```

Manakah antara panggilan *function* dan paparan output yang diperolehi adalah benar?

Panggilan <i>function</i>	
I	cetak(3)
II	cetak(3, 7)
III	cetak(25, c = 24)
IV	cetak(c = 50, a = 100)



## Output yang diperoleh

I	a ialah 3 b ialah 5 dan c ialah 10
II	a ialah 3 b ialah 7 dan c ialah 10
III	a ialah 25 b ialah 5 dan c ialah 24
IV	a ialah 100 b ialah 5 dan c ialah 50

- A I  
 B I dan II  
 C I, II, dan III  
 D I, II, III dan IV

## E-PRAKTIS



[goo.gl/qH1owq](http://goo.gl/qH1owq)

## Soalan Berstruktur

1. Berikut menunjukkan jadual bagi entiti BUKU dan entiti PENERBIT.

ISBN	Kod_Penerbit	Jenis_Bahan
978-983-47-1417-8	P120	Novel
978-983-46-0651-0	A13	Novel
978-983-46-1942-8	A13	Novel
978-967-61-2455-5	P222	Novel

Kod_Penerbit	Nama_Penerbit	Alamat_Penerbit
P120	Penerbitan Warna	4, Jalan Warni, Taman Palet, Selangor.
A13	Buku Bersama	4-5, Jalan Sama-sama, PJ, Selangor.
P185	Rakan Buku	G15-1, Jalan Mesin, Serdang, Selangor.
P222	Penerbit ABC	11-4, Jalan SS11/5, PJ, Selangor.

Berdasarkan entiti BUKU dan entiti PENERBIT, jawab soalan-soalan berikut:

- Nyatakan kekunci primer dan kekunci asing bagi entiti BUKU dan entiti PENERBIT.
  - Berikan penerangan bagi jawapan anda di (a).
  - Apakah kekardinalan antara entiti PENERBIT dan entiti BUKU?
2. Berikut menunjukkan jadual untuk entiti NILAM.

ID_Murid	Tahun_Terbitan	Tarikh	Judul	Penerbit	Pengarang
SM2101	2009	1/10/2017	SI PENGEMIS BERTUAH	SRI CEMPAKA	WAHEEDA SALEH
SM2102	2009	1/10/2017	MUTIARA AJAIB	BINTANG CEMERLANG	ABU BAKAR ARIF
SM2103	2010	1/10/2017	AKU MAHU BANYAK LIKE	BUMI HIJAU	ALIF OMAR
SM2104	2010	1/10/2017	SAYANGKU JALUR GEMILANG	TUNAS HARAPAN	FATIMAH BASAR
SM2105	2011	1/10/2017	JUARA KAMPUNG	PINTAR	ISMAIL HARUN

Tuliskan pernyataan-pernyataan SQL berdasarkan keadaan berikut:

- Perlukan medan Tahun\_Terbitan, Judul dan Penerbit dengan syarat judul diterbitkan pada 2009. Paparan maklumat mesti disusun mengikut Judul secara menaik.
  - Perlukan medan ID\_Murid, Judul dan Tarikh. Paparan maklumat mesti disusun mengikut Tarikh dari awal bulan sehingga ke akhir bulan.
3. Tuliskan satu segmen kod yang menggunakan *built-in function* untuk memaparkan setiap nama dalam *list* nama yang berikut.

nama = ['Fauzi bin Ayub', 'Paulsta ak Vincent', 'Ambo Anre', 'Marimuthu a/l Maniapan']

Setiap nama mesti bermula pada baris baharu seperti yang ditunjukkan di sebelah.

## Contoh output:

```
Fauzi bin Ayub
Paulsta ak Vincent
Ambo Anre
Marimuthu a/l Maniapan
```

4. Kaji dan uji kod arahan berikut secara manual.



```
def cetak(teks, bilangan):  
    while bilangan > 0:  
        print(teks)  
        bilangan = bilangan - 1  
cetak("Salam", 4)
```

- (a) Apakah output yang diperoleh?  
(b) Apakah kelemahan kod ini?  
(c) Baiki dan tuliskan semula kod yang telah dibaiki.
5. Hasilkan satu atur cara yang menggunakan *function* untuk menyusun secara songsang rentetan yang diberikan.
- Contoh rentetan: "1234abcd"  
Output: "dcba4321"



### Soalan Penyelesaian Masalah

1. Guru yang menguruskan biasiswa di sekolah anda telah mengalami masalah untuk mendapatkan maklumat murid yang layak menerima biasiswa dengan cepat. Sebagai murid yang mengambil mata pelajaran Asas Sains Komputer, anda diminta untuk membangunkan Sistem Maklumat Biasiswa (SMB) supaya dapat membantu guru tersebut. SMB perlu menjana laporan yang mengandungi ID Murid, Nama Murid, Nama Kelas, Gred A bagi mata pelajaran Bahasa Melayu, Bahasa Inggeris, Matematik dan Sains. Dalam proses membangunkan sistem tersebut. Jawab soalan-soalan berikut.
- (a) Bina pangkalan data yang mempunyai dua jadual yang dinamakan MURID dan PENCAPAIAN. Pangkalan data tersebut perlu menyimpan ID Murid, Nama Murid, Nama Kelas, Gred Bahasa Melayu, Gred Bahasa Inggeris, Gred Matematik, dan Gred Sains.  
(b) Bina hubungan bagi dua jadual yang dibina.  
(c) Masukkan enam rekod dalam jadual MURID dan PENCAPAIAN.  
(d) Masukkan sekurang-kurangnya dua rekod bagi Nama Kelas yang sama.  
(e) Hasilkan pertanyaan yang menghubungkan dua jadual. Pertanyaan tersebut perlu berupaya untuk mendapatkan data bagi ID Murid, Nama Murid, Nama Kelas dan Gred A.  
(f) Hasilkan satu borang berdasarkan jadual-jadual di (a). Berikan nama yang sesuai bagi borang-borang tersebut.  
(g) Hasilkan laporan berdasarkan pertanyaan di (e). Laporan perlu memaparkan ID Murid, Nama Murid, Nama Kelas dan Gred A bagi semua mata pelajaran.
2. Tuliskan satu atur cara bagi permainan meneka nama buah-buahan yang dipilih secara rawak oleh komputer daripada satu senarai buah-buahan. Nama buah-buahan itu ialah epal, oren, ciku, limau, durian, delima, zaitun dan pisang. Dalam permainan ini, bilangan tekaan dihadkan kepada panjang nama buah-buahan yang sedang diteka. Atur cara anda harus boleh menggunakan *built-in functions* dan *user-defined functions* yang sesuai untuk berinteraksi dengan pengguna.



# JAWAPAN

## BAB 1 Konsep Asas Pemikiran Komputasional



Imbas QR code ini untuk mendapatkan jawapan bagi aktiviti yang terpilih.

[goo.gl/EzyVh2](https://goo.gl/EzyVh2)

### Uji Kendiri 1.1

- 01 Fasa analisis masalah  
02 Fasa reka bentuk atur cara  
03 Fasa pengkodan  
04 Fasa pengujian dan penyahpejatan  
05 Fasa dokumentasi
- Fasa analisis masalah
- pseudokod, carta alir
- pengkodan
- Ralat sintaks, ralat logik dan ralat masa larian
- Dokumentasi luaran dan dokumentasi dalaman
- Teknik-teknik pemikiran komputasional
  - Teknik leraian
  - Teknik pengecaman corak
  - Teknik peniskalaan
  - Teknik pengitlakan
- leraian
- peniskalaan
- pengecaman corak
- Maksud ralat-ralat
  - Ralat sintaks – ralat yang disebabkan penulisan oleh atur cara yang tidak memenuhi sintaks sesuatu bahasa pengaturcaraan
  - Ralat logik – ralat yang disebabkan oleh kesilapan logik pengatur cara sehingga menyebabkan atur cara menghasilkan output yang salah
  - Ralat masa larian – ralat yang disebabkan oleh kemasukan data yang tidak menepati kehendak arahan. Ralat ini akan menyebabkan pelaksanaan atur cara terhenti secara tiba-tiba dengan mengeluarkan paparan mesej tertentu
- Penulisan komen yang betul adalah dengan
  - meletakkan simbol *hash* (#) di permulaan sesuatu ayat tunggal
  - meletakkan tiga tanda petikan ("" "" """) pada komen berbilang yang melebihi satu baris
- Dokumen pustaka memudahkan pengatur cara mengingat semula atau pengatur cara lain memahami struktur, logik dan kegunaan atur cara serta keperluan perkakasan dan perisian sekiranya ada keperluan untuk menyenggara atur cara tersebut.
- Dokumentasi dalaman membantu pengatur cara itu sendiri mengingat semula atau pengatur cara lain memahami maksud dan kegunaan sesuatu baris kod atau segmen kod. Seterusnya, ini akan memudahkan pengatur cara menyenggara atur cara apabila perubahan perlu dilakukan kelak.

### PENTAKSIRAN 1

#### Soalan Objektif

1. B    2. A    3. D    4. B    5. C  
6. C    7. B    8. B    9. D    10. D

#### Soalan Berstruktur

- Atur cara untuk mencari jumlah dan purata dua nombor
  - Pemboleh ubah input
    - nombor1
    - nombor2
  - Pernyataan dalam baris yang berikut mengandungi ralat:
    - Baris 5
    - Baris 9
  - Baris 5  
jumlah = nombor1 + nombor2  
Baris 9  
print("Purata bagi dua nombor ialah", purata)

2. (a) Menulis pernyataan kod untuk  
 (i) meminta input daripada pengguna

