

## 4 OPERATOR DAN UNGKAPAN

### 4.1 Operator

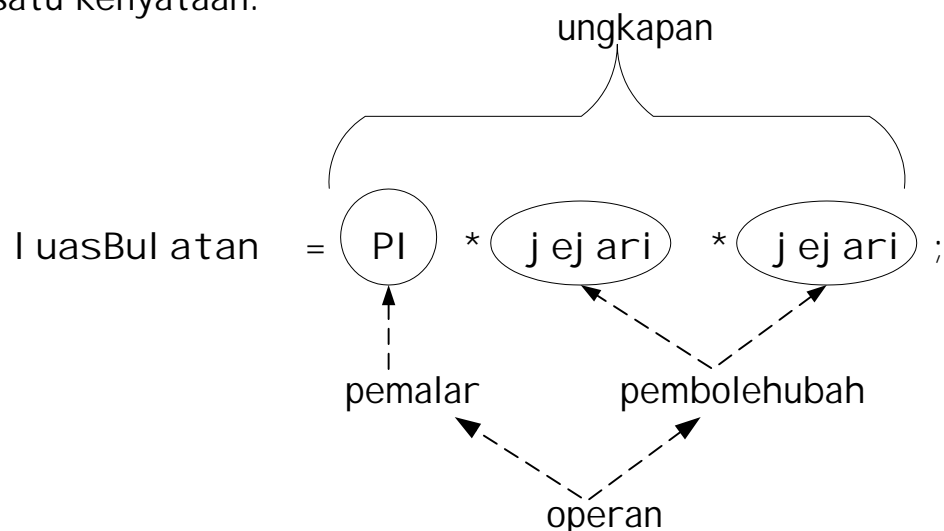
- Data yg disimpan di dlm ingatan (lihat bab 3) boleh diolah dgn menggunakan **operator**. C mempunyai 45 operator utk mengolah berbagai jenis data.
- Beberapa operator telah diperkenalkan sebelum ini:

+ operator tambah  
 - operator tolak  
 = operator umpukan

- Operator digunakan utk menggabungkan operan (pembolehubah@pemalar) menjadi ungkapan. Ungkapan yg ditamatkan dgn ; menjadi kenyataan.:

**ungkapan**      gaji + el aun - hutang  
**kenyataan**      pendapatan = gaji + el aun - hutang;

- Cth satu kenyataan:



- Operator terbahagi kepada:-
  - operator aritmetik
  - operator hubungan
  - operator logik
- Dlm bab ini kita akan belajar fungsi operator<sup>2</sup> tertentu & syarat<sup>2</sup> penggunaannya.

## 4.2 Operator aritmetik

○ Operator aritmetik dalam C terbahagi kepada 2:-

- i) unari                      ii) binari

### 4.2.1 Operator Aritmetik unari

Operator unari dikenakan kpd **satu** pembolehubah shj.

Operator	Fungsi
+	operator positif
-	operator negatif
++	operator tokok
--	operator susut
!	operator TIDAK/NOT
~	operator salingan/pelengkap satu
&	operator alamat bagi sesuatu pembolehubah
*	operator kandungan sesuatu alamat
sizeof	saiz sesuatu pembolehubah dlm bait

```
mai n()
{
    int x, c;

    c = -64;
    x = +c;           /* x = -64 */
    x = -c;           /* x = 64 */
    x = ++c;         /* x = 65 */
    x = --c;         /* x = 64 */
    c = sizeof x;   /* c = 2 */
    c = ~c;         /* c = 253 */
}
```

Jika operator tokok dan susut dikenakan **selepas** (posfix) satu pembolehubah, nilai pembolehubah tersebut yg asal akan digunakan **dan selepas** itu baru nilai pembolehubah tersebut ditokok @ disusut.

```
mai n()
{
    int x, xcc, xtt;
    x = 24;
    xcc = x++;      xtt = ++x;
    printf( "x= %d, xcc= %d , xtt= %d", x, xcc, xtt);
}
```

Keluaran skrin:

**x= 26, xcc= 24, xtt= 26**

## 4.2.2 Operator Aritmetik binari

- operator yang terletak di antara dua operan.
- Simbol perwakilan operator binari:

+	operator tambah
-	operator tolak
*	operator darab
/	operator bahagi
%	operator hasil bahagi @ modulus (integer shj.)

