

## 3 PENGENALAN KEPADA C

### 3.0 Latar Belakang

- Asal C bermula daripada bahasa BCPL (Basic Combined Programming Language) oleh Martin Richards pada 1967
- BCPL diubah suai oleh Ken Thompson menjadi bahasa B
- B pula diubah suai ke C oleh Dennis Ritchie pada 1972
- Digelar bahasa pertengahan kerana menggabungkan ciri-ciri bahasa aras tinggi dan aras rendah, antara ciri<sup>2</sup> gabungan C
  - ☆ mempunyai kemudahan yang biasa disediakan oleh bahasa peringkat rendah contohnya pengolahan bit dan penggunaan pembolehubah daftar
  - ☆ mempunyai pembinaan kawalan dan jenis struktur data yang mana biasa terdapat dalam bahasa peringkat tinggi

#### ○ Kenapa C ?

Kenapa C menjadi pilihan untuk menyelesaikan masalah saintifik dan masalah kejuruteraan ?

- Mudah alih : C diapiawaikan (ANSI C), mudah dialihkan dari satu sistem ke sistem yg lain.
- Kecekapan C tinggi berbanding bahasa lain.
- Struktur C bermodular
- Berkuasa utk mengungkapkan pengolahan data dan aritmetik.
- Sintaks yg mudah
- Fungsi<sup>2</sup> perpustakaan yg banyak utk menyokong pengaturcaraan.

### 3.1 Contoh Aturcara C yang Mudah - mencetak satu mesej di atas skrin komputer

```

/* Contoh aturcara C: cetak satu mesej di skrin*/

#include <stdio.h>

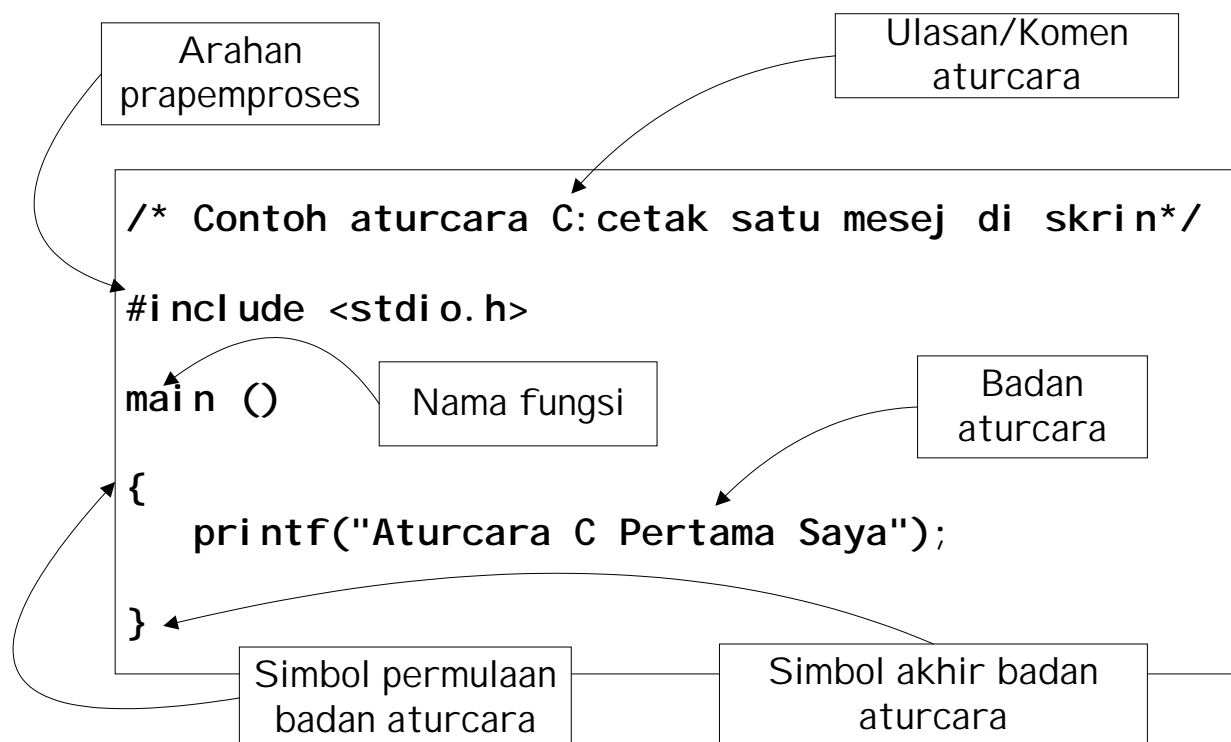
main ()
{
    printf("Aturcara C Pertama Saya");
}