```
nombor1 = float(input
("Masukkan nombor pertama:
"))
nombor2 = float(input
("Masukkan nombor kedua:
"))
```

- (ii) formula yang digunakan

```
tambah = nombor1 + nombor2
tolak = nombor1 - nombor2
darab = nombor1 * nombor2
bahagi = nombor1 / nombor2
```

- (iii) output yang akan dihasilkan

```
print("\nHasil tambah bagi
dua nombor ialah ",tambah)
print("\nHasil tolak bagi
dua nombor ialah ",tolak)
print("\nHasil darab bagi
dua nombor ialah ",darab)
print("\nHasil bahagi bagi
dua nombor ialah ",bahagi)
```

- (b) Lakaran paparan output

Masukkan nombor 1: ...  
 Masukkan nombor 2: ...  
 Hasil tambah bagi dua nombor ialah ...  
 Hasil tolak bagi dua nombor ialah ...  
 Hasil darab bagi dua nombor ialah ...  
 Hasil bahagi bagi dua nombor ialah ...

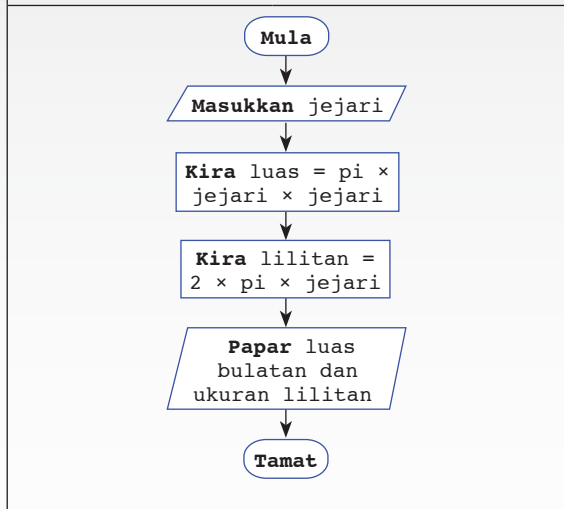
### Soalan Penyelesaian Masalah

- (a) Pelaporan

Fasa 1: Analisis Masalah
<b>Teknik Leraian</b> Membina atur cara mudah bagi mengira luas dan lilitan sebuah bulatan Kira luas bulatan      Kira ukur lilit bulatan
Fasa 2: Reka bentuk atur cara
<b>Teknik Pengecaman Corak</b> Item input bagi pengiraan luas dan lilitan bulatan, iaitu jejari dan pi.

Teknik Peniskalaan	
Pengiraan luas	Nilai jejari dan pi
	Formula matematik: Luas bualatan = $\pi r^2$
Pengiraan lilitan	Nilai jejari dan pi
	Lilitan bulatan = $2\pi r$

Teknik Pengitlakan
1. <b>Mula</b> 2. <b>Masukkan</b> jejari 3. <b>Kira</b> 3.1 luas = pi × jejari × jejari 3.2 lilitan = 2 × pi × jejari 4. <b>Papar</b> luas bulatan dan ukuran lilitan 5. <b>Tamat</b>



### Fasa 3: Pengekoden

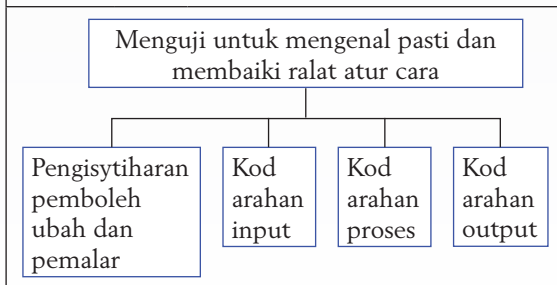
Teknik Leraian
Mengekod algoritma, mengira luas dan lilitan sebuah bulatan Pengisytiharan pemboleh ubah dan pemalar      Input      Proses      Output

Teknik Pengecaman Corak
Pengisytiharan pemboleh ubah dan pemalar Pemboleh ubah: jejari, luas dan lilitan bulatan Pemalar: pi yang boleh diimport dari <i>module math</i>

Input	Pengguna memasukkan satu nombor untuk jejari
Proses	Kira luas bulatan = $\pi \times \text{jejari} \times \text{jejari}$ Kira lilitan bulatan = $2 \times \pi \times \text{jejari}$
Output	Papar luas dan lilitan bulatan
<b>Teknik Peniskalaan</b>	
Aspek-aspek penting	
Pengisytiharan pemboleh ubah dan pemalar	Pemboleh ubah jejari ditetapkan <i>float</i> sebagai jenis datanya ( <i>data type</i> ). Import <i>module math</i> agar dapat menggunakan pemalar pi.
Input	Kod arahan mesej dorongan: <code>jejari = float(input ("\nMasukkan jejari: "))</code>
Proses	Kod arahan mengira luas dan lilitan bulatan: <code>luas = pi * jejari * jejari</code> <code>lilitan = 2 * pi * jejari</code>
Output	Kod arahan memaparkan nilai luas dan lilitan bulatan: <code>print("\nLuas bulatan ialah", luas)</code> <code>print("Lilitan bulatan ialah", lilitan)</code>

#### Fasa 4: Pengujian dan Penyahpepajian

##### Teknik Leraian



##### Teknik Pengecaman Corak

Pengisytiharan pemboleh ubah dan pemalar	Murid melaporkan sebarang ralat atau mesej ralat yang didapati. Jika ralat yang dijumpai mempunyai corak tertentu, huraikan corak itu. Misalnya, <i>NameError: name 'pi' is not defined</i> atau <i>NameError: name 'math' is not defined</i>
--	--

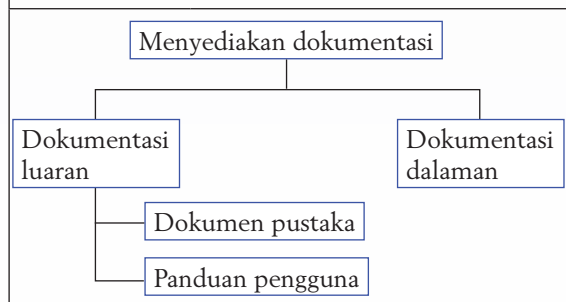
Input	
Proses	
Output	
<b>Teknik Peniskalaan</b>	
Aspek-aspek penting	
Pengisytiharan pemboleh ubah dan pemalar	Pemboleh ubah diberi; • jenis data yang sesuai. • nama yang sesuai dan nama dieja dengan betul. • nilai permulaan.
Input	Memastikan pemboleh ubah jejari dan pemalar pi digunakan untuk menyimpan nilai yang betul.
Proses	• Formula yang digunakan adalah betul. • Logik dan susunan adalah betul.
Output	• Output yang diperoleh adalah tepat dan betul. • Format output adalah seperti yang dikehendaki.

##### Teknik Pengitlakan

Menghasilkan satu senarai semak yang dapat digunakan semasa pengujian dan penyahpepajian sesuatu atur cara.

#### Fasa 5: Dokumentasi

##### Teknik Leraian



##### Teknik Pengecaman Corak

Dokumentasi dalaman	Kaji beberapa contoh atur cara, dokumen pustaka dan panduan pengguna untuk melihat corak dan jenis maklumat yang perlu dimasukkan dalam dokumen-dokumen yang berkenaan.
Dokumen pustaka	
Panduan pengguna	

<b>Teknik Peniskalaan</b>	
Aspek penting	
Dokumentasi dalaman	Penulisan komen untuk menerangkan fungsi setiap bahagian atau sesuatu barisan kod.
Dokumentasi pustaka	Maklumat yang diperlukan oleh seorang pengatur cara: <ul style="list-style-type: none"> <li>• Tarikh</li> <li>• Nama pengatur cara</li> <li>• Tujuan/Kegunaan atur cara</li> <li>• Keperluan sistem pengendalian</li> <li>• Keperluan perkakasan</li> <li>• Keperluan perisian</li> <li>• Algoritma (pseudokod dan carta alir)</li> <li>• Kod sumber (<i>source code</i>)</li> </ul>
Panduan pengguna	Maklumat yang diperlukan oleh seorang pengguna: <ul style="list-style-type: none"> <li>• Nama dan tujuan atur cara</li> <li>• Keperluan sistem pengendalian, perkakasan dan perisian.</li> <li>• Cara penggunaan</li> </ul>
<b>Teknik Pengitlakan</b>	
Menghasilkan dokumentasi dalaman dan dokumentasi luaran berdasarkan aspek-aspek penting yang dikenal pasti.	

(b) Kod atur cara mengira luas dan lilitan sebuah bulatan

```

Soalan_Bab 1.py - C:\Documents\Tutorial Python\Soalan...
File Edit Format Run Options Window Help
""" Atur cara bagi mengira luas dan lilitan sebuah
    bulatan"""

""" Tujuan: Mengira luas dan lilitan bulatan berdasarkan
    nilai jejari yang diinput oleh pengguna."""
# Disediakan oleh Ahmad b. Firdaus pada 28 Feb 2018.

