

Pemikiran komputasional ialah satu proses pemikiran bagi tujuan menyelesaikan masalah oleh manusia sendiri berbantuan mesin atau kedua-duanya sekali dengan menggunakan konsep asas sains komputer.

Proses Pemikiran Komputasional

Kemahiran yang diperlukan :

- a. Kemahiran berfikiran secara logik
- b. Kemahiran membina algoritma

**Masalah
dipecahkan**

**Kenalpasti corak
yang sama**

**Perkara tidak
penting
ditinggalkan**

**Sediakan 1 model
penyelesaian
masalah**

4 TEKNIKASAS DALAM PEMIKIRAN KOMPUTASIONAL

Leraian (Decomposition)

- Pemecahan suatu masalah atau sistem yg kompleks kpd bahagian kecil bagi memudahkan pemahaman dan penyelesaian.
- Bahagian- bahagian kecil boleh diteliti, diselesaikan atau direka secara berasingan akan membolehkan masalah yang besar dapat diselesaikan dengan mudah

Pengecaman corak (Pattern Recognition)

- Kenal pasti corak tertentu yang terdiri daripada kesamaan atau ciri-ciri yang sama untuk masalah yang lebih kecil dapat membantu menyelesaikan masalah kompleks dengan lebih berkesan

Peniskalaan (Abstraction)

- Meninggalkan aspek kurang penting yang terdapat dalam corak-corak yang dicamkan dan memfokus kepada aspek-aspek yang penting.

Pengitlakan (Generalisation)

- Pembinaan model dalam bentuk formula, Teknik , peraturan atau langkah-langkah bagi menyelesaikan masalah.

1.1.1 Menggunakan konsep pemikiran komputasional dalam fasa pembangunan atur cara bagi membangunkan atur cara aritmetik .

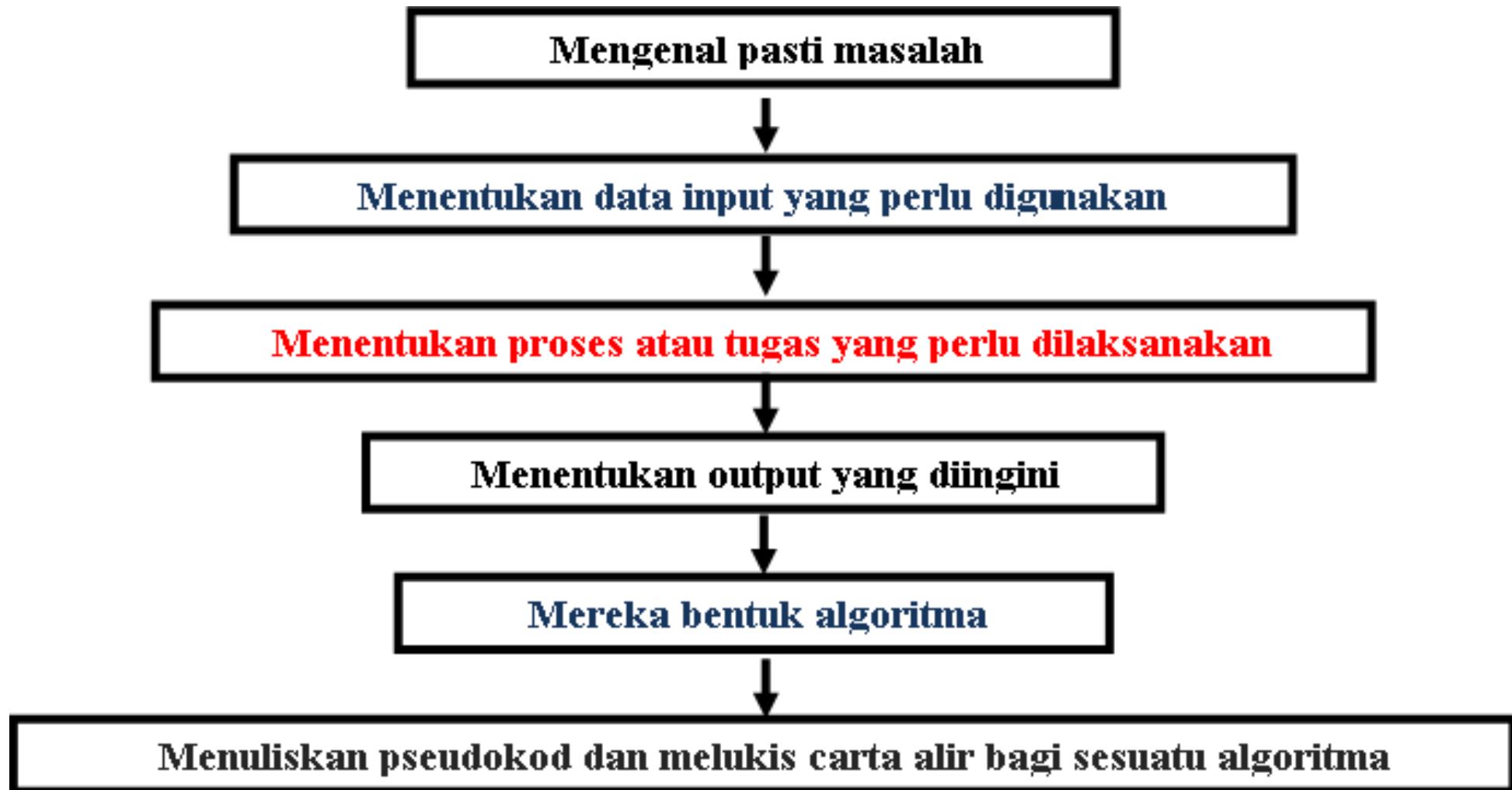
(i)Fasa Analisis Masalah

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

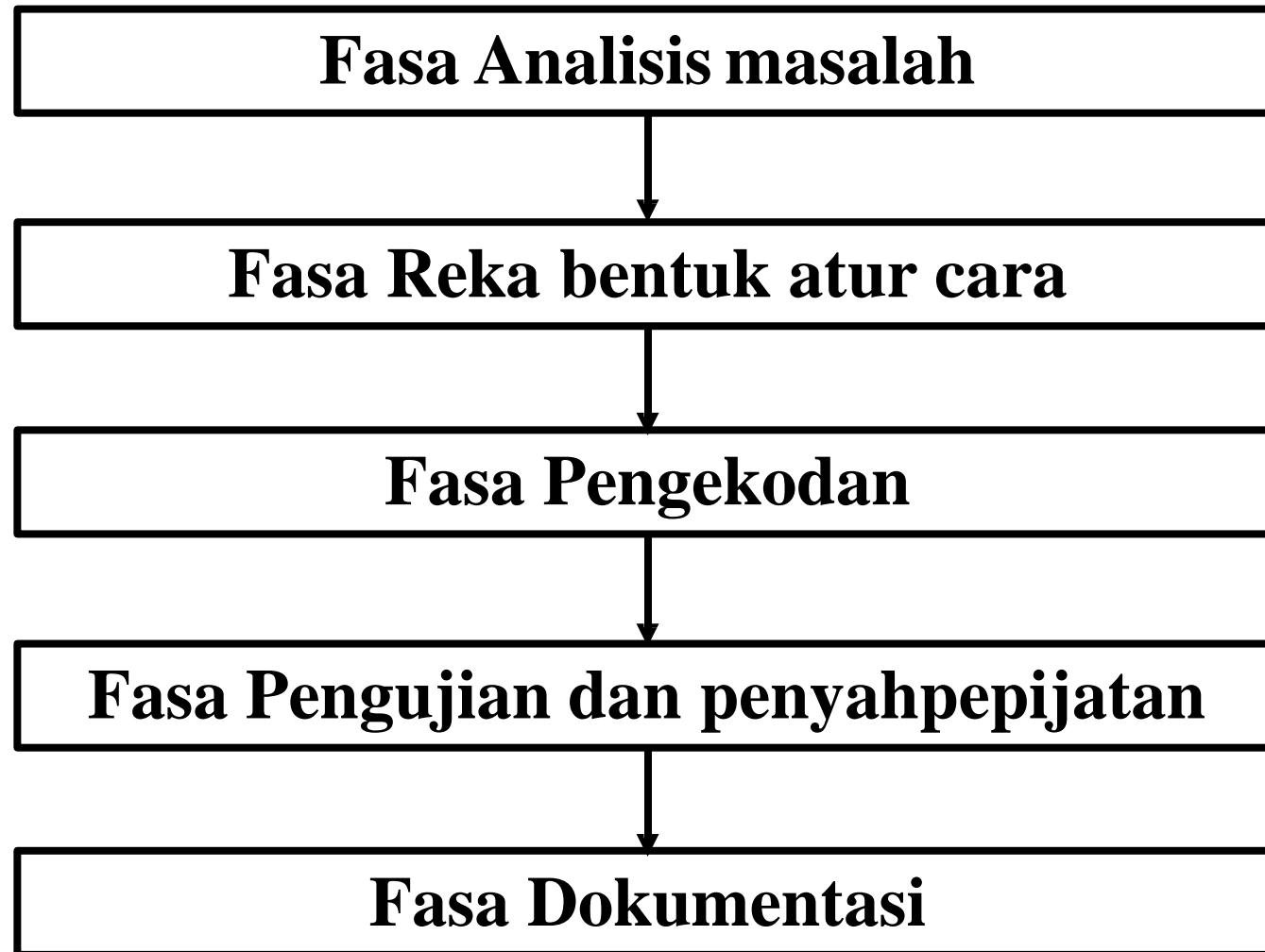
(ii) Fasa Reka bentuk Atur Cara

- a) Menulis pseudokod.
- b) Melukis carta alir.
- c) Mereka bentuk antara muka pengguna.

LANGKAH-LANGKAH MENGHASILKAN PSEUDOKOD DAN CARTA ALIR BAGI SESUATU ALGORITMA



Fasa Pembangunan Atur Cara



(iii)Fasa Pengekodan

- a) Memilih bahasa pengaturcaraan (programming language) : Python dan perisian pengaturcaraan (Software) Pyscripter dan IDLE.
- b) Mengekod atur cara berdasarkan pseudokod dan atur cara.



Penterjemah (*translator*)

Penghimpuan (*assembly*)

Basic Assembly Language yang digunakan oleh sistem IBM .

Low-level languages those languages which are extremely close to machine language. They are also known as Assembly languages. The closest languages after Assembly to Machine language are C and C++.

Pengkompil (*compilier*)

Bahasa pengaturcaraan : C, C++, Java dan Pascal

Translator high-level programming language to lower-level language to create an executable program

Pentaksir (*interpreter*)

Bahasa pengaturcaraan : Python, PHP , Perl dan Ruby

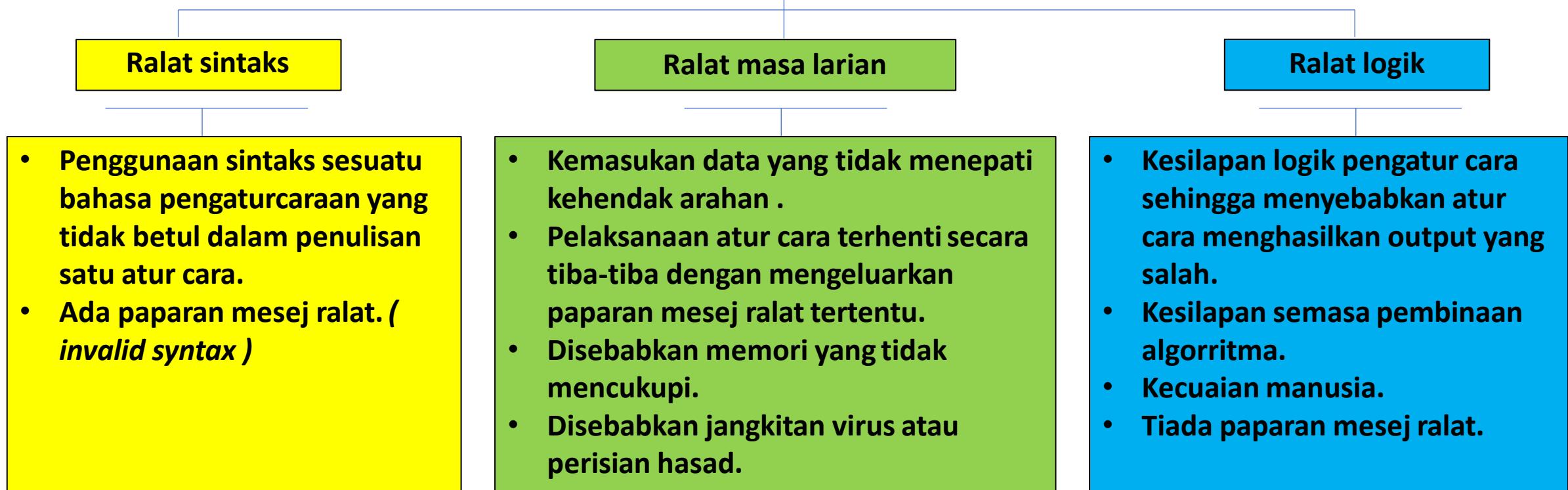
- Proses mengalihkan kod arahan kepada bahasa mesin yang difahami oleh komputer.

(iv) Fasa Pengujian Atur Cara dan Penyahpepijatan Ralat

- a) Menguji atur cara yang telah ditulis bagi mengesan dan membetulkan sebarang ralat yang terdapat pada atur cara.
- b) Mengenal pasti ralat-ralat yang ada iaitu **Ralat Sintaks (Syntax error)** , **Ralat masa larian (runtime error)** , **Ralat logik (logical error)**

- > **RALAT – kesilapan** atau **kesalahan** yang menyebabkan sesuatu algoritma atau atur cara tidak berfungsi seperti yang diharapkan atau mengeluarkan **output yang salah**.

RALAT



- > **kesilapan** atau **kesalahan** wujud dalam atur cara sering dirujuk sebagai **pepijat**.
- > **Proses mencari dan menghapuskan ralat** dipanggil **penyahpepijatan**.



Ralat = kesilapan atau kesalahan dalam pembangunan aturcara atau segmen kod.

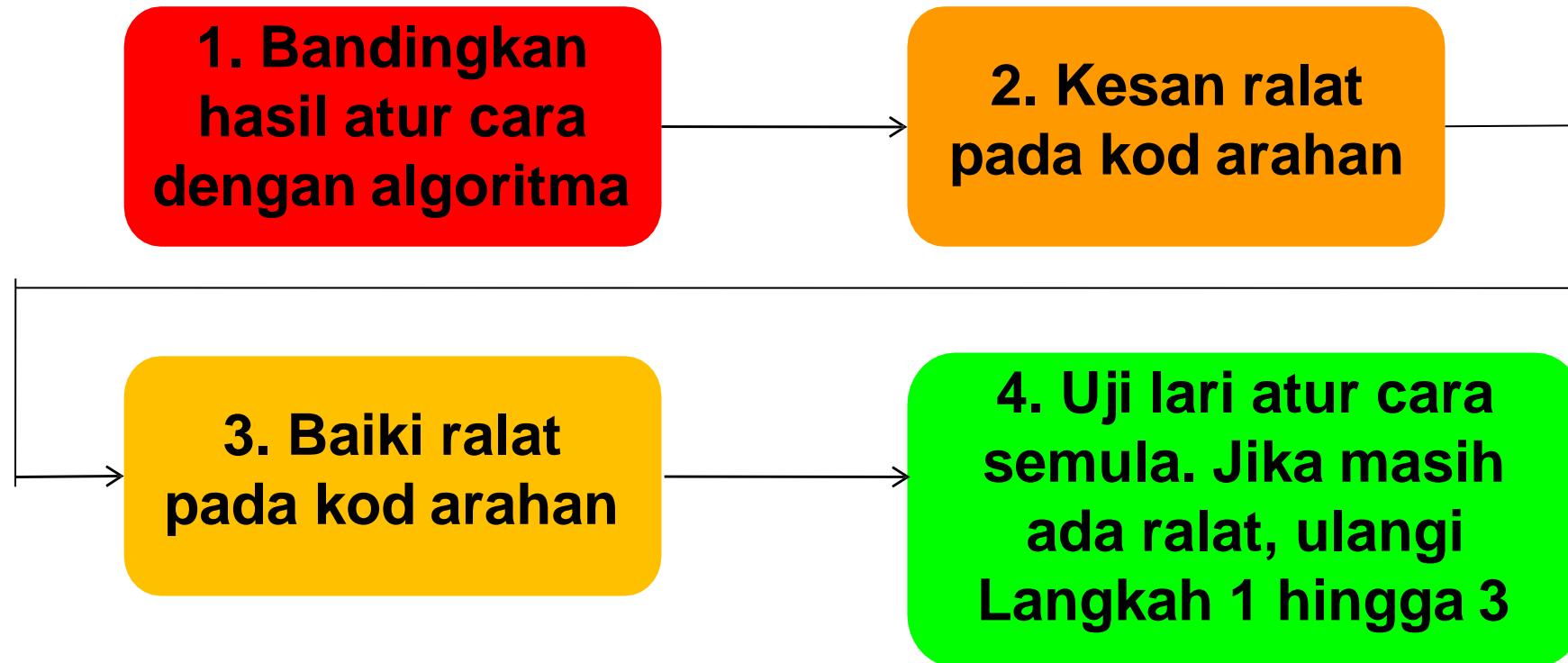
• **Ralat masa larian** - segmen kod aturcara berhenti secara tiba-tiba **disebabkan memori tidak mencukupi , jangkitan virus atau perisian hasad.**

• **Ralat logik** - disebabkan kecuaian pengaturcara seperti kesalahan menggunakan operator. mengganggu perlaksanaan segmen kod tetapi akan mengeluarkan output yang salah.

- - a. Penggunaan sintaks yang tidak betul,
 - b. Perkataan sintaks / arahan dieja salah
 - c. Tertinggal simbol seperti titik bertindih, tanda petikan atau tanda kurungan,
 - d. Cara penggunaan sintaks yang tidak betul seperti meletak sintaks di tempat yang salah.
 - e. Penggunaan inden yang salah.

- - a. Pembahagian dengan sifar,
 - b. Penambahan pemboleh ubah yang berlainan jenis data.
 - c. Tidak mentakrifkan pemboleh ubah, fungsi, modul atau objek sebelum digunakan
 - d. Mengakses nilai pustaka yang tidak wujud atau objek atribut yang tidak wujud.
 - e. Mengakses fail yang tidak wujud.

- - a. Pemanggilan pemboleh ubah yang salah.
 - b. Penggunaan nombor bulat atau nombor perpuluhan yang tidak betul atau bukan di tempat sepatutnya.
 - c. Penggunaan operator pengendali yang salah.
 - d. Kesilapan dalam ungkapan Boolean.
 - e. Penggunaan ruang jarak yang salah.



Langkah-langkah menguji dan menyahpepijatan

A . Pengujian dan penyahpepijatan secara manual

**a. Semakan kesesuaian
nama boleh 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.**

Langkah-langkah menguji dan menyahpepijatan

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.

Semakan kualiti

Teknik semakan meja (desk check)

Semakan meja ialah satu teknik manual untuk menyemak logik setiap langkah serta logik aliran langkah-langkah algoritma.

Satu set sampel data input digunakan dan memproses data itu secara manual mengikut aliran langkah-langkah algoritma.

Teknik langkah demi langkah (step through)

Teknik langkah demi langkah ialah satu teknik manual untuk memeriksa logik aliran algoritma secara langkah demi langkah

Sampel data input tidak digunakan dan memproses data itu secara manual mengikut aliran langkah-langkah algoritma.

(v) Fasa Dokumentasi

a) Menyediakan dokumentasi dalaman dan dokumentasi luaran .

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

Algoritma merupakan set langkah atau prosedur dan urutan tindakan untuk menyelesaikan masalah .

Kaedah Algoritma	Penerangan
a) Pseudokod	Satu set aturan yang ditulis dalam bahasa pertuturan dan beberapa kod asas untuk menyelesaikan sesuatu masalah. Set aturan ini menunjukkan langkah demi langkah untuk melaksanakan sesuatu tugas dari mula hingga tamat.
b) Carta Alir	Kaedah grafik yang menunjukkan langkah-langkah bagi penyelesaian masalah dan hubung kait antara satu sama lain. Aktiviti ini akan dilaksanakan mengikut turutan dari atas ke bawah menggunakan carta alir. Carta alir mengandungi simbol geometrik yang dihubungkan menggunakan anak panah.

Kod Arah



Satu arahan yang dituliskan dalam bahasa komputer untuk mengarahkan komputer melaksanakan sesuatu tugas .

Segmen Kod



Satu set arahan yang digunakan untuk membina satu atur cara bagi menyelesaikan masalah.

Jadual 1 Simbol-simbol dalam Carta Aliran

Simbol	Makna / Fungsi	Penerangan
	Mula / tamat	Mula dan tamat sesuatu aktiviti
	Proses	Aktiviti atau proses
	Pilihan	Menguji sama ada benar atau salah
	Aliran	Aliran Aktiviti
	Penyambung	Menggabungkan aliran yang berlainan
	Input / output	Operasi Input / Output

1. Struktur Linear – Urutan

2. Struktur Kawalan Pilihan

- > Struktur kawalan Pilihan Tunggal (IF)
- > Struktur kawalan Dwipilihan (Double Selection) (IF-ELSE)
- > Struktur Kawalan Pelbagai Pilihan (Multi Selection) (IF-ELSE-IF)
- > Struktur Kawalan Pilihan bersarang (Nested-IF)

2. Struktur Kawalan Ulangan

- > Struktur kawalan Ulangan UNTUK (For)
- > Struktur kawalan Ulangan SELAGI (While)
- > Struktur Kawalan Ulangan SEHINGGA (REPEAT-UNTIL)



Struktur kawalan ulangan for

Struktur kawalan ulangan while

Persamaan

Pernyataan syarat diuji sebelum pernyataan yang berulang dilaksanakan

Perbezaan

Bilangan ulangan adalah
ditetapkan

Bilangan ulangan **tidak diketahui** dan
ditentukan sama ada syarat dipenuhi
atau tidak



- Bertindak mengikut syarat yang membolehkan atur cara yang lebih mesra pengguna dan bermanfaat dihasilkan .
- Satu syarat akan menentukan satu keputusan melalui satu atau dua atau pelbagai cabang pilihan .
- Membolehkan masalah yang lebih rumit diselesaikan dengan atur cara yang interaktif

Struktur kawalan ulangan

Justifikasi mengapa gunakan struktur kawalan ulangan

Struktur kawalan ulangan for	Struktur kawalan ulangan while
Persamaan	
Pernyataan syarat diuji sebelum pernyataan yang berulang dilaksanakan	
Perbezaan	
Bilangan ulangan adalah ditetapkan	Bilangan ulangan tidak diketahui dan ditentukan sama ada syarat dipenuhi atau tidak

- Membolehkan pengatur cara mengarahkan komputer untuk melaksanakan langkah-langkah tertentu secara berulang kali
- Kod yang ditulis tidak perlu panjang dan memakan ruang.
- Lebih mudah berbanding dengan kod atur cara yang tidak menggunakan gelung.
- Sintaks pemboleh ubah yang bertindak sebagai **Pembilang** untuk struktur dikawal dengan bilangan.
- Bahagian kod yang mengandungi syarat dan langkah-langkah untuk berulang disebut **gelung (loop)**.
- Setiap pusingan ulangan disebut **lelaran (iteration)**

Struktur kawalan UNTUK

Sintaks for

for pemboleh ubah gelung **in** [1, 2, 3, dll.]:
Pernyataan

Fungsi	Tujuan
Tanda [] atau [:]	Mencapai satu aksara atau lebih dalam string
len ()	Menghitung bilangan aksara dalam string
Simbol +	Menggabungkan dua string tanpa ruang kosong
Simbol ,	Menggabungkan dua string dengan satu ruang kosong
Simbol *	Mengulang sting dengan bilangan yang dinyatakan
join ()	Menambahkan aksara di antara setiap aksara

Fungsi yang memanipulasi nilai string dan tujuannya

Tanda [] / [a: b]

Output

```
bintang4.py x Contoh 3.4 ms 81 x String x String [] x
lagu = 'Negaraku'

# mencapai sesuatu aksara dalam string mula dari 0
print(lagu[0])
print(lagu[1])
print(lagu[2])
print(lagu[3])
print(lagu[4])
print(lagu[5])
print(lagu[6])
print(lagu[7])

# print aksara dari aksara ketiga hingga keenam
print("\n" + lagu[3:6])
```

Shell

```
>>> %Run 'String []'
N
e
g
a
r
a
k
u
ara
>>>
```

FUNGSI range ()

range (henti)

- **henti** ialah integer bagi urutan berakhir
- urutan bermula dengan **sifar** sehingga satu digit sebelum integer **henti**
- *Contoh :*
- $\text{range}(3) == [0,1,2]$
maka ulangan sebanyak 3 kali

range (mula ,henti)

- **mula** ialah integer bagi urutan bermula
- **henti** ialah integer bagi urutan berakhir
- urutan bermula dengan **integer mula** sehingga satu digit sebelum integer **henti**
- *Contoh :*
- $\text{range}(3, 7) == [3,4,5,6]$ *maka ulangan sebanyak 4 kali*
 $(7 - 3 = 4)$

range (mula,henti,langkau)

- **mula** ialah integer bagi urutan bermula
- **henti** ialah integer bagi urutan berakhir
- **langkau** ialah integer yang menunjukkan perbezaan antara dua digit berselahan dalam urutan.
- urutan bermula dengan **integer mula** sehingga satu digit sebelum integer **henti**
- *Contoh :*
- $\text{range}(3,10,2) == [3,5,7,9]$ *maka ulangan sebanyak 4 kali*

- Penggunaan fungsi `range()` amat berguna kerana fungsi `range()` menyatakan bilangan lelaran struktur kawalan `for` harus berulang.
- Nombor yang digunakan di dalam fungsi `range()` disebut **parameter**.
- Parameter dalam fungsi `range()` mestilah integer terdiri daripada **nombor positif** atau **nombor negatif** .