```

- Keluaran pada skrin :

Aturcara C Pertama Saya

- Penjelasan aturcara di atas:



### 3.1 Contoh Aturcara C yang Mudah (samb.....)

#### ◎ Komen/ulasan aturcara

1. Tujuan untuk menjelaskan maklumat mengenai aturcara.
2. Untuk dibaca oleh pembaca - diabaikan oleh komputer !
3. Bermula dengan /\* dan diakhiri dengan \*/

#### ◎ Arahan prapemproses (akan dibincangkan kemudian !)

1. Memasukkan kandungan fail lain.
2. Setiap arahan prapemproses dimulakan dengan simbol #

#### ◎ Nama fungsi

1. Aturcara C terdiri dari satu atau lebih fungsi.
2. Fungsi utama dipanggil fungsi **mai n**.
3. Selain fungsi **mai n** kita bebas untuk memberi nama pada fungsi<sup>2</sup> lain. **printf** ialah contoh fungsi lain yg dipanggil oleh **mai n** utk mencetak mesej.
4. Tujuan ( ) bagi fungsi adalah untuk pemboleh ubah masukan dan keluaran sesuatu fungsi.
5. Pelaksanaan sesuatu aturcara bermula dari fungsi **mai n**.

#### ◎ Badan aturcara/fungsi

Semua kenyataan/arahan sesuatu fungsi ditulis selepas simbol { dan ditamatkan dgn simbol }.

```
{
    kenyataan/arahan2;
    :
}
```

Setiap kenyataan/arahan dlm badan fungsi **mesti** ditamatkan dgn semikolon (;).

### 3.2 Contoh Aturcara C dgn Pembolehubah - mencetak satu mesej di atas skrin komputer

```

1: /* Aturcara C dgn pembolehubah */
2: #include <stdio.h>
3:
4: main ( )
5: {
6:     int umur;
7:     umur = 25;
8:     printf("Aturcara C Pertama Saya\n");
9:     printf("Selepas umur saya %d tahun", umur);
10: }

```

Perhatikan ! 1:, 2: -- 10: bukan sebahagian dari aturcara. Ia dimasukkan utk menunjukkan baris<sup>2</sup> dlm aturcara.

○ Keluaran pada skrin :

```

Aturcara C Pertama Saya
Selepas umur saya 25 tahun

```

○ Penjelasan aturcara

Baris 1-2: bhg ulasan & prapemproses - telah dibincangkan

Baris 3: kosong ! - guna utk menyenangkan pembacaan aturcara

Baris 4-10: mentakrifkan satu fungsi bernama **main** yg terdiri dari 4 kenyataan/arahan di baris 6-9.

Baris 6: mengisytiharkan satu pembolehubah bernama **umur** yg boleh menyimpan nilai **integer** shj.

Baris 7: memberikan nilai 25 kpd pembolehubah **umur** - iaitu **umur diumpukkan** (assigned) dgn nilai 25.

Baris 8: panggil fungsi **printf** utk cetak mesej dlm "...". Simbol **\n** mengarahkan supaya mesej yg seterusnya dicetak bermula di baris yg baru.

Baris 9: cetak mesej dlm "...". dan nilai pembolehubah **umur**.

Simbol **%d** menyatakan jenis dan format utk mencetak nilai pembolehubah **umur**.

Baris 5&10: memulakan & menamatkan takrifan fungsi **main**.