import math # supaya boleh menggunakan pemalar pi

# Meminta pengguna memasukkan input
jejari = float(input("Masukkan jejari bulatan (cm): "))

# Mengira luas dan lilitan
luas = math.pi * jejari * jejari
lilitan = 2 * math.pi * jejari

# Paparan output
print("\nLuas bulatan ialah", round(luas,3),"sentimeter padu.")
print("Ukuran lilitan ialah", round(lilitan,3),"sentimeter.")
Ln:1 Col:0

```

## BAB 2 Perwakilan Data



Imbas QR code ini untuk mendapatkan jawapan bagi aktiviti yang terpilih.  
[goo.gl/oDr6Xz](https://goo.gl/oDr6Xz)

### Uji Kendiri 2.1

1. Kriptografi dihuraikan sebagai sains atau kajian tentang teknik kerahsiaan atau dikenali sebagai keselamatan komunikasi. Tujuan utama kriptografi adalah untuk mengekod komunikasi bagi menghalang kandungannya daripada didedahkan atau dipintas oleh pihak lain.
2. Pada zaman moden hari ini, manusia amat bergantung kepada sistem pengkomputeran digital. Kriptografi penting untuk melindungi maklumat penting atau sulit yang disimpan dalam bentuk digital yang mudah diakses melalui laman sesawang.
3. Pengesahan – Pengenalpastian pihak-pihak yang terlibat dalam suatu komunikasi. Kerahsiaan – Kesulitan sesuatu maklumat yang dihantar melalui rangkaian komputer dapat dipelihara. Integriti – Perkhidmatan keselamatan yang mengenal pasti sebarang perubahan terhadap maklumat atau data oleh pihak lain. Tiada sangkalan – Membuktikan penghantar dan penerima maklumat masing-masing tidak menafikan bahawa mereka menghantar dan menerima maklumat tersebut.
4. (a) Teks asal yang boleh dibaca dan difahami dengan mudah. Teks sebelum disulitkan.  
(b) Teks yang tidak membawa makna apabila dibaca. Teks setelah disulitkan.  
(c) Satu proses penukaran teks biasa (*plaintext*) kepada teks sifer (*ciphertext*) yang tidak membawa makna apabila dibaca.  
(d) Satu proses menukarkan teks sifer (*ciphertext*) menjadi teks biasa (*plaintext*) semula.  
(e) Satu nilai atau perkataan yang perlu digunakan oleh sesetengah kaedah sifer untuk melakukan penyulitan dan penyahsulitan.



## BAB 3 Algoritma



Imbas QR code ini untuk mendapatkan jawapan bagi aktiviti yang terpilih.

[goo.gl/dY6nP2](https://goo.gl/dY6nP2)



Imbas QR code ini untuk mendapatkan jawapan bagi carta alir dalam soalan-soalan Uji Kendiri 3.1 dan Latihan Sumatif 3.

[goo.gl/iwv1JR](https://goo.gl/iwv1JR)

### Uji Kendiri 3.1

- Linear search* ialah suatu teknik carian yang mudah dan sesuai digunakan pada senarai yang tidak diisih. Teknik carian ini juga lebih efisien jika digunakan pada senarai yang tidak mengandungi item yang banyak. Walau bagaimanapun, *linear search* ini tidak begitu cekap kerana setiap item dalam senarai akan disemak.
  - Binary search* sesuai digunakan pada senarai yang diisih. Teknik carian ini lebih menjimatkan masa kerana semakan tidak perlu dilakukan pada setiap item dalam senarai. Bagi senarai yang mempunyai bilangan item yang banyak, teknik carian ini lebih efisien digunakan.
  - Bubble sort* sesuai digunakan pada senarai yang tidak mengandungi item yang banyak.
  - Isihan baldi sesuai digunakan pada senarai yang mengandungi item yang banyak.
- Pseudokod:

```
1 Mula
2 Setkan senarai = [6, 8, 10, 12,
14, 16, 18, 20]
3 Isytihar pemboleh ubah i, j, m,
T
4 Setkan n = 8
5 Setkan i = 0
6 Setkan j = 7
7 Masukkan satu item carian t
```

```
8 while i < j
8.1 Setkan m = (i + j)/2
8.2 Jika t == Lm
8.2.1 Papar "Item ada dalam
senarai"
8.2.2 Keluar gelung
8.3 Jika tidak
8.3.1 Jika t < Lm
8.3.1.1 Kira j = m - 1
8.3.2 Jika tidak
8.3.2.1 Kira i = m + 1
8.3.3 Tamat jika
8.4 Tamat jika
9 Tamat
```

### 3. Bubble sort

- Teknik *binary search* ialah teknik untuk mencari nilai pertengahan dalam senarai. Item carian akan dibandingkan dengan item pertengahan. Semakan akan dilakukan sama ada item carian lebih besar atau lebih kecil daripada item pertengahan. Dengan cara ini, hanya separuh daripada item-item dalam senarai yang akan disemak lagi dan separuh lagi akan diabaikan. Proses ini akan berulang sehingga item carian dijumpai dalam senarai. Oleh itu, *binary search* ini adalah lebih pantas dan sesuai digunakan.
- Linear search*:

```
1 Mula
2 Setkan senarai = [A, B, C, D,
E, F, G, H]
3 Isytihar pemboleh ubah i, n,
t
4 Setkan i = 0
5 Setkan n = 8
6 Masukkan satu nilai carian t
7 for i < n
7.1 Jika Li == t
7.1.1 Papar "Item ada
dalam senarai"
7.2 Jika tidak
7.2.1 Kira i = i + 1
8 Jika i >= n
8.1 Papar "Item tiada dalam
senarai"
9 Tamat
```



Binary search:

```

1 Mula
2 Setkan senarai = [A, B, C, D, E, F, G, H]
3 Isytihar pemboleh ubah i, j, m, K
4 Setkan n = 8
5 Setkan i = 0
6 Setkan j = 7
7 Masukkan satu item carian K
8 while i < j
  8.1 Setkan m = (i + j)/2
  8.2 Jika K == Lm
    8.2.1 Papar "Item ada dalam senarai"
    8.2.2 Keluar gelung
  8.3 Jika tidak
    8.3.1 Jika K < Lm
      8.3.1.1 Kira j = m - 1
    8.3.2 Jika K >= Lm
      8.3.2.1 Kira i = m + 1
    8.3.3 Tamat jika
  8.4 Tamat jika
9 Tamat
  
```

- (b) – Carian dilakukan pada senarai [A, B, C, D, E, F, G, H]
- Sebelum gelung bermula, kedudukan item pertama dalam senarai disetkan kepada 0
  - Satu nilai carian perlu dikenal pasti

6. (a) *Linear search* kerana bilangan nombor yang tidak begitu besar (100 nombor), dan semua nombor perlu disemak untuk melihat sama ada nombor mempunyai digit 5.

*Bubble sort* kerana bilangan nombor yang perlu diisih adalah kecil, iaitu 5, 15, 25, 35, 45, 55, 65, 75, 85 dan 95

(b) Algoritma:

1. Cari semua kad nombor yang mempunyai digit 5.
2. Kumpulkan semua kad nombor ini.
3. Isih kad-kad nombor dengan digit 5 mengikut urutan menurun.

### PENTAKSIRAN 3

#### Soalan Objektif

1. D    2. B    3. A    4. C    5. B  
6. C    7. C    8. D    9. B    10. B

#### Soalan Berstruktur

1. – Item-item dalam senarai berada dalam keadaan terisih.  
– Senarai mengandungi item yang banyak.
2. Teknik *bubble sort* akan digunakan kerana hanya terdapat 5 item dalam senarai. Teknik *bubble sort* juga lebih menjimatkan masa bagi senarai yang mempunyai item yang sedikit.

3.

	Perbezaan	Persamaan
<i>Linear search</i>	• Setiap item dalam senarai akan disemak.	• Carian dilakukan pada senarai[ ].
<i>Binary search</i>	• Nilai carian akan dibandingkan dengan nilai yang berada di tengah-tengah senarai.	• Satu nilai carian perlu dikenal pasti.

4.