OPERATOR DAN PEMBOLEH UBAH



Operator	Penerangan	Contoh a = 5 dan b = 10
<code>==</code>	Jika nilai dua operan adalah sama , maka keadaan menjadi benar (比较a 和b 的value两个是一样)	<code>(a == b)</code> <code>False (a == 5)</code> <code>True</code>
<code>!=</code>	Jika nilai dua operan adalah tidak sama , maka keadaan menjadi benar (比较a 和b 的value两个是不一样)	<code>(a != b)</code> <code>True</code> <code>(a != 5)</code> <code>False</code>
<code>></code>	Jika nilai operan kiri adalah lebih besar daripada nilai operan kanan, maka keadaan menjadi benar 大 > 小	<code>(a > b)</code> <code>False</code> <code>(b > a)</code> <code>True</code>
<code><</code>	Jika nilai operan kiri adalah kurang daripada nilai operan kanan, maka keadaan menjadi benar 小 < 大	<code>(a < b)</code> <code>True</code> <code>(b < a)</code> <code>False</code>
<code>>=</code>	Jika nilai operan kiri adalah lebih besar daripada atau sama dengan nilai operan kanan, maka keadaan menjadi benar 大或相等一样> = 小	<code>(a >= b)</code> <code>False (b >= a)</code> <code>True</code>
<code><=</code>	Jika nilai operan kiri adalah kurang daripada atau sama dengan nilai operan kanan , maka keadaan menjadi benar 小或相等一样 < 大	<code>(a <= b)</code> <code>True</code> <code>(b <= a)</code> <code>False</code>



Operator	Penerangan	Contoh $a = \text{benar}$ dan $b = \text{palsu}$ $d = 5, c = 10$
AND (logik DAN)	Jika nilai dua operan adalah sama , maka keadaan menjadi benar (两的condition一定要TRUE 才会TRUE)	($a \text{ AND } b$) False ($d == 5 \text{ AND } c > 5$) True
OR (logik ATAU)	Jika nilai dua operan adalah tidak sama , maka keadaan menjadi benar (只要其中一个condition TRUE , 就TRUE)	($a \text{ OR } b$) True
NOT (logik TIDAK)	Jika nilai operan kiri adalah lebih besar daripada nilai operan kanan, maka keadaan menjadi benar (相反condition , TRUE 变 FALSE , FALSE 变 TRUE)	NOT (a AND b) 代表 NOT (TRUE AND FALSE) NOT (FALSE) 就是TRUE



Nama Operator	Simbol operator matematik	Simbol operator dalam komputer	Contoh pengiraan dalam atur cara
Tambah	+	+	$88 + 22 = 110$
Tolak	-	-	$42 - 25 = 17$
Darab	x	*	$23 * 8 = 184$
Bahagi	÷	/	$66 / 3 = 22$ $11 / 2 = 5.5$

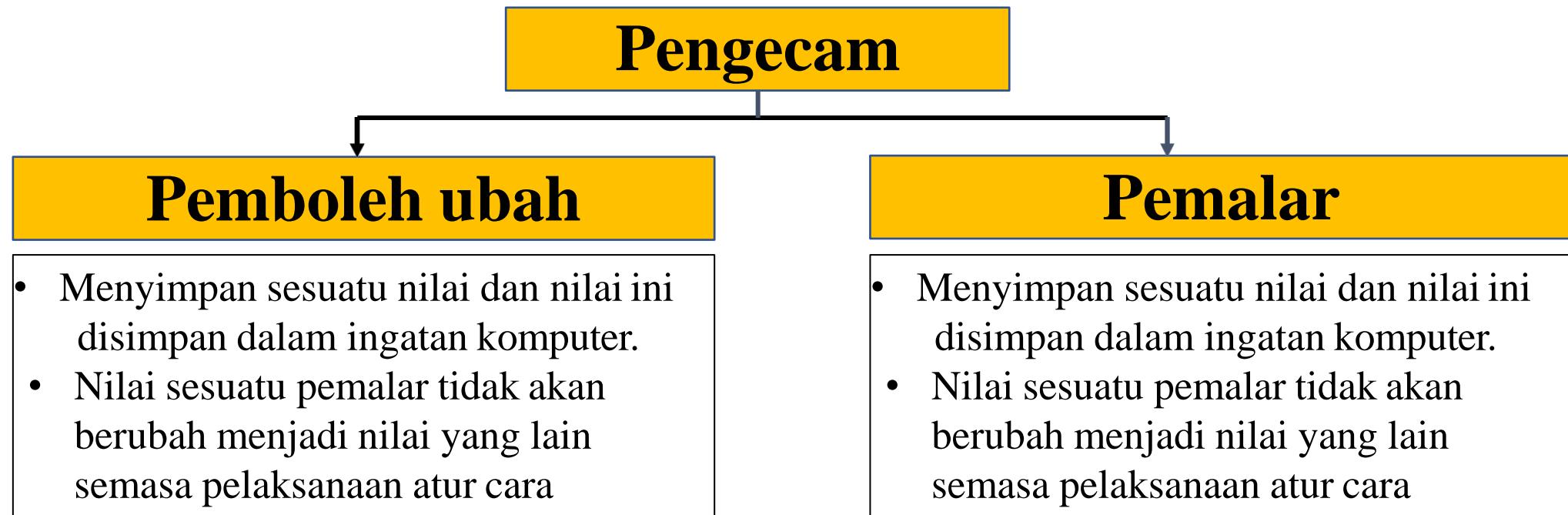
Nama Operator	Simbol operator	Penerangan	Contoh
Modulus	%	Baki pembahagi dipaparkan	$10 \% 2 = 0$ $11 \% 2 = 1$
Eksponen	**	Operator kiri dikuasakan mengikut nilai operan Kanan	$10 ** 3 = 10^3$
Floor Division	//	Hasil Pembahagian tanpa nilai baki .	$10 // 2 = 5$ $11 // 2 = 5$

Floor Division (除出来的平均整)

$$\begin{array}{r} 5 \\ \overline{)11} \\ 10 \\ \hline 1 \end{array}$$

Modulus (余数)

Jenis-jenis Pengecam



nama = str (“ Cikgu Amanda ”)

Pengecam bagi
pemboleh ubah

Jenis data string
digunakan

Nilai yang disimpan
oleh pemboleh ubah



Perbezaan antara boleh ubah dan pemalar

Ciri-ciri	Pemboleh ubah	Pemalar
Perubahan nilai	Mempunyai nilai boleh berubah-ubah	Mempunyai nilai yang tetap dan tidak berubah
Nilai yang perlu diingat	Pengaturcara perlu mengingati label dan nilai yang digunakan	Pengaturcara tidak perlu mengingati label dan nilai yang digunakan
Standard penggunaan nilai	Tiada nilai standard	Nilai yang standard



Jenis data digunakan semasa pengisytiharan suatu boleh ubah untuk :

- membenarkan atur cara mengenal pasti kegunaan boleh ubah tersebut
- membolehkan boleh ubah tersebut disimpan dalam storan ingatan dengan berkesan .

Jenis DATA

Nombor Berangka

Nombor bulat (int)

Contoh : -5 , 0 , 100

Nombor perpuluhan

(double , float)

Contoh : 0.09 , -4.5 ,
45.1

Teks

Aksara (char)

Contoh :

‘1’ , ‘2’ , ‘A’ , ‘B’ , ‘z’ , ‘y’ , ‘%’ , ‘@’

Rentetan (string)

Contoh :

“Malaysia”

“Saya suka ASK”

Nilai Logik (Boolean)

BENAR

- Dipapar sebagai **true** dalam aturcara

PALSU

-Dipapar sebagai **false** dalam aturcara

Contoh	Jenis Data	Jenis data yang menyimpan
bilPelajar = 30	integer	Nombor bulat
gaji = 1500.60	float	Nombor perpuluhan
sudahBayar = False	boolean	true atau false
gred ='A'	string	Satu aksara tetap dianggap string dalam python.
nama = "Siti Aisyah"	string	gabungan aksara-aksra

- **Pboleh ubah** ialah storan ingatan dalam komputer yang digunakan oleh atur cara . Cth :

Nama

- Digunakan untuk mewakili sesuatu data yang perlu digunakan berulang kali oleh atur cara yang dibangunkan.
- Pboleh ubah yang diisytiharkan akan diberikan label yang disebut **pengecam**.

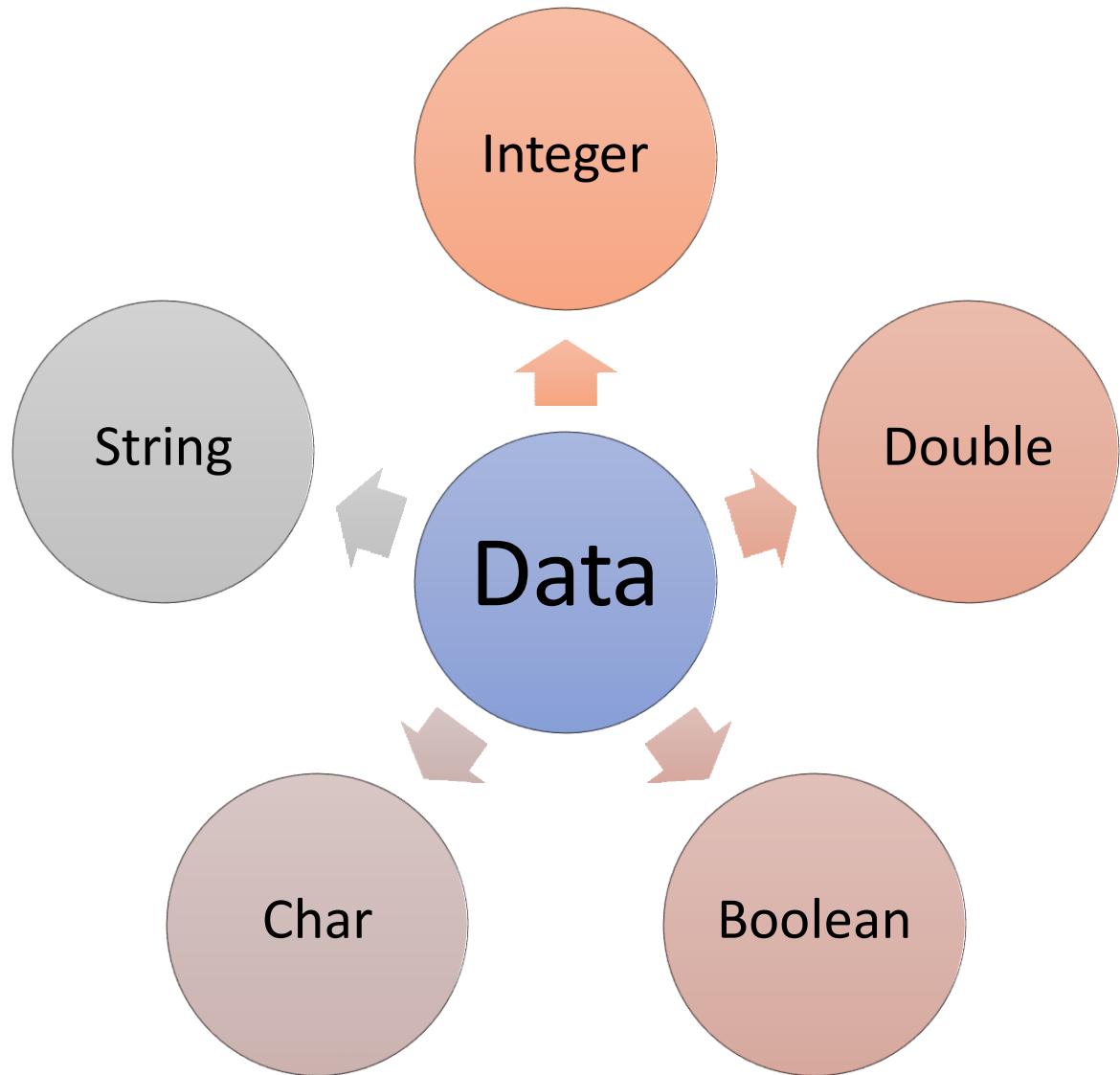
(有 assign value 的 pboleh ubah 叫 pengecam)

Pengecam 2 jenis :

Pboleh ubah , cth : Nama = “ AMANDA LAI ”

Pemalar , cth : Pi = 3.142

3.1.1 Jenis data dalam segmen kod



Jenis data digunakan semasa pengisytiharan suatu pemboleh ubah untuk membenarkan atur cara mengenal pasti kegunaan pemboleh ubah tersebut dan membolehkan pemboleh ubah tersebut disimpan dalam storan ingatan dengan berkesan .



Data	Ciri –ciri	Jenis	Contoh
Integer	<ul style="list-style-type: none">• Nombor tidak mengandungi pecahan dan nombor perpuluhan .• Nombor bulat	Nombor Positif	1,2,3,4,5
		Sifar	0
		Nombor Negatif	-1,-2,-3,-4,-5
Double	<ul style="list-style-type: none">• Jenis data mewakili nombor yang mengandungi nilai nombor pecahan atau nombor perpuluhan .	Double -64 bit	0.1234567890123
		Float – 32 bit	0.12345678
Char	<ul style="list-style-type: none">• Jenis data mewakili aksara (character) .• Aksara terdiri daripada huruf , abjad, angka, digit, simbol berdasarkan Kod Piawai ASCII	Huruf	A,B,C,D,E,F,G.....Z
		Digit	0,1,2,3,4,5,6,7,8,9
		Simbol	!, @ , # , \$, % , ^ , & ?
String	<ul style="list-style-type: none">• Satu perkataan yang dibina daripada aksara.• Terdiri daripada satu urutan aksara-aksara yang berjujukan yang membentuk rentetan (string).• Nilai string kepada sesuatu pemboleh ubah perlu diletakkan dengan , Petikan tunggal (‘ ’) Petikan berganda (“ ”)		‘ASK’ “Mari Belajar ASK ”
Boolean	<ul style="list-style-type: none">• Untuk membandingkan nilai sebelah kiri dan sebelah kanan .• Hubungan hasil logik iaitu “BENAR ” atau “PALSU ”• Digunakan dalam ungkapan Boolean untuk pernyataan bersyarat untuk struktur kawalan pilihan dan struktur kawalan ulangan .		Username == “MEB2101” a < 10 b != ‘1’

PERWAKILAN

DATA

BAB 2 PERWAKILAN DATA – SISTEM NOMBOR

Nombor Decimal	Nombor Binari	Nombor Octal	Nombor Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

	Sistem Nombor Perpuluhan	Sistem Nombor Perduaan	Sistem Nombor Perlapanan	Sistem Nombor Perenambelasan
Dikenali sebagai	Sistem Asas 10	Sistem Asas 2	Sistem Asas 8	Sistem Asas 16
Sistem Nombor	Sistem nombor desimal	Sistem nombor binari	Sistem nombor oktal	Sistem nombor heksadesimal
Nombor ditandakan dengan subskrip	subskrip 10	Subskrip 2	Subskrip 8	Subskrip 16
Contoh	58₁₀	11100₂	112₈	8AF₁₆
Pilihan digit	0,1,2,3,4,5,6,7,8,9	0,1	0,1,2,3,4,5,6,7	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
Nilai Tempat	1,10,100,1000	1,2,4,8,16,32,64,128....	1,8,64,512,1024.....	1,16,256,4096...

$$x_2 \rightarrow x_{10}$$

$$\begin{array}{r}
 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 \hline
 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 \hline
 128 + 0 + 32 + 0 + 8 + 0 & = 168
 \end{array}$$

$$x_{8/16} \rightarrow x_2$$

$$\begin{array}{r}
 5 | 7 | 3 \\
 \hline
 101 & 111 & 011 \\
 \hline
 \end{array} \rightarrow$$

$$x_2 \rightarrow x_8/x_{16}$$

$$\begin{array}{r}
 011 | 001 \\
 \hline
 1 \\
 \hline
 \end{array}$$

or (kaedah pengiraan)

$$x_{8/16} \rightarrow x_{10}$$

$$\begin{array}{c|c|c|c}
 8^3 & 8^2 & 8^1 & 8^0 \\
 \hline
 512 & 64 & 8 & 1 \\
 \hline
 5 & 6 & 3 & 0
 \end{array}$$

$$x_{10} \rightarrow x_{2/8/16}$$

$$\begin{array}{c}
 2/8/16 [x_{10}] \text{ baki } x \uparrow \\
 \hline
 0
 \end{array}$$

Kepentingan Sistem nombor perenambelasan :

- Mewakili **bahasa mesin selain nombor perduaan** .
- Digunakan untuk **pembangunan asas sistem mikropemproses**.
- Menunjukkan **ralat tentang lokasi ingatan komputer** . Ini memudahkan pembangunan sistem mengenal pasti dan membetulkan ralat tersebut.
- Penting **untuk mewakili warna pada alat digital dalam model warna RGB**
- Membolehkan komputer menyimpan data dengan efisien .

Hubungkait Nombor Perenambelasan heksadesimal dengan kod ASCII dalam bahasa mesin

- Komputer hanya memahami **bahasa mesin yang terdiri daripada nilai 1 dan 0 sahaja**. Setiap aksara di papan kekunci diwakili oleh satu rentetan nombor yang terdiri daripada digit 1 dan 0.
- **Pengekodan ASCII** adalah bahasa penterjemahan yang dapat menterjemahkan bahasa mesin yang boleh difahami oleh manusia. Skema pengekodan terdiri **daripada set gabungan 0 dan 1 yang berbeza** dalam **1 bait** boleh mewakili sehingga 256 aksara yang berbeza.
- Sistem piawai ASCII **membolehkan pelbagai komponen komputer beroperasi dengan betul dalam ses sebuah komputer**.
- **Kegunaan utama nombor perenambelasan ialah mewakili nombor perduaan dalam bentuk lebih pendek, lebih mudah dibaca dan diingati oleh manusia**.
- Mengapakah nombor perenambelasan dipilih ?

Nombor perenambelasan dipilih **kerana proses penukaran daripada nombor perduaan lebih mudah**.

KOD ASCII (Nombor Perpuluhan)	Nombor Perduaan	Aksara
48	0010 0000	0
65	0100 0001	A
97	0110 0001	a

Kepentingan Kod ASCII

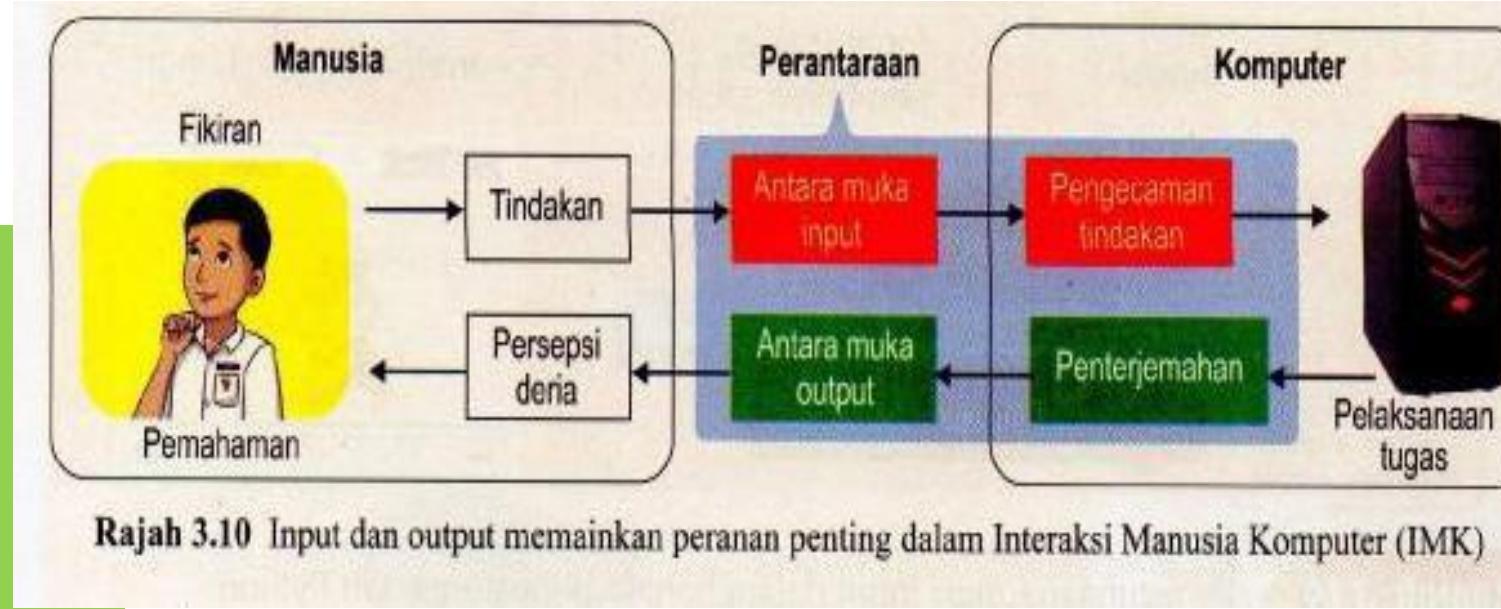
1. ASCII membolehkan manusia berinteraksi dengan sebuah komputer.
2. ASCII membolehkan pengeluar-pengeluar komponen komputer mengeluarkan komponen yang boleh beroperasi dengan betul di dalam sebuah komputer.
3. ASCII juga berfungsi untuk mencapai kesesuaian di antara pelbagai jenis peralatan pemprosesan data. Seterusnya ia membolehkan peralatan ini berkomunikasi di antara satu sama lain dengan sempurna.

3.1.3 (i) Menghasilkan segmen kod menggunakan fungsi input dan fungsi output



- Fungsi input dan fungsi output merupakan kaedah interaktif yang digunakan dalam **Interaksi Manusia Komputer (IMK)**

- Membolehkan mesej yang hendak disampaikan dan diterima dapat difahami oleh pengguna juga komputer.
- IMK melibatkan cara manusia berkomunikasi dengan komputer dengan ciri-ciri perantaraan dalam aspek yang lebih luas.