### 3.3 Aturcara C Yg Jelas di Baca

C adalah bahasa 'Gaya bebas'. Contoh aturcara yg sah:

Format 1:

```
#include <stdio.h>
main ( )
{
    int umur;
    umur = 125;
    printf("umur saya %d tahun", umur);
}
```

Format 2:

```
#include <stdio.h> main ( ) {    int umur;
    umur = 125; printf("umur saya %d
tahun", umur); }
```

Format 3:

```
#include <stdio.h>
main ( ) { int
    umur;
    umur
    = 125;
    printf("
        umur saya %d tahun", umur
    );
}
```

Format 4:

```
#include <stdio.h>
main ( ) {

    int umur;
    umur = 125;
    printf("umur saya %d tahun", umur);
}
```

Format 1 & 4 mudah dibaca -- **digalakkan** penggunaannya !

**Latihan:**

1. Tulis aturcara C untuk menyelesaikan masalah berikut:

a) cetak mesej - Hurey ! Aturcara C saya.

b) cetak butir<sup>2</sup> diri anda.

nama

umur - *simpan dlm pembolehubah*

alamat

c) cetak mesej

**20hb Mac Tahun 2005**

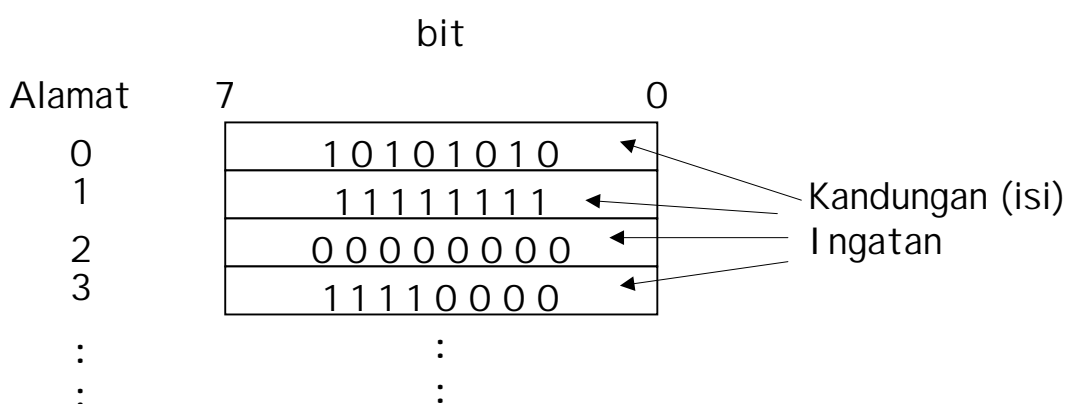
*simpan maklumat yg dihitamkan dlm pembolehubah.*

2. Tentukan ralat dlm aturcara berikut:

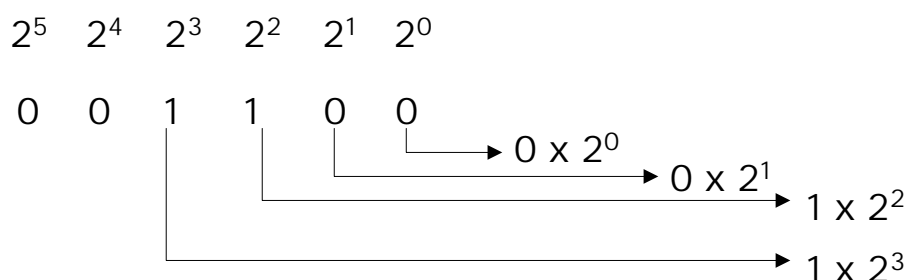
```
#include <stdio.h>
/* contoh aturcara */
main ( )
{
    int umur
    umur = 125;
    printf "umur= %d tahun", umur);
```

### 3.4 Ingatan Dan Simpanan Data Dlm Komputer

- Oleh kerana komputer adalah sistem digital, unit asas yg boleh disimpan dlm ingatan ialah nombor perduaan '0' @ '1'.  
 Satu digit perduaan dipanggil **bit**.  
 8 bit = 1 **bait** (byte)  
 2 bait = 1 **kata** (word)  
 2 kata = 1 **kata panjang** (long word)
- Ingatan dlm komputer terdiri dari senarai bait<sup>2</sup> (bytes).  
 Setiap bait mempunyai satu alamat.