```

7.2.1 Jika Lj > Lj+1
  7.1.1 temp = Lj
  7.1.2 Lj = Lj+1
  7.1.3 Lj+1 = temp
7.2.2 Tamat jika
7.2.3 Kira j = j + 1
7.3 Kira i = i + 1
  
```

#### Soalan Penyelesaian Masalah

1. (a) *Bubble sort*

1. Semak item pertama dan item kedua dalam senarai.
2. Bandingkan kedua-dua item tersebut.
3. Jika kedua-dua item tidak mengikut urutan, tukar kedudukan kedua-duanya.
4. Ulang langkah 2 hingga langkah 3 dengan item kedua dan item ketiga, item ketiga dan item keempat dan seterusnya hingga akhir senarai.
5. Semak senarai untuk melihat sama ada semua item berada pada kedudukan yang betul.
6. Ulang langkah 1 hingga 5 sehingga semua item dalam senarai telah diisih mengikut urutan.

(b) Pseudokod:

```
1 Mula
2 Setkan senarai = [30, 22, 33,
55, 42]
3 Isytihar pemboleh ubah i, j,
n, temp
4 Setkan n = 5
5 Setkan i = 0
6 Setkan temp = 0
7 for i < n - 1
7.1 Setkan j = 0
7.2 for j < n - i - 1
7.2.1 Jika  $L_j > L_{j+1}$ 
7.2.1.1 temp =  $L_j$ 
7.2.1.2  $L_j = L_{j+1}$ 
7.2.1.3  $L_{j+1} = temp$ 
7.2.2 Tamat jika
7.2.3 Kira  $j = j + 1$ 
7.3 Kira  $i = i + 1$ 
8 Papar senarai
9 Tamat
```

2. Pseudokod:

```
1 Mula
2 Setkan senarai = [3, 4, 7, 10,
15, 24, 36, 72, 90]
3 Isytihar pemboleh ubah i, j, m, K
4 Setkan n = 10
5 Setkan i = 0
6 Setkan j = 9
7 Masukkan satu item carian K
8 while i < j
8.1 Setkan  $m = (i + j)/2$ 
8.2 Jika  $K == L_m$ 
8.2.1 Papar "Item ada dalam
senarai"
8.2.2 Keluar gelung
8.3 Jika tidak
8.3.1 Jika  $K < L_m$ 
8.3.1.1 Kira  $j = m - 1$ 
8.3.2 Jika tidak
8.3.2.1 Kira  $i = m + 1$ 
8.3.3 Tamat jika
8.4 Tamat jika
9 Tamat
```

3. Aplikasi ini akan mencari nombor gandaan positif (tidak termasuk sifar) daripada satu senarai nombor yang dimasukkan oleh pengguna. Setelah mencari semua nombor gandaan positif yang terdapat dalam senarai, aplikasi akan memaparkan senarai nombor gandaan positif tersebut. Jika tiada nombor gandaan positif dalam senarai, aplikasi akan tamat.

Pengguna perlu memasukkan:

- Bilangan item dalam senarai
- Item-item dalam senarai
- Nombor yang hendak digandakan

Algoritma:

```
1. Isytihar dua senarai, senarai[]
dan senaraiBaharu[]
2. Isytihar pemboleh ubah i, j,
noGanda
3. Setkan i = 0
4. Pengguna taipkan satu set nombor
positif untuk dimasukkan ke
dalam senarai[]
5. Pengguna taipkan satu nilai
sebagai nombor yang hendak
digandakan
6. Setkan noGanda = nombor yang
hendak digandakan
7. Setkan n = bilangan item dalam
senarai
8. Semak item pertama dalam
senarai[]
9. Jika item dalam senarai boleh
dibahagi dengan noGanda tanpa
baki
Masukkan item dalam
senaraiBaharu[ ]
10. Jika masih ada item lain dalam
senarai[]
Semak item yang berikut
Ulang langkah 6
11. Jika senaraiBaharu[] == kosong
Papar "Tiada nombor gandaan"
Jika lain
Papar senaraiBaharu[]
12. Tamat
```

## BAB 4 Kod Arahan



Imbas QR code ini untuk mendapatkan jawapan bagi aktiviti yang terpilih.

[goo.gl/eQLj6T](https://goo.gl/eQLj6T)

### Uji Kendiri 4.1

- Data
- berpusat
- unik
- entiti
- Mengelakkan pertindihan antara rekod lama dengan rekod baharu
  - Memastikan setiap rekod data yang disimpan mempunyai satu nilai unik sebagai pengenalan diri
- Kekardinalan merujuk kepada jenis perhubungan antara dua entiti dalam sesebuah organisasi. Kekardinalan menyatakan bilangan hubungan yang boleh wujud antara dua entiti itu secara khusus. Dalam pangkalan data, kekardinalan merujuk kepada bilangan pautan yang dapat dibentuk antara setiap rekod dalam sesuatu jadual entiti dengan rekod-rekod dalam jadual entiti yang lain dan sebaliknya.
  -



(ii)



- Jadual (*Table*)
  - Borang (*Form*)
  - Pertanyaan (*Query*)
  - Laporan (*Report*)
- Borang digunakan untuk memasukkan, mengubah suai dan memadamkan data serta rekod.
  - Borang mempunyai antara muka pengguna yang interaktif.

- Structured Query Language* (SQL)
  - Pernyataan SELECT...FROM  
Pernyataan SELECT...WHERE  
Pernyataan SELECT...ORDER BY

- Nama entiti: MURID  
Nama atribut: ID\_Murid, Nama\_Murid

11.

P	Q	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0

P	Q	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

- Menjana laporan dengan menggunakan *Report Wizard*
- SELECT Nama\_Ibu, Nama\_Bapa  
FROM BUTIR\_PENJAGA;
  - SELECT ID\_Murid, Nama, Kaum  
FROM BUTIR\_MURID  
ORDER BY Nama ASC;
  - SELECT ID\_Murid, Nama, Kaum  
FROM BUTIR\_MURID  
WHERE Kaum = "CINA";
  - SELECT Nama, Jantina, Kaum  
FROM BUTIR\_MURID  
WHERE Jantina = "LELAKI"  
ORDER BY Nama ASC;

### Uji Kendiri 4.2

- Suatu masalah besar lebih mudah diselesaikan jika masalah tersebut diselesaikan secara modular.
  - Memudahkan proses penyahpepajian dilakukan.
  - Function* dan *procedure* yang dibina dapat digunakan semula dalam atur cara lain jika diperlukan.
- User-defined function 1*

```

def pow1(x,y):
    return (x**y)
print (pow1(2,8))
    
