



TEKNIK MEMBINA ATUR CARA DENGAN BAHASA C

DAYANG NORHAYATI ABANG JAWAWI
ROSBI MAMAT



Bab 2: Pengenalan Pengaturcaraan C

© Copyright Universiti Teknologi Malaysia

innovative • entrepreneurial • global

2.0 Latar Belakang

- Kenapa memilih C?

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

- > Mudah alih : C dipeawakan (ANSI C), mudah dialihkan dari satu sistem ke sistem yg lain
- > Kecekapan C tinggi berbanding bahasa lain.
- > Berkuasa utk mengungkapkan pengolahan data dan aritmetik.
- > Struktur C bermodular
- > Sintaks yg mudah
- > Fungsi2 perpustakaan yg banyak utk menyokong pengaturcaraan.

© Copyright Universiti Teknologi Malaysia

2.1 Bahasa Pengaturcaraan

Atur cara computer -> arahan-arahan yang diberikan kepada komputer untuk melaksanakan tugas-tugas tertentu.

Bahasa pengaturcaraan -> notasi dan bahasa formal yang boleh difahami dan seterusnya diproses oleh komputer.



Bahasa pengaturcaraan aras rendah

Cth:- Bahasa Mesin dan Bahasa Himpunan

Bahasa pengaturcaraan aras tinggi

Cth:- C, C++, Java, Fortran, Visual C++, Python dan C#

2.2 Atur Cara C Mudah dan Cara Memproses Atur Cara

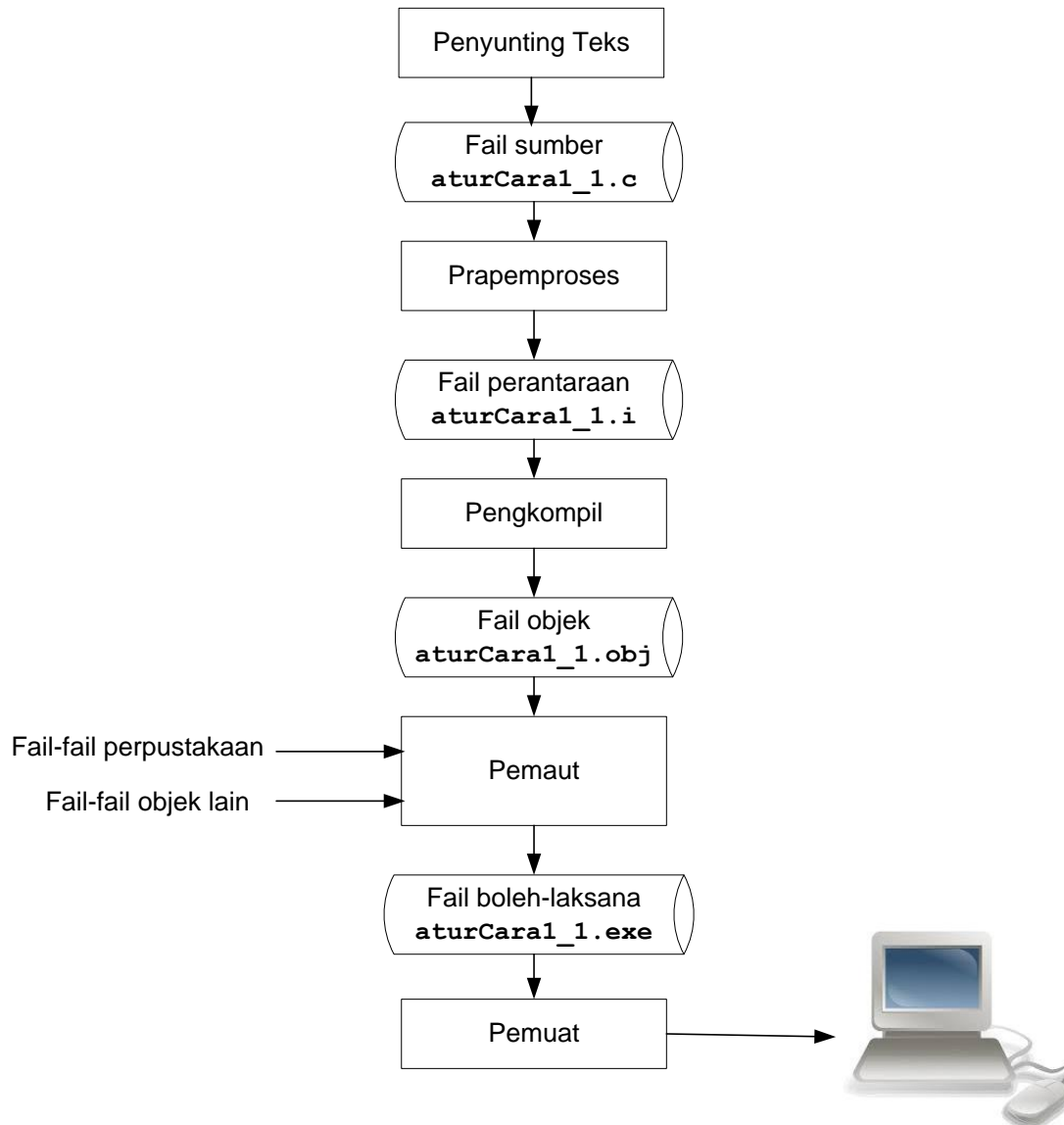
- ❖ Hasil pelaksanaan Atur cara akan mencetak mesej “Atur Cara C Pertama Saya”

```
1:  /* Contoh atur cara C:cetak satu mesej di skrin
2:  Penulis: Dayang Norhayati and Rosbi Mamat, UTM */
3:
4:  #include <stdio.h>
5:
6:  int main ()
7:  {e
8:    printf("Atur Cara C Pertama Saya");
9:    return 0;
10: }
```