Contoh:

```
main()
{
    int x, y, z;

    x = 10; y = 17;
    z = x + y;    /* z di umpukkan dgn. nilai 27 */
    y = y - x;    /* y di umpukkan dgn. nilai 7 */
    x = y * z;    /* x di umpukkan dgn. nilai 189 */
    z = x / 20;   /* z di umpukkan dgn. nilai 9 */
    y = z % x;    /* y di umpukkan dgn. nilai 9 */
}
```

### i) Operator / & %

- / - Melakukan operasi bahagi ttp. bila digunakan dgn pembolehubah integer ia akan memberikan hasil bhg integer(no. bulat).

Contoh:

```
7.0/2.0 hasil 3.5
7.0/2   hasil 3.5
7/2.0   hasil 3.5
7/2     hasil 3
```

- %- Memberi nilai baki bg pembahagian 2 no. HANYA boleh digunakan bersama nilai atau operan berjenis integer

Contoh:

```
7%2     hasil 1
```

## 4.2.2 Operator Aritmetik binari (samb....)

Contoh operasi / & % :

$$\begin{array}{r}
 3 \longleftarrow 7/2 \\
 2 \overline{) 7} \\
 \underline{6} \\
 1 \longleftarrow 7\%2
 \end{array}
 \qquad
 \begin{array}{r}
 2 \longleftarrow 299/100 \\
 100 \overline{) 299} \\
 \underline{200} \\
 99 \longleftarrow 299\%100
 \end{array}$$

ii) Penggunaan Jenis Data Pelbagai dlm Ungkapan Aritmetik binari

- Dlm satu unkanan aritmetik binari, min. 2 operan & 1 operator

Format:

$$\boxed{\text{operan}_1 \text{ operator}_1 \text{ operan}_2 \dots \text{operator}_{n-1} \text{ operan}_n}$$

- Operan `int` & `double`  
jika kesemua operan adalah `integer` nilai terhasil adalah `integer`,  
jika tidak, akan menghasilkan `double`.

Contoh:

```

main()
{
    int i1=3, i2=2, hasil1, hasil2;
    double d=2.0, hasil3, hasil4;

    hasil1 = i1/i2;      /* 3/2 */
    hasil2 = i1/d;      /* 3/2.0 */
    hasil3 = i1/i2;      /* 3/2 */
    hasil4 = i1/d;      /* 3/2.0 */
}

```

hasil1	hasil2	hasil3	hasil4
1	1	1.0	1.5

## 4.2.2 Operator Aritmetik binari (samb....)

### iii) Cast

- Dari contoh berlaku pertukaran jenis data dlm satu ungkapan - pertukaran jenis data secara automatik.  
Pertukaran jenis data boleh diarahkan menggunakan operasi casting.

Format cast:

(j eni s data) pembol eubah@ungkapan

Contoh i:

```
double a=3.0, b=2.0, hasil;
hasil = a % b; /*Ralat Sintaks!!!*/
```

Penyelesaian:

```
hasil = (int)a % (int)b; /*Bebas Ralat*/
```

Contoh 2:

```
main()
{
    int jum_markah = 456, bil_pelajar = 5;
    double purata_markah1, purata_markah2;

    purata_markah1=jum_markah/bil_pelajar;
    purata_markah2=(double)jum_markah / bil_pelajar;
}
```

purata\_markah1

91.0
------

purata\_markah2

91.2
------

### 4.3 Kenyataan dgn pelbagai operator

- Setakat ini kita membincangkan ungkapan menggunakan satu operator pd satu kenyataan.