Rajah 3.10 Input dan output memainkan peranan penting dalam Interaksi Manusia Komputer (IMK)

Fungsi Input - untuk mendapatkan **data input daripada pengguna**

Fungsi Output – merujuk kepada maklumat yang **dipaparkan pada skrin komputer**

Interaksi Manusia Komputer (IMK)

Faktor komputer

- Teknik grafik komputer
- Sistem pengoperasian
- Bahasa pengaturcaraan
- Persekuturan pembangunan

Faktor Manusia

- Teori komunikasi
- Bahasa
- Sains Sosial
- Psikologi kognitif
- Psikologi sosial
- Perasaan manusia – kepuasan penggunaan komputer

Perantaraan (Medium)

- Antara muka suatu atur cara – *user interface*
- Cara data dimasukkan
- Reka bentuk grafik yang mesra pengguna

**Atur cara komputer
(bahasa pengaturcaraan)**

Scratch

C++

Java

Visual Basic

Python

Kod-kod atur
cara diterjemah
kepada bahasa
mesin yang
terdiri daripada
nilai 1 dan 0
sahaja oleh
penterjemah

Bahasa Mesin

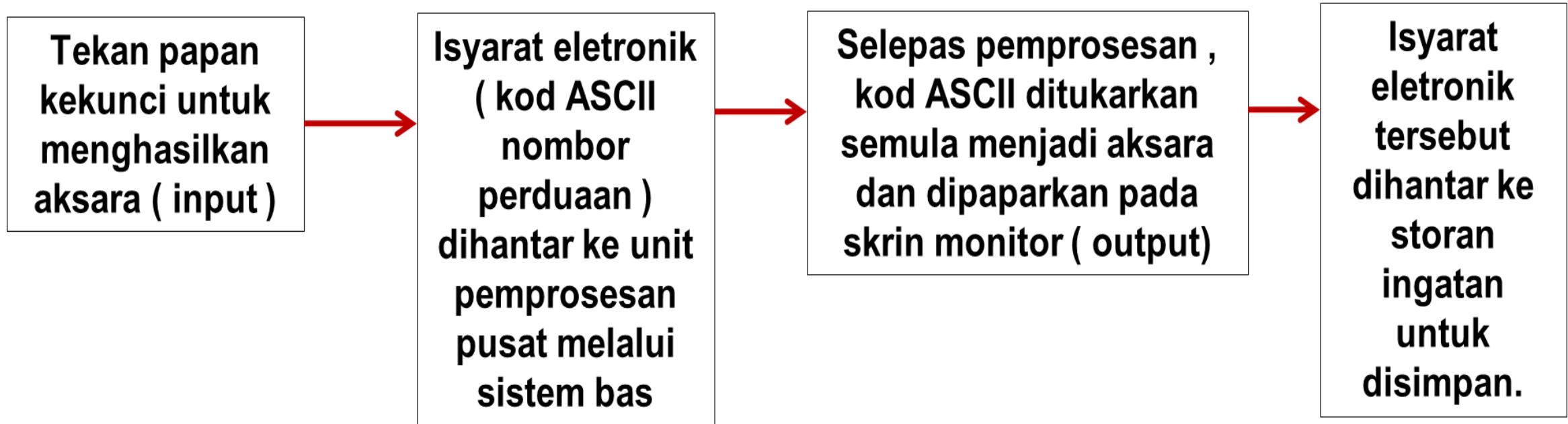
**Komputer hanya
memahami bahasa
mesin yang terdiri
daripada nilai 1 dan
0 sahaja.**

1.1.4 Hubung kait nombor perlapanan dengan kod ASCII dalam bahasa mesin

Bahasa
Mesin

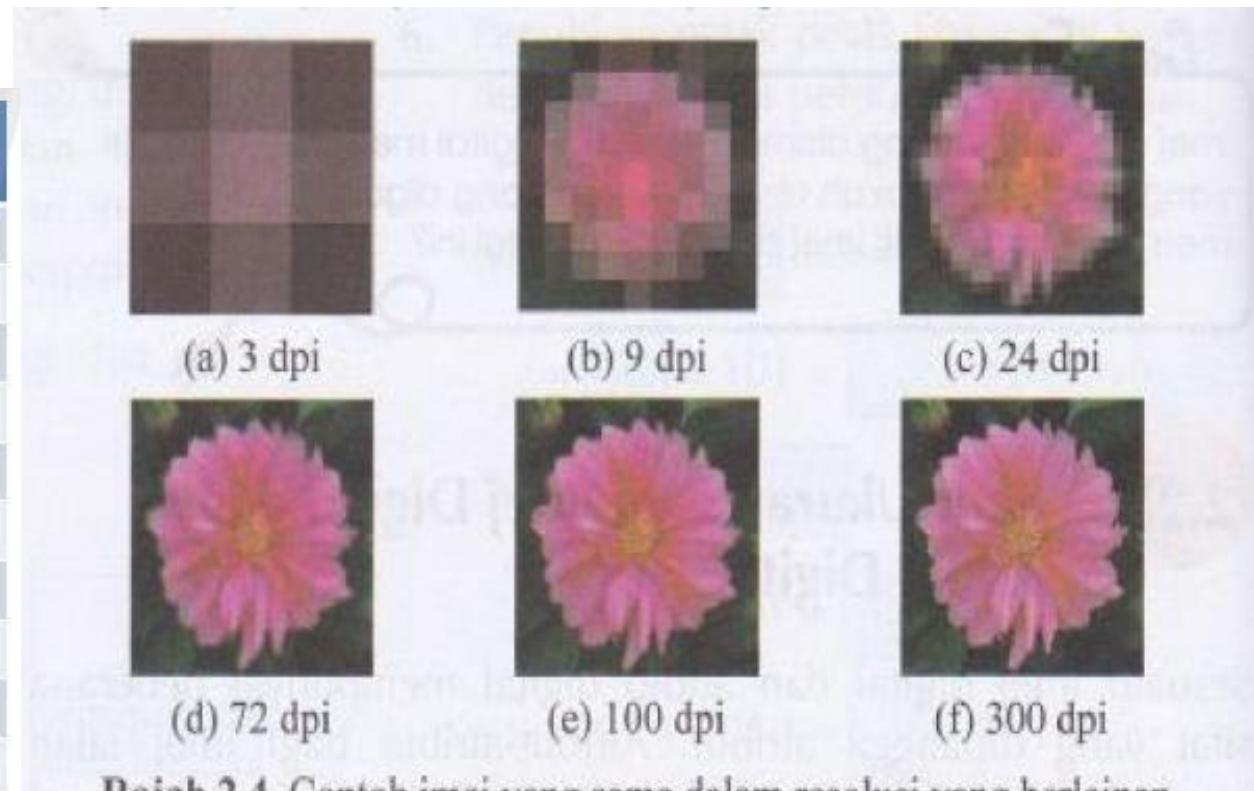
- Kod mesin iaitu bahasa asas yang difahami oleh mesin.
- Terdiri daripada urutan digit 0 dan 1 yang disebut nombor perduaan.
- Atur cara komputer (*program*) boleh ditulis oleh pelbagai bahasa pengaturcaraan (*programming language*).
- Komputer tidak boleh memahami kod-kod bahasa pengaturcaraan secara terus dan perlu diterjemahkan kepada kod mesin dahulu oleh penterjemah (*Translator*) iaitu penghimpun (*assembly*), pengkompil (*compiler*), pentafsir (*interpreter*).

Bagaimanakah satu aksara dari papan kekunci boleh dipaparkan pada skrin monitor ?



AUDIO DAN IMAGE

Kedalaman Warna	Bit bagi setiap piksel	Formula	Julat warna
1 bit	1	2^1	2
2 bit	2	2^2	4
3 bit	3	2^3	8
4 bit	4	2^4	16
5 bit	5	2^5	32
6 bit	6	2^6	64
7 bit	7	2^7	128
8 bit	8	2^8	256
16 bit	16	2^{16}	65 536
24 bit	24	2^{24}	16 777 216



Rajah 2.4 Contoh imej yang sama dalam resolusi yang berlainan

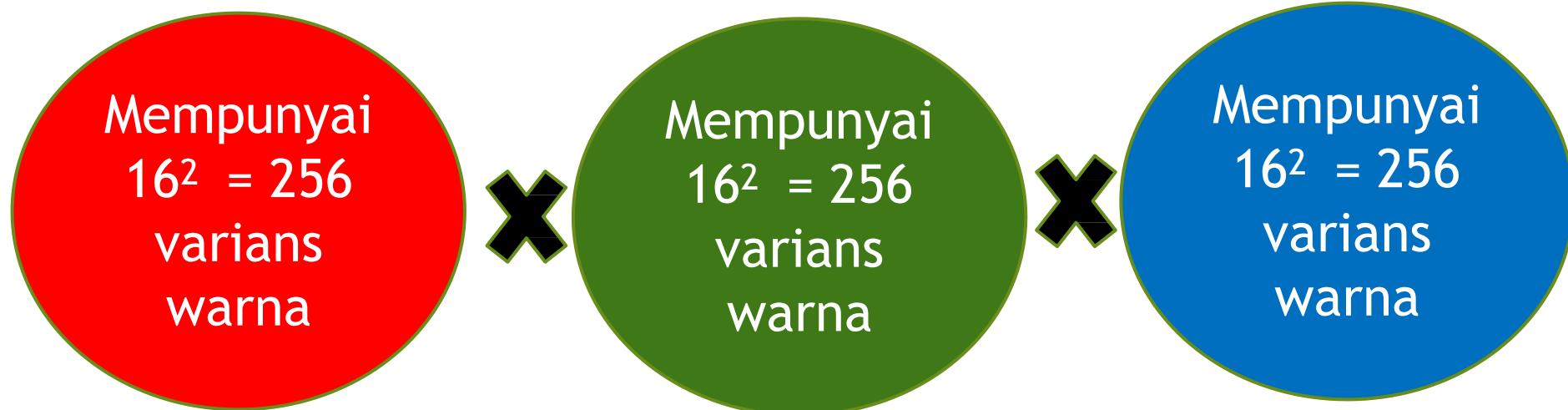
	Resolusi (dot per inch)	Dimensi (piksel)	Kedalaman warna (bit)
Kualiti	Resolusi Tinggi	Dimensi Tinggi	Kedalaman warna tinggi
Saiz Fail Imej	<ul style="list-style-type: none"> Lebih banyak piksel digunakan untuk membentukkan imej. Lebih banyak piksel bermaksud lebih banyak bit dan bait diperlukan untuk menyimpan maklumat tentang piksel-piksel dalam imej . Saiz file menjadi besar 		Lebih tinggi kedalaman bit warna , semakin tinggi piksel per bit (ppb) imej .

Resolusi , dimensi , kedalaman warna lebih tinggi maka lebih tinggi keperincian sesuatu imej.

Kesan : Imej lebih jelas, tajam dan menarik

Saiz Fail : Lebih besar

Jumlah Spektrum warna



Kedalaman bit warna

$$2^8 = 256$$

8 bit mempunyai 256 warna

Saiz fail : 1 warna = 8bit = 1 bait = 1 piksel

Saiz fail Imej berwarna (RGB) : $2^{24} = 24$ bit = 1 piksel x 3 bait





Gambar foto ini mempunyai resolusi **300 dpi** dan berukuran **17inci x 13 inci** . Saiz Imej ?

$$\begin{aligned}\text{Dimensi} &= \text{Panjang} \times \text{Lebar} \\ &= (17 \times 300) \times (13 \times 300) \text{ piksel} \\ &= 5100 \times 3900 \\ &= 19890000 \text{ piksel}\end{aligned}$$

$$\begin{aligned}\text{Saiz fail} &= 19890000 \times 3 \text{ bait} \\ &= 59670000 \text{ bait}\end{aligned}$$

Tips :

1 piksel = 3 bait (3 warna asas)

Resolusi - 300 dpi

Saiz = Dimensi (piksel)

1 piksel = 1 bait (gambar kelabu)

Kadar kedalaman bit
bagi sesuatu
audio digital



Jumlah bilangan bit (0 dan 1)
yang digunakan untuk mewakili
setiap saat masa main balik
(play back time)

Unit ukuran kadar kedalaman bit ialah bit per saat (bps)

$$\text{Kadar Kedalaman bit} = \frac{\text{Kadar sampel (Hz)}}{\text{Kedalaman bit (bit)}} \times \text{Saluran (} 1 = \text{Mono, } 2 = \text{Stereo) }$$

Contoh 2.13 (BT m/s 53)

Satu fail audio mempunyai **kadar sampel 44 100 Hz**, **kedalaman 16 bit** dan **dua saluran audio**. Apakah kadar kedalaman bit bagi fail audio ini ?

Tips :
考试 ukuran unit
没有写直接 0 分
bps

$$\text{Kadar Kedalaman bit} = \frac{\text{Kadar sampel (Hz) } \times \text{Kedalaman bit (bit) } \times \text{Saluran (1 = Mono , 2 = Stereo)}}{}$$

Kadar Kedalaman bit
= 44100 x 16 x 2
= 1 411 200 bit per saat

Saiz fail Audio digital

$$\text{Saiz fail audio} = \boxed{\text{Kadar sampel (Hz)}} \times \boxed{\text{Kedalaman bit (Bait)}} \times \boxed{\text{Bilangan saluran (1 mono) 2 stereo}} \times \boxed{\text{Durasi (saat)}}$$

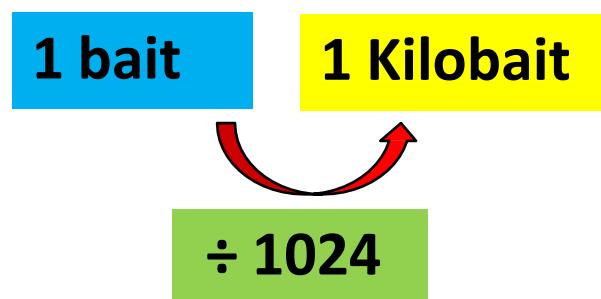
Saiz fail Audio digital

Satu fail audio mempunyai Kedalaman bit = **16 bit** , Saluran = **stereo** , Kadar Sampel = **192 KHz** dan Durasi = **46 minit** .
Berapakah saiz fail Audio ? (Kilobait)

$$\text{Saiz Fail Audio} = \text{Kadar Sampel (Hz)} \times \text{kedalaman bit (Bait)} \times \text{Saluran Audio (1,2)} \times \text{Durasi Audio (Saat)}$$

$$\begin{aligned}\text{Saiz fail audio} &= (192 \times 1000) \text{ Hz} \times (16 \div 8) \times 2 \times (46 \times 60) \text{ bait} \\ &= 192000 \times 2 \times 2 \times 2760 \text{ bait} \\ &= 2119680000 \text{ bait}\end{aligned}$$

$$\begin{aligned}\text{Saiz fail audio} &= 2119680000 \div 1024 \text{ Kilobait} \\ &= 2070000 \text{ Kilobait}\end{aligned}$$



HTML

Melakar papan cerita

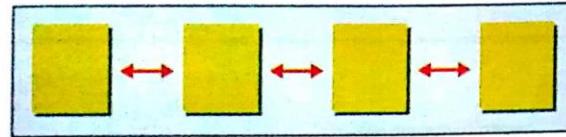
- Papan cerita ialah **satu siri lakaran yang menggambarkan isi kandungan untuk setiap paparan skrin suatu laman sesawang**

- Kebaikan melakar papan cerita :

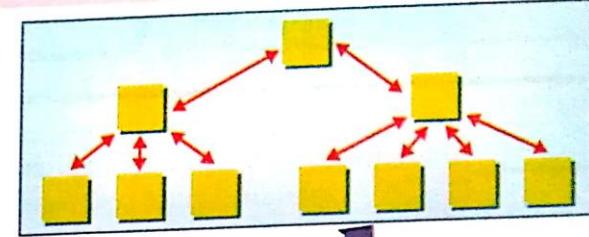
- Mengetahui bilangan laman sesawang yang diperlukan.
- Memberikan penjelasan kasar tentang struktur laman sesawang
- Mengetahui aliran antara laman sesawang.
- Mengetahui maklumat yang harus dimasukkan.

Rajah 4.10 Panduan melakar papan cerita laman sesawang

Struktur linear menghubungkan setiap laman dengan dua pilihan pengemudian iaitu, bergerak ke hadapan atau ke belakang.

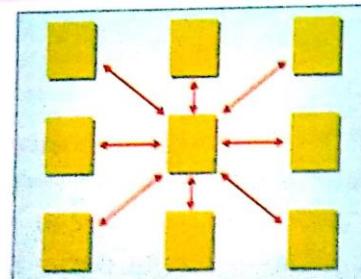


Struktur cabang bersifat dinamik dan berkeupayaan mengemudi ke laman lain dengan mudah. Laman utama dipautkan ke laman-laman lain melalui sistem menu.

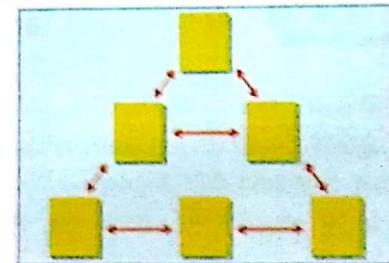


Struktur maklumat dan pautan laman sesawang

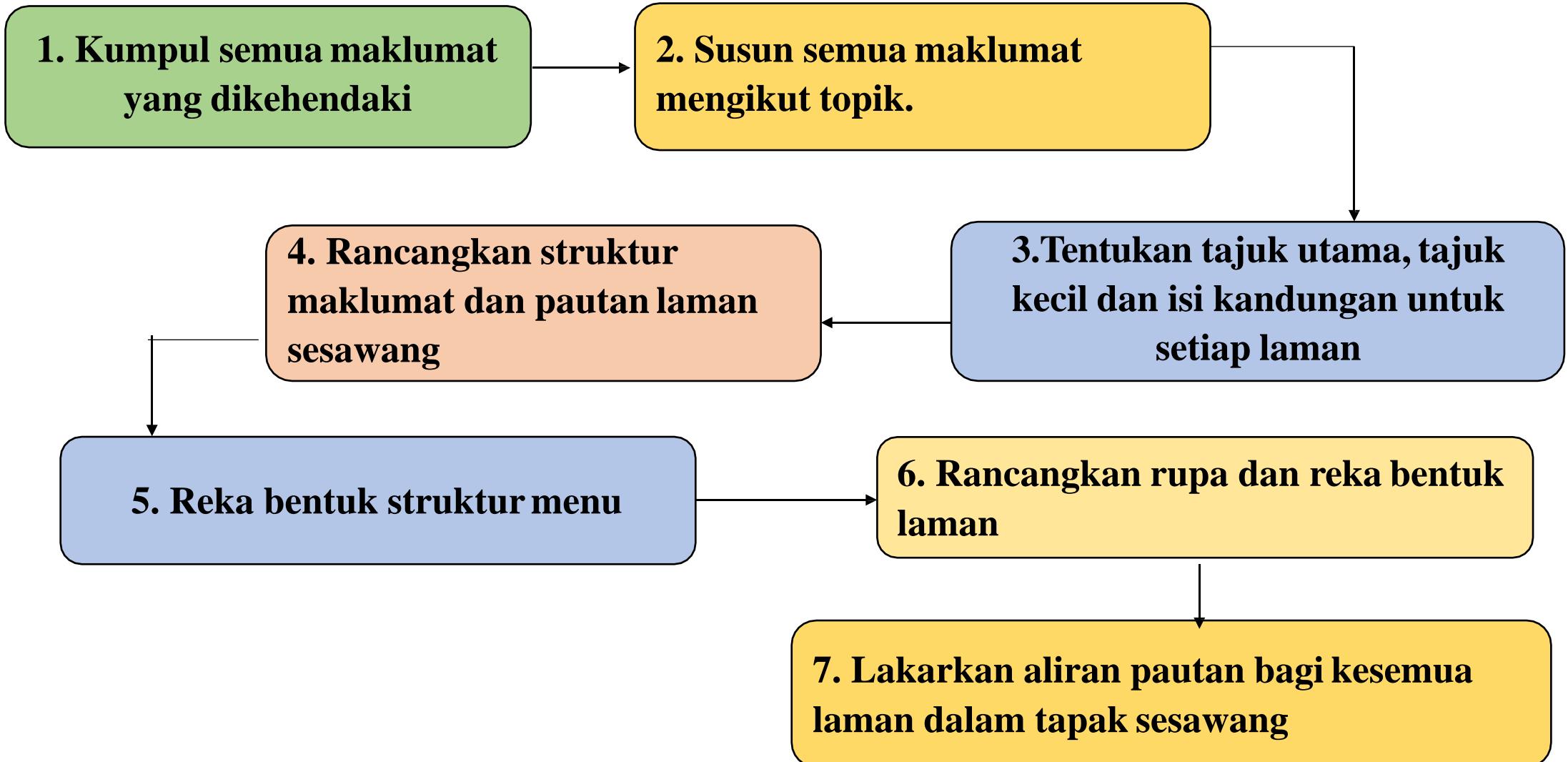
Struktur roda berbentuk seperti ruji basikal. Laman utama dipautkan ke laman-laman lain tetapi pengunjung terpaksa kembali ke laman utama sebelum dipautkan ke laman seterusnya.



Struktur hierarki mempunyai model pengemudian atas ke bawah. Bermula dengan laman utama, pengunjung dipautkan ke laman-laman lain. Pengunjung ada pilihan untuk ke laman-laman seterusnya tanpa kembali ke laman utama.



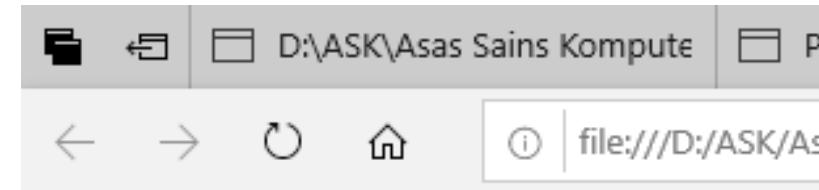
Langkah-langkah melakar papan cerita



Pull down menu

Pull down Menu

```
<html>
<head>
<title> PULL DOWN MENU </title>
</head>
<body>
<p> Pilih Kelas ASK </p>
<select>
<option value = " 3 Red"> 3 Red </option>
<option value = " 3 Orange"> 3 Orange </option>
<option value = " 3 Yellow"> 3 Yellow </option>
<option value = " 3 Green"> 3 Green </option>
</select>
</body>
```



Pilih Kelas ASK

3 Red ▾

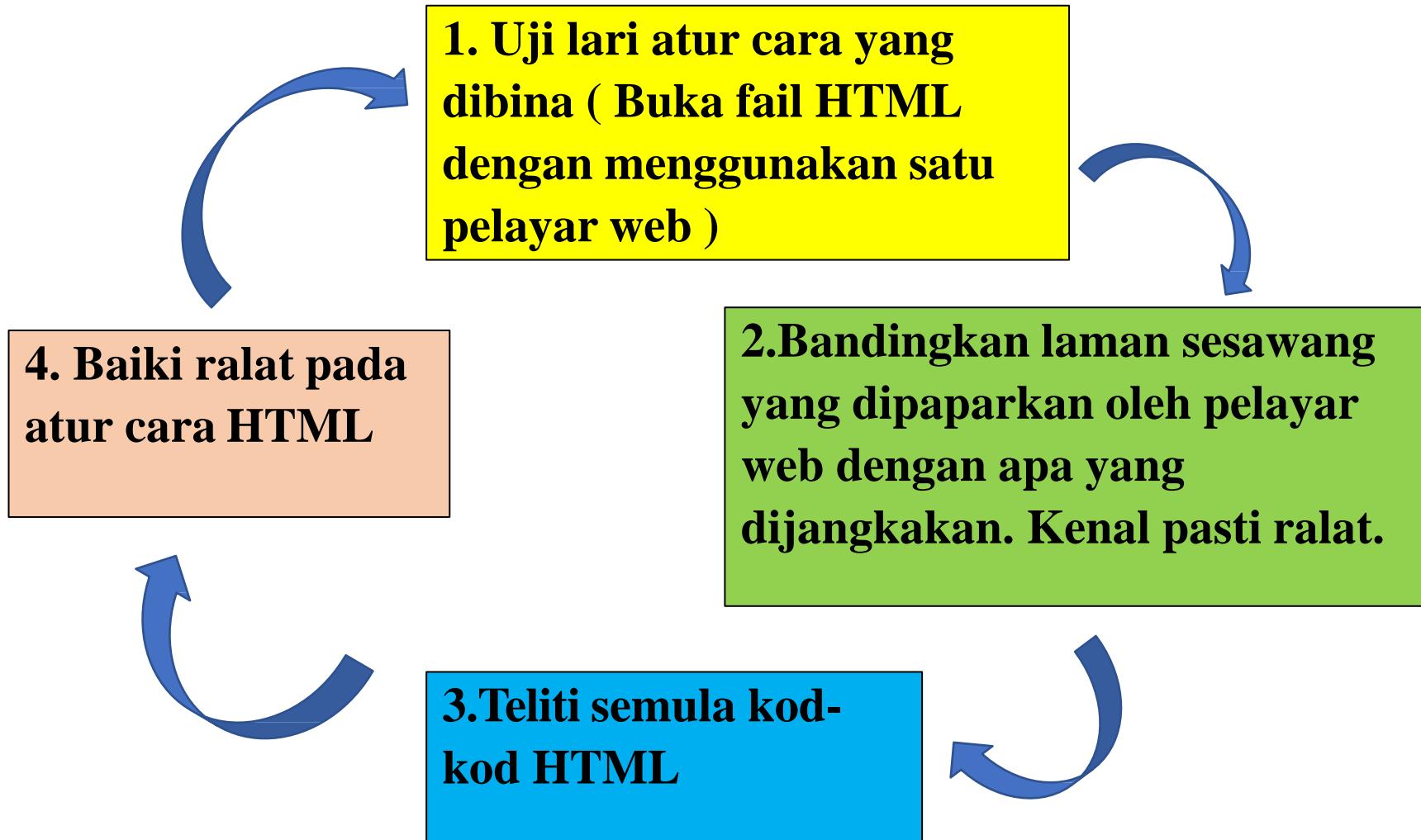


Pilih Kelas ASK

3 Red
3 Orange
3 Yellow
3 Green

Langkah-langkah mengesan ralat dalam atur cara HTML

BT T1 m/s 137



Ralat sintaks :

- Ralat yang biasa dijumpai dalam pengaturcaraan HTML adalah ralat sintaks dan penggunaan tag di tempat yang salah.
- Ralat sintaks – Kesilapan dalam cara kod arahan diulis. Cth : kesalahan ejaan, penggunaan tanda baca yang tidak betul.
- Kesilapan ini akan menyebabkan pelayar web tidak dapat melaksanakan kod-kod atur cara yang dihasilkan.