Atur Cara C Pertama Saya

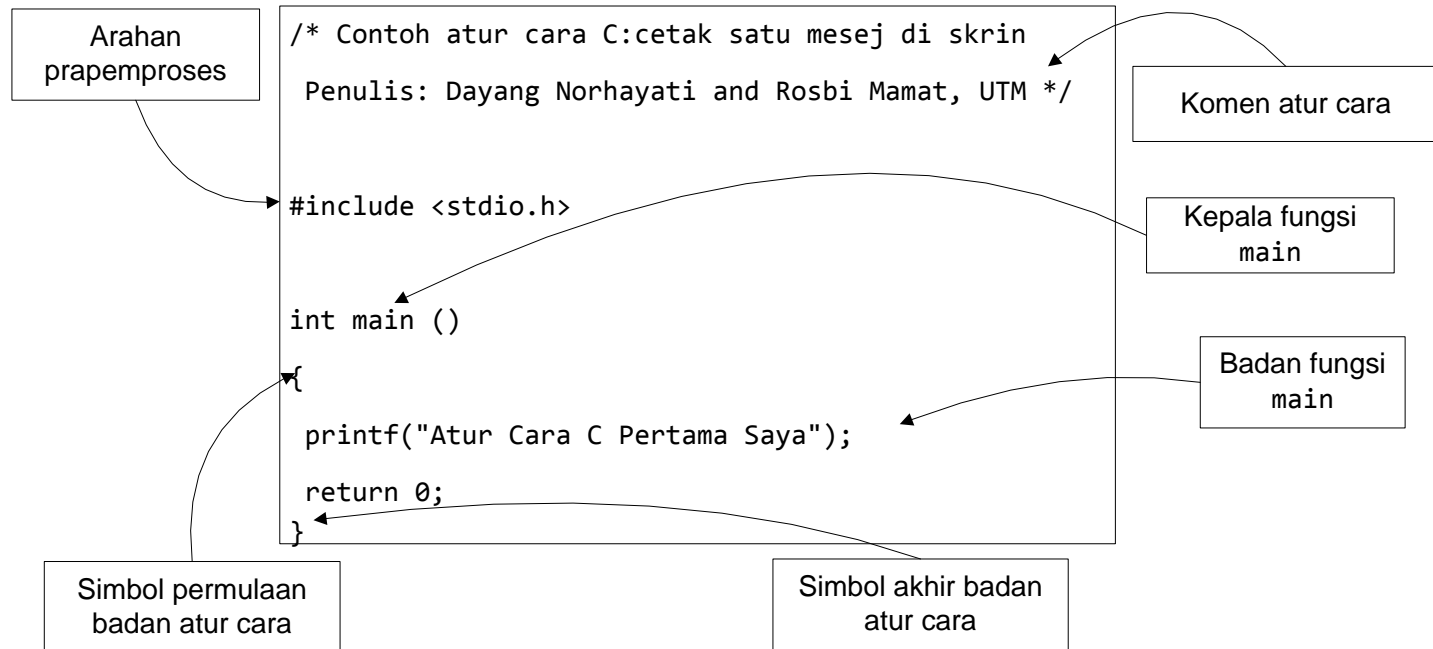
2.2 Atur Cara Mudah dan Cara Memproses Atur Cara