- Nilai max yg boleh disimpan di satu alamat ialah  $255_{10}$  ( $11111111_2$ ).  
 Nilai min " " " "  $0_{10}$  ( $00000000_2$ )
- Penukaran perduaan ke desimal



$$001100_2 = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5$$

$$= 12_{10}$$

### 3.4 Ingatan Dan Simpanan Data Dlm Komputer (samb...)

- Menulis nilai integer besar dlm btk perduaan memerlukan bil. bit yg banyak - mudah membuat kesilapan.

Nilai integer besar lebih ringkas jika ditulis dlm btk perenambelasan (heksadesimal -Hex).

- Jadual penukaran Hex-Binari-Desimal

Simbol Hex	Binari	Desimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

$$\begin{aligned}
 4FC_{16} &= 4 \times 16^2 + 15 \times 16^1 + 12 \times 16^0 = 1276_{10} \\
 &= 0100 \ 1111 \ 1100_2
 \end{aligned}$$



### 3.4 Ingatan Dan Simpanan Data Dlm Komputer (samb...)

- **Integer tak bertanda** (unsigned integer) - integer dgn nilai positif shj.

**Integer bertanda** (signed integer) - integer dgn nilai positif dan negatif. Bit paling kiri (MSB) digunakan sbg tanda.

MSB = 1 : integer negatif

MSB = 0 : integer positif

Contoh: dgn perwakilan 8 bit

Integer tak bertanda			Integer bertanda		
11111111 <sub>2</sub>	255 <sub>10</sub>	FF <sub>16</sub>	01111111 <sub>2</sub>	+127 <sub>10</sub>	7F <sub>16</sub>
00000000 <sub>2</sub>	0 <sub>10</sub>	00 <sub>16</sub>	00000000 <sub>2</sub>	0 <sub>10</sub>	0 <sub>16</sub>
			10000000 <sub>2</sub>	-128 <sub>10</sub>	80 <sub>16</sub>

- **Aksara** (character)

Simbol bagi angka (0..9), abjad (A..Z) & simbol istimewa spt @ # % dsbnya diwakili sbg integer 8 bit dlm komputer. Nilai simbol<sup>2</sup> ini disimpan sbg Kod ASCII (American Standard Code for Information Interchange) di dlm ingatan.

Contoh: Kod ASCII bagi 'A' = 65

Kod ASCII bagi '\$' = 36

- **Integer besar.**

Utk menyimpan nilai<sup>2</sup> lebih besar dari 8 bit, lebih dari satu ingatan perlu digunakan.

Saiz ingatan

16 bit

32 bit

Nilai integer

0 ... 65535

-32768 .. +32767

0 ... 4,294,967,295

-2,147,483,648 ... +2,147,483,647

**Latihan:**

1. Tukar nombor perduaan berikut ke hex dan desimal:

a)  $10001110_2$       b)  $10101010_2$       c)  $00100111_2$

2. Tukar nombor hex berikut ke desimal dan binari:

a) 25      c) 1FF      b) E6

### 3.6 Pencam (identifiers)

- Dlm cth aturcara C yg lepas, fungsi & pembolehubah diberi nama simbolik spt **printf** & **umur**.  
Nama simbolik ini dipanggil **pencam** bagi fungsi & pembolehubah tersebut.
- Dlm C pencam digunakan sbg nama rasmi bagi : fungsi, pembolehubah, pemalar dan label.
- Pencam dlm C mesti memenuhi syarat<sup>2</sup> berikut:
  - ▷ bermula dgn abjad @ aksara 'underscore' ( \_ )
  - ▷ terdiri dari abjad (A...Z, a...z), digit (0...9) @ aksara ( \_ ) shj
  - ▷ saiz max 30/1/2 aksara
  - ▷ bukan katakunci<sup>2</sup> C. Katakunci<sup>2</sup> C :

auto	extern	si zeof	#defi ne
break	fl oat	stati c	#i ncl ude
case	for	struct	
char	goto	swi tch	
const	i f	typedef	
conti nue	i nt	uni on	
defaul t	l ong	unsi gned	
do	regi ster	voi d	
doubl e	return	vol ati le	
el se	short	whi l e	
enum	si gned		

contoh<sup>2</sup> pencam:

#### Sah

utm  
\_bek  
rekod\_201  
i fi

#### Tak sah

2utm  
meow?  
rekod-301  
i f

- C membezakan diantara pencam huruf besar atau pencam huruf kecil @ gabungan huruf besar & huruf kecil.

contoh 4 pencam yg berbeza:

ba BA Ba bA

- Apakah perbezaan 2 pencam berikut:

dayang\_norhayati\_binti\_abang\_jawawi  
dayang\_norhayati\_binti\_abang\_jawawi2

### 3.6 Pencam (samb....)

○ Pencam terdiri dpd:

- i. Pencam Piawai
- ii. Pencam Pengguna

#### i. Pencam Piawai

- Nama pencam yg digunakan didlm perpustakaan C. Mempunyai maksud tertentu spt : `printf` & `scanf`
- Pencam ini boleh ditakrifkan semula ttp fungsi asal pencam tersebut tidak ditukar semula, cth aturcara:

```
int printf; /*printf menyimpan nilai integer*/
printf("Cuba pasti salah\n");
```

#### ii. Pencam Takrifan Pengguna

- Nama pencam yg direka oleh pengguna utk 3 tujuan penakrifan:
  1. Pembolehubah
  2. Pemalar
  3. Fungsi
- Pembolehubah -pencam digunakan utk menyimpan nilai.  
-Format pengisytiharan :

```
j eni s_data nama_pembolehubah;
```

- cth:

```
int umur;
double jejari, luas, ukurlilit;
```

### 3.6 Pencam (samb....)

- Pemalar -pencam digunakan utk menyimpan nilai yg sama sepanjang masa.

-2 cara pengisytiharan pemalar :

```
const j eni s_data nama_pemal ar = ni l ai _pemal ar;
```

@

```
#defi ne nama_pemal ar ni l ai _pemal ar
```

- cth pemalar  $\pi$ :

```
const doubl e PI = 3. 1459;
```

@

```
#defi ne PI 3. 1459
```

-jika satu nilai pemalar diisytiharkan & pengguna cuba mengumpukkan nilai lain kpd pemalar - RALAT Sintaks!

- Nama Fungsi -satu kumpulan arahan yg ditulis oleh pengguna utk tujuan tertentu & setiap fungsi mempunyai nama yg unik.

-cth :

pencam nama fungsi

```
voi d cetak_al amat_saya (voi d)
{
    pri ntf("Dayang Norhayati Abang Jawawi \n");
    pri ntf("Fakul ti Sai ns Komputer");
    pri ntf("dan Si stem Makl umat\n");
    pri ntf("Skudai , Johor\n");
}
```

-Fungsi akan dipelajari dgn terperinci dlm bab dihadapan ....Tunggu....

### 3.7 Jenis-jenis Data

- Dlm aturcara mudah yg lepas kenyataan :  
`int umur;`  
 katakunci `int` mengisytiharkan pembolehubah `umur` untuk menyimpan data jenis `integer` shj.
- Dlm C katakunci<sup>2</sup> berikut digunakan utk mengisytiharkan jenis data yg. boleh disimpan oleh sesuatu pembolehubah :

katakunci (jenis data)	saiz dlm bait	julat no yg disimpan
<code>char</code>	1	-128 - 127
<code>int</code>	2	-32768 - 32767
<code>short</code>	2	-32768 - 32767
<code>long</code>	4	-2147483648 - 2147483647
<code>float</code>	4	$\pm 1.18 \times 10^{-38}$ - $\pm 3.4 \times 10^{38}$
<code>double</code>	8	$\pm 9.46 \times 10^{-308}$ - $\pm 1.79 \times 10^{308}$