Tingkatan 3

2.1 Kriptografi dalam keselamatan Data

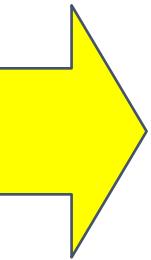
- **Keselamatan data** amat penting pada era siber.
- Salah satu mekanisme keselamatan data ialah **kriptografi**.
- Pelbagai kaedah sifer diwujudkan **untuk menjamin keselamatan data dan memastikan mesej hanya dapat dibaca dan difahami oleh penerima**.
- Kriptografi ialah kajian atau penggunaan **penyulitan data dalam komunikasi dan pemindahan data**, bertujuan supaya **data tidak dapat difahami atau digunakan oleh pihak lain**.

- **Encryption = Penyulitan**
Decryption = Nyahsulit
Plaintext = Teks biasa
Ciphertext = Teks sifer

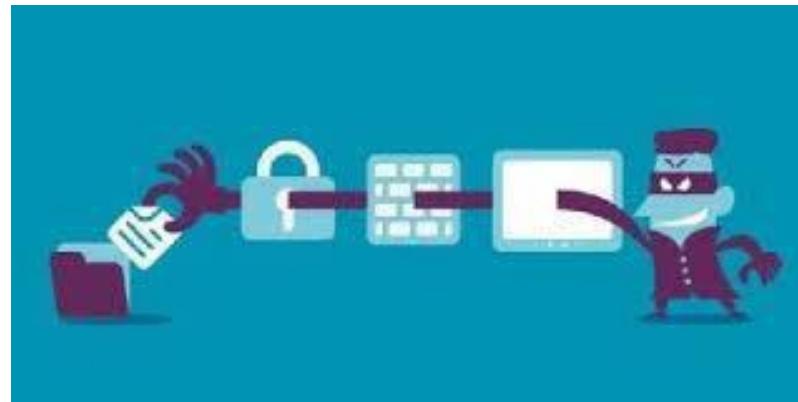


2.1.1 Kriptografi dalam Pengkomputeraan

Sistem
pengkomputeran
digital



Kebanyakan maklumat penting atau sulit individu, masyarakat dan negara disimpan dalam bentuk digital boleh diakses melalui laman sesawang



Pencerobohan maklumat 信息侵入



**Untuk melindungi
informasi daripada
terdedah dan dipintas
oleh pihak lain.**

**Untuk mengelakkan
sebarang
penyamaran atau
penipuan berlaku.**

**Untuk menjaga
kerahsiaan
maklumat.**



2.1.1 Kriptografi dalam Pengkomputeraan

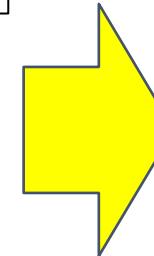
BT m/s 30

Bahasa Yunani

Kriptografi
(cryptography)

“kriptos”
(sembunyi)

“graphein”
(tulis)



Kajian tentang teknik kerahsiaan atau sebagai keselamatan komunikasi data

Tujuan Utama = untuk melindungi informasi daripada terdedah dan dipintas oleh pihak lain.

4 kepentingan perkhidmatan keselamatan data kriptografi



Perkhidmatan Keselamatan

a. Kerahsiahан (Confidentiality) 保密

- Kesulitan sesuatu maklumat yang dihantar melalui rangkaian komputer yang perlu dilindungi supaya maklumat tersebut tidak diketahui oleh pihak lain selain penerima.

b. Pengesahan (Authentication) 认证

- Pengenalpastian pihak-pihak yang terlibat dalam suatu komunikasi.
- Maklumat atau data yang diterima perlu dikenal pasti agar maklumat dihantar oleh pihak yang disahkan.
- Cth : Maybank2U , TAC

c. Integriti (Integrity) 廉正

- Perkhidmatan keselamatan yang mengenal pasti sebarang perubahan terhadap maklumat atau data oleh pihak lain.
- Cth : transaksi perbankan

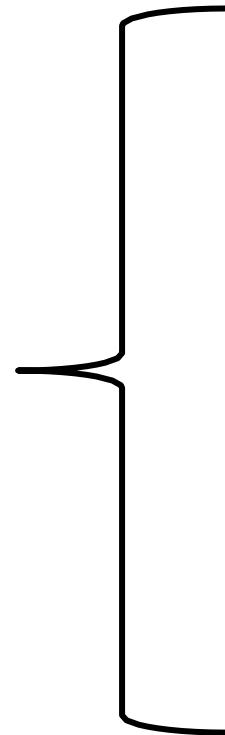
d. Tiada sangkalan (Non-repudiation) 不可否认

- Untuk membuktikan bahawa penghantar dan penerima maklumat tidak menafikan bahawa mereka menghantar dan menerima maklumat tersebut.

2.1.2 Proses Sifer (*Cipher*)

Kriptografi

Penyamaran demi menjaga kerahsiaan dan keselamatan sesuatu data .



Penyulitan (*encryption*) 加密

Proses penukaran teks biasa kepada teks sifer, iaitu dalam bentuk yang tidak bermakna apabila dibaca.



Nyahsulit (*decryption*) 解码

Proses untuk mengembalikan teks sifer menjadi teks biasa yang dapat dibaca dan mempunyai maksud

Algoritma Kriptografi / Sifer

=> Proses melakukan penyulitan dan nyahsulit



BT m/s 32

Sifer

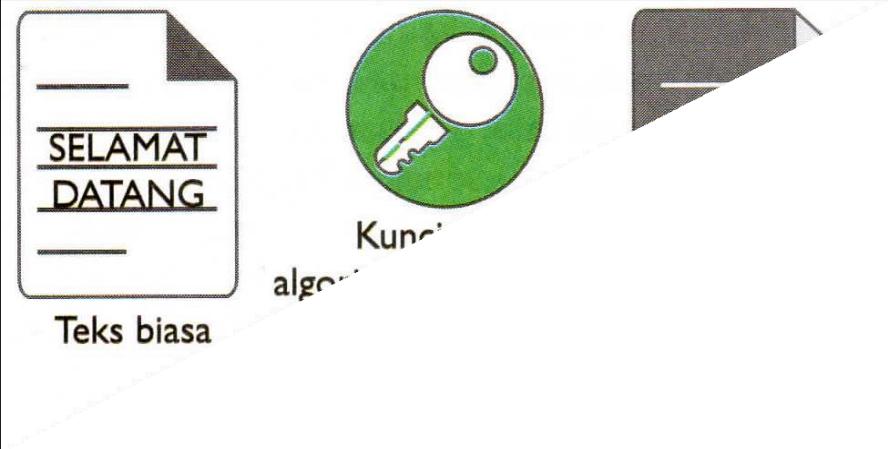
Sifer Kunci Simetri

- Termasuk semua **sifer klasik** – **hanya untuk penyulitan dan nyahsulit teks tulisan.**
- Menggunakan kunci yang sama untuk melakukan **penyulitan** dan **nyahsulit**.
- Kunci yang digunakan boleh dalam bentuk nombor, huruf, perkataan atau simbol.
- Maklumat kunci mesti disimpan secara rahsia.
- Penghantar dan penerima boleh menjana kunci rahsia baharu.

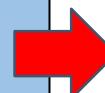
Sifer Kunci Tidak Simetri

- Digunakan dalam **sifer moden – sifer yang kompleks digunakan dalam sebarang bentuk data.**
- Kompleks dan melibatkan pengiraan matematik.
- Melibatkan **2 kunci** iaitu **kunci awam** dan **kunci persendirian**.
- Kunci awam boleh digunakan oleh sesiapa sahaja untuk penyulitan.
- Kunci persendirian disimpan secara rahsia oleh penenerima sahaja.
- Nyahsulit memerlukan algoritma yang sepadan dan dengan kunci persendirian.

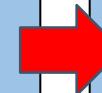
(i) Penyulitan (*Encryption*)



1. Pengirim menghantar mesej (teks biasa) kepada penerima

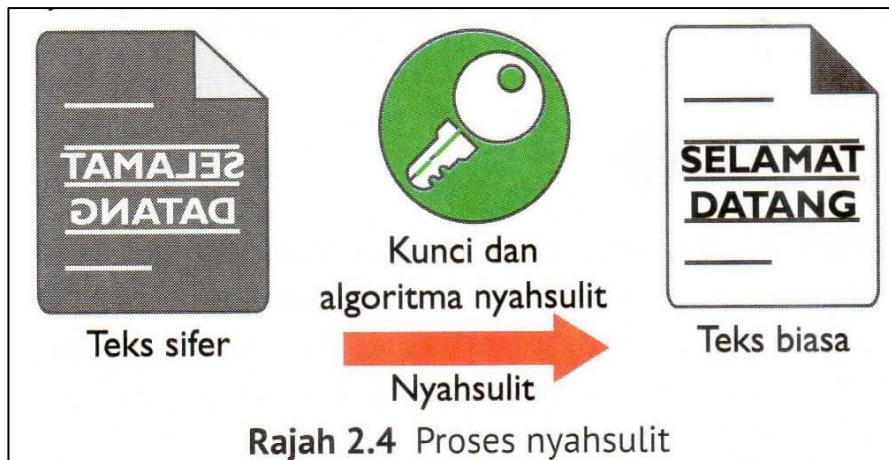


2. Mesej ini akan disulitkan menggunakan satu kunci dan algoritma penyulitan serta ditukar ke teks sifer.



3. Teks sifer, iaitu teks yang tidak bermakna akan terhasil

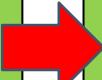
(ii) Nyahsulitan (*Decryption*)



1. Teks sifer yang telah disulitkan akan dinyahsulit menggunakan satu kunci dan algoritma nyahsulit.



2. Teks sifer yang akan menjadi teks biasa.



3. Penerima akan menerima teks biasa yang dapat dibaca.

Rajah 2.4 Proses nyahsulit

Kaedah sifer

KAEDAH – KAEDAH SIFER

Reverse cipher

- a. Songsangan dilakukan berdasarkan perkataan
- b. Songsangan dilakukan berdasarkan seluruh mesej.

Substitution cipher

- a. Caesar Cipher
- b. Pigpen Cipher
- c. Atbash Cipher -
Songsangan dilakukan berdasarkan abjad (A-Z)

Transposition cipher

- a. Columnar Transposition
- b. Rail Fence Cipher

Reverse cipher

- Kaedah Sifer paling mudah kerana proses penyulitan dilakukan dengan keadaan songsang.
- Menggunakan cara songsang untuk menyulit mesej.

KAEDAH SIFER	Songsangan dilakukan berdasarkan perkataan	Songsangan dilakukan berdasarkan seluruh mesej.
Proses Penyulitan (Encryption)	<p><i>Ejaan setiap perkataan dalam mesej diterbalikkan</i></p> <p><i>Teks biasa (plaintext) :</i></p> <p>ASK PN LAI YEN WEI</p> <p><i>Teks sifer (ciphertext) :</i></p> <p>URUG KSA NP IAL NEY IEW</p>	<p><i>Mesej dipaparkan secara terbalik, termasuk kedudukan dan ejaan setiap perkataan bermula dari abjad terakhir pada satu mesej yang hendak disulitkan</i></p> <p><i>Teks biasa (plaintext) :</i></p> <p>GURU ASK PN LAI YEN WEI</p> <p><i>Teks sifer (ciphertext) :</i></p> <p>IEW NEY IAL NP KSA URUG</p>
Proses Nyahsulit (Decryption)	<p>Songsangkan abjad-abjad setiap perkataan dalam teks sifer.</p> <p><i>Teks sifer (ciphertext) :</i></p> <p>URUG KSA NP IAL NEY IEW</p> <p><i>Teks biasa (plaintext) :</i></p> <p>GURU ASK PN LAI YEN WEI</p>	<p>Songsangkan abjad-abjad seluruh mesej dalam teks sifer bermula dari perkataan terakhir.</p> <p><i>Teks sifer (ciphertext) :</i></p> <p>IEW NEY IAL NP KSA URUG</p> <p><i>Teks biasa (plaintext) :</i></p> <p>GURU ASK PN LAI YEN WEI</p>
Kedudukan Teks biasa	<ul style="list-style-type: none"> • Kedudukan abjad-abjad teks biasa akan berubah dari kedudukan asalnya 	
Identiti Teks biasa	<ul style="list-style-type: none"> • Identiti abjad-abjad teks biasa tidak berubah. Kekal huruf asal. 	
KEKUATAN	<ul style="list-style-type: none"> • Mudah digunakan . 	
KELEMAHAN	<ul style="list-style-type: none"> • Tidak menggunakan kunci. Tahap kerahsiaannya rendah. • Mesej mudah untuk menyulit atau dinyahsulit dengan hanya menyongsangkannya 	

Substitution Cipher

- Mesej disulitkan dengan menggantikan satu unit teks biasa dengan satu unit teks yang lain (sifer) terdiri daripada satu abjad atau kumpulan abjad atau simbol.
- Paling mudah kerana proses penyulitan dilakukan dengan menggantikan setiap abjad teks biasa dengan abjad yang lain atau simbol .

KAEDAH SIFER	Atbash Cipher – Songsangan dilakukan berdasarkan abjad (A- Z)																																																						
Proses Penyulitan (Encryption)	Kedudukan abjad A-Z diterbalikkan susunan sebagai sifer, seterusnya huruf di dalam mesej digantikan dengan abjad sifer. <table border="1"><tr><td>Teks Biasa</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr><tr><td>Teks Sifer</td><td>Z</td><td>Y</td><td>X</td><td>W</td><td>V</td><td>U</td><td>T</td><td>S</td><td>R</td><td>Q</td><td>P</td><td>O</td><td>N</td><td>M</td><td>L</td><td>K</td><td>J</td><td>I</td><td>H</td><td>G</td><td>F</td><td>E</td><td>D</td><td>C</td><td>B</td><td>A</td></tr></table> <i>Teks biasa (plaintext) :</i> GURU ASK PN LAI YEN WEI <i>Teks sifer (ciphertext) :</i> TFIF ZHP KM OZR BVM DVR	Teks Biasa	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	Teks Sifer	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Teks Biasa	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																													
Teks Sifer	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A																													
Proses Nyahsulit (Decryption)	Kedudukan abjad A-Z diterbalikkan susunan sebagai sifer, seterusnya huruf mesej di dalam teks sifer digantikan dengan abjad teks biasa. <i>Teks sifer (ciphertext) :</i> TFIF ZHP KM OZR BVM DVR <i>Teks biasa (plaintext) :</i> GURU ASK PN LAI YEN WEI																																																						
Kedudukan Teks biasa	Kedudukan abjad-abjad teks biasa TIDAK akan berubah .																																																						
Identiti Teks biasa	Identiti abjad-abjad teks biasa berubah dengan digantikan oleh abjad-abjad tertentu.																																																						
KEKUATAN	<ul style="list-style-type: none">• Mudah digunakan .																																																						
KELEMAHAN	<ul style="list-style-type: none">• Tidak menggunakan kunci. Tahap kerahsiaannya rendah.• Mesej mudah untuk sulit dan dinyahsulit dengan hanya menyongsangkannya susunan abjad teks biasa dari A-Z ke teks sifer Z-A																																																						

KAEDAH SIFER	Caesar Cipher Paling awal digunakan oleh manusia untuk menyulitkan mesej dan disebut <i>shift cipher</i> (sifer anjakan)																																																												
Proses Penyulitan (Encryption)	<ul style="list-style-type: none"> Penyulitan teks biasa (plaintext) dilakukan dengan menggantikan setiap abjad dalam teks itu dengan abjad lain dalam susunan abjad , iaitu dengan menganjakkan abjad dalam susunan abjad. Bilangan anjakan dikenali sebagai kunci. Contoh : Kunci = 5 Bina jadual penggantian teks sifer dengan anjakan 5 abjad huruf ke kiri. Gantikan abjad teks biasa dengan abjad teks sifer mengikut jadual. <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">2</td><td style="text-align: center;">3</td><td style="text-align: center;">4</td><td style="text-align: center;">5</td><td style="border-left: 1px solid black;"></td> </tr> <tr> <td>Teks Biasa</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr> <tr> <td>Teks Sifer</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> </table> <p><i>Teks biasa (plaintext) :</i> GURU ASK PN LAI YEN WEI <i>Teks sifer (ciphertext) :</i> LZWZ FXP US QFN DJS BJJN</p>	1	2	3	4	5		Teks Biasa	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	Teks Sifer	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
1	2	3	4	5																																																									
Teks Biasa	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																																			
Teks Sifer	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E																																			
Proses Nyahsulit (Decryption)	<ul style="list-style-type: none"> Bina jadual penggantian teks sifer dengan anjakan 5 abjad huruf ke kiri. Nyahsulitkan abjad teks sifer dengan abjad teks biasa mengikut jadual penggantian yang disediakan . <p><i>Teks sifer (ciphertext) :</i> LZWZ FXP US QFN DJS BJJN <i>Teks biasa (plaintext) :</i> GURU ASK PN LAI YEN WEI</p>																																																												
Kedudukan Teks biasa	Kedudukan abjad-abjad teks biasa TIDAK akan berubah .																																																												
Identiti Teks biasa	Identiti abjad-abjad teks biasa berubah dengan digantikan oleh abjad-abjad tertentu.																																																												
KEKUATAN	<ul style="list-style-type: none"> Menggunakan kunci. Tahap kerahsiaan sederhana dan tinggi Abjad dalam teks biasa digantikan dengan abjad yang lain. 																																																												
KELEMAHAN	<ul style="list-style-type: none"> Kaedah cuba-jaya boleh digunakan untuk meramalkan nilai kunci. 																																																												

KAEDAH SIFER	<p>Pigpen Cipher</p> <ul style="list-style-type: none"> • Substitution cipher geometri yang mudah • Menukar abjad asal kepada simbol grafik yang sepadan seperti rajah di bawah. <table border="1" style="margin-bottom: 10px;"> <tr> <td>A</td><td>B</td><td>C</td><td></td><td>J</td><td>K</td><td>L</td><td></td> </tr> <tr> <td>D</td><td>E</td><td>F</td><td></td><td>M</td><td>N</td><td>O</td><td></td> </tr> <tr> <td>G</td><td>H</td><td>I</td><td></td><td>P</td><td>Q</td><td>R</td><td></td> </tr> </table>	A	B	C		J	K	L		D	E	F		M	N	O		G	H	I		P	Q	R																													
A	B	C		J	K	L																																															
D	E	F		M	N	O																																															
G	H	I		P	Q	R																																															
Proses Penyulitan (Encryption)	<ul style="list-style-type: none"> • Bina jadual penggantian abjad bagi simbol pigpen. • Gantikan abjad teks biasa kepada teks sifer dengan simbol grafik mengikut jadual perwakilan simbol pigpen yang dibinakan . <table style="margin-top: 10px; border-collapse: collapse;"> <tr> <td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td><td style="text-align: center;">D</td><td style="text-align: center;">E</td><td style="text-align: center;">F</td><td style="text-align: center;">G</td><td style="text-align: center;">H</td><td style="text-align: center;">I</td><td style="text-align: center;">J</td><td style="text-align: center;">K</td><td style="text-align: center;">L</td><td style="text-align: center;">M</td><td style="text-align: center;">N</td><td style="text-align: center;">O</td><td style="text-align: center;">P</td><td style="text-align: center;">Q</td><td style="text-align: center;">R</td><td style="text-align: center;">S</td><td style="text-align: center;">T</td><td style="text-align: center;">U</td><td style="text-align: center;">V</td><td style="text-align: center;">W</td><td style="text-align: center;">X</td><td style="text-align: center;">Y</td><td style="text-align: center;">Z</td> </tr> <tr> <td style="text-align: center;"></td><td style="text-align: center;"></td> </tr> </table> <p><i>Teks biasa (plaintext) :</i></p> <p>GURU ASK PN LAI YEN WEI</p> <p><i>Teks sifer (ciphertext) :</i></p> <p></p>	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																										
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																												
Proses Nyahsulit (Decryption)	<ul style="list-style-type: none"> • Bina jadual penggantian abjad bagi simbol pigpen. • Nyahsulitkan simbol grafik teks sifer kepada abjad teks biasa mengikut jadual perwakilan simbol pigpen yang dibinakan . <p><i>Teks sifer (ciphertext) :</i></p> <p></p> <p><i>Teks biasa (plaintext) :</i></p> <p>GURU ASK PN LAI YEN WEI</p>																																																				
Kedudukan Teks biasa	Kedudukan abjad-abjad teks biasa TIDAK akan berubah .																																																				
Identiti Teks biasa	Identiti abjad-abjad teks biasa berubah dengan digantikan oleh simbol grafik																																																				
KEKUATAN	<ul style="list-style-type: none"> • Menggunakan kunci. Tahap kerahsiaan rendah. • Abjad dalam teks biasa digantikan dengan simbol grafik yang lain 																																																				
KELEMAHAN	<ul style="list-style-type: none"> • Mudah dinyahsulitkan jika mengetahui cara perwakilan simbol. 																																																				

Transposition Cipher

- **Mengubahkan kedudukan abjad** dalam teks biasa. Abjad-abjad yang terkandung dalam **teks biasa adalah tetap.**
- Perubahan ini dikenali sebagai pemutaran atau pencampuran (scrambling). **Memerlukan kunci dalam penyulitan dan Nyahsulit.**

Columnar Tranposition

1. Sifer tranposisi yang menyulitkan satu mesej dengan menulisnya dalam **jadual secara mendatar (Horizontal)** dengan bilangan lajur (columnar) yang telah ditetapkan.
2. Satu perkataan dipilih sebagai kunci penyulitan.
3. Bilangan kunci akan menentukan **bilangan lajur (排)** dalam jadual. (Key = 格子)
4. Abjad dalam kunci akan menentukan cara bagi mengubah urutan abjad-abjad dalam mesej.
(加密的排法依照 KEY 的 A 到 Z 顺序)
5. **Bilangan baris (行) = Bilangan abjad dalam tek biasa ÷ bilangan abjad dalam kunci**
6. **Proses Penyulitan (Encryption) :**
 - a. Kenal pastikan kunci. Dapatkan kunci . Bilangan lajur = Bilangan kunci . Isikan kunci di baris pertama kemudian isikan baris kedua dengan susunan abjad kunci berdasarkan urutan dari A-Z.
 - b. Hitungkan bilangan baris untuk teks biasa dengan formula :
Bilangan baris = Bilangan abjad dalam teks biasa ÷ bilangan abjad dalam kunci
 - c. Kembangkan jadual berdasarkan Bilangan baris yang didapatkan daripda langkah b.
 - d. Isikan teks biasa (plaintext) ke dalam jadual dari kiri ke kanan , baris demi baris secara mendatar (橫左去右) .
 - e. Selepas semua teks dimasukkan, Petak-petak kosong diisikan dengan sebarang abjad demi bagi memenuhi jadual.
 - f. Teks sifer (ciphertext) dihasilkan dengan menuliskan dari Lajur 1 mula atas ke bawah diikuti Lajur seterusnya.