- ❖ Proses menulis dan mentejemahkan atur cara



2.3 Elemen Asas Atur Cara C

❖ Penjelasan atur cara:-



2.3 Elemen Asas Atur Cara C

- Komen/ulasan aturcara
 - Tujuan untuk menjelaskan maklumat mengenai aturcara.
 - Untuk dibaca oleh pembaca - diabaikan oleh komputer !
 - Bermula dengan `/*` dan diakhiri dengan `*/`
- Arahan prapemproses
 - Memasukkan kandungan fail lain.
 - Setiap arahan prapemproses dimulakan dengan simbol `#`
- Nama fungsi
 - Aturcara C terdiri dari satu atau lebih fungsi.
 - Fungsi utama dipanggil fungsi **main**.

2.3 Elemen Asas Atur Cara C

- Nama fungsi (sambungan)
 - Selain fungsi **main** kita bebas untuk memberi nama pada fungsi2 lain. **printf** ialah contoh fungsi lain yg dipanggil oleh **main** utk mencetak mesej.
 - Tujuan () bagi fungsi adalah untuk pemboleh ubah masukan dan keluaran sesuatu fungsi.
 - Pelaksanaan sesuatu aturcara bermula dari fungsi **main**

- Badan aturcara/fungsi

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

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


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

2.3 Elemen Asas Atur Cara C

- Contoh Aturcara C dengan pemboleh ubah

```
1:  /* Contoh atur cara C:cetak mesej di skrin
2:  Penulis: Dayang Norhayati and Rosbi Mamat, UTM */
3:
4:  #include <stdio.h>
5:
6:  int main ()
7:  {
8:      int umur;
9:      umur = 20;
10:     printf("Atur Cara C Pertama Saya\n");
11:     printf("Selepas umur saya %d tahun", umur);
12:     return 0;
13: }
```

Keluaran pada
skrin:



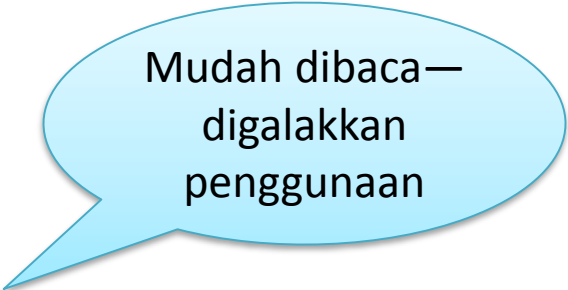
**Aturcara C Pertama Saya
Selepas umur saya 20 tahun**

2.4 Gaya Pengaturcaraan C

- 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);
}
```



Mudah dibaca—
dialakkan
penggunaan

Format 2:

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

2.4 Gaya Pengaturcaraan C

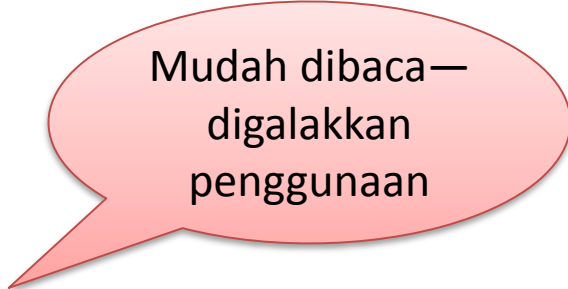
❑ C adalah bahasa 'Gaya bebas'. Contoh aturcara yg sah :-

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);
}
```



Mudah dibaca—
dibalakkan
penggunaan

2.4 Gaya Pengaturcaraan C

- Gaya penulisan yang membantu penghasilan atur cara C dengan berjaya adalah:

1.	Menggunakan baris kosong untuk memisahkan bahagian atur cara
2.	Menulis satu kenyataan pada satu baris dan elakkan penulisan satu kenyataan pada banyak baris.
3.	Gunakan jumlah bilangan jidar yang sama untuk kumpulan arahan yang sama, contohnya kenyataan-kenyataan badan fungsi di baris 4 hingga 6 dalam Atur cara format 1 dianjak ke kanan dengan sama.
4.	Menggunakan ruang kosong untuk memudahkan pemahaman sesuatu kenyataan, seperti di baris 5 Atur cara format 1, walaupun kenyataan ini boleh ditulis tanpa ruang kosong seperti umur=20; tetapi penggunaan ruang kosong di antara kenyataan umpukan lebih menampakkan pemboleh ubah dan nilai umpukannya.