- Perhatikan :
  1. `char`, `int`, `short` & `long` adalah utk menyimpan **nombor integer** (.. -2, -1,0,1,2,3....)
  2. `float` & `double` adalah utk menyimpan **nombor nyata** (-2.1, 1.1 dsb)
- Utk membezakan nombor integer **tak bertanda** (positif shj) & nombor **bertanda** (negatif & positif) C menggunakan katakunci **unsigned** bagi nombor<sup>2</sup> integer tak bertanda.

katakunci (jenis data)	julat no yg disimpan
<code>unsigned char</code>	0 - 255
<code>char</code>	-128 - 127
<code>unsigned int</code>	0 - 65535
<code>int</code>	-32768 - 32767
<code>unsigned long</code>	0 - 4294967295
<code>long</code>	-2147483648 - 2147483647

### 3.7 Jenis-jenis Data (samb....)

- **char** biasanya digunakan utk menyimpan kod<sup>2</sup> ASCII bagi aksara spt 'a', 'M', '#', '!', '%' .....
- Cth mengisytiharkan pembolehubah dlm C :

```

unsigned char kod;
char alfa, c_kod, no;
int hari, tahun;
unsigned int luas, isi padu;
long int bil_pelajar;
long bilBiri_biri;
float gaji_pm;
double luas_planet;

```

- Pembolehubah boleh diberikan nilai awalnya semasa ia diisytiharkan :
 

```

int hari = 28, tahun = 201;
float gaji_pm = 654321.25;

```
- Nilai<sup>2</sup> pemalar (constant) dlm C boleh ditulis spt berikut :

Jenis Pemalar	Contoh nilai pemalar
Perpuluhan (desimal)	65
Perenambelasan (heksadesimal)	0x41, 0X41
Perlapanan (oktal)	0101
Aksara	'A'
Perpuluhan panjang	65L
Nombor nyata	65.0, 65., 65e0, 650.E-1

- Cthnya aksara 'A' boleh ditulis dlm berlainan format spt berikut :

	Pemalar aksara	Desimal	Hex	Oktal
Format	'A'	65	0x41	0101
Format dlm ingatan (binari)	01000001	01000001	01000001	01000001

### 3.7 Jenis-jenis Data (samb....)

- Dlm C terdapat beberapa pemalar aksara istimewa yg mempunyai maksud tertentu.

Pemalar Aksara	Maksud
' \n'	newline - pergi ke baris seterusnya pd lajur 1
' \t'	horizontal tab - anjak ke kanan
' \v'	vertical tab - anjak ke bawah pd lajur yg sama
' \r'	carriage return - baris seterusnya
' \x41'	nombor hexa 0x41
' \101'	nombor oktal 101
' \0'	null - aksara nol digunakan utk menandakan penghujung rentetan aksara (string)
' \''	tanda (')
' \"'	tanda (")
' \\'	tanda (\)
' \b'	backspace - undur ruang
' \f'	formfeed - halaman seterusnya (utk pencetak)
' \a'	alert - membunyikan loceng

- Pembolehubah jenis **const** **tidak boleh diubah** nilainya oleh aturcara. Ia biasanya digunakan utk melindungi pembolehubah yg dihantar kpd sesuatu fungsi drpd diubah nilainya.