Bilangan Lajur = Bilangan abjad kunci
(Key 的字母数量)

Kunci = PINTU (5个字母 所以就有5排上到
下) **L1 , L2, L3, L4, L5**

**4. Ciphertext dibacakan mula lajur 1 dari
atas ke bawah sehingga lajur terakhir.**
**Cipher的读法是从第一排 (L1) 上到下到最
后一排。**

**1. Susunan abjad
kunci mengikut
urutan (key 的顺序
排法 a-z)**

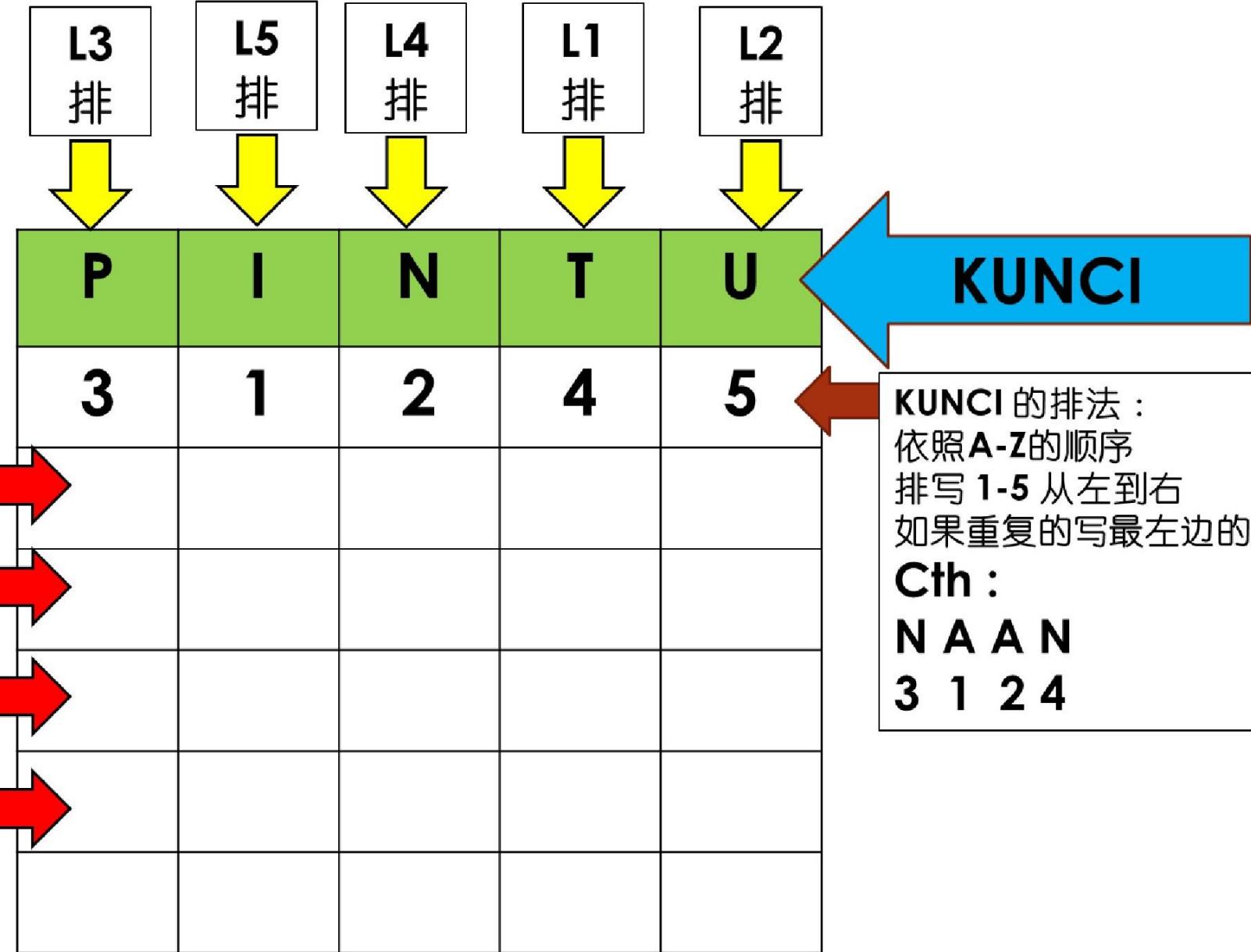
P	I	N	T	U
3	1	2	4	5

**3. Tuliskan mesej
plaintext secara
mendatar dari kiri ke
kanan mula baris
pertama hingga ke
baris terakhir.**

Sekiranya , terdapat
petak kosong, isikan
petak kosong dengan
sebarang abjad (A-Z)

**2. Baris = bil abjad plaintext ÷ bil
abjad kunci**
行 = Mesej 的字母数量 (÷)
KUNCI 的字母数量

把MESEJ从左到右填写在格子从第一行写到最后一行。
如果有多余的空格就随便填写 A - Z



KUNCI 的排法：
依照**A-Z**的顺序
排写**1-5**从左到右
如果重复的写最左边的先

Cth :

NAAN

3 1 2 4

KUNCI = PINTU

Plaintext = “GURU ASK PN LAI YEN WEI”

Ciphertext = ?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
G	U	R	U	A	S	K	P	N	L	A	I	Y	E	N	W	E	I

Bilangan kunci = 5 等于 Bilangan lajur = 5

Bilangan abjad = 18

$$\begin{aligned}\text{Bilangan Baris} &= 18 \div 5 \\ &= 3.6 \\ &= 4\end{aligned}$$

KUNCI = PINTU Plaintext = “GURUASK PN LAI YEN WEI”

L1	L2	L3	L4	L5
UKIE	RPYI	GSAW	UNEX	ALNX

CIPHER TEXT (根据第L1的格子写从上到下) 一直填写到L5

PLAINTEXT 加密第一行
baris开始从左到右到最后一行。

P	I	N	T	U
3	1	2	4	5
G	U	R	U	A
S	K	P	N	L
A	I	Y	E	N
W	E	I	X	X

你会发现剩下的两个空格随便填写A-Z“X”。

CIPHER TEXT : UKIE RPYI GSAW UNEX ALNX

Proses Nyahsulit (Decryption) :

1. Kenal pastikan kunci. Dapatkan kunci . Bilangan lajur = Bilangan kunci . Isikan kunci di baris pertama kemudian isikan baris kedua dengan susunan abjad kunci berdasarkan urutan dari A-Z.
2. Hitungkan bilangan baris untuk teks biasa dengan formula :
Bilangan baris = Bilangan abjad dalam tek sifer ÷ bilangan abjad dalam kunci
3. Kembangkan jadual berdasarkan Bilangan baris yang didapatkan daripada langkah b.
4. Isikan teks sifer (ciphertext) ke dalam jadual pada Lajur pertama mula dari atas ke bawah secara menegak (Vertical) (直上到下) diikuti lajur seterusnya.
5. Selepas semua teks sifer dimasukkan, bacakan teks biasa (plaintext) secara mendatar dari kiri ke kanan baris demi baris mula baris pertama hingga ke baris terakhir . (PLAINTEXT 破解第一行开始从左到右解码到最后一行)

KUNCI = PINTU

Ciphertext = “UKIE RPYI GSAW UNEX ALNX”

Plaintext = “? ”

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
U	K	I	E	R	P	Y	I	G	S	A	W	U	N	E	X	A	L	N	X

Bilangan kunci = 5 等于 Bilangan lajur = 5

Bilangan abjad = 20

Bilangan Baris = $20 \div 5$

Bilangan kunci = 5 (五排 = L1,L2, L3, L4, L5)

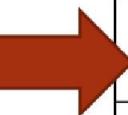
Bilangan baris = 4 (4行 等于 1 排 L 有 4 个字母)

CIPHER TEXT : UKIE RPYI GSAW UNEX ALNX

L1	L2	L3	L4	L5
UKIE	RPYI	GSAW	UNEX	ALNX

1. CIPHER TEXT (根据第L1的格子写从上到下) 一直填写到L5

2. PLAINTEXT 破解第一行
baris开始读从左到右到
最后一行。



P	I	N	T	U
3	1	2	4	5
G	U	R	U	A
S	K	P	N	L
A	I	Y	E	N
W	E	I	X	X

你会发现剩下的两个格XX,没有意思不用管它。

KAEDAH SIFER	Columnar Transposition
Proses Penyulitan (Encryption)	<p>Teks biasa = “GURU ASK PN LAI YEN WEI”</p> <ol style="list-style-type: none"> 1. KUNCI = PINTU. Bilangan lajur = Bilangan kunci . Isikan kunci di baris pertama kemudian isikan baris kedua dengan susunan abjad kunci berdasarkan urutan dari A-Z. PINTU = 3, 1, 2, 4, 5 2. Bilangan baris = Bilangan abjad dalam teks biasa ÷ bilangan abjad dalam kunci $= 18 \div 5$ $= 3.6$ $= 4$ 3. Kembangkan jadual berdasarkan Bilangan baris yang didapatkan daripda langkah b. 4. Isikan teks biasa (plaintext) ke dalam jadual dari kiri ke kanan , baris demi baris secara mendatar (橫左去右) . 5. Selepas semua teks dimasukkan, Petak-petak kosong diisikan dengan sebarang abjad demi bagi memenuhi jadual. 6. Teks sifer (ciphertext) dihasilkan dengan menuliskan dari Lajur 1 mula atas ke bawah diikuti Lajur seterusnya. <p>Teks sifer = UKIE RPYI GSAW UNEX ALNX</p>
Proses Nyahsulit (Decryption)	<p>Teks sifer = UKIE RPYI GSAW UNEX ALNX</p> <ol style="list-style-type: none"> 1. KUNCI = PINTU. Bilangan lajur = Bilangan kunci . Isikan kunci di baris pertama kemudian isikan baris kedua dengan susunan abjad kunci berdasarkan urutan dari A-Z. PINTU = 3, 1, 2, 4, 5 2. Bilangan baris = Bilangan abjad dalam teks sifer ÷ bilangan abjad dalam kunci $= 20 \div 5$ $= 4$ 3. Kembangkan jadual berdasarkan Bilangan baris yang didapatkan daripda langkah b. 4. Isikan teks sifer (ciphertext) ke dalam jadual pada Lajur pertama mula dari atas ke bawah secara menegak (Vertical) diikuti lajur seterusnya. 5. Selepas semua teks dimasukkan, bacakan teks biasa (plaintext) secara mendatar dari kiri ke kanan baris demi baris mula baris pertama hingga ke baris terakhir . <p>Teks biasa = “GURU ASK PN LAI YEN WEI”</p>

Kedudukan Teks biasa	<ul style="list-style-type: none"> • Kedudukan abjad-abjad teks biasa berubah dari kedudukan asalnya. • Memasukkan mengikut bentuk petak dari kiri ke kanan
Identiti Teks biasa	<ul style="list-style-type: none"> • Identiti abjad-abjad teks biasa tidak berubah . Kekal huruf asalnya
KEKUATAN	<ul style="list-style-type: none"> • Menggunakan kunci iaitu perkataan.Tahap kerahsiaan sederhana dan tinggi. • Nombor pada lajur berdasarkan kunci sukar diramalkan.
KELEMAHAN	<ul style="list-style-type: none"> • Abjad-abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.

Rail Fence Cipher

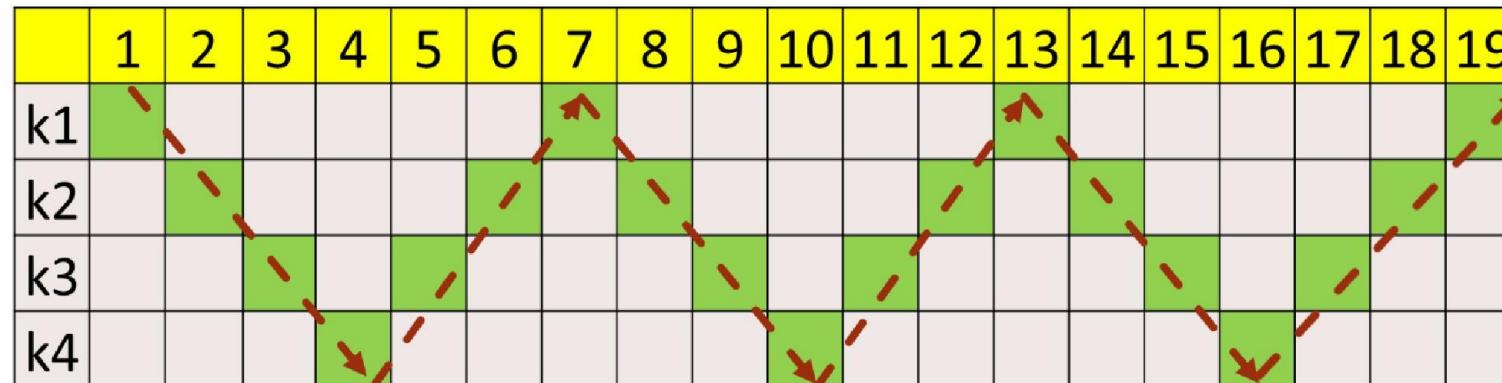
1. Penyulitan abjad-abjad dalam teks biasa ditulis semula mengikut susunan baris demi baris dan mengikut corak zig-zag
 2. Memerlukan kunci untuk menyatakan baris yang dikehendaki. (行=KEY 的数字)
 3. Lajur adalah mengikut jumlah bilangan abjad dalam teks biasa. (格子=plaintext 的字母)
 4. Lukiskan Petak Fence dan tandakan petak dalam corak zig-zag

5. Proses Penyulitan (Encryption) :

- a. Tuliskan Plaintext dalam susunan bentuk zig-zag (pagar)
 - b. Teks sifer dibacakan melintang mengikutkan susunan baris . (写从第一行左到右写)

6. Proses Nyahsulit (Decryption) :

- a. Tuliskan teks sifer baris demi baris mengikut petak yang telah ditanda
 - b. Plaintext dibacakan dalam bentuk zig-zag (pagar)



PLAINTEXT

CHIPIERTEXT

ZIGZAG

Penyulitan Rail Fence Cipher

Kunci = 6

Plaintext = “**GURU ASK PN LAI YEN WEI**” CIPHER TEXT = ?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
G	U	R	U	A	S	K	P	N	L	A	I	Y	E	N	W	E	I

Lajur (column) 格子 = PlainText 的字数量 = 18

Baris (row) 行 = Key 的字数量 = 6

cth : Kunci = 6 就代表有 6 行

1 . Lukiskan jadual 6 baris (6 行上到下) 18 petak lajur (18 排格子左到右) 。

2. Tandakan petak dengan corak zig-zag .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
k1																		
k2																		
k3																		
k4																		
K5																		
K6																		

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
k1	G																	
k2		U									L		I					
k3			R								N			Y				
k4				U						P				E			I	
K5					A		K								N	E		
K6						S										W		

1. 用 zig-zag 排法的方式把 Plaintext 写进格子.

2. 跟着每一行的写出 ciphertext 从上到下。

K1 : GA

K2 : ULI

K3 : RNY

K4 : UPEI

K5 : AKNE

K6 : SW

3. Cipher text : **GAULIRNYUPEIAKNESW**

Nyahsulit Rail Fence Cipher

Kunci = 6

CIPHER TEXT = “ **GAULIRNYUPEIAKNESW**”

Bilangan baris = 6

Bilangan lajur = 18 (bilangan abjad dalam teks sifer)

1. Lukiskan petak pagar , 6 baris dan 18 lajur
2. Tandakan petak pagar mengikut corak zig-zag.
3. Isikan teks sifer ke dalam petak yang telah ditandakan baris demi baris mula pada baris teratas hingga baris terakhir isikan petak dari kiri ke kanan .

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
k1	G									A								
k2		U							L		I							
k3			R						N			Y						
k4				U				P				E				I		
K5					A	K						N		E				
K6					S								W					

4. Bacakan teks biasa mengikut corak zig-zag.

Plaintext = “ **GURU ASK PN LAI YEN WEI** ”

KAEDAH SIFER	Rail Fence Cipher
Proses Penyulitan (Encryption)	<p>Kunci = 6 Teks biasa = “GURU ASK PN LAI YEN WEI”</p> <ol style="list-style-type: none"> 1. Penyulitan abjad-abjad dalam teks biasa ditulis semula mengikut susunan baris demi baris dan mengikut corak zig-zag. 2. Memerlukan kunci untuk menyatakan baris yang dikehendaki. (行=KEY 的数字) 3. Lajur adalah mengikut jumlah bilangan abjad dalam teks biasa. (格子=plaintext 的字母的数量) 4. Lukiskan Petak Fence baris = 6 , lajur = 18 petak dan tandakan petak dalam corak zig-zag. 5. Tuliskan Plaintext dalam susunan bentuk zig-zag (pagar) 6. Teks sifer dibacakan melintang mengikutkan susunan baris . (从第一行左到右写) <p>Teks sifer = “GAULIRNYUPEIAKNESW”</p>
Proses Nyahsulit (Decryption)	<p>Kunci = 6 Teks sifer = “GAULIRNYUPEIAKNESW”</p> <ol style="list-style-type: none"> 1. Nyahsulit abjad-abjad dalam teks sifer ditulis semula mengikut susunan baris demi baris dan mengikut corak zig-zag. 2. Memerlukan kunci untuk menyatakan baris yang dikehendaki. (行=KEY 的数字) 3. Lajur adalah mengikut jumlah bilangan abjad dalam teks sifer. (格子=ciphertext 的字母的数量) 4. Lukiskan Petak Fence baris = 6 , lajur = 18 petak dan tandakan petak dalam corak zig-zag 5. Tuliskan teks sifer baris demi baris mengikut petak yang telah ditanda. (写从第一行左到右写) 6. Teks biasa dibacakan dalam bentuk zig-zag (pagar) <p>Teks biasa = “GURU ASK PN LAI YEN WEI”</p>
Kedudukan Teks biasa	<ul style="list-style-type: none"> • Kedudukan abjad-abjad teks biasa berubah dari kedudukan asalnya. • Memasukkan mengikut bentuk pagar zig-zag
Identiti Teks biasa	<ul style="list-style-type: none"> • Identiti abjad-abjad teks biasa tidak berubah . Kekal huruf asalnya
KEKUATAN	<ul style="list-style-type: none"> • Menggunakan kunci iaitu baris.Tahap kerahsiaan sederhana dan tinggi. • Bilangan baris sukar diramalkan.
KELEMAHAN	<ul style="list-style-type: none"> • Abjad-abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.



Kaedah Sifer	Tahap Kerahsiaan	Kekuatan	Kelemahan
Reverse cipher	Rendah	<ul style="list-style-type: none">Mudah digunakan	<ul style="list-style-type: none">Tidak menggunakan kunci.Mesej mudah dinyahsulit dengan hanya menyongsangkannya
Caesar Cipher	Sederhana dan Tinggi	<ul style="list-style-type: none">Menggunakan kunci.Abjad dalam teks biasa digantikan dengan abjad yang lain.	<ul style="list-style-type: none">Keadah cuba-jaya boleh digunakan untuk meramalkan nilai kunci.
Pigpen Cipher	Rendah	<ul style="list-style-type: none">Menggunakan kunci.Abjad dalam teks biasa digantikan dengan simbol lain.	<ul style="list-style-type: none">Mudah dinyahsulitkan jika mengetahui cara perwakilan simbol.
Rail Fence Cipher	Sederhana dan Tinggi	<ul style="list-style-type: none">Menggunakan kunci , iaitu baris.Bilangan baris sukar untuk diramalkan.	<ul style="list-style-type: none">Abjad-Abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.
Columnar Transposition	Sederhana dan Tinggi	<ul style="list-style-type: none">Menggunakan kunci.Nombor pada lajur yang berdasarkan kunci sukar diramal.	<ul style="list-style-type: none">Abjad-Abjad dalam teks sifer terdiri daripada abjad-abjad dalam teks biasa.