```

### User-defined function 2

```
def pow2(x,y):  
    sum = x  
    for i in range(1,y):  
        sum = sum * x  
    return sum  
print (pow2(2,8))
```

3. Mula-mula, umpukan kepada pemboleh ubah nilai yang dipulangkan oleh *function* input. Kemudian, gunakan *function type* untuk mengenal pasti jenis data yang disimpan dalam pemboleh ubah tersebut. Misalnya,

```
Nombor 1 = int(input("Masukkan  
satu nombor integer:"))  
Nombor 2 = float(input("Masukkan  
satu nombor perpuluhan:"))
```

4. (a) *User-defined function* (a)

```
def corak():  
    for i in range (6):  
        print("**" * i)  
  
# Pemanggilan function  
corak()
```

- (b) *User-defined function* (b)

```
def corak(aksara):  
    for i in range (6):  
        print(aksara * i)  
  
# Pemanggilan function  
corak("%")
```

- (c) *User-defined function* (c)

```
def corak(aksara, bilangan):  
    for i in range (bilangan + 1):  
        print(aksara * i)  
  
# Pemanggilan function  
corak("%", 7)
```

- (d) *User-defined function* (d)

```
def corak(aksara, bilangan):  
    for i in range(bilangan+1):  
        print(aksara * i)  
  
aksaral = input("Masukkan aksara  
pilihan anda: ")  
baris = int(input("Nyatakan  
bilangan baris : "))  
  
# Pemanggilan function  
corak(aksaral, baris)
```

### PENTAKSIRAN 4

#### Soalan Objektif

1. D    2. A    3. B    4. D    5. C  
6. B    7. A    8. D    9. B    10. A  
11. B    12. C    13. B    14. D    15. D

#### Soalan Berstruktur

1. (a) Entiti BUKU  
Kunci primer ialah ISBN  
Kunci asing ialah Kod Penerbit  
Entiti PENERBIT  
Kunci primer ialah Kod Penerbit  
Kunci asing ialah ISBN
- (b) Entiti BUKU  
ISBN ialah nilai unik bagi setiap rekod.  
Kod Penerbit sesuai menjadi kunci asing  
kerana atribut ini berupa kunci primer  
bagi jadual (entiti) PENERBIT.  
Entiti PENERBIT  
Kod Penerbit ialah nilai unik yang  
mewakili setiap penerbit. ISBN ialah  
kunci asing kerana atribut ini ialah kunci  
primer bagi jadual (entiti) BUKU.
- (c) Penerbit boleh menerbitkan banyak buku.  
Oleh itu, jenis kekardinalan ialah *one to many* (1:M)



2. (a) SELECT Tahun \_Terbitan, Judul, Penerbit  
FROM NILAM  
WHERE Tahun \_Terbitan = "2009"  
ORDER BY Judul ASC;  
(b) SELECT ID\_Murid, Judul, Tarikh  
FROM NILAM  
ORDER BY Tarikh ASC;
3. nama = ["Fauzi bin Ayub",  
"Paulsta ak Vincent", "Ambo  
Anre", "Marimuthu a/l Maniapan"]  
print("\n".join(nama))
4. (a) Output
- ```
Salam  
Salam  
.  
.  
.
```
- (b) Output dipaparkan secara tak terhingga  
(c) Kod yang dibaiki

```
def cetak(teks, bilangan):
    while bilangan > 0:
        print(teks)
        bilangan = bilangan - 1

# -----Atur cara utama-----
cetak("Salam", 4)
```

5. Kod atur cara menggunakan *built-in function*

```
# -----Atur cara utama-----
senarai = list("1234abcd")
senarai.reverse()
print(str(senarai))
```

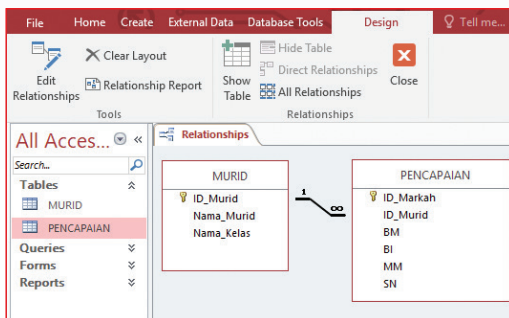
Kod atur cara *user-defined function*

```
def songsang(rentetan):
    srentetan = ""
    index = len(rentetan)
    while index > 0:
        srentetan += rentetan[index-1]
        index = index - 1
    return srentetan

print(songsang("1234abcd"))
```

### Soalan Penyelesaian Masalah

- (a) Dua jadual: MURID dan PENCAPAIAN  
(b) Hubungan *one to many* (1:M)



(c) Enam rekod dalam jadual

| ID_Murid | Nama_Murid     | Nama_Kelas |
|----------|----------------|------------|
| M1002    | AMIR BIN HAMSA | 3 ADIL     |
| M1003    | LEE LIAN TEEN  | 3 ADIL     |
| M1004    | ALIA BT AFZAN  | 3 BIJAK    |
| M1005    | WONG LEE TAN   | 3 BIJAK    |
| M1006    | ARUN A/L MUTHU | 3 CERDIK   |
| M1007    | ZAZA BT ZAMRI  | 3 CERDIK   |

| ID_Markah | ID_Murid | BM | BI | MM | SN |
|-----------|----------|----|----|----|----|
| PT1002    | M1002    | A  | A  | A  | A  |
| PT1003    | M1003    | B  | C  | B  | A  |
| PT1004    | M1004    | A  | C  | D  | A  |
| PT1005    | M1005    | A  | A  | A  | A  |
| PT1006    | M1006    | D  | B  | A  | A  |
| PT1007    | M1007    | A  | A  | A  | A  |

(d) Dua rekod bagi nama kelas yang sama

| Nama_Kelas |
|------------|
| 3 ADIL     |
| 3 ADIL     |
| 3 BIJAK    |
| 3 BIJAK    |
| 3 CERDIK   |
| 3 CERDIK   |

(e) SELECT MURID.ID\_Murid, MURID>Nama\_Murid, MURID>Nama\_Kelas, PENCAPAIAN.BM, PENCAPAIAN.BI, PENCAPAIAN.MM, PENCAPAIAN.SN  
FROM MURID INNER JOIN  
PENCAPAIAN ON MURID.ID\_Murid =  
PENCAPAIAN.ID\_Murid  
WHERE PENCAPAIAN.BM="A"  
AND PENCAPAIAN.BI="A" AND  
PENCAPAIAN.MM="A" AND  
PENCAPAIAN.SN="A";

| ID_Murid | Nama_Murid     | Nama_Kelas | BM | BI | MM | SN |
|----------|----------------|------------|----|----|----|----|
| M1002    | AMIR BIN HAMSA | 3 ADIL     | A  | A  | A  | A  |
| M1005    | WONG LEE TAN   | 3 BIJAK    | A  | A  | A  | A  |
| M1007    | ZAZA BT ZAMRI  | 3 CERDIK   | A  | A  | A  | A  |

(f) Borang

(g) Laporan

| PENCAPAIAN GRED A |                |            |    |    |    |    |
|-------------------|----------------|------------|----|----|----|----|
| ID_Murid          | Nama_Murid     | Nama_Kelas | BM | BI | MM | SN |
| M1002             | AMIR BIN HAMSA | 3 ADIL     | A  | A  | A  | A  |
| M1005             | WONG LEE TAN   | 3 BIJAK    | A  | A  | A  | A  |
| M1007             | ZAZA BT ZAMRI  | 3 CERDIK   | A  | A  | A  | A  |

## 2. Kod atur cara