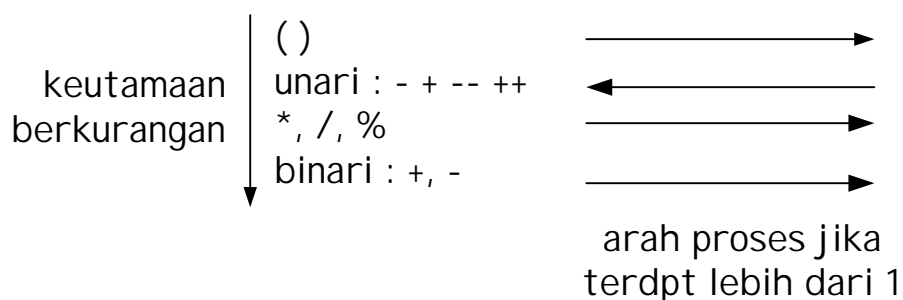
contoh :  $x = -y$ ;  $x = y + z$ ;

- Jika pelbagai operator dlm satu ungkapan spt:

$x = -y + z * 7 \% 2 / ++z$ ;

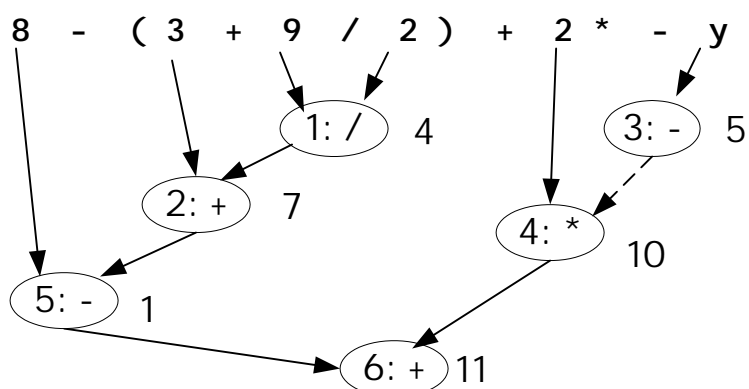
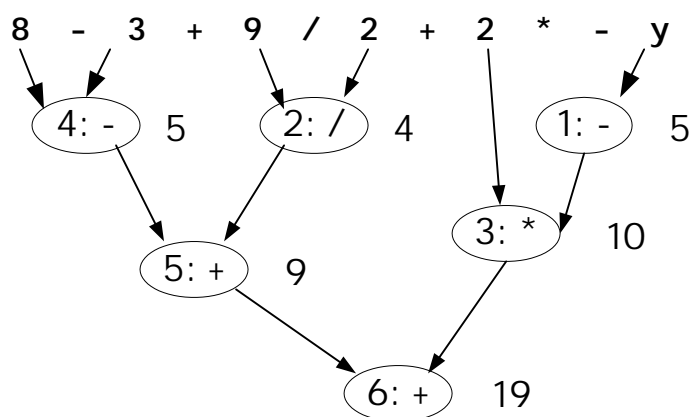
Perlukan syarat keutamaan utk memproses aritmetik tersebut.

Kadar Keutamaan operator unari adalah seperti berikut:



Contoh:

```
int z, y=-5;
z= 8 - 3 + 9 / 2 + 2 * - y;
z= 8 - (3 + 9 / 2) + 2 * - y;
```



### 4.3 Ungkapan dgn pelbagai operator (samb....)

- Mengetahui syarat keutamaan utk memproses aritmetik membantu kita memahami bgm C memproses kenyataan. Ttp utk menghasilkan kod aturcara yang mudah dibaca penggunaan ( ) amat digalakkan.

contoh  $x*y*z+y/b-c$

lebih mudah difahan jika  $x*((y*z)+y)/(b-c)$

Contoh formula math & kenyataan C:

Formula	kenyataan C
$b^2 - 4ac$	$b * b - 4 * a * c$
$\frac{a + b}{c + d}$	$(a + b) / (c + d)$
$\frac{1}{1 + x^2}$	$1 / (1 + x * x)$

### 4.4 Pernyataan Umpukan =

- pernyataan umpukan menggunakan simbol =. Operator umpukan ini juga boleh digunakan dalam 2 cara:
  - pernyataan umpukan berganda
  - pernyataan umpukan kompaun

i) pernyataan umpukan berganda

- Terdapat lebih dari satu operator (=) dalam sesuatu pernyataan umpukan berganda.

