



# **NOTA**

## **BAB 5**

# **PENGATURCARAAN**



## **5.1 KONSEP ASAS PENGATURCARAAN**

**5.1.1 MENDEFINISIKAN ATURCARA DAN BAHASA  
PENGATURCARAAN**

**5.1.2 GENERASI DAN ARAS BAHASA PENGATURCARAAN**

**5.1.3 PENDEKATAN BAHASA PENGATURCARAAN**

**5.1.4 PENTERJEMAHAN**

**5.1.5 ELEMEN ASAS DALAM PENGATURCARAAN**



**5.1.1**

**MENDEFINISIKAN**

**ATURCARA**

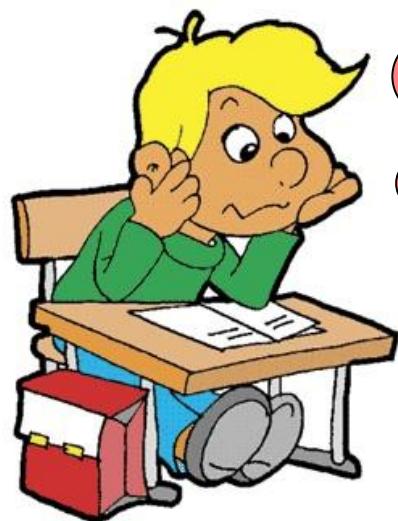
**DAN BAHASA**

**PENGATURCARAAN**

# MAKSUD ATURCARA



Apa yang  
dimaksudkan  
dengan  
aturcara???



Arahan – arahan tersusun  
Yang mengarahkan komputer  
Melakukan tugas tertentu

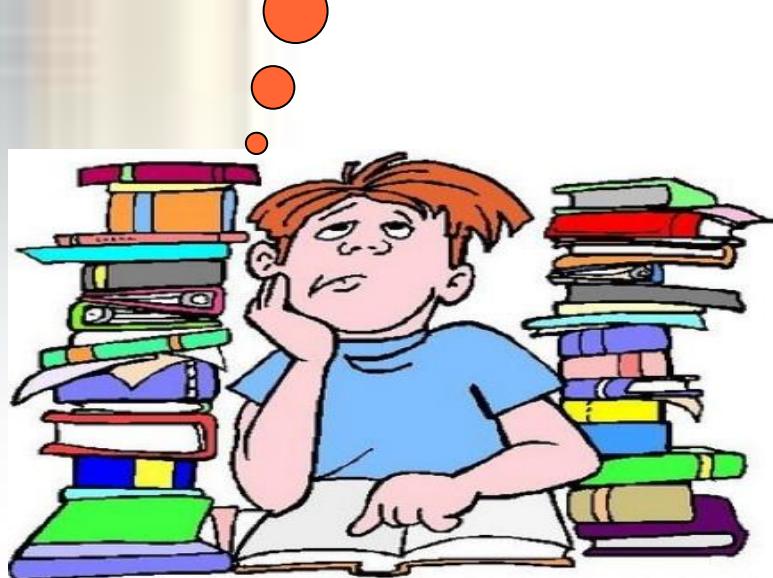


# MAKSUD BAHASA PENGATURCARAAN

Bahasa pengaturcaraan itu apa???

Bahasa buatan yang digunakan untuk menulis arahan yang boleh diterjemahkan ke dalam bahasa mesin dan kemudian dilaksanakan oleh komputer