```
# Atur cara Meneka Nama Buah-buahan
import random
import collections
#----- Function cetakBuah -----
def cetakBuah(buah):
    for char in buah:
        if char in abjadTeka:
            print(char, end = " ")
        else:
            print("_", end = " ")
    print()
#----- Atur cara utama -----
senarai = ["epal", "oren", "ciku",
"limau", "durian", "delima",
"zaitun", "pisang"]
buah = random.choice(senarai)
print("\nTeka nama
buah (Petanda: namanya
mempunyai",len(buah),"abjad)")
abjadTeka = ""
cetakBuah(buah)

hadCubaan = 2*len(buah)

print("\nAnda diberi",hadCubaan,
"cubaan untuk mendapat nama dengan
betul.")
bilang = 1
```

```
while (bilang <= hadCubaan):
    print("\nCubaan",bilang,"/",
hadCubaan)
    teka=str(input("Masukkan abjad :
"))
    if not teka.isalpha():
        print("Masukkan hanya
abjad.")
    elif len(teka) > 1:
        print("Masukkan hanya abjad
tunggal.")
    elif teka in abjadTeka:
        print("Abjad ini sudah
diteka.")
    elif teka in buah:
        abjadTeka += teka
        cetakBuah(buah)

    if collections.
Counter(abjadTeka) ==
collections.Counter(buah):
        print("\nTahniah, anda
menang!")