2.4 Gaya Pengaturcaraan C

- ❑ **Kenyataan komen** membantu penghasilan dokumen perisian berkualiti

- ❑ Cara penulisan komen atur cara yang boleh membantu penghasilan dokumen perisian berkualiti::

1.	Mengandungi maklumat asas atur cara termasuk nama atur cara dan kenyataan hakcipta, maklumat penting pengatur cara, tujuan atur cara dan penerangan lanjut mengenai proses pelaksanaan atur cara.
2.	Menulis komen yang bermakna, jelas, penting dan terkini.
3.	Tidak mengulang apa yang telah jelas dan boleh dibaca daripada atur cara.

© Copyright Universiti Teknologi Malaysia

2.5 Pengenal Pasti

- ❖ **Pengenal pasti** merupakan nama kepada pemboleh ubah, fungsi dan pemalar.
- ❖ syarat-syarat menamakan pengenal pasti ialah:

1.	terdiri daripada abjad A hingga Z, a hingga z, digit 0 hingga 9 atau aksara garis bawah “_” sahaja.
2.	Aksara pengenal pasti mesti bermula dengan abjad atau aksara garis bawah “_”.
3.	Walaupun tiada had saiz pengenal pasti, tetapi kebanyakan pengkompil hanya mengambil kira 32 aksara pertama sahaja sebagai pengenal pasti.

2.5 Pengenal Pasti

❖ syarat-syarat menamakan pengenal pasti ialah (sambungan):

4.	Pengenal pasti adalah sensitif huruf, C membezakan antara pengenal pasti huruf besar atau pengenal pasti huruf kecil atau gabungan huruf besar dan huruf kecil.
5.	Kata simpanan C tidak boleh digunakan sebagai pengenal pasti. Cth pengenal pasti yang sah:- utm, _umur, rekod_201, ifi

2.5 Pengenal Pasti

- Contoh pengenal pasti yang tidak sah

Pengenal pasti tidak sah	Sebab tidak sah
2utm	Mula dengan digit
umur?	Aksara “?” tidak dibenarkan
Rekod-301	Aksara “-” tidak dibenarkan
if	Kata simpanan atau kata kunci

2.5 Pengenal Pasti

- **Kata simpanan** adalah kata kunci untuk mengenal pasti identiti bahasa C.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

2.5 Pengenal Pasti

- ❑ Penggunaan huruf besar atau huruf kecil pada pengenal pasti menghasilkan pengenal pasti yang berbeza. Contoh empat pengenal pasti berikut adalah berbeza:

nama, Nama, NAMA, NaMa

- ❑ dua pengenal pasti berikut adalah sama kerana pengkompil hanya mengambil kira 32 aksara pertama sebagai pengenal pasti.

nilai_bacaan_pencemaran_udara_bulan_Januari
nilai_bacaan_pencemaran_udara_bulan_Mac

2.5 Pengenal Pasti

➤ Kategori Pengenal Pasti :

Pengenal Pasti Piawai

- digunakan dalam perpustakaan C, seperti fungsi printf(), telah ditakrifkan dalam perpustakaan C.

- Cth:-

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

- Kenyataan selepas takrifan integer printf, kenyataan seterusnya menggunakan pengenal pasti printf sebagai fungsi perpustakaan C untuk mencetak mesej, ini tidak dibenarkan dan ralat sintaks akan diperolehi.

2.5 Pengenal Pasti

➤ Kategori Pengenal Pasti (Sambungan) :

Pengenal Pasti Takrifan Pengguna

- nama pengenal pasti direka oleh pengatur cara untuk tiga tujuan pentakrifan iaitu pentakrifan pemboleh ubah, pemalar dan fungsi.

- Cth:-

```
void cetak_alamat (void) {  
    printf("Universiti Teknologi Malaysia\n");  
    printf("81310 UTM, Skudai, Johor\n");  
    printf("Malaysia");  
}
```

2.5 Pengenal Pasti

- Garis panduan memilih nama pengenal pasti takrifan pengguna supaya penghasilan gaya atur cara mudah difahami adalah:

1. Elakkan penggunaan pengenal pasti yang terlalu singkat tanpa memberi makna seperti `x` dan `xy`, guna nama yang memberi gambaran maksud seperti `nilaiPenderia` atau `jejari`.