Contoh 1:

```
int a=b=c=d=e=250;
```

Contoh 2:

```
int b=2, nombor=0, jum=0, purata=3;
nombor= b = 10 + (jum=10-purata);
printf("\n%d %d %d %d", b , nombor, jum, purata);
```

Output:

```
17 17 7 3
```

#### 4.4 Pernyataan Umpukan = (samb....)

ii) pernyataan umpukan kompaun

- Kenyataan yg memudahkan kenyataan berikut:  
 pembol eubah = pembol eubah operator ungkapan;  
 kepada:  
 pembol eubah operator = ungkapan;

- Gabungan beberapa operator yang dibenarkan adalah seperti berikut:

+=  
 %=  
 -=  
 /=  
 \*=

Contoh:

masa = masa - 1;	/*sama*/	masa -= 1;
bil = bil + 1;	/*sama*/	bil += 1;
jum_ti = jum_ti + ti;	/*sama*/	jum_ti += ti;
n = n * (x + 1);	/*sama*/	n *= x + 1;

#### 4.5 Fungsi Perpustakaan Matematik

- Setakat ini kita mengenali operator math yg mudah spt +, -, \*, / dan %. Ttp ini tidak cukup utk menyelesaikan masalah spt:

$$\sqrt{x} \quad |q+z| \quad \left[ \frac{h}{12.3} \right]^3$$

- Rutin/fungsi utk sesetengah pengiraan penting ini telah disediakan C. Fungsi ini boleh dipanggil bersama dengan arahan prapemproses #i ncl ude. Contoh 2 panggilan perpustakaan matematik:  
 #i ncl ude <math. h>  
 #i ncl ude <stdl i b. h>

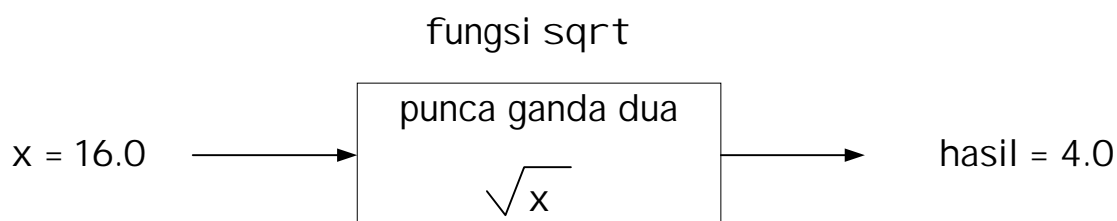


## 4.5 Fungsi Perpustakaan Matematik (samb....)

○ Beberapa fungsi dpd perpustakaan matematik:

fungsi	fail perpus.	tujuan@nilai dikembalikan & cth	argumen	hasil
abs(x)	stdlib.h	mutlak x abs(-5) hasil 5	int	int
exp(x)	math.h	$e^x$ exp(1.0) hasil 2.71828	double	double
log(x)	math.h	$\log_e(x)$ log(2.71828) hasil 1.0	double	double
pow(x, y)	math.h	$x^y$ pow(0.16, 0.5) hasil 0.4	double, double	double
sqrt(x)	math.h	$\sqrt{x}$ dan $x \geq 0.0$ sqrt(2.25) hasil 1.5	double	double

○ Contoh operasi fungsi sqrt dlm kotak hitam:



## 4.6 Operator hubungan

Operator hubungan digunakan utk melakukan perbandingan. Ia digunakan oleh suruhan<sup>2</sup> yg membuat keputusan. Ia memerlukan 2 pembolehubah.

<	kurang dari
<=	kurang dari atau sama dgn
>	lebih besar dari
>=	lebih besar dari atau sama dgn
==	sama dgn
!=	tidak sama dgn

Perhatikan:

$x = 5$  bermaksud berikan nilai 5 kpd pembolehubah  $x$

$x == 5$  bermaksud bandingkan samada nilai  $x$  sama dgn 5

**Jangan terkeliru operator umpukan dgn operator sama dgn !**

Hasil dari operator hubungan ialah samada **BENAR (TRUE)** atau **PALSU (FALSE)**. Dlm C, PALSU diwakili oleh nombor/nilai sifar (0) dan BENAR oleh sebarang nombor bukan sifar (biasanya 1).