        break
    bilang += 1

if bilang > hadCubaan:
    print("\nAnda kalah.")
    print("Nama buah itu ialah {}".
format(buah))
```

# GLOSARI

## **Argumen**

Nilai sebenar kepada pemboleh ubah yang digunakan oleh *function* atau *procedure*, iaitu parameter.

## **Atribut**

Merujuk kepada ciri-ciri yang menerangkan sesuatu entiti.

## **Binary search**

Satu teknik carian yang melibatkan keputusan dwipilihan. Sebelum melakukan *binary search*, item-item dalam senarai perlu diisih dalam urutan menaik.

## **Boolean**

Jenis data yang mempunyai hanya dua nilai logik, iaitu *True* dan *False*.

## **Bubble sort**

Satu teknik isihan yang menggunakan kaedah membandingkan dua item secara bergilir untuk mengisihkan item-item dalam senarai mengikut urutan menaik atau menurun.

## **Bucket sort**

Satu teknik isihan yang mengasingkan item-item dalam senarai tertentu ke dalam beberapa baldi (*bucket*), seterusnya item-item di dalam baldi akan diisih. Semua item di dalam baldi yang telah diisih disusun semula mengikut urutan yang diinginkan.

## **Built-in function**

Sejenis *function* yang sedia ada dan disimpan dalam pustaka bahasa pengaturcaraan.

## **Caesar Cipher**

Satu kaedah sifer yang menggantikan setiap abjad dalam teks biasa dengan abjad yang lain melalui anjakan dalam susunan abjad.

## **Data**

Maklumat yang dianggap sebagai aset penting bagi sesebuah organisasi.

## **Dokumentasi dalaman**

Disediakan untuk kegunaan pengatur cara lain atau pihak syarikat bagi memudahkan penyenggaraan sesuatu aplikasi atau atur cara.

## **Dokumentasi luaran**

Menerangkan cara-cara menggunakan aplikasi melalui penghasilan manual pengguna.

## **Entiti**

Merujuk kepada suatu objek atau satu jenis objek yang mempunyai data disimpan mengenainya.

## **Function**

Subatur cara yang sesuai melaksanakan tugas yang khusus dan berulang kali serta mampu memulangkan satu nilai selepas tugas diselesaikan.

## **Input**

Data yang dimasukkan ke dalam komputer atau atur cara.

## **Kekardinalan**

Merujuk kepada perhubungan antara entiti-entiti dalam satu pangkalan data.

## **Kekunci asing**

Satu atribut atau medan dalam satu jadual hubungan, tetapi pada masa yang sama atribut ini berupa kunci primer bagi satu jadual hubungan yang lain.

## **Kekunci primer**

Satu atribut dalam satu jadual hubungan yang mempunyai nilai-nilai yang unik.

## **Kriptografi**

Kajian tentang teknik kerahsiaan atau dikenali sebagai keselamatan komunikasi.

## **Leraian**

Teknik pemikiran komputasional yang memecahkan masalah yang besar kepada bahagian-bahagian masalah yang lebih kecil.

### ***Linear search***

Satu teknik carian untuk mendapatkan item yang dikehendaki dalam satu senarai linear.

### **Maklumat**

Maklumat merupakan hasil daripada pengumpulan, pemprosesan dan penganalisan data yang boleh digunakan untuk membuat sesuatu.

### **Nyahsulit**

Satu proses penukaran teks sifer kepada teks biasa.

### **Output**

Output merujuk kepada data yang dipaparkan pada skrin komputer.

### **Pangkalan data**

Tempat pengumpulan dan penyimpanan data secara berpusat bagi sistem maklumat yang berasaskan komputer.

### **Parameter**

Pemboleh ubah yang digunakan oleh sesuatu *function* atau *procedure* untuk menerima dan menyimpan data daripada atur cara yang memanggilnya dan memulangkan data kepada atur cara yang memanggilnya.

### **Pembangunan atur cara**

Melibatkan lima fasa, iaitu fasa analisis masalah, fasa reka bentuk atur cara, fasa pengekodan, fasa pengujian dan penyahpejijatan serta fasa dokumentasi.

### **Pemikiran komputasional**

Proses berfikir yang membenarkan seseorang mencari penyelesaian secara sistematik.

### **Pengecaman corak**

Teknik pemikiran komputasional yang mengenal pasti ciri-ciri persamaan corak pada bahagian-bahagian masalah yang kecil.

### **Pengitlakan**

Teknik pemikiran komputasional yang menghasilkan model bagi menyelesaikan masalah yang serupa.

### **Peniskalaan**

Teknik pemikiran komputasional yang mengenal pasti aspek-aspek penting dan meninggalkan aspek-aspek kurang penting.

### **Penyulitan (*Encryption*)**

Satu proses penukaran teks biasa kepada teks sifer, iaitu teks dalam bentuk yang tidak bermakna apabila dibaca.

### ***Pigpen Cipher***

Satu kaedah sifer yang menggantikan setiap abjad dalam teks biasa dengan simbol.

### ***Procedure***

Subatur cara yang sesuai melaksanakan tugas yang khusus dan berulang kali tanpa memulangkan sebarang nilai selepas tugas diselesaikan.

### **Proses**

Satu siri tindakan atau langkah yang dilaksanakan untuk melengkapkan satu tugas atau satu tujuan tertentu.

### ***Query***

Pertanyaan (*query*) bermaksud pertanyaan kepada pangkalan data hubungan dalam bahasa *query* untuk mencapai data dan maklumat yang disimpan dalamnya.