Contoh aturcara menggunakan pembolehubah & pemalar

```

/* Contoh aturcara kira luas segi empat & bulatan */
/* Penulis : Rosbi Mamat Tarikh : 20.02.2020 */
#include <stdio.h>

main ()
{
    const float pi = 3.1415;
    int panjang, lebar, luas_segi4;
    float jejari, luas_bulatan;

    panjang = 12; lebar = 3; luas_segi4 = panjang * lebar;
    jejari = 3.5; luas_bulatan = pi * jejari * jejari;

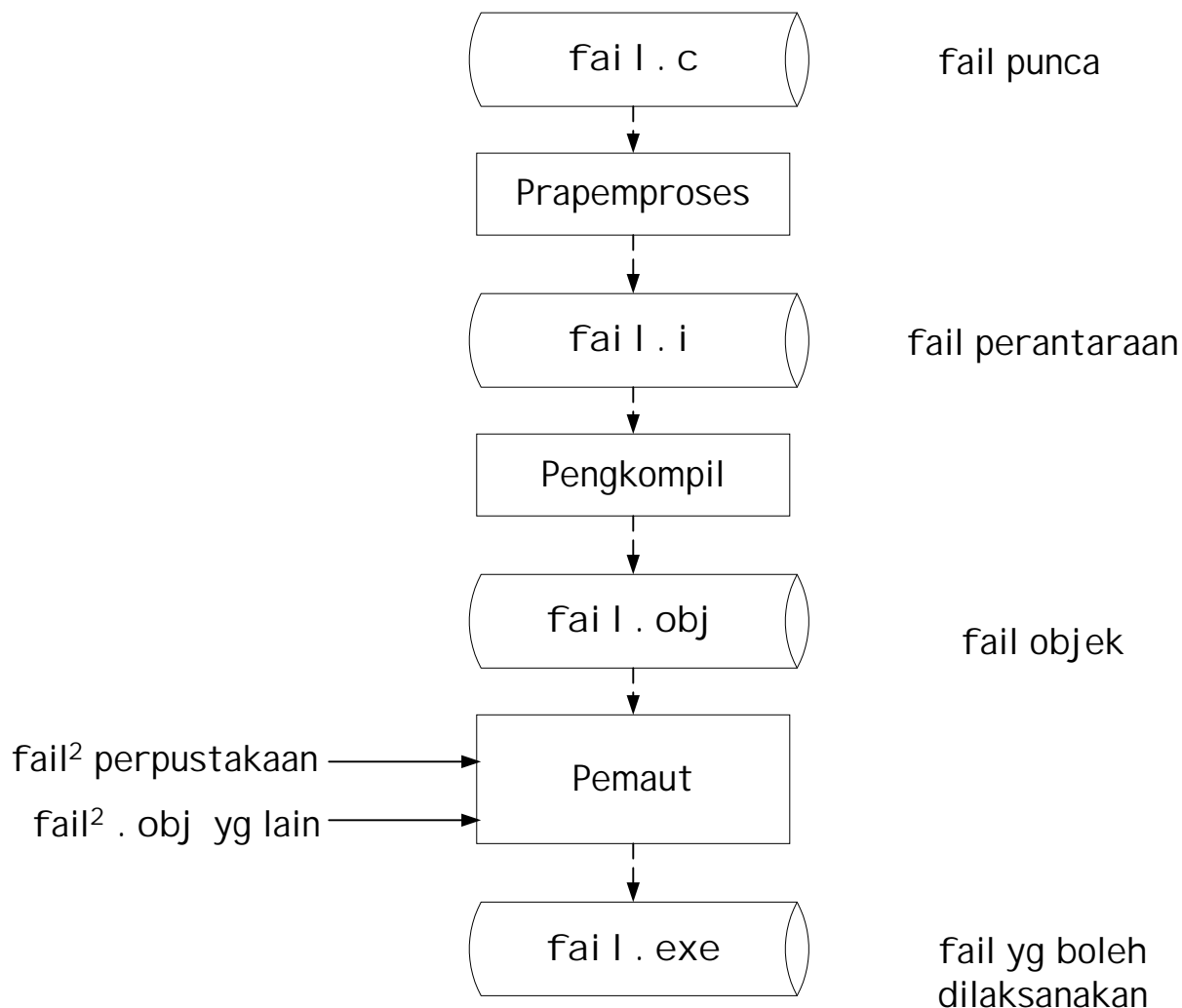
    printf("Luas bulatan = %f\t
           Luas segi 4 = %d\n", luas_bulatan, luas_segi4 );
    printf("\a Luas bulatan = %f\r
           Luas segi 4 = %d\n", luas_bulatan, luas_segi4 );
}

```



### 3.8 Arahan Prapemproses

- Arahan C prapemproses akan diproses dahulu sebelum dikompil. Perhatikan proses yg dilalui oleh fail punca/sumber sebelum boleh dilaksanakan.