Faedah-faedah pembangunan algoritma :

- a. Memahami bagaimana komputer memproses urutan tindakan-tindakan yang perlu dilaksanakan.**
- b. Mempelajari cara menulis algoritma yang teratur, mudah dibaca dan difahami.**
- c. Mempelajari cara menulis algoritma dengan cekap.**
- d. Mempelajari cara melakukan penambahan pada algoritma.**

Algoritma dalam masalah pengaturcaraan

Search (Carian)

Proses untuk mendapatkan suatu item tertentu yang terkandung dalam senarai

Linear Search

Teknik untuk mendapatkan suatu item yang dikehendaki dalam satu senarai linear

Binary Search

Teknik carian yang melipatkan keputusan dwipilihan

Sort (isihan)

Proses mengisih atau menyusun item-item dalam suatu senarai linear mengikut urutan tertentu

Bubble sort

Teknik pengisihan untuk mengisih item-item dalam senarai mengikut urutan menaik atau menurun

Bucket sort

Teknik mengasingkan item-item dalam senarai tertentu ke dalam baldi (bucket), seterusnya item-item dalam baldi akan diisih dan disusun semula ke dalam senarai

- *Search (Linear, Binary) 寻找*
 - Pencarian data bertujuan untuk mencari dan mendapatkan data yang terdapat dalam sekelompok data.
- *Sort (Bubble, Bucket) 排法*
 - Proses untuk menyusun kembali himpunan objek menggunakan aturan tertentu

Search dan sort digunakan dalam senarai atau pangkalan data

Perbandingan antara *Linear Search* dan *Binary Search*

By Pn. LAI YEN WEI SMJKYB

	Linear Search	Binary Search
Ciri- ciri / Syarat-Syarat	<ul style="list-style-type: none"> Satu teknik untuk mendapatkan item yang dikehendaki dalam satu senarai linear. Item-item dalam senarai tidak perlu diisih mengikut urutan dahulu sebelum membuat carian. 	<ul style="list-style-type: none"> Teknik carian yang melipatkan keputusan dwipilihan. Sebelum melakukan binary search, item-item dalam senarai perlu diisih dalam urutan menaik. Lebih sesuai digunakan pada senarai yang mempunyai item yang banyak.
Langkah-langkah	<div style="border: 1px solid black; padding: 5px;"> <p>Bandingkan item pertama dengan item yang dikehendaki . Jika item pertama sama, carian akan tamat.</p> </div> <div style="text-align: center; margin-top: 10px;"> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Jika tidak, carian akan dilakukan pada item seterusnya sehingga dijumpai.</p> </div> <div style="text-align: center; margin-top: 10px;"> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Jika item tidak dijumpai sehingga item terakhir, proses carian akan tamat tanpa hasil.</p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>Carian bermula dengan perbandingan di tengah senarai. Jika item carian sama dengan item di tengah senarai, carian tamat.</p> </div> <div style="text-align: center; margin-top: 10px;"> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Jika item carian lebih besar daripada item di tengah senarai, item di tengah senarai dan item-item di sebelah kiri akan diabaikan.carian akan dilakukan pada senarai yang tinggal.</p> </div> <div style="text-align: center; margin-top: 10px;"> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Jika item carian lebih kecil daripada item di tengah senarai, item di tengah senarai dan item-item di sebelah kanan akan diabaikan.carian akan dilakukan pada senarai yang tinggal.</p> </div>
Elemen pertama perbandingan dalam carian	Elemen paling pertama sekali dalam senarai .	Elemen di pertengahan senarai.
Keberkesanan dan kelajuan	<p><i>Linear search</i> sangat mudah, tidak begitu efisien kerana teknik ini akan menyemak setiap item yang terdapat dalam senarai serta memerlukan masa yang panjang jika senarai mengandungi terlalu banyak item. Jika item yang dicari berada di akhir atau item yang dicari tidak berada dalam senarai, carian akan tetap dilakukan sehingga item yang akhir.</p>	<p><i>Binary search</i> lebih efisien dan menjimatkan masa berbanding dengan <i>linear search</i> kerana <i>binary search</i> tidak perlu menyemak setiap item dalam senarai.</p>

Perbandingan antara *Bubble sort* dan *Bucket Sort*

By Pn. LAI YEN WEI SMJKYB

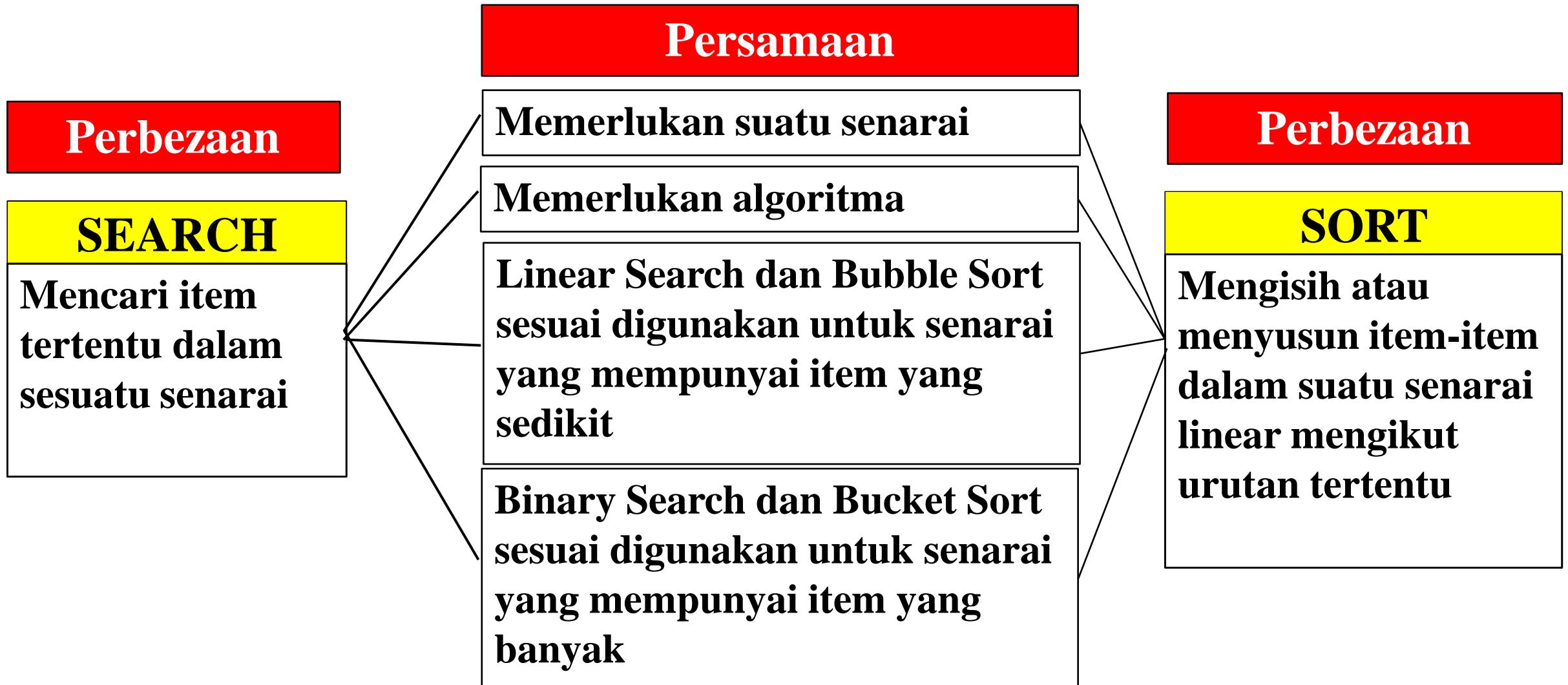
	Bubble Sort	Bucket Sort
Ciri- ciri / Syarat-Syarat	<ul style="list-style-type: none">Teknik pengisian untuk mengisih item-item dalam senarai mengikut urutan menaik atau menurun.<i>Bubble sort</i> sesuai digunakan pada senarai yang mempunyai item yang sedikit dan item-item yang terdiri daripada integer dan abjad.	<ul style="list-style-type: none">Teknik mengasingkan item-item dalam senarai tertentu ke dalam baldi (bucket), seterusnya item-item dalam baldi akan diisih dan disusun semula ke dalam senaraiBilangan baldi yang diperlukan bergantung kepada pengatur cara dan bilangan item dalam senarai yang perlu diisih.Hanya sesuai untuk senarai nombor yang berada dalam julat yang ditetapkan.
Langkah-langkah	<p>Bandingkan item pertama dengan item kedua. Jika item pertama lebih besar daripada item kedua, tukar kedudukan kedua-dua item tersebut.</p> <p style="text-align: center;">↓</p> <p>Jika item pertama lebih kecil daripada item kedua, kekalkan kedudukan kedua-dua item.</p> <p style="text-align: center;">↓</p> <p>Bandingkan item kedua dengan item ketiga, dan seterusnya sehingga dua item terakhir. Tukarkan kedudukan jika item-item tidak berada dalam urutan yang betul.</p> <p style="text-align: center;">↓</p> <p>Perbandingan ini berulang sehingga item-item dalam senarai berada dalam urutan menaik.</p>	<p>Sediakan beberapa buah baldi dan tetapkan julat yang sesuai bagi item yang akan diletakkan ke dalam baldi.</p> <p style="text-align: center;">↓</p> <p>Masukkan item-item yang perlu diisih ke dalam baldi mengikut julat yang betul.</p> <p style="text-align: center;">↓</p> <p>Isih item-item dalam baldi mengikut urutan dan masukkan semua item-item yang telah diisih ke dalam senarai.</p>

Perbandingan antara *Bubble sort* dan *Bucket Sort*

	<i>Bubble sort</i>	<i>Bucket Sort</i>
Bilangan algoritma yang digunakan.	Hanya satu algoritma digunakan iaitu algoritma <i>bubble sort</i> .	Dua algoritma digunakan iaitu algoritma <i>bucket sort</i> dan algoritma <i>bubble sort</i> .
Keberkesanan dan kelajuan	<ul style="list-style-type: none">• <i>Bubble Sort</i> paling ringkas dan mudah.• <i>Bubble Sort</i> melaksanakan isihan satu persatu dari awal sehingga semua nombor tersusun.• Kaedah ini tidak efektif dan kelajuan isihan adalah perlahan.	<p><i>Bucket sort</i> tidak melaksanakan isihan terus menerus dari mula sampai akhir, tetapi dengan cara mengumpulkan elemen yang sama pada bucket tertentu kemudian isihan dilakukan.</p> <p>Kaedah ini lebih efektif dan kelajuan isihan menjadi lebih pantas berbanding dengan bubble sort.</p>

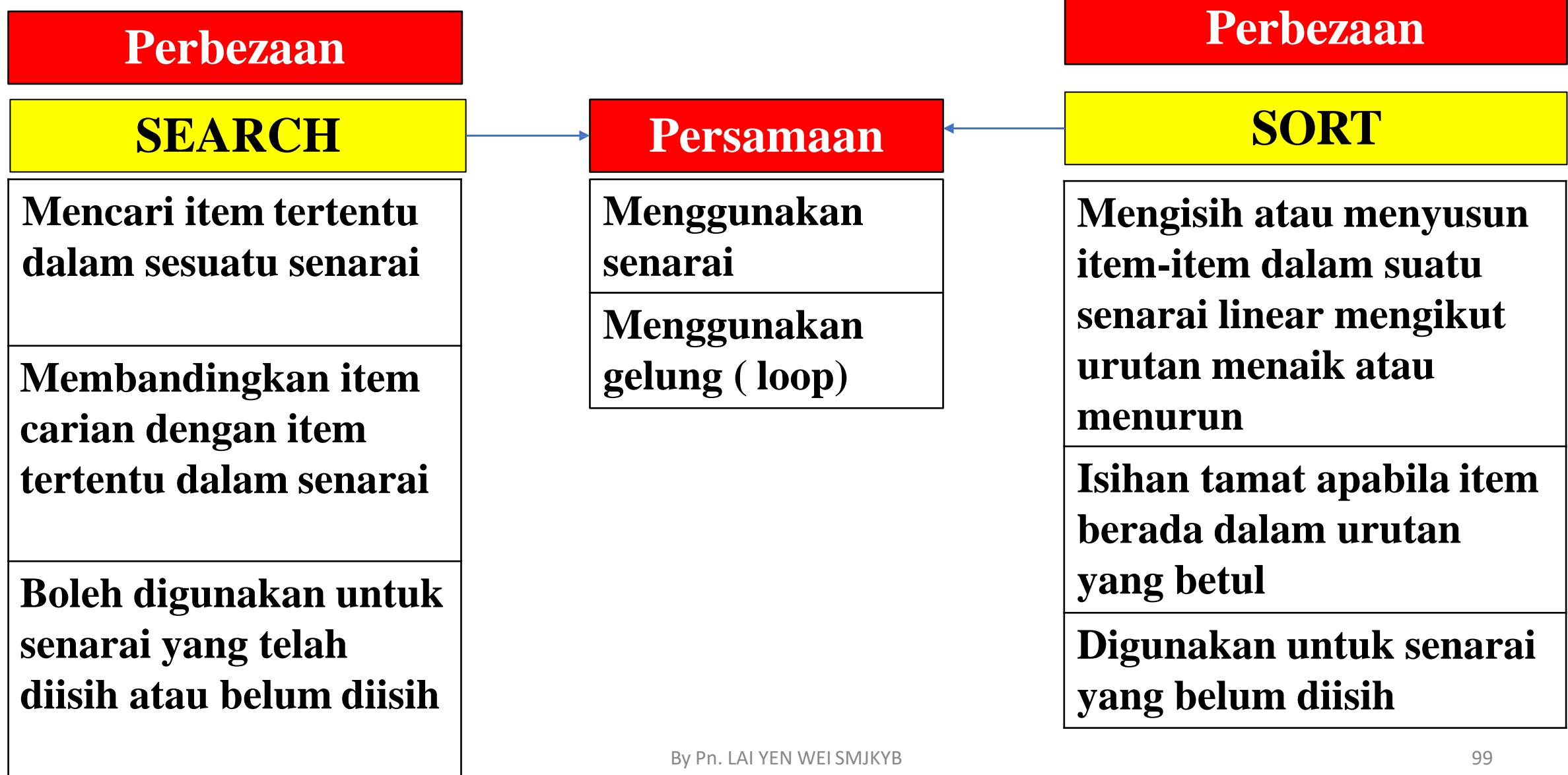
3.1 Persamaan dan Perbezaan Ciri-ciri Search dan Sort

BT m/s 72



3.1.5 Perbandingan ciri-ciri algoritma Search dan Sort

BT m/s 101





Latihan Linear Search

RUJUK BTEKS m/s 74-75

Menulis pseudokod dan melukis carta alir

Tips :

(L = List, T = Target , i = indeks, n = bilangan item)

L ialah senarai yang mengandungi n item.

T ialah pemboleh ubah yang mewakili **nilai item carian**.

i ialah pemboleh ubah yang mewakili **kedudukan item** dalam senarai L

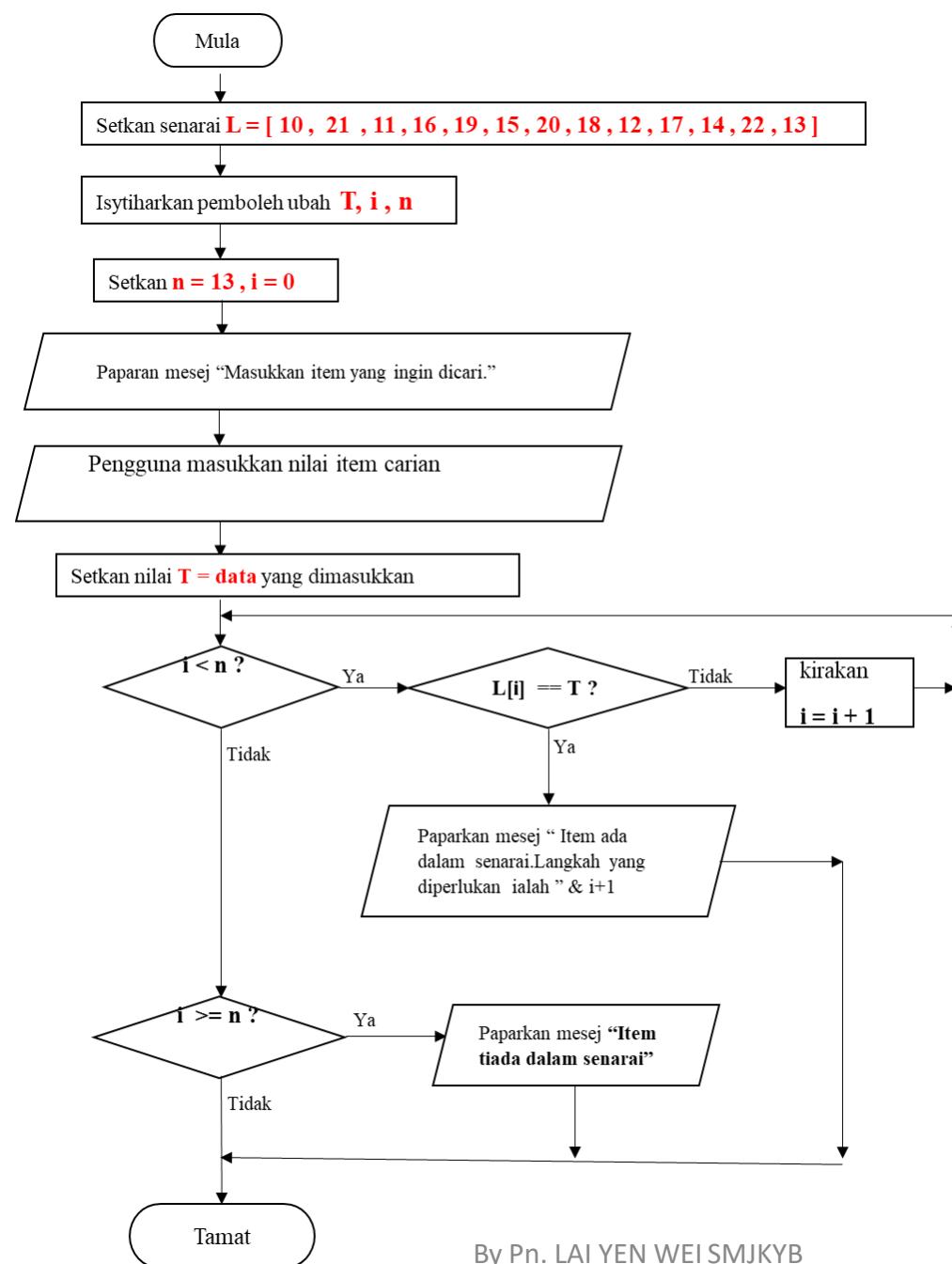
n ialah pemboleh ubah yang mewakili **bilangan item** dalam senarai L.

- Satu teknik untuk mendapatkan item yang dikehendaki dalam satu senarai linear.
- Item-item dalam **senarai tidak perlu diisih** mengikut urutan dahulu sebelum membuat carian.



Pseudokod Linear Search

1. Mula
 2. Setkan senarai $L = [10 , 21 , 11 , 16 , 19 , 15 , 20 , 18 , 12 , 17 , 14 , 22 , 13]$
 3. Isytiharkan **pemboleh ubah T, i , n**
 4. Setkan $n = 13 , i = 0$
 5. Paparan mesej "**Masukkan item yang ingin dicari.**"
 6. Pengguna masukkan nilai item carian
 7. Setkan **nilai T = data yang dimasukkan**
 8. **for** $i < n$
 9. **Jika** $L[i] == T$
 10. Paparkan mesej "**Item ada dalam senarai.** Langkah yang diperlukan
 ialah ", $i+1$
 Langkau ke Langkah 19
 11. **Jika Tidak**
 12. kirakan $i = i + 1$
 13. **Tamat Jika**
 14. Ulang Langkah 8
 15. **Jika** $i >= n$
 16. Paparkan mesej "**Item tiada dalam senarai**"
 17. **Tamat Jika**
 18. **Tamat**





Binary Search

senarai L = [10 , 21 , 11 , 16 , 19 , 15 , 20 , 18 , 12 , 17 , 14 , 22 , 13]

Mula mesti isihkan senarai secara menaik

Setkan senarai L = [10 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20 , 21 , 22]

Menulis pseudokod dan melukis carta alir

Tips :

(L = List , T = Target , i = indeks , n = bilangan item , j , m=indeks pertengahan)

L ialah senarai yang mengandungi n item.

T ialah pemboleh ubah yang mewakili **nilai item carian**.

i ialah pemboleh ubah yang mewakili **kedudukan item pertama** dalam senarai L

n ialah pemboleh ubah yang mewakili **bilangan item** dalam senarai L.

j = ialah pemboleh ubah yang mewakili **kedudukan item terakhir** dalam senarai L

m = (i + j) //2

jika L[m] > T , kedudukan item terakhir ialah j= m-1

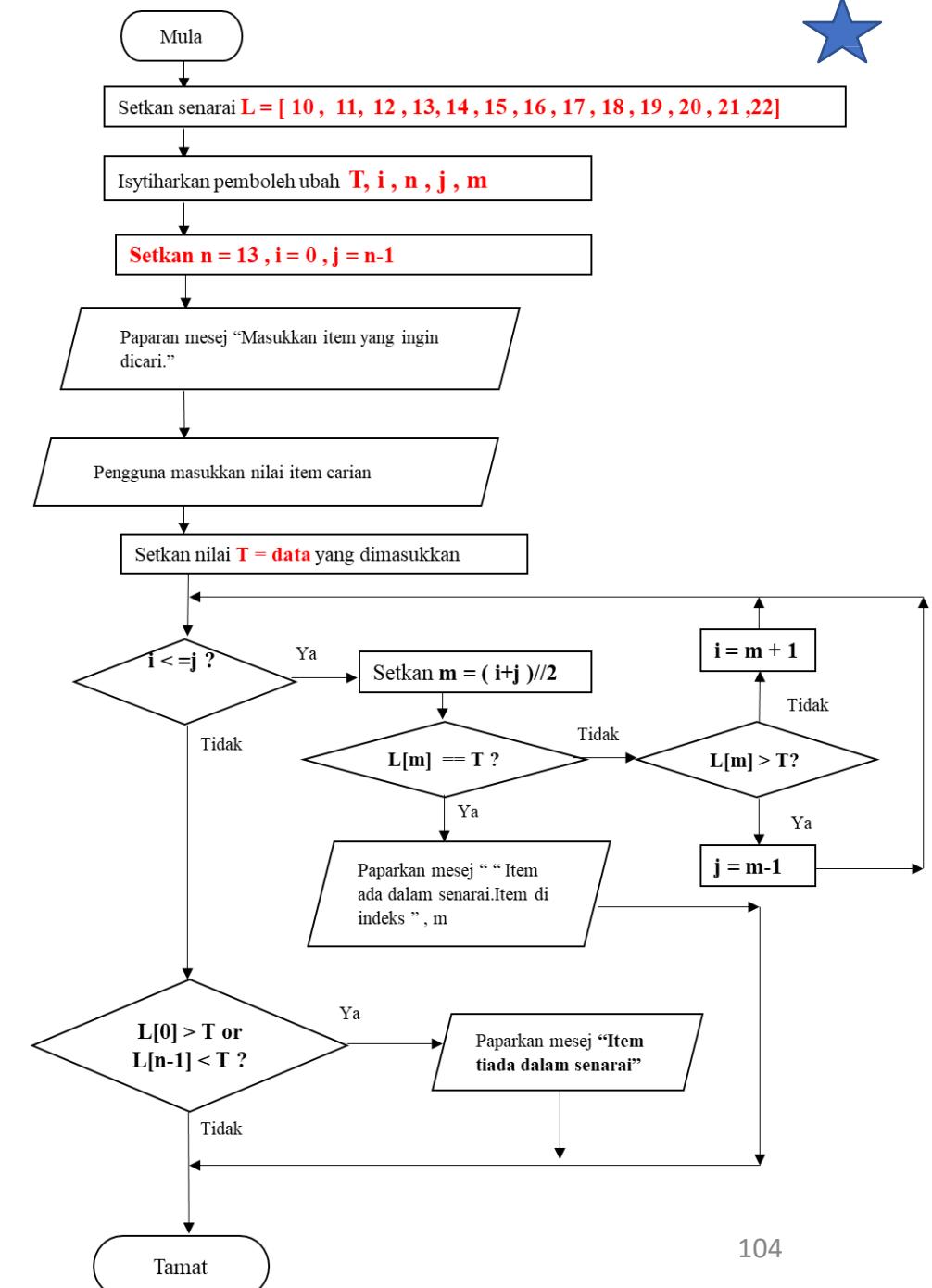
jika L[m] < T , kedudukan item pertama ialah i = m+1

- Teknik carian yang melipatkan keputusan dwipilihan.
- Sebelum melakukan binary search, item-item dalam senarai perlu **diisih dalam urutan menaik**.
- Lebih sesuai digunakan pada **senarai yang mempunyai item yang banyak**.



Pseudokod Binary Search

1. Mula
2. Setkan senarai $L = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]$
3. Isytiharkan **pemboleh ubah** T, i, n, j, m
4. Setkan $n = 13, i = 0, j = n-1$
5. Paparan mesej "Masukkan item yang ingin dicari."
6. Pengguna masukkan nilai item carian
7. Setkan nilai $T = \text{data yang dimasukkan}$
8. **while** $i \leq j$
9. Setkan $m = (i+j) // 2$
10. **Jika** $L[m] == T$
 - 11. Paparkan mesej " Item ada dalam senarai.Item di indeks ", m
 - 12. Keluar Gelung. Langkau ke Langkah 24
13. **Jika Tidak**
 - 14. Jika $L[m] > T$
 - 15. $j = m-1$
 - 16. **Jika Tidak**
 - 17. $i = m+1$
 - 18. Tamat Jika
 - 19. Tamat Jika
 - 20. Ulang Langkah 8
 - 21. **Jika** $L[0] > T \text{ or } L[n-1] < T$
 - 22. Paparkan mesej "Item tiada dalam senarai"
 - 23. Tamat Jika
 - 24. Tamat





Sort

Proses mengisih atau menyusun item-item dalam suatu senarai linear mengikut urutan tertentu

Kepentingan sort :

- Data yang telah diisih (*sort*) menjadikan **carian lebih cekap , percantuman lebih efisien** dan **memudahkan pemprosesan data dalam tertib yang dikehendaki.**
- Lebih efisien **untuk menghapuskan atau mencantumkan data duplikasi .**



BUBBLE SORT

- Teknik pengisihan untuk mengisih item-item dalam senarai mengikut urutan menaik atau menurun.
- *Bubble sort* sesuai digunakan pada senarai yang mempunyai **item yang sedikit** dan **item-item yang terdiri daripada integer dan abjad**.