Contohnya: C++, C, Phyton





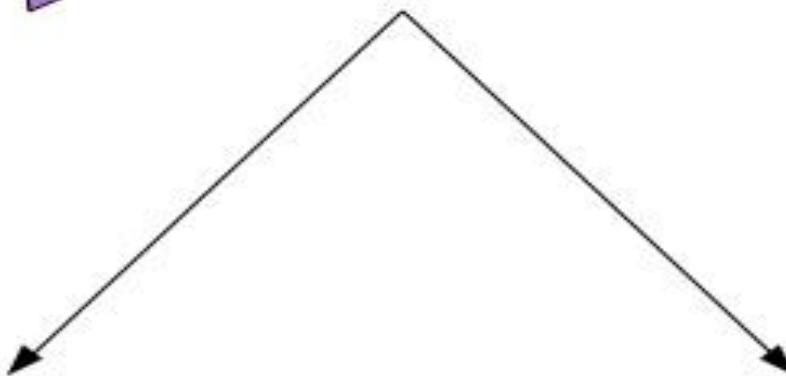
## **5.1.2**

# **GENERASI DAN ARAS BAHASA PENGATURCARAAN**

# GAMBARAN KESELURUHAN



## BAHASA PENGATURCARAAN ARAS RENDAH



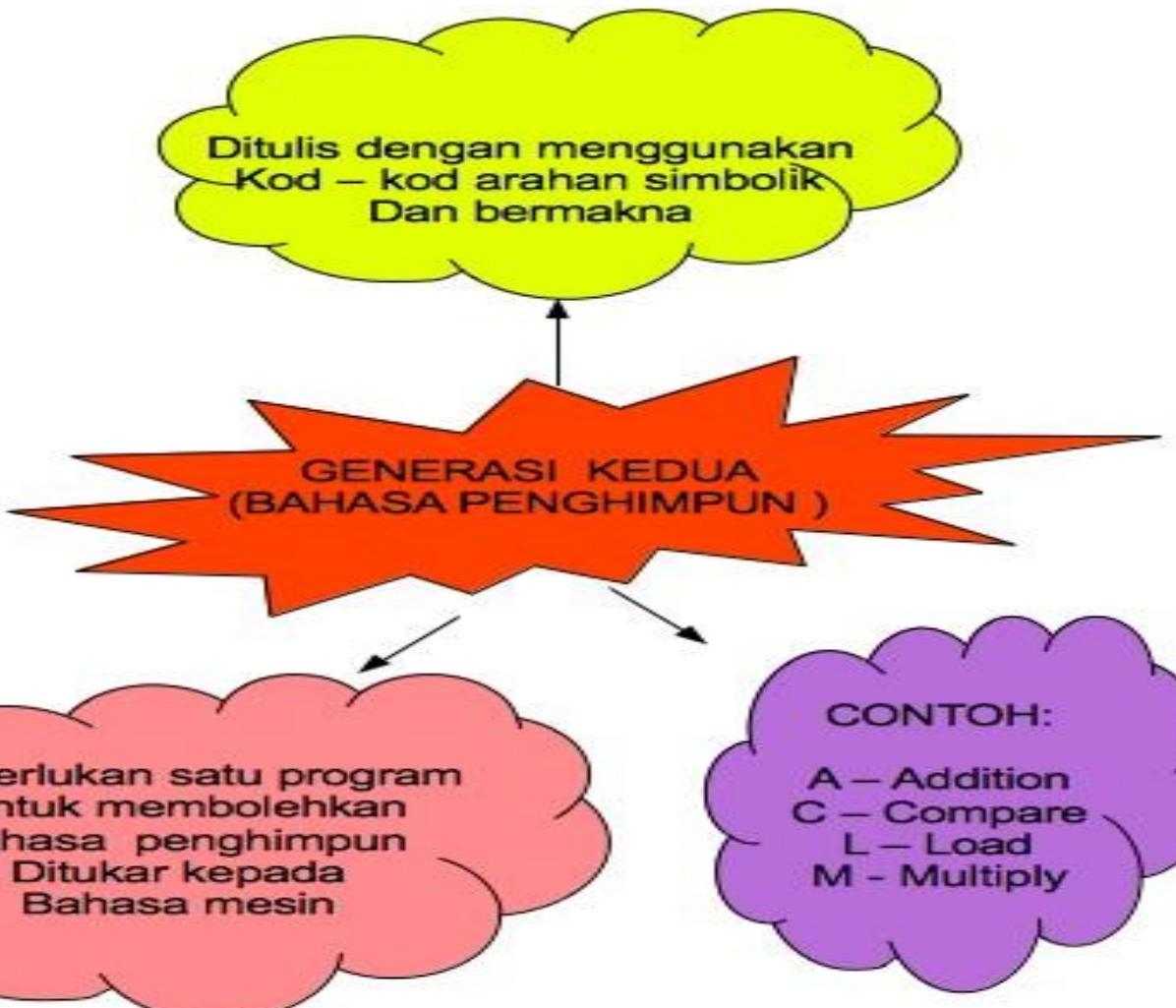
GENERASI PERTAMA  
(BAHASA MESIN)