- Format panggilan kepada prapemproses :

```
#j eni s_prapemproses arahan²_berkai tan
```

- 2 prapemproses yg biasa digunakan #define & #include

### 3.8 Arahan Prapemproses (samb....)

#### i. Arahan prapemproses #define

- untuk mengisytiharkan pemalar simbolik

- Format panggilan prapemproses #define rujuk 3.6.

#### ii. Arahan prapemproses #include

- digunakan untuk menggabungkan fail luaran ke dalam satu aturcara sumber/punca dgn memanggil fail pengepala (namafail.h), 2 jenis fail kepala:

- ◆ piawai - utk mencapai perpustakaan C
- ◆ takrifan pengguna - utk mencapai aturcara punca lain dlm fail yg berlaian yg ditulis oleh pengguna.

- Format panggilan prapemproses #include piawai :

```
#include <nama_pegapala>
```

- Format panggilan prapemproses #include takrifan pengguna :

```
#include "nama_pegapala"
```

- Cth aturcara dgn #include piawai & #define:

```
#include <stdio.h>
#define PI 3.14159
#define NAMPENULIS "Dayang Norhayati"

int main(void)
{
    double jejari, luas, ukurlilit;

    printf("Penulis aturcara : %s\n", NAMPENULIS);

    jejari = 1.0;
    luas = PI * jejari * jejari;
    ukurlilit = 2 * PI * jejari;

    printf("Luas bulatan ialah %f\n", luas);
    printf("Ukurlilit bulatan ialah %f", ukurlilit);

    return 0;
}
```

### 3.8 Arahan Prapemproses (samb....)

- output aturara:

```
Penulis aturcara : Dayang Norhayati
Luas bulatan ialah 3.141590
Ukurlilit bulatan ialah 6.283180
```

- Cth aturcara dgn #include takrifan pengguna & #define:

```
/* Nama Fail : bulat.c*/

#include "bulat.h"

int main(void)
{
    printf("Penulis aturcara: %s\n", NAMAPENULIS);

    jejari = 1.0;
    luas = PI * jejari * jejari;
    ukurlilit = 2 * PI * jejari;

    printf("Luas bulatan ialah %f\n", luas);
    printf("Ukurlilit bulatan ialah %f", ukurlilit);

    return 0;
}
```

```
/* Nama Fail : bulat.h*/

#include <stdio.h>
#define PI 3.14159
#define NAMAPENULIS "Dayang Norhayati "

double jejari, luas, ukurlilit;
```

### 3.3 Latihan

1. Tentukan jenis data yg akan digunakan utk mewakili data<sup>2</sup> berikut:

- jumlah penduduk Johor Bahru
- berat sebiji kelapa
- bilangan pelajar UTM yang berbasikal
- jumlah import Malaysia sepanjang 5 tahun yg lepas
- warna kereta

2. Apakah keluaran pd skrin jika suruhan<sup>2</sup> berikut ditambah dlm aturcara dlm m/s 16:

- `printf("bulatan\b\b\n '\ '");`
- `panjang= '\0x64'; lebar = 'A'; luas_segi4=panjang*lebar; printf("%d", luas_segi4);`

3. Cari 7 kesilapan dlm aturcara berikut:

```
KiraHutang ( )
{
    int nama_yang_tersangat_panjang_jenis_int;
    float nama_yang_tersangat_panjang_jenis_float;
    const float kadar = 25.23, goto=1.3;
    float pinjambank, pinjamkawan, samanpolis, hutang;
    char kod;
    int bil_guli = 5.0;

    Hutang= pinjambank*kadar+pinjambak+pinjamkawan+samanpolis;
    kadar=20.1; nama_yang_tersangat_panjang_jenis_int =80000;
    kod = 66;
}
```