2. Elakkan juga penggunaan pengenal pasti yang terlalu panjang untuk mengelakkan kesilapan menaip pengenal pasti yang panjang, singkatan yang mudah difaham seperti `bil_pelajar`.

3. Guna huruf besar atau aksara “`_`” untuk mengasingkan pengenal pasti yang mempunyai lebih daripada satu perkataan seperti `namaPelajar` atau `nama_pelajar`.

2.6 Jenis Data dan Pengisytiharan

- ❖ **Jenis data** adalah bentuk data yang akan digunakan atau diproses.
- ❖ Dlm aturcara mudah yg lepas kenyataan :

```
int umur;
```

 kata kunci **int** mengisytiharkan pembolehubah **umur** untuk menyimpan data jenis **integer** sahaja.
- ❖ Dalam C kata kunci berikut digunakan untuk mengisytiharkan jenis data yang boleh disimpan oleh sesuatu pembolehubah :

Kata kunci jenis data	Saiz bait	Julat nombor yang disimpan
char	1	-128 hingga 127
int	2	-32768 hingga 32767
short	2	-32768 hingga 32767
long	4	-2147483648 hingga 2147483647
float	4	1.18x10 ⁻³⁸ hingga 3.4x10 ³⁸
double	8	9.46x10 ⁻³⁰⁸ hingga 1.79x10 ³⁰⁸

2.6 Jenis Data dan Pengisytiharan

- ❖ 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)

- ❖ Untuk membezakan nombor integer **tak bertanda** (positif sahaja) dan nombor **bertanda** (negatif & positif) C menggunakan katakunci **unsigned** bagi nombor-nombor integer tak bertanda.

Kata kunci jenis data	Julat nombor yang disimpan
unsigned char	0 hingga 255
char	-128 hingga 127
unsigned int	0 hingga 65535
int	-32768 hingga 32767
unsigned long	0 hingga 4294967295
long	-2147483648 hingga 2147483647

2.6 Jenis Data dan Pengisytiharan

- ❖ **char** biasanya digunakan utk menyimpan kod-kod ASCII bagi aksara seperti 'a', 'M', '#', '!', '%'
- ❖ Contoh mengisytiharkan pemboleh ubah dalam C adalah seperti berikut:

```
unsigned char kod;  
char alfa, kod_aksara, no;  
int hari, tahun;  
unsigned int luas, isipadu;  
long int bil_pelajar;  
float laju;  
double luas_planet;
```

- ❖ Sintaks untuk memberi nilai awal kepada pemboleh ubah ketika pengisytiharan ialah:

```
jenis_data nama_pemboleh_ubah = nilai_awal;  
cth: int hari = 28;
```


2.6 Jenis Data dan Pengisytiharan

- ❖ Kata kunci **const** digunakan ketika mengisytiharkan sesuatu pemalar, sintaks pengisytiharan pemalar dengan kata kunci ini adalah seperti berikut.

```
const jenis_data nama_pemalar = nilai_pemalar;
```

- ❖ Contoh pengisytiharan pemalar yang sah ialah:

```
const double PI = 3.1459;
```

- ❖ Satu lagi cara pemalar boleh diisytiharkan ialah dengan menggunakan arahan prapemproses **#define**. Sintaksnya adalah seperti berikut:

```
#define nama_pemalar nilai_pemalar
```

- ❖ Contoh pemalar PI yang diisytihar dengan arahan prapemproses adalah:

```
#define PI 3.1459
```

2.6 Jenis Data dan Pengisytiharan

- ❖ Menulis nilai pemalar:

Format penulisan nilai malar	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

- ❖ Menulis aksara 'A' dalam format penulisan nilai malar berbeza

Format	Aksara	Desimal	Hex	Oktal
Nilai malar	'A'	65	0x41	0101
Nilai binary	01000001	01000001	01000001	01000001

2.6 Jenis Data dan Pengisytiharan

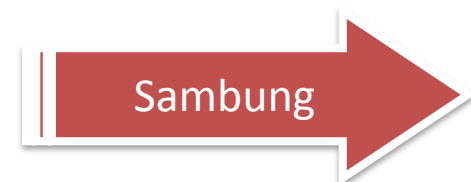
- ❖ **nilai malar aksara** istimewa iaitu urutan keluar menggunakan urutan simbol khas untuk mewakili maksud tertentu nilai malar aksara.