GENERASI KEDUA  
(BAHASA PENGHIMPUN)

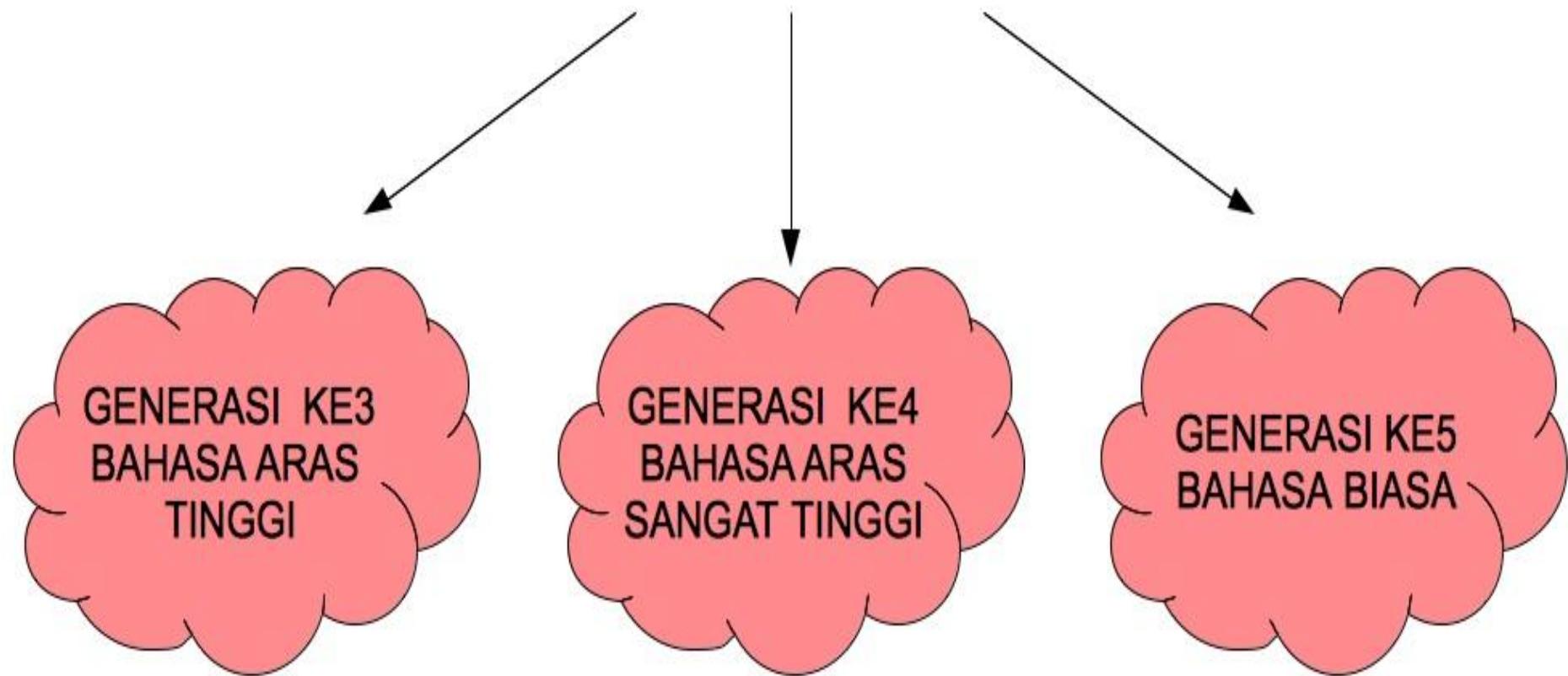
# GENERASI PERTAMA

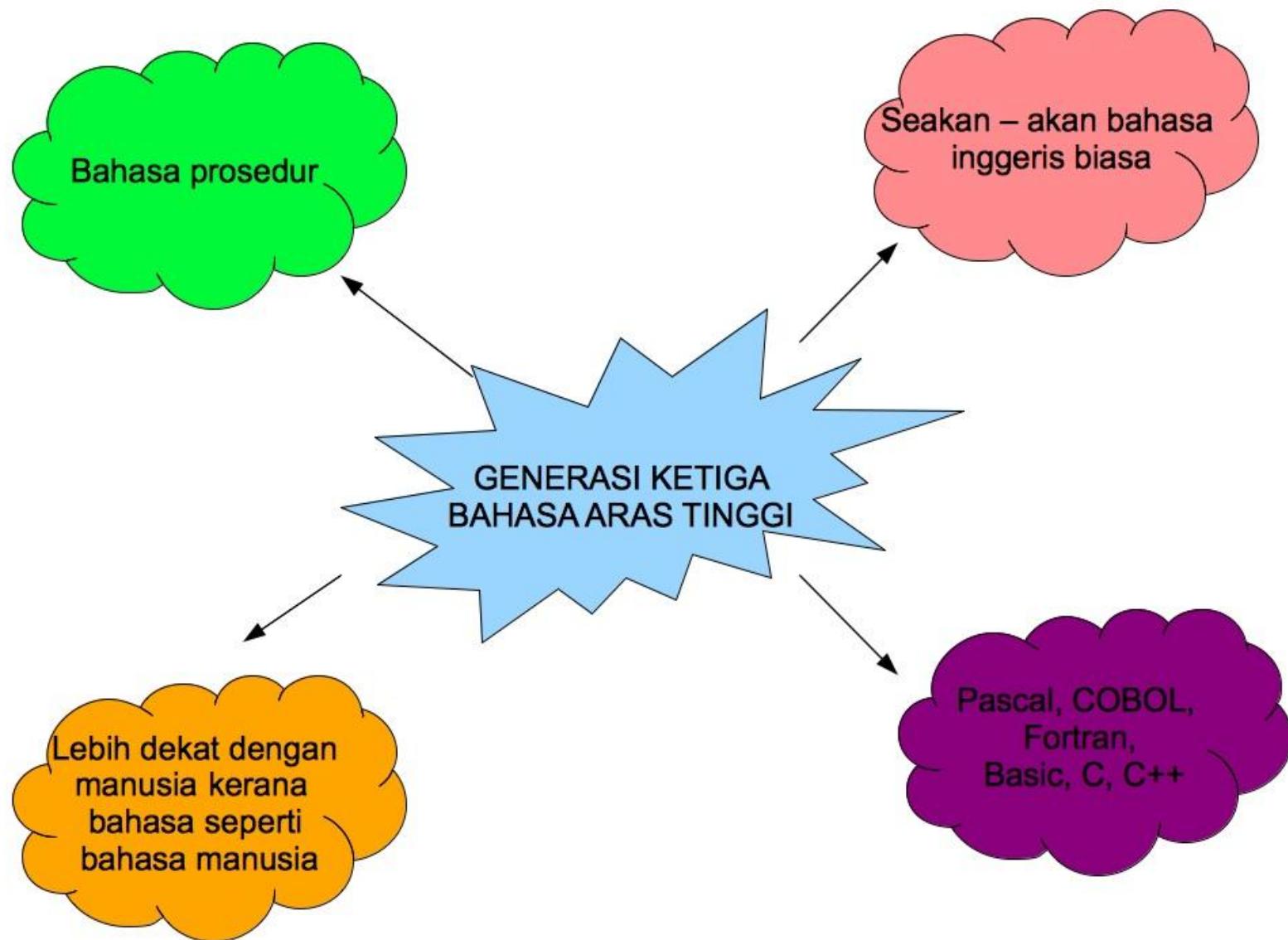


# GENERASI KEDUA



## BAHASA PENGATURCARAAN ARAS TINGGI





## GENERASI KEEMPAT BAHASA ARAS SANGAT TINGGI

Sangat terhad  
kepada  
aplikasi tertentu  
saaja

Menggunakan sintaks  
(berbentuk bahasa)  
yang  
tidak digunakan  
dalam  
bahasa pengaturcaraan  
yang lain

SQL,  
Nomad and focus





# **5.1.3**

## **PENDEKATAN**

## **BAHASA**

# **PENGATURCARAAN**

A cartoon illustration of a young boy with black hair and glasses, wearing a white shirt, sitting at a desk and reading an open orange book. He has a thoughtful expression, indicated by three purple thought bubbles above his head.