### ***Reverse cipher***

Satu kaedah sifer yang menggunakan cara songsangan untuk menyulitkan mesej.

### ***Search***

Proses untuk mendapatkan satu item tertentu yang terdapat dalam satu senarai.

### ***Sort***

Proses menyusun item-item dalam senarai linear mengikut urutan tertentu.

### **SQL**

Satu bahasa pengaturcaraan yang ringkas dan mudah untuk mencapai data dan maklumat dalam pangkalan data.

### ***Substitution cipher***

Satu kaedah sifer yang menyulitkan teks biasa dengan menggantikan satu unit teks biasa dengan satu unit teks yang lain.

### ***Transposition cipher***

Satu kaedah sifer yang mengubah kedudukan abjad dalam teks biasa. Abjad-abjad yang terkandung dalam teks biasa adalah tetap sama.



# INDEKS

## A

Argumen 169, 176  
Atribut 118–120

## B

Binary search 65–68, 77–80  
Boolean 157  
    operator OR 157  
    operator AND 159  
Bubble sort 69–72, 81–85  
Bucket sort 71–72, 86–89

## C

Caesar Cipher 36, 48, 49  
Columnar Transposition 39, 46

## D

Data 114  
Dokumentasi dalaman 2, 3, 14  
Dokumentasi luaran 2, 3  
    dokumen pustaka 13, 15  
    panduan pengguna 14, 16–18

## E

Entiti 118–120

## F

Function 167–169  
    built-in function 170, 171  
    user-defined function 172–174  
    pernyataan 176, 179

## I

Integriti 31

## K

Kekardinalan 123–126  
Kekunci asing 121  
Kekunci primer 121  
Kerahsiaan 31  
Kriptografi 30

## L

Linear search 63–65, 73–76

## M

Maklumat 114

## N

Nyahsulit (*Decryption*) 33

## P

Pangkalan data 114, 116  
    membina pangkalan data 127–138  
Parameter 169, 172, 174, 176, 179  
Pembangunan atur cara 2–3  
    analisis masalah 5, 20  
    reka bentuk atur cara 5, 20  
    pengekodan 7, 21  
    pengujian dan penyahpejijatan 8, 22  
    dokumentasi 13, 23  
Pemikiran komputasional 4  
Pengesahan 31  
Penyulitan (*Encryption*) 33  
Pigpen Cipher 37, 46  
Procedure 167–169, 179  
    pernyataan 179–180,

## Q

Query 117  
    membina query 150–155

## R

Ralat 87–97  
    sintaks 3, 9, 190  
    masa larian 3, 9, 195  
    logik 3, 9, 197  
Rail Fence Cipher 43, 46  
Reverse cipher 34, 46

## S

Search 62, 90, 101–105  
Sort 69, 95, 99, 101–105  
SQL 117, 150–153, 154–157  
Substitution cipher 34, 46

## T

Teknik leraian 5–14, 100  
Teknik pengecaman corak 5–14, 98, 99, 100  
Teknik peniskalaan 6–14, 101  
Teknik pengitlakan 6–14, 101  
Tiada sangkalan 31  
Transposition cipher 39, 46

# RUJUKAN

Roffey, C., 2013. Coding Club Python Basics. Cambridge: Cambridge University Press.

Roffey, C., 2013. Coding Club Python: Next Steps. Cambridge: Cambridge University Press.

Cormen, T. H. and Leiserson, C. E., July 31, 2009. Introduction to Algorithms 3rd Edition. MIT Press.

Dasgupta, S. and Papadimitriou, C., Sept. 13, 2006. Algorithms. McGraw Hill.

Harel, D. and Feldman, Y., 1987. Algorithmics The Spirit of Computing 3rd Edition. Addison Wesley.

Dorling, M. and Rouse, G., 2014. Compute-IT 1: Computing for KS3. London: Hodder Education.

Dorling, M. and Rouse, G., 2014. Compute-IT 2: Computing for KS3. London: Hodder Education.

Mel, H.X. and Baker, D., 2001. Cryptography Decrypted. Addison Wesley.

Chai, I. and White J. D., 2004. Structuring Data & Building Algorithm. McGraw Hill (Malaysia) Sdn. Bhd.

Berman, K.A. and Paul, J.L., 2005. Algorithm: Sequential, Parallel and Distributed. Thomson Course Technology.

Omar Z. dan Laiha, M.K., 2002. Pengenalan kepada Keselamatan Komputer. McGraw Hill (Malaysia) Sdn. Bhd.

"Python for Beginners," dicapai pada 2 Januari 2018, <https://www.python.org/about/gettingstarted/>

Sedgewick, R. and Wayne, K., 2011. Algorithms 4th Edition. Pearson Education Inc.

"The Python Tutorial," dicapai pada 21 Disember 2017, <https://docs.python.org/3.5/tutorial/index.html>

Chun, W. J., 2001. Core Python Programming. Prentice Hall PTR.

Dengan ini **SAYA BERJANJI** akan menjaga buku ini dengan baiknya dan bertanggungjawab atas kehilangannya serta mengembalikannya kepada pihak sekolah pada tarikh yang ditetapkan

### Skim Pinjaman Buku Teks

Sekolah \_\_\_\_\_

| Tahun | Tingkatan | Nama Penerima | Tarikh Terima |
|-------|-----------|---------------|---------------|
|       |           |               |               |
|       |           |               |               |
|       |           |               |               |
|       |           |               |               |
|       |           |               |               |
|       |           |               |               |

Nombor Perolehan: \_\_\_\_\_

Tarikh Penerimaan: \_\_\_\_\_

**BUKU INI TIDAK BOLEH DIJUAL**

**RM 12.45**

ISBN 978-967-466-319-3



9 789674 663193 >

**FT303001**