Nilai malar aksara	Maksud
'\n'	pergi ke baris seterusnya pada lajur 1
'\t'	anjak ke kanan
'\v'	anjak ke bawah pada lajur yang sama
'\r'	pergi ke baris seterusnya
'\x41'	nombor hexa 0x41
'\101'	nombor oktal 101
'\0'	aksara nol digunakan untuk menandakan penghujung rentetan aksara
'\''	tanda (')
'\"'	tanda (")
'\\'	tanda (\)
'\b'	undur ruang
'\f'	ke halaman seterusnya untuk cetakan
'\a'	membunyikan loceng

2.6 Jenis Data dan Pengisytiharan

- ❖ Atur cara mengira luas dibawah menunjukkan contoh penggunaan pemboleh ubah dan pemalar dengan jenis data berbeza.

```
1:  /* Contoh atur cara kira luas segiempat & bulatan */
2:  /* Penulis : Rosbi Mamat Tarikh : 20.02.2020 */
3:  #include <stdio.h>
4:
5:  int main () {
6:      const float pi = 3.1415;
7:      int panjang, lebar, luas_segi4;
8:      float jejari, luas_bulatan;
9:
```



2.6 Jenis Data dan Pengisytiharan

- ❖ Atur cara mengira luas dibawah menunjukkan contoh penggunaan pemboleh ubah dan pemalar dengan jenis data berbeza. (Sambungan)

```
10:     panjang = 12;
11:     jejari = 3.5;
12:     lebar = 3;
13:
14:     luas_segi4 = panjang * lebar;
15:     luas_bulatan = pi * jejari * jejari;
16:
17:     printf("Luas bulatan = %f \t'Luas segi4 = %d\n",
18:           luas_bulatan, luas_segi4 );
19:     return 0;
20: }
```

2.7 Ringkasan Bab

- 1) C adalah bahasa pengaturcaraan aras yang mempunyai ciri-ciri: kebolehlenturan, kemudahalihan, kecekapan dan bermodul, untuk penghasilan perisian yang berkualiti.
- 2) Proses menterjemahkan atur cara C kepada arahan yang boleh dilaksanakan oleh komputer memerlukan beberapa langkah dengan bantuan peralatan penyunting teks, prapemproses, pengkompil, pemaut dan pemuat.
- 3) Elemen asas atur cara C ialah komen atur cara, arahan prapemproses, fungsi `main()` dan pengisytiharan jenis data.
- 4) Kenyataan komen atau ulasan atur cara yang ditulis oleh pengatur cara untuk menjelaskan maklumat mengenai atur cara dan membantu pembaca memahami atur cara yang dibangunkan. Komen ditulis antara dua simbol `/*` dan `*/` atau dimulakan barisan komen dengan simbol `/**`.

2.7 Ringkasan Bab

- 5) Arahan prapemproses memberi suruhan kepada prapemproses C untuk menukar kod sumber dengan memasukkan fail lain ke dalam kod sumber sebelum atur cara dikompil. Format panggilan kepada prapemproses adalah:

```
#jenis_prapemproses arahan_berkaitan
```

- 6) Setiap atur cara C perlu mempunyai fungsi utama yang dipanggil main(), format takrifan fungsi utama adalah:

```
jenis_puLangan main()  
{  
    kenyataan-kenyataan C;  
    :  
}
```

2.7 Ringkasan Bab

- 7) Pengisytiharan menyatakan jenis entiti pemboleh ubah atau pemalar di dalam sesuatu atur cara dengan menggunakan jenis data. Format pengisytiharan pemboleh ubah dan pemalar adalah:

```
jenis_data nama_pemboleh_ubah;
```

```
const jenis_data nama_pemalar = nilai_pemalar;
```

- 8) Gaya pengaturcaraan adalah kaedah-kaedah dan teknik-teknik yang membolehkan atur cara ditulis dengan gaya yang mudah dibaca dan difahami dan seterusnya menyokong penyenggaraan perisian secara berkesan.

2.7 Ringkasan Bab

- 9) Pengenal pasti merupakan nama kepada pemboleh ubah, fungsi dan pemalar, dan terdapat syarat-syarat bagi menamakan pengenal pasti dalam atur cara C.

- 10) Jenis data adalah satu set nilai data untuk mewakili data yang akan diproses dan digunakan untuk pengisytiharan pemboleh ubah atau pemalar.