Kelebihan & kelemahan BUBBLE SORT

- *Bubble Sort* paling ringkas dan mudah.
- Kaedah ini tidak efektif dan kelajuan isihan adalah perlahan



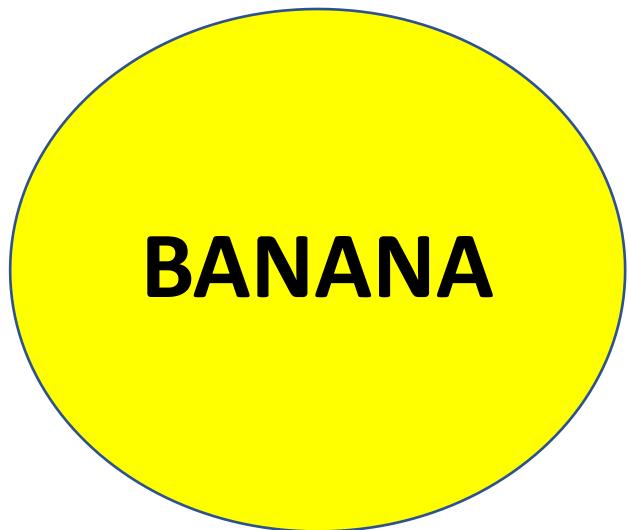
BUBBLE Sort

Bandingkan item pertama dengan item kedua. Jika item pertama lebih besar daripada item kedua, tukar kedudukan kedua-dua item tersebut.

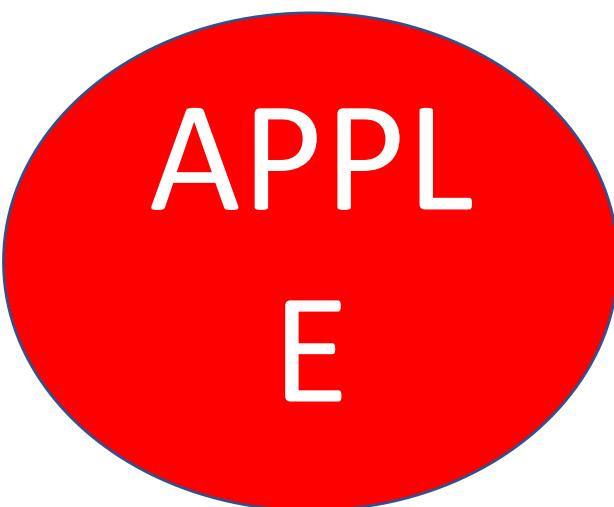
Jika item pertama lebih kecil daripada item kedua, kekalkan kedudukan kedua-dua item.

Bandingkan item kedua dengan item ketiga, dan seterusnya sehingga dua item terakhir. Tukarkan kedudukan jika item-item tidak berada dalam urutan yang betul.

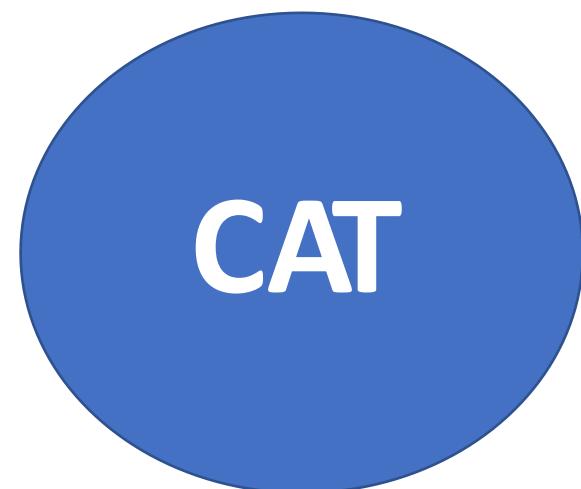
Perbandingan ini berulang sehingga item-item dalam senarai berada dalam urutan menaik.



Item 1



Item 2



Item 3



Menulis pseudokod dan melukis carta alir Bubble Sort Menaik

Setkan senarai $L = [4,1,9,5,3]$

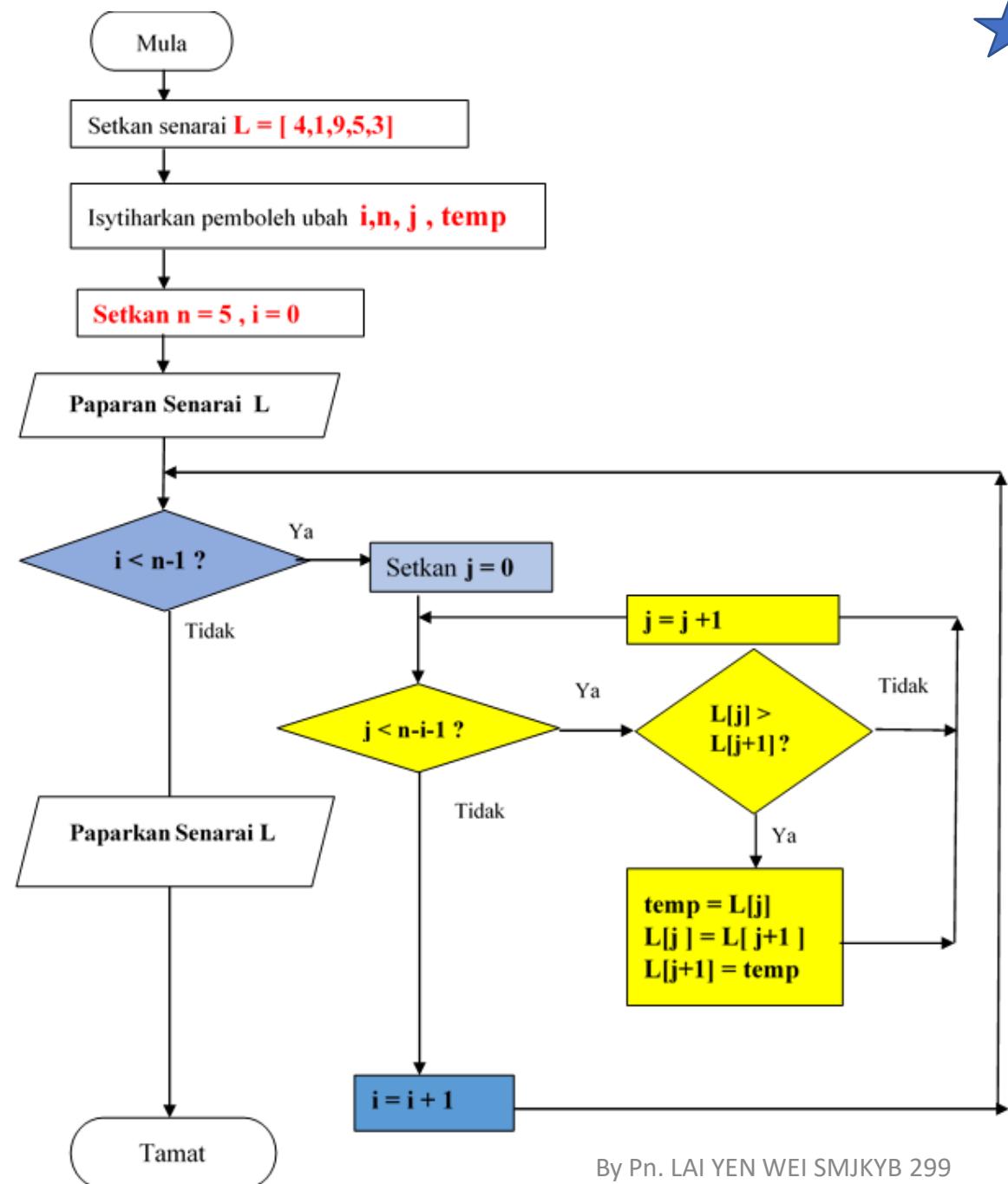
(L = List, n = bilangan item)

Pemboleh ubah	Penerangan
L	senarai yang mengandungi n item.
n	pemboleh ubah yang mewakili bilangan item dalam senarai L.
i	pemboleh ubah bilangan lelaran (loop) dalam gelung.
j	pemboleh ubah yang mewakili kedudukan item dalam senarai L
temp	pemboleh ubah yang menyimpan data sementara membuat perbandingan.



Pseudokod Bubble Sort

1. Mula
2. Setkan senarai $L = [4,1,9,5,3]$
3. Isytiharkan pemboleh ubah i, n, j, temp
4. Setkan $i = 0, n = 5, \text{temp} = 0$
5. Paparkan Senarai L
6. for $i < n-1$
7. $j = 0$
8. for $j < (n-i-1)$
 - 9. Jika $L[j] > L[j+1]$
 - 10. $\text{temp} = L[j]$
 - 11. $L[j] = L[j+1]$
 - 12. $L[j+1] = \text{temp}$
 - 13. Tamat Jika
 - 14. Kirakan $j = j + 1$
15. Ulangkan Langkah 8
16. kirakan $i = i + 1$
17. Ulangkan Langkah 6
18. Paparkan Senarai L
19. Tamat





Bucket sort

- Suatu **isihan** yang **menyusun item-item** dalam senarai **mengikut kategori** .
- Item-item akan dimasukkan ke dalam baldi mengikut kategori yang ditetapkan.
- Menggunakan 2 algoritma BUCKET SORT dan BUBBLE SORT
- Item-item akan diisihkan mengikut urutan menaik atau urutan menurun.
- Item-item dalam baldi yang telah diisih akan dicantumkan semula ke dalam senarai.



Formula untuk menentukan Baldi

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

Bilangan Baldi = had atas

*bilangan item
dalam senarai*

L senarai list

n ialah pemboleh ubah yang mewakili bilangan item dalam senarai L.

nB ialah pemboleh ubah yang mewakili bilangan baldi yang diperlukan.



Senarai L=[2 , 9 , 4 , 6 , 7 , 1 , 10 , 5 , 3 , 8]

Senarai Baharu =[]

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

$$n = 10$$

$$\begin{aligned} nB &= \text{had atas } \sqrt{10} \\ &= 3.162 \end{aligned}$$

$$nB = 4$$



Tips Pseudokod Cikgu Amanda :

- Senarai $L = [2, 9, 4, 6, 7, 1, 10, 5, 3, 8]$
- n = Bilangan item dalam senarai
- Bilangan baldi ditentukan dengan menggunakan formula $nB = \text{had atas } \sqrt{n}$
- Contoh, $n = 10$, $nB = \text{ceil}(\sqrt{10}) = 3.162$ maka bilangan baldi = 4
- Divider = $\text{ceil}[(\max(L) + 1) / nB]$
= $\text{ceil}[(10 + 1) / 4]$
= $\text{ceil}[11/4]$
= $\text{ceil}[2.75]$
= 3
- Wujudkan baldi kosong, SenaraiBaru = [[], [], [], []]
- Tentukan item ke baldi = $L[a]//\text{Divider}$
- Jika baldi tidak kosong, isihkan item

Divider 是拿来决定item 丢进哪一个桶



Menulis pseudokod dan melukis carta alir Bubble Sort Menaik

Tips :

(**L = List, n = bilangan item**)

L ialah senarai yang mengandungi item-item perlu diisih, iaitu senarai nama ahli .

i ialah pemboleh ubah yang mewakili kedudukan item dalam senarai L.

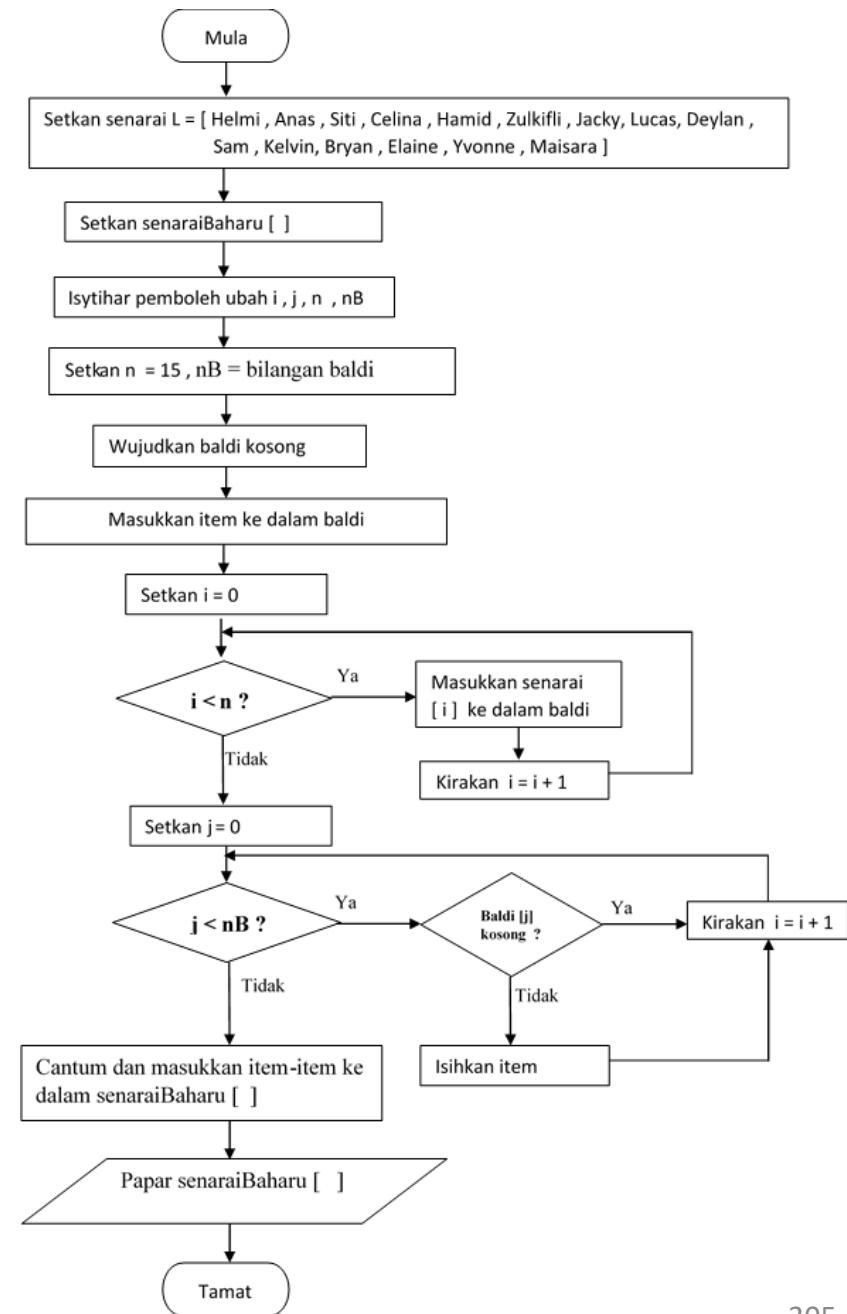
n ialah pemboleh ubah yang mewakili **bilangan item** dalam senarai L.

nB ialah pemboleh ubah yang mewakili **bilangan baldi yang diperlukan**.

j ialah pemboleh ubah yang mewakili kedudukan baldi.



1. Mula
2. Setkan senarai $L = [\text{Helmi}, \text{Anas}, \text{Siti}, \text{Celina}, \text{Hamid}, \text{Zulkifli}, \text{Jacky}, \text{Lucas}, \text{Deylan}, \text{Sam}, \text{Kelvin}, \text{Bryan}, \text{Elaine}, \text{Yvonne}, \text{Maisara}]$
3. Setkan senaraiBaru []
4. Isytihar pemboleh ubah i, j, n, nB
5. Setkan $n = 15$
6. Setkan $nB = \text{bilangan baldi}$
7. Wujudkan baldi kosong
8. Masukkan item ke dalam baldi
9. Setkan $i = 0$
10. for $i < n$
 - 11. Masukkan senarai $[i]$ ke dalam baldi
 - 12. Kirakan $i = i + 1$
13. Ulang langkah 10
14. Setkan $j = 0$
15. for $j < nB$
 - 16. **Jika** Baldi $[j]$ kosong
Kirakan $j = j + 1$
 - 17. **Jika Tidak**
Isihkan item
 - 18. Kirakan $j = j + 1$
19. **Tamat Jika**
20. Ulang langkah 15
21. Cantum dan masukkan item-item ke dalam senaraiBaru []
22. Papar senaraiBaru []
23. Tamat



4.11 Pangkalan DATA

Pangkalan data :

- Tempat pengumpulan dan penyimpanan data secara berpusat bagi sistem maklumat yang berasaskan komputer.
- Koleksi data yang saling berhubung antara satu sama lain yang disimpan **tanpa pengulangan data yang tidak dikehendaki**
- Dapat memudahkan data dan maklumat untuk disimpan , dicapai dan dikemas kini.

4.11 Pangkalan DATA

- Pangkalan data pertama diciptakan pada tahun 1960.
- SABRE System yang digunakan oleh International Business Machine (IBM) untuk membantu American Airlines bagi mengatasi berkenaan data penempahan penerbangan.

Cara pengurusan data

Pengurusan
Data

Sistem Fail

1. Sistem fail secara manual
2. Sistem fail elektronik

Sistem Pangkalan Data

- Untuk membina sistem pemprosesan data yang cekap dan menyatupadukan penyimpanan, pemprosesan, capaian dan keselamatan data



KEMENTERIAN PENDIDIKAN MALAYSIA

Pendidikan Berkualiti Insan Terdidik Negara Sejahtera

SISTEM ANALISA PEPERIKSAAN SEKOLAH (SAPS)

SEMAK SLIP PEPERIKSAAN PELAJAR

SILA HUBUNGI PIHAK SEKOLAH UNTUK MENGETAHUI TARikh SEMAKAN KEPUTUSAN PEPERIKSAAN

LANGKAH 1: CARIAN MURID

SILA MASUKKAN NOMBOR SIJIL LAHIR/ KAD PENGENALAN PELAJAR (**TANPA "-" ATAU "SPACE"**)

No KP / Sijil Lahir :

LANGKAH 2 : CARIAN SEKOLAH

SILA MASUKKAN MAKLUMAT SEKOLAH PELAJAR, SEKOLAH TERKINI ATAU SEBELUM INI

Negeri :

Nama Sekolah :

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

BAB 4.1 Pangkalan DATA DAN SQL

- Data memainkan peranan yang penting dalam proses membuat keputusan.
- Maklumat adalah hasil daripada pengumpulan, pemprosesan dan penganalisaan data yang boleh digunakan untuk membuat sesuatu keputusan.

Pangkalan data :

- **Tempat pengumpulan dan penyimpanan data secara berpusat** bagi sistem maklumat yang berasaskan komputer.
- Koleksi data yang saling berhubung antara satu sama lain yang disimpan **tanpa pengulangan data yang tidak dikehendaki**
- Dapat **memudahkan data dan maklumat untuk disimpan , dicapai dan dikemas kini.**
- Pangkalan data pertama diciptakan pada tahun 1960.
- SABRE System yang digunakan oleh International Business Machine (IBM) untuk membantu American Airlines bagi mengatasi berkenaan data penempahan penerbangan.

Pengurusan Data

Sistem Fail

1. **Sistem fail secara manual**
2. **Sistem fail elektronik**

Sistem Pangkalan Data

- Untuk membina sistem pemprosesan data yang cekap dan menyatupadukan penyimpanan, pemprosesan, capaian dan keselamatan data

Ciri-ciri sistem fail secara manual dan elektronik

BT m/s 115

Ciri-ciri	Sistem Fail Secara Manual	Sistem Fail Secara Elektronik
Cara data disimpan	Data disimpan dalam <u>fail kertas</u>	<u>Menggunakan komputer</u> Data direkod dalam <u>fail komputer</u> .
Tempat simpanan fail	Fail kertas disimpan dalam <u>kabinet fail</u> <u>Ruang yang diperlukan untuk menyimpan fail</u>	Data disimpan dalam <u>peranti storan komputer</u>
Kemaskini data	<u>Rekod sukar diubah</u>	<u>Rekod senang diubah</u>
Capaian data	Capaian ke atas data <u>mengambil masa yang lama</u>	Capaian ke atas data <u>memakan masa yang singkat.</u>

Fungsi, kebaikan dan contoh kegunaan sistem pangkalan data

Fungsi	Kebaikan	Contoh
<ul style="list-style-type: none"> Membina sistem pemprosesan data yang cekap Menyepadukan koleksi data 	<ul style="list-style-type: none"> Memudahkan capaian, pengurusan dan pengemaskinian data Mengelakkan pengulangan data Menjaga integriti data Menjaga keselamatan data 	<ul style="list-style-type: none"> Penggunaan Sistem Inventori Barang di pasar raya Penggunaan Sistem Analisis Peperiksaan Sekolah (SAPS) untuk semua sekolah di Malaysia

Kebaikan evolusi dari sistem fail ke sistem pangkalan data.

Sistem Fail

Capaian ke atas data tidak boleh dilakukan secara serentak

Keselamatan data tidak terjamin

Berlaku pengulangan data

Perkongsian data adalah terhad

Sistem Pangkalan Data.

Data boleh dicapai oleh semua pihak secara serentak dalam masa yang singkat

Mempunyai kata laluan yang khusus dan sandaran data yang mantap

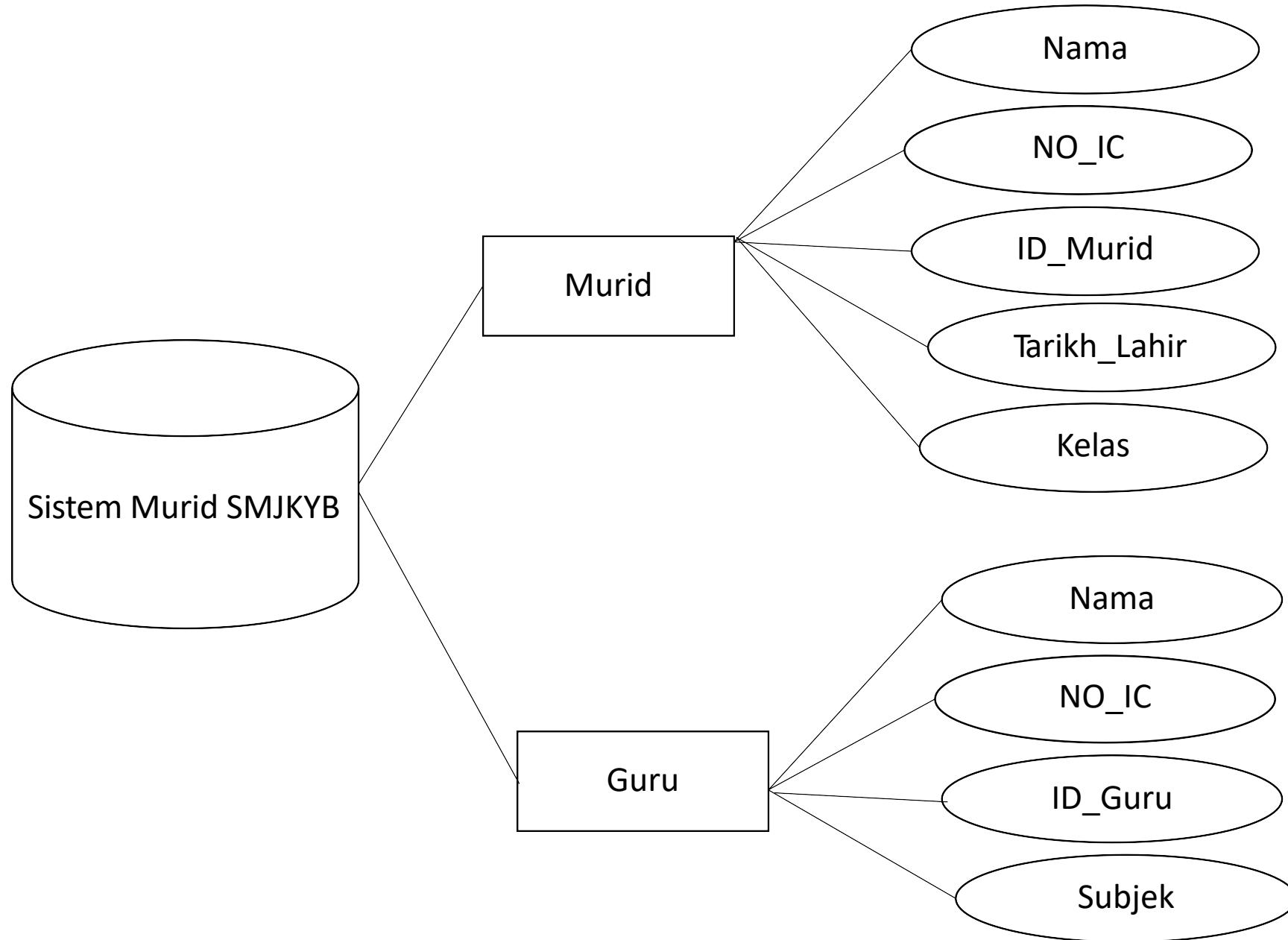
Kawalan data secara berpusat menjadikan pengulangan data yang sama dapat dielakkan

Data dapat dikongsikan dengan semua pihak yang terlibat secara efisien .