Pendekatan bahasa pengaturcaraan ada dua iaitu:

- Pendekatan berstruktur
- Pendekatan berorientasikan objek.

## **DEFINISI PENDEKATAN BERSTRUKTUR**

Menggunakan model  
dari atas ke bawah

Membahagikan struktur  
program kepada  
sub – sub seksyen  
dari atas ke bawah

Dilukis dalam  
bentuk segi empat tepat



# **DEFINISI PENDEKATAN BERORIENTASI KAN OBJEK**

- **Satu program yang menggabungkan data dan fungsi bagi menghasilkan objek**
  - **Objek pula merupakan item yang mengandungi data dan prosedur yang boleh bertindak ke atas data.**
  - **Boleh digunakan untuk memanggil semula objek yang telah digunakan sebelumnya.**
  - **Contoh : c++**
- 



# **Perbezaan Antara Pendekatan berstruktur dan Pendekatan Berorientasikan Objek**





# Pendekatan Berorientasikan Objek

- **Pengaturcaraan berorientasi objek ialah, sebuah atur cara Komputer yang terdiri daripada sekumpulan unit-unit atau objek. Untuk membolehkan perkomputeran berlaku, setiap objek berupaya untuk menerima dan menghantar mesej (pesan) kepada objek lain.**
- 

# Pendekatan Berstruktur

- **Adalah satu subset kepada pengaturcaraan prosedural yang membentuk struktur logik pada program yang sedang ditulis bagi menjadikan ia lebih efisien dan lebih mudah untuk difahami dan diubah suai.**

Pendekatan Berorientasikan Objek	Pendekatan Berstruktur
Guna objek - objek gabungkan data manakala fungsinya adalah untuk merekacipta objek	Guna rekabentuk atas bawah kesemua program dipecahkan kepada bahagian - bahagian kecil dan dikenali sebagai modul
setiap objek mempunyai hubungan antara satu sama lain.	Selalu digunakan untuk mengurus dan meletakkan kod - kod program di mana bekerja sebagai hieraki kepada modul
Kelas (class) — sebuah kelas mentakrifkan ciri-ciri abstrak bagi sesuatu benda. Ini termasuklah sifat-sifat yang ada padanya dan peranannya.	kawalan adalah berdasarkan kepada hieraki
Objek (object) — tentuan (instance) bagi suatu kelas.	Mengguna teknik pemecahan masalah yang logik dan sistematik
Kaedah (method) — kebolehan bagi sebuah objek.	Memuat algoritma yang efisien, efektif dan sederhana

<p>Pewarisan (inheritance) — Lazimnya sebuah kelas boleh memiliki "subkelas" yang mengkhususkan kelas tersebut.</p>	<p>Program disusun dengan logik yang mudah dipahami</p>
<p>Pengkapsulan (encapsulation) — Mengasingkan pelaksanaan (implementasi) daripada antaramuka</p>	<p>Tidak menggunakan perintah GOTO</p>
<p>Pengabstrakan (abstraction)-- Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti.</p>	<p>Biaya pengujian program relatif rendah</p>
<p>Polimorfisme (polymorphism) — Menggunakan nama yang sama untuk memulakan operasi yang berlainan pada objek yang menggunakan jenis data berbeza.</p>	<p>Memiliki dokumentasi yang baik</p>
<p>contoh: smalltalk,java visual basic, c++</p>	<p>e.contoh: ada, pascal, fotran</p>

## **5.1.4**

# **PENTERJEMAH**

- PENGHIMPUN**
- PENTAFSIR**
- PENGKOMPIL**



# **PENGHIMPUN**

- **Arahan untuk penghimpun ditulis menggunakan kod arahan berbentuk simbolik atau singkatan yang bermakna atau mnemonik.**
- **Menukar aturcara kepada bahasa mesin.**
- **Contoh : macro 80 assembler**

# PENTAFSIR

- **Menterjemah aturcara baris demi baris kepada kod mesin dan tersu melaksanakannya.**
- **Setiap baris aturcara akan diterjemah satu persatu sehingga tamat.**
- **Jika berlaku ralat sintaksis, aturcara tidak akan terlaksana dan mesej ralat akan dipaparkan di skrin.**
- **Contoh : basic dan logo**