Contoh:

kenyataan :  $y = (4 == 5)$  akan memberikan  $y$  nilai sifar

kenyataan :  $y = (x <= z)$

memberikan  $y$  nilai bukan sifar jika  $x$  lebih kecil atau sama dgn  $z$

memberikan  $y$  nilai sifar jika  $x$  lebih besar dari  $z$

## 4.7 Operator logik

Operator logik digunakan utk menggabungkan beberapa ungkapan yg mengandungi operator hubungan. Berguna utk melakukan perbandingan kompleks bagi membuat keputusan.

&&	DAN (AND)
	ATAU (OR)
!	TIDAK (SALINGAN)

## 4.7 Operator logik (samb...)

Jika **ungkapan1** dan **ungkapan2** adalah ungkapan<sup>2</sup> yg mengandung operator hubungan mudah spt  $x < 10$ ,  $z == 0$  dsbnya, maka:

<b>ungkapan1 &amp;&amp; ungkapan2</b>	adalah BENAR jika & hanya jika kedua <sup>2</sup> ungkapan adalah BENAR
<b>ungkapan1    ungkapan2</b>	adalah BENAR jika satu atau kedua <sup>2</sup> ungkapan adalah BENAR
<b>! ungkapan1</b>	adalah BENAR jika ungkapan1 adalah PALSU dan sebaliknya

Aturan **ungkapan1** dan **ungkapan2** dinilai adalah dari kiri ke kanan. Oleh itu utk memastikan ungkapan<sup>2</sup> dinilai spt yg dikehendaki, gunakan kurungan '(' )' utk menentukan aturan ungkapan<sup>2</sup> dinilai.

Contoh:

- a)  $(6 > 2) \&\& (3 == 3)$                       adalah BENAR
- b)  $(6 < 2) \&\& (3 == 3)$                       adalah PALSU
- c)  $! ( (6 < 2) \&\& (3 == 3) )$                       adalah BENAR
- d)  $j != 0 \&\& (x/10 < 3)$                       adalah BENAR jika j bukan sifar dan x lebih kecil dari 30
  
- e)  $(12 != 12) || ((2 > 1) \&\& 5 == 2)$
- f)  $!(10 == 10) || (20 <= 22) || 2000$
- g)  $-100 \&\& -155$
- h)  $x = ( (9 == 0) \&\& (1 < 1) )$

## Latihan

1. Diberi pengenalan pemalar dan pembolehubah

```
#define PI 3.14159
#define MAX_I 1000
.....
double x, y;
int a, b, i;
```

yang mana menunjukkan kepada pernyataan yang sah (cari nilai yang disimpan oleh setiap pernyataan) dan yang mana tidak sah (kenapa).

anggap pembolehubah a adalah 3, b adalah 4, dan y adalah -1.0.

a. $i = a \% b;$	j. $i = (\text{MAX\_I} - 990) / a;$
b. $i = (989 - \text{MAX\_I}) / a;$	k. $x = a / y;$
c. $i = b \% a;$	l. $i = \text{PI} * a;$
d. $x = \text{PI} * y;$	m. $x = \text{PI} / y;$
e. $i = a / -b;$	n. $x = b/a;$
f. $x = a / b;$	o. $i = (\text{MAX\_I} - 990) \% a;$
g. $x = a \% (a / b);$	p. $i = a \% 0;$
h. $i = b / 0;$	q. $i = a \% (\text{MAX\_I} - 990);$
i. $i = a \% (990 - \text{MAX\_I});$	

2. Anggapkan anda mempunyai pengenalan pembolehubah berikut:

```
int color, lime, straw, kuning, merah, oren;
double hitam, putih, hijau, biru, ungu, crayon;
```

Cari nilai setiap pernyataan a.-f. yg mana nilai pembolehubah 'color' adalah 2, 'hitam' adalah 2.5, 'crayon' adalah -1.3, 'straw' adalah 1, 'merah' adalah 3, 'ungu' adalah  $0.3E + 1$ .

```
a. putih = color * 2.5 / ungu;
b. hijau = color / ungu;
c. oren = color / red;
d. biru = (color + straw) / (crayon + 0.3);
e. lime = merah / color + red % color;
f. ungu = straw / merah * color;
```