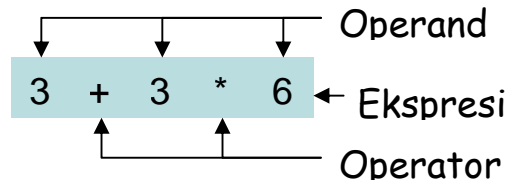


MODUL 3 - 4

OPERATOR DAN EKSRESI



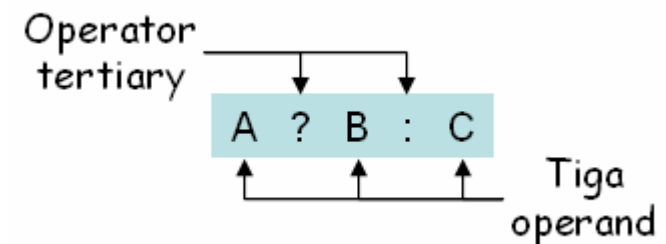
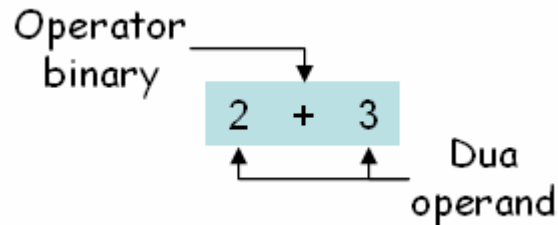
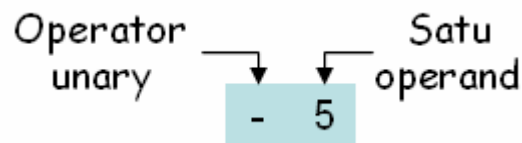
Eksresi/ungkapan : Suatu bentuk yang menghasilkan suatu nilai.
 Operator : simbol atau karakter khusus (mis. +,*) yang digunakan dalam suatu eksresi untuk menghasilkan suatu nilai.

Operator :

- ❖ Operator Aritmetika
- ❖ Operator penambahan (increment) dan pengurangan (decrement)
- ❖ Operator penugasan
- ❖ Operator perbandingan
- ❖ Operator logika
- ❖ Operator bit

Menurut jumlah operand yang dilibatkan, operator diklasifikasikan menjadi menjadi :

- Operator unary, yaitu operator yang hanya melibatkan satu buah operand
- Operator binary, yaitu operator yang melibatkan dua buah operand
- Operator tertiary, yaitu operator yang melibatkan tiga buah operand



Operator Aritmetika:

Operator	Keterangan
+	Penjumlahan
	Tanda plus
-	Pengurangan
	Tanda minus
*	Perkalian
/	Pembagian
%	Sisa pembagian(Modulus)

Kode sumber	Hasil
<pre>//aritmetika.cpp #include<iostream.h> void main(void) { int a=11; int b=3; cout<<a+b<<endl; cout<<a*b<<endl; cout<<a-b<<endl; cout<<a/b<<endl; cout<<a%b<<endl; }</pre>	 14 33 8 3 2

Urutan pelaksanaan operator aritmetika :

Operator	Urutan
()	Dievaluasi pertama. Jika berkalang maka kalang paling dalam dievaluasi paling pertama. Jika ada beberapa pasangan kurung dalam satu level maka dievaluasi dari kiri ke kanan
*, /, atau %	Dievaluasi kedua. Jika ada beberapa maka dievaluasi dari kiri ke kanan
+ atau -	Dievaluasi terakhir. Jika ada beberapa maka dievaluasi dari kiri ke kanan

Contoh : Aljabar : $z = pr\%q + w/x - y$

C: z = p * r % q + w / x - y ;

 6 1 2 4 3 5

Tentukan urutan operasi aritmetika dari persamaan berikut :

$y = a(b+c) + c(a-b(d+a)(a\%b))$, jika $a=20$, $b=10$, $c=7$, $d=2$

Operator Penambahan dan Pengurangan :

$x = x + 1 \rightarrow x++$ atau $++x$

$x = x - 1 \rightarrow x--$ atau $--x$

x semula	Pernyataan	Hasil y	Hasil x
5	$y = x++;$	5	6
5	$y = ++x;$	6	6
5	$y = x--;$	5	4
5	$y = --x;$	4	4

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    int x,y;
```

```
    x = 5;
```

```
    y = 8 - x++;
```

```
    cout<<x<<endl;
```

```
    cout<<y<<endl;
```

} $x = 6$
 $y = 3$

```
    x = 5;
```

```
    y = 8 - ++x;
```

```
    cout<<x<<endl;
```

```
    cout<<y<<endl;
```

} $x = 6$
 $y = 2$

$x = 5$
 $y = 8 - x++$
↑
3
x=6

$x = 5$
 $y = 8 - ++x$
↑
2
x=6