# PENGKOMPIL

- **Menukar kod sumber kepada kod mesin binari iaitu 1 dan 0 sehingga keputusan diperoleh.**
  - **Sewaktu pengkompilan, jika terdapat ralat sintaksis mesej akan dipaparkan di skrin.**
  - **Pembetulan boleh dilakukan, tetapi perlu dikompil semula bagi memastikan aturcara dapat dijalankan.**
  - **Contoh : c++**
- 

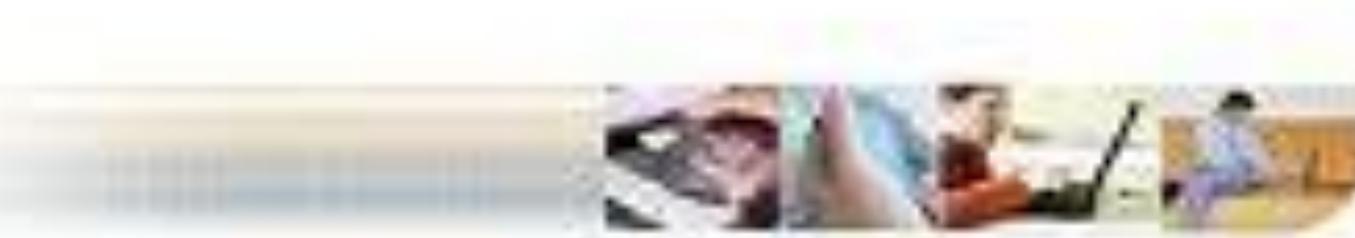


**5.1.5**

## **ELEMEN ASAS DALAM PENGATURCARAAN**



# **MEMBEZAKAN ANTARA PEMALAR DENGAN PEMBOLEHUBAH**





# **Pemalar**

- Mempunyai nilai tetap iaitu tidak berubah sepanjang perlaksanaan turcara**
  - Apabila diisytharkan, nilai lain tidak boleh diumpukan kepadanya**
- 

# **Pembolehubah**

- Nilai pembolehubah boleh berubah pada bila-bila masa semasa program berjalan**
- Untuk menyimpan nilai input atau hasil perlaksanaan satu pernyataan**
- Mesti diisytiharkan sebelum digunakan**

# JENIS-JENIS DATA

Boolean

Integer

Double

String

Date





**Membezakan antara  
jenis data:  
Boolean, Integer,  
Double, String, Date.**

# Boolean

**Nilai yang diistiharkan sebagai  
TRUE dan FALSE**

## Contoh

- |              |   |               |
|--------------|---|---------------|
| $( 10 > 9 )$ | = | Benar (True)  |
| $( 6 < 4 )$  | = | Palsu (False) |
| $( 6 == 7 )$ | = | Palsu (False) |



# **String**

## **Data dalam bentuk teks**

### Contoh

Nama = Abu

Alamat = Tanjung Malim, Perak



# **Double**

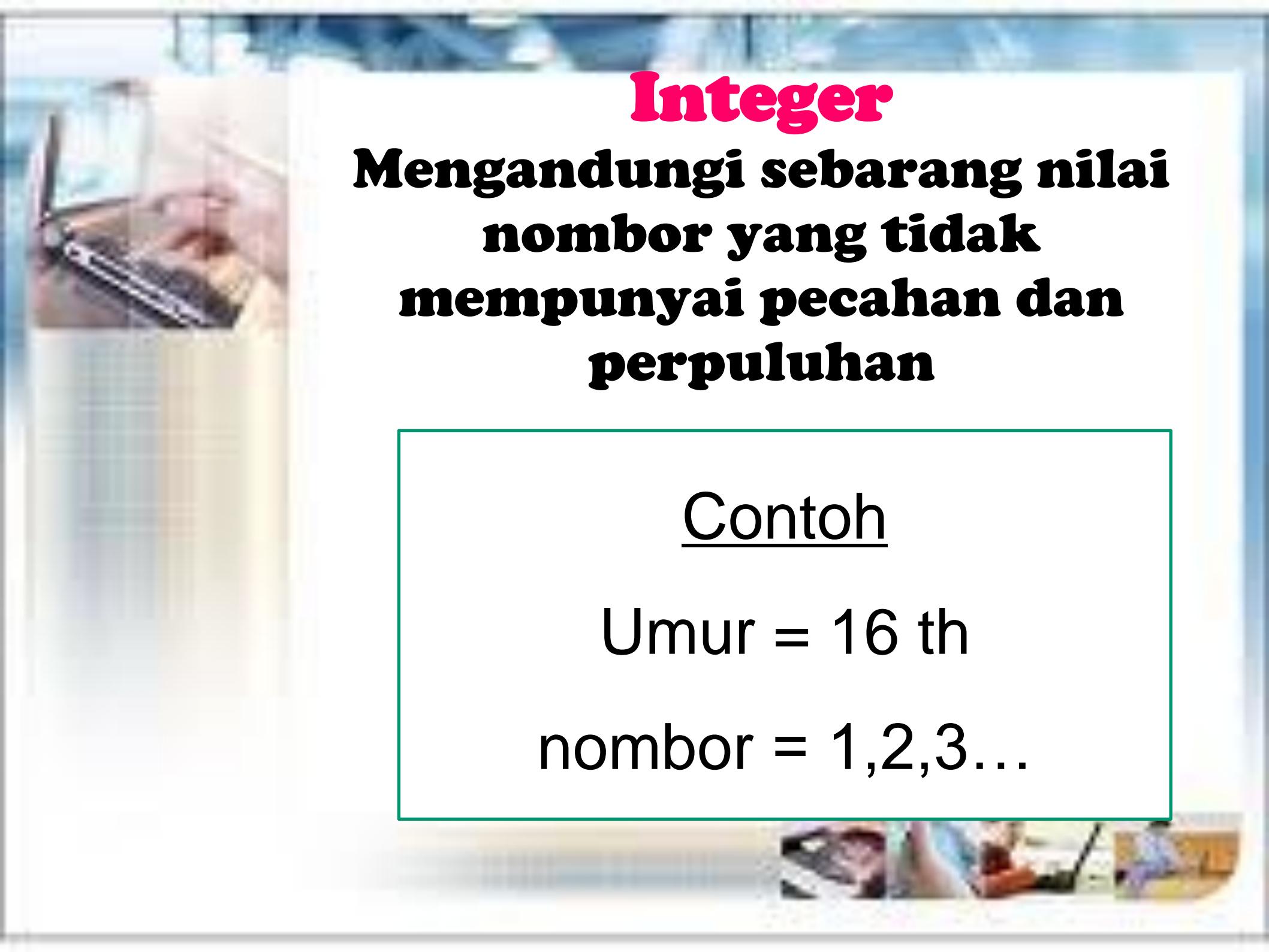
## **Semua nilai nombor termasuk yang mempunyai pecahan atau perpuluhan**

### Contoh

Umur = 20th

Markah = 80.5

Kg = 1/2kg



# **Integer**

## **Mengandungi sebarang nilai nombor yang tidak mempunyai pecahan dan perpuluhan**

Contoh

Umur = 16 th

nombor = 1,2,3...



# Date

## Data dalam bentuk tarikh dan masa

Contoh

Tarikh = 12.12.2012

Masa = 12.12 p.m



# **Membezakan antara pengendali matematik dengan logik**



## PENGENDALI MATEMATIK

**Simbol:**

+ (tambah)

- (tolak)

\* (darab)

/ (bahagi)

$$10 + 10 = 20$$

$$10 - 5 = 5$$

$$10 \times 10 = 100$$

$$10 \div 5 = 2$$

## PENGENDALI LOGIK

**Simbol:**

1. Operator logik.

&& (AND)

|| (OR)

dengan

! (NOT)

dengan

2. Operator hubungan

> Lebih besar daripada

>= Lebih besar atau sama

< Kurang daripada

<= Kurang atau sama

= = Sama dengan

!= Tidak sama

$$A = B$$

$$M > N$$

$$P < Q$$

$$I \leq j$$



# **Membezakan antara struktur kawalan jujukan dengan struktur kawalan pilihan**



**~TERIMA KASIH~**