BT m/s 117

4.1.1 SQL (Structured Query Language)

- *Structured Query Language (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. Koleksi data yang saling berhubung antara satu sama lain yang disimpan tanpa pengulangan data yang tidak dikehendaki
- Contoh pernyataan SQL : SELECT... FROM, SELECT...WHERE dan SELECT...ORDER BY.



4.1.2 Entiti dan atribut

Komponen dalam pangkalan data

Jadual / Table

Entiti

Maklumat dalam Jadual

Atribut

- Satu kumpulan objek yang sama jenis
- Bersifat unik
- Wujud dalam persekitaran, iaitu tempat di mana pangkalan data berada atau tempat di mana anda hendak membangunkan pangkalan data
- Contoh entiti: orang, tempat atau benda, Maklumat Murid
- Mempunyai data mengenainya yang boleh dikutip dan disimpan
- Diwakili oleh simbol segi empat dalam gambar rajah perhubungan entiti

- Menerangkan ciri-ciri entiti
- Mempunyai satu atau lebih ciri
- Merupakan lajur dalam jadual pangkalan data
- Diwakili oleh bentuk bulatan bujur dalam gambar rajah perhubungan entiti

Lengkapkan gambar rajah perhubungan antara Entiti dan atribut bagi jadual berikut ?

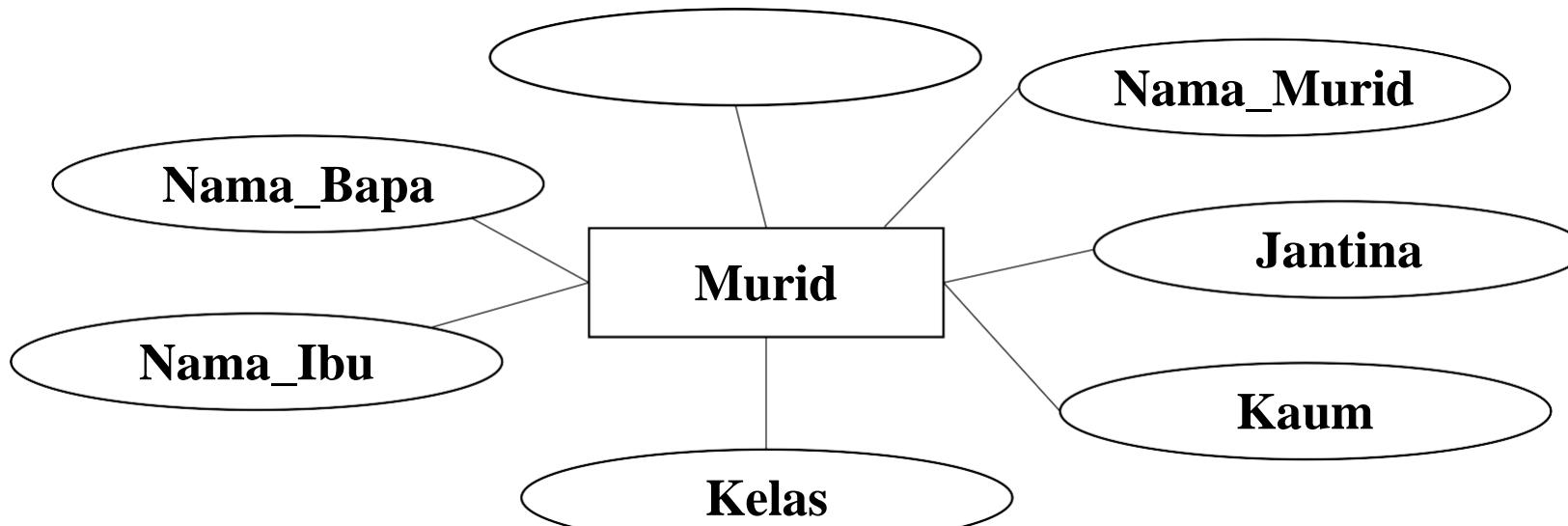
Entiti

Atribut (Lajur atau medan)

REKOD

MURID							
ID_Murid	Nama_Murid	Jantina	Kaum	Kelas	Nama_Bapa	Nama_Ibu	
P001	Ali bin Azmi	L	Melayu	3R	Azmi Bin Saifuddin	Siti Nor Binti Baharuddin	
P002	Aminah binti Yusof	P	Melayu	3V	Yusof Bin Jaafar	Khatijah Kassim Bin Nor	
P003	Chong Yee Liang	P	Cina	3Y	Chong Ah Fatt	Tan Hui Hui	
P004	Dayang Bin Minsu	P	Melayu	3I	Minsu Bin Allif	Maminah Binti Yusof	
P005	Denish A/L Kathigasu	L	India	3O	Kahtigasu A/L Muthusarr	Nomitha A/P Rajandran	

Jawapan :



4.1.3 Kekunci PRIMER , KEKUNCI ASING

Dua Jenis
Kekunci
dalam
sistem
pangkalan
data

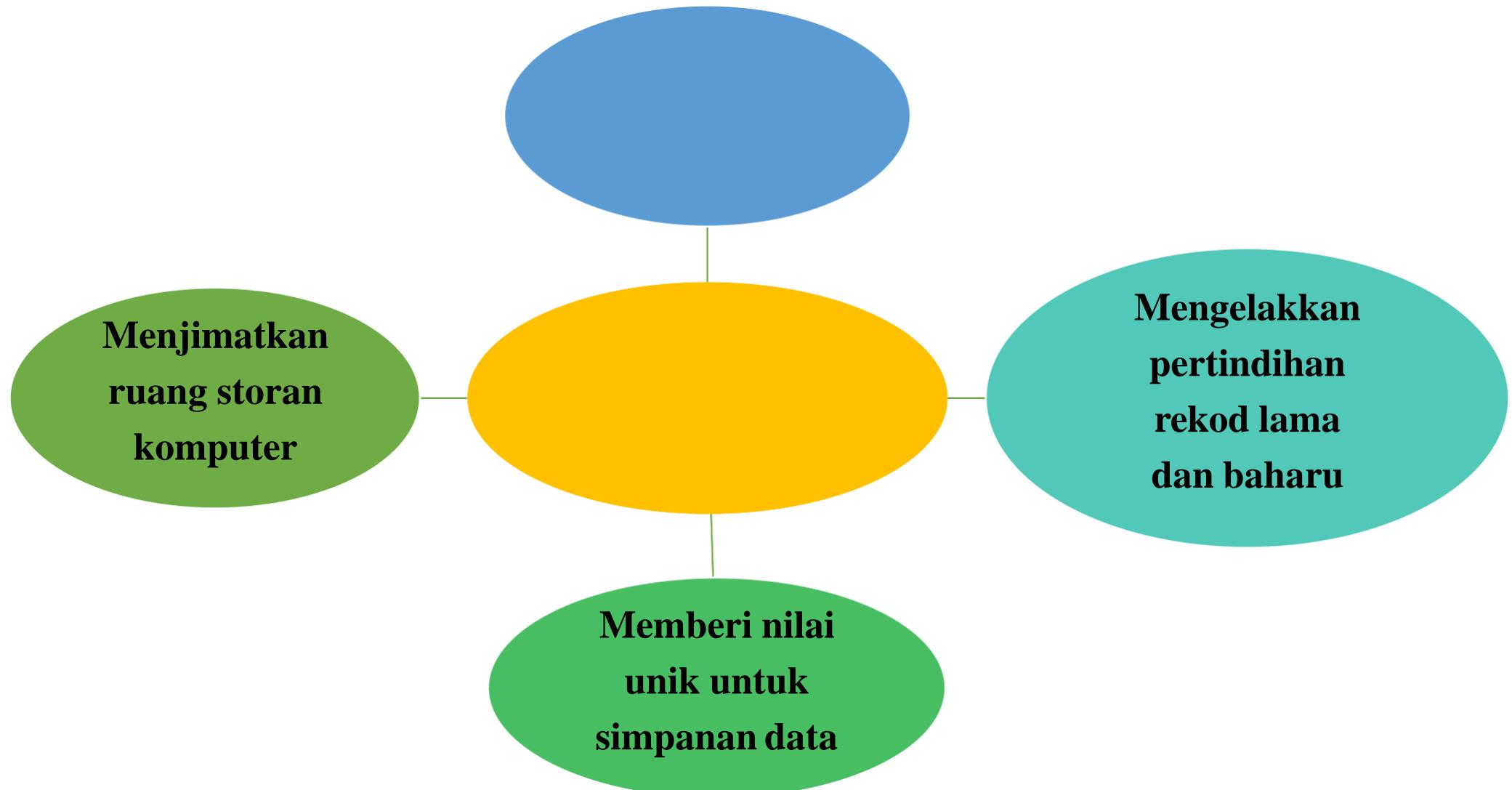
Kekunci
Primer

- Atribut dengan **nilai unik**
- Tidak boleh mengandungi data yang berulang/ sama/ Tidak ditinggalkan kosong (*null*)
- Digunakan untuk mengenal pasti rekod dalam jadual
- Setiap entiti mempunyai sekurang-kurangnya **satu kekunci primer**

Kekunci
Asing

- Atribut dalam satu jadual hubungan
- Atribut yang merupakan **kekunci primer bagi satu jadual hubungan yang lain**
- Mengaitkan dua atau lebih entiti yang mempunyai hubungan
- Mengaitkan rekod-rekod dari dua atau lebih daripada jadual yang mempunyai hubungan

Kepentingan kekunci primer dalam satu sistem pangkalan data



4.1.4 Kekardinalan (Cardinality) antara entiti dan hubungan

- **Kekardinalan** merujuk kepada **perhubungan antara entiti-entiti dalam satu pangkalan data.**
- Jenis kekardinalan :
 - a. **One to one (1:1)**
 - b. **One to many (1:M)**
 - c. **Many to many (M:N)**
- Perhubungan antara dua entiti lazimnya diwakili dengan menggunakan kata kerja.
- Dalam gambar rajah perhubungan entiti (*Entity Relationship Diagram —ERD*), hubungan dilukis dengan simbol rombus.



4.1.4 KeKARDinalan (Cardinality) antara entiti dan hubungan

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.



Contoh:

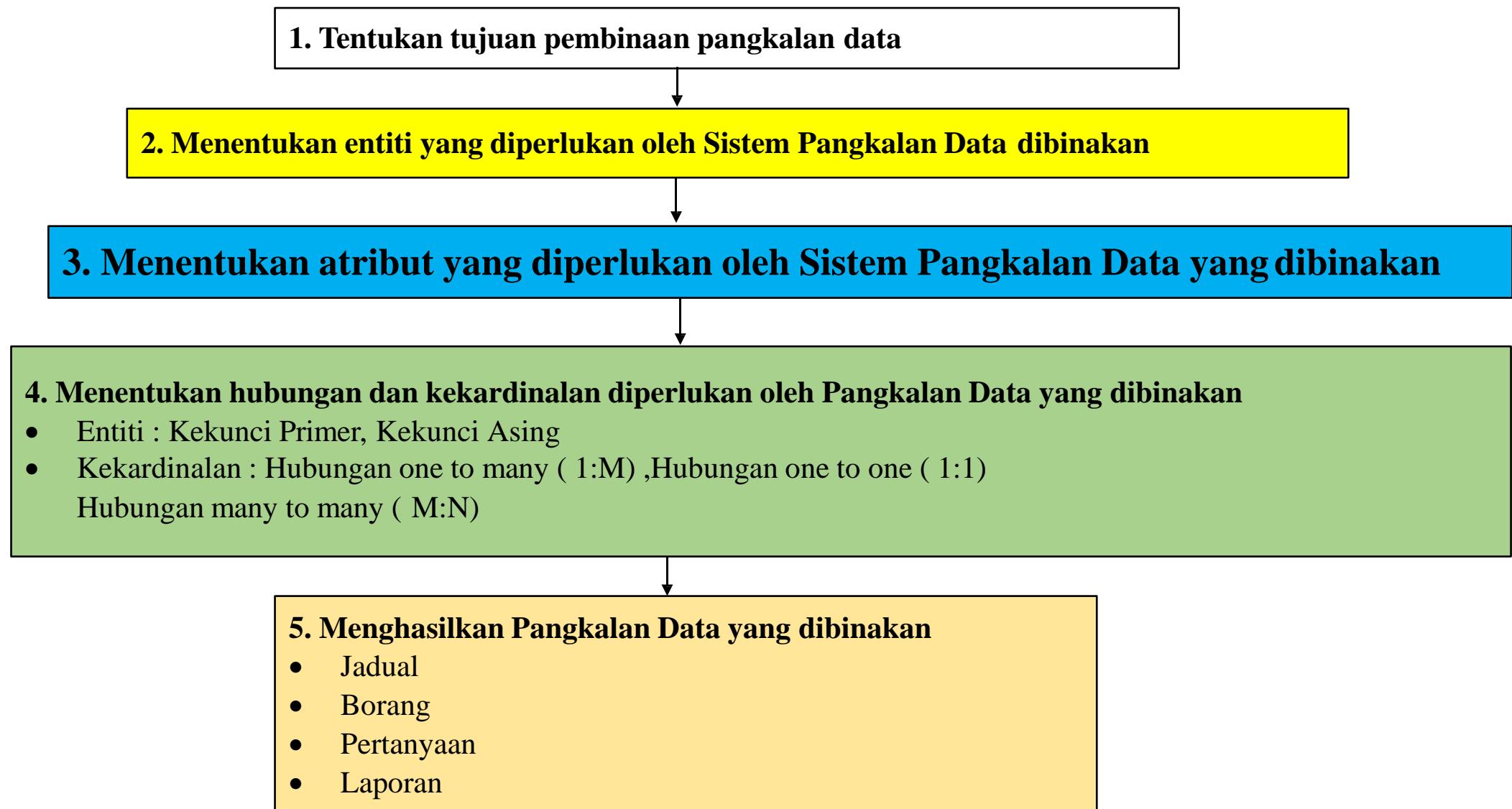
Seorang murid dapat meminjam banyak buku tetapi sebuah buku hanya boleh dipinjam oleh seorang murid sahaja. Rajah berikut menunjukkan perhubungan antara satu entiti dengan lebih daripada satu entiti yang lain.



4.1.5 Membina pangkalan Data

1. Pangkalan data dibinakan dengan menggunakan **perisian sistem pengurusan pangkalan data , DBMS (Database Management System).**
2. Perisian (Software) DBMS :
 - a. Microsoft Access
 - b. Microsoft SQL Server
 - c. Oracle
 - d. Informix
 - e. DB2
 - f. Sybase
 - g. Postgress
 - h. MySQL

Langkah-langkah untuk membinakan pangkalan data



Objek-objek pangkalan data dalam perisian aplikasi Microsoft Access

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

Capaian kepada perisian aplikasi Microsoft Access

Langkah 1:

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



Langkah 2:

Klik ikon Blank database 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.

Langkah-langkah membina pangkalan data yang terdiri daripada entiti (table) MURID

Langkah 1:

Setelah melengkapkan langkah-langkah dalam pencapaian kepada perisian aplikasi Microsoft Access , paparan di bawah diperhatikan.

Langkah 2:

Klik ikon **Design View** untuk mendapatkan paparan table

Langkah 3:

Ubahkan nama Table1 kepada **MURID** dan klik **OK**

Langkah-langkah membina pangkalan data yang terdiri daripada atribut (field)

Langkah 1:

Klik ikon **View** dan pilih **Design View**. Kemudian, 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.

Jenis data yang boleh ditetapkan dalam perisian aplikasi Microsoft Access

Jenis data	Penerangan jenis data
<i>AutoNumber</i>	Menjana pembilangan 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.

4.1.6 Menghasilkan Borang dan Memasukkan Data Melalui Borang

- Perisian aplikasi *Microsoft Access* membolehkan anda menghasilkan borang dijana secara automatik menggunakan **Form Wizard**.
- Borang yang dihasilkan digunakan untuk memasukkan, mengedit atau memaparkan data dan rekod dalam jadual hubungan sesuatu entiti.
- 4 jenis susunan atur FORM
 - Columnar – Rekod dipaparkan satu demi satu
 - Tabular – Semua rekod dipaparkan dalam bentuk jadual yang menarik
 - Databasesheet - Semua rekod dipaparkan dalam bentuk helaian data yang menarik
 - Justified – Semua medan field disusunkan pada baris-baris blok yang mempunyai margin kiri dan kanan yang tetap

Penerangan butang-butang yang terdapat pada bahagian bawah borang

Butang	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
	Carian rekod.

4.1.9 Menjana Laporan Berdasarkan Hasil Pertanyaan (Query)

Dalam perisian aplikasi *Microsoft Access*, cara yang paling mudah untuk menghasilkan laporan adalah dengan menggunakan **Report Wizard**.

Langkah-langkah menjana laporan dengan menggunakan Report Wizard

1. Pada menu bar, klik Create. Kemudian, klik **Report Wizard**.
2. Paparan tetingkap Report Wizard akan muncul. Pilih jadual atau *query* yang anda ingin gunakan untuk menghasilkan laporan.
3. Setelah jadual atau pertanyaan dipilih, tetingkap **Available Fields** menunjukkan semua medan yang ada di dalam jadual. Klik ikon >> untuk memilih semua medan.
4. Medan yang dipilih akan muncul dalam tetingkap **Selected Fields**. Klik **Next**.
5. Anda boleh klik ikon > atau < untuk menentukan susunan medan yang dipaparkan dalam laporan. Klik **Next**.
6. Anda boleh memilih medan untuk tujuan isihan dan klik **Ascending** atau **Descending** untuk menentukan isihan secara menaik atau menurun. Klik **Next**.
7. Seterusnya, pilih cara susun atur (*Layout*) dan orientasi (*Orientation*) untuk menetapkan format laporan anda. Klik **Next**. Terdapat **3 jenis layout : Stepped , Block , Outline**
8. Masukkan nama laporan anda. Klik **Finish**.



```
SELECT column1, column2, ...
FROM table_name;
```

```
SELECT * FROM table_name;
```

WHERE sintaks , AND , OR, NOT , ORDER BY ..ASC|DESC

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

```
SELECT Nama_Murid
FROM MURID
ORDER BY Nama_Murid ASC ;
```

Nama_Murid
Ali bin Azmi
Aminah binti Yusof
Arul A/L Param
Chong Yee Liang
Dayang Bin Minsu
Denish A/L Kathigasu
Lim Chen Wei
Priya David
Rachel Lim Yee Ling
Siti Noor Binti Ali
Tan Chen Hong
Tan Mei Mei
Tang Siang Heng
Thulasi A/P Mariappan
*

```
SELECT Nama_Murid
FROM MURID
ORDER BY Nama_Murid DESC ;
```

Nama_Murid
Thulasi A/P Mariappan
Tang Siang Heng
Tan Mei Mei
Tan Chen Hong
Siti Noor Binti Ali
Rachel Lim Yee Ling
Priya David
Lim Chen Wei
Denish A/L Kathigasu
Dayang Bin Minsu
Chong Yee Liang
Arul A/L Param
Aminah binti Yusof
Ali bin Azmi



INSERT INTO Sintaks

INSERT INTO *table_name* (*column1*, *column2*, *column3*, ...)
VALUES (*value1*, *value2*, *value3*, ...);

INSERT INTO MURID (ID_Murid, Nama_Murid, Jantina, Kaum, Kelas)

VALUES ('P015', 'Yong Kai Le', 'P', 'Cina', '3R');

UPDATE *table_name*
SET *column1* = *value1*, *column2* = *value2*, ...
WHERE *condition*;

SELECT MIN(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT MAX(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT SUM(*column_name*)
FROM *table_name*
WHERE *condition*;

SELECT AVG(*column_name*)
FROM *table_name*
WHERE *condition*;

Bab 4.2 Struktur Kod Arahan

4.2.1 Menerangkan fungsi struktur beikut dalam :

- a. Function
- b. Procedure

Pengekodan Algoritma

Aturcara Utama

Subatur cara

Function

- Dugunakan bagi subtugas yang perlu **memulangkan satu nilai** selepas tugas itu diselesaikan.
- Sesuai digunakan bagi proses pengiraan kerana kod yang sama tidak perlu dituliskan berulang kali.

Procedure

- Dugunakan bagi subtugas yang perlu dilaksanakan berulang kali **tanpa pemulangan nilai.**



?

Ia menjadikan kod arahan seluruh atur cara menjadi **lebih kemas , teratur , sistematisik , bersifat modular dan lebih mudah diselenggara**

Persamaan	
Mempunyai nama khusus yang esklisit .	
Berparameter atau tidak berparameter	
Melaksanakan satu urutan arahan bagi menyelesaikan sesuatu tugas khusus	
Boleh digunakan berulang kali dalam atur cara yang sama atau berbeza sekiranya perlu	
Perbezaan	
Function	Procedure
Ada dua jenis : a. Function dalaman (built-in) b. Function dihasilkan sendiri (user-defined)	Dituliskan oleh pengguna.
Memulangkan satu atau beberapa nilai def nama_function (parameter): badan function return (nilai)	Tidak memulangkan sebarang nilai kecuali nilai secara lalai : none def nama_procedure (parameter): badan procesure
Perlu menerima nilai input (argumen)	Tidak semestinya menerima nilai input (argumen)

4.2.2 Penggunaan Pernyataan Function

Function

Function dalaman (built-in)

- Fungsi yang **sedia ada dan disimpan dalam library bahasa pengaturcaraan**.
- Kod sumber untuk setiap built-in function **tidak boleh dilihat oleh pengatur cara**.

Contoh :

```
print ("Mari belajar BUILD_IN FUNCTION !")
M1 = int(input ("Masukkan markah 1 : "))
M2 = int(input ("Masukkan markah 2 : "))
M3 = int(input ("Masukkan markah 3 : "))

print ("Markah tertinggi ialah : ", max(M1,M2,M3))
```

```
print( ), input( ), int( ), max( )
```

Function dihasilkan sendiri (user-defined)

- Satu set arahan bagi melaksanakan suatu tugas khas yang akan berulang dalam arah utama.
- Mempunyai nama yang **deskriptif (描述性的)**, iaitu nama yang dapat mencerminkan tugasnya.

Contoh :

```
def checkstatus ( a ) :
    if a% 2 == 0:
        p = "Nombor Genap"
    else :
        p = "Nombor Ganjil"

    return p
```

```
Number = int(input ("Masukkan satu number : "))
Status = checkstatus ( Number )
print ( Number , "ialah" , Status )
```

Perbezaan dan Persamaan *Function dalaman (built-in)* dan *Function dihasilkan sendiri (user-defined)*

BT T3 ms 174

Persamaan

Mempunyai nama (bahagian pengepala)

Mempunyai badan function set arahan)

Mempunyai perkataan **return**

<u>Function dalaman (built-in)</u>	Perbezaan	<u>Function dihasilkan sendiri (user-defined)</u>
Tidak boleh dilihat	Kod Sumber	Boleh dilihat dan diubah
Perlu ada argumen	Argumen	Boleh ada atau tanpa argumen
Unik , deskriptif , terpelihara dan tidak boleh diubah.	Nama	Boleh diberi dengan sebarang nama yang tidak bercanggah dengan kata kunci
Datang sekali dengan bahasa pengaturcaraan	Function	Ditulis oleh pengatur cara

```
#你define一个新的 function ASK ( a )
#有 a - parameter , , 代表你callfunction 要送 variable 过来这里。
def ASK2 (a) : # a - parameter, 你用在你def function 头里面的
    print(a)
    print( "Ini ialah FUNCTION")
    b = "I LOVE CIKGU AMANDA!"
    return b # pemulangan nilai 把 b 送回去 main program
```

```
#main program
# 你callfunction ASK2 ( "ASK F3") - send argument "ASK F3" 去function ASK2
a = input ( "Masukkan kelas :")
result = ASK2( a) # result 是boleh ubah 你 hold着从 function ASK2 return 送回来的 value b
print ( result )
```

a 是argumen