```

    x = 5;
    y = 8 - x--;
    cout<<x<<endl;
    cout<<y<<endl;
}

    x = 5;
    y = 8 - --x;
    cout<<x<<endl;
    cout<<y<<endl;
    return 0;
}

```

Diagram illustrating the execution of the two code snippets:

Snippet 1 (x = 5; y = 8 - x--):

- Initial state: $x = 5$
- Expression: $y = 8 - x--$. The value of x is 5, and then it is decremented to 4.
- Result: $y = 3$
- Final state: $x = 4, y = 3$

Snippet 2 (x = 5; y = 8 - --x):

- Initial state: $x = 5$
- Expression: $y = 8 - --x$. The value of x is first decremented from 5 to 4, and then used in the calculation.
- Result: $y = 4$
- Final state: $x = 4, y = 4$

Operator Bit

C++ mendukung 6 buah operator yang beroperasi pada level bit (0 dan 1), yaitu :

- ❖ & (AND)
- ❖ | (OR)
- ❖ ^ (XOR)
- ❖ ~ (NOT)
- ❖ << (Geser Kiri)
- ❖ >> (Geser Kanan)

Operator &

Bit 1	Bit 2	Hasil
0	0	0
0	1	0
1	0	0
1	1	1

Contoh :

9 & 10 memberikan hasil 8

1	0	0	1	→	9
1	0	1	0	→	10
1	0	0	0	→	8

Operator |

Bit 1	Bit 2	Hasil
0	0	0
0	1	1
1	0	1
1	1	1

Contoh :

9 | 10 memberikan hasil 11

1	0	0	1	→	9
1	0	1	0	→	10
1	0	1	1	→	11

Operator ^

Bit 1	Bit 2	Hasil
0	0	0
0	1	1
1	0	1
1	1	0

Contoh :

9 ^ 10 memberikan hasil 3

1	0	0	1	→	9
1	0	1	0	→	10
0	0	1	1	→	3

Operator ~

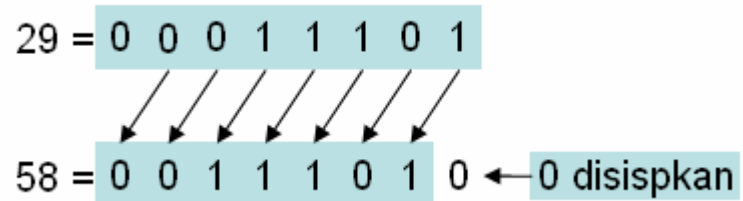
~185 memberikan hasil -186

0	1	0	1	1	1	0	0	1	→	185
↓	↓	↓	↓	↓	↓	↓	↓	↓		
1	0	1	0	0	0	1	1	0	→	-186

Operator <<

$x \ll n$ identik dengan $x * 2^n$

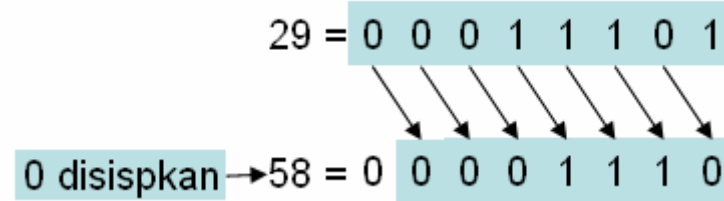
$29 \ll 1$ memberikan hasil 58



Operator >>

$x \gg n$ identik dengan $x / 2^n$

$29 \gg 1$ memberikan hasil 14



Tentukan output dari program berikut?

//Operasi Bit

```
#include<iostream.h>
```

```
void main(void)
```

```
{
```

```
    int x=185;
```

```
    int y=5;
```

```
    cout<<"~" << x <<" = "<<~x<<endl;
```

```
    cout<<x<<" | "<<y<<" = "<<(x|y)<<endl;
```

```
    cout<<x<<" & "<<y<<" = "<<(x&y)<<endl;
```

```
    cout<<x<<" ^ "<<y<<" = "<<(x^y)<<endl;
```

```
    cout<<x<<" << "<<y<<" = "<<(x<<y)<<endl;
```

```
    cout<<x<<" >> "<<y<<" = "<<(x>>y)<<endl;
```

```
}
```

Operator Penugasan

Operator	Keterangan	Contoh
=	Pemberian nilai	
+=	Penambahan bilangan	$a += 2$ sama dengan $a = a + 2$
-=	Pengurangan bilangan	$a -= 2$ sama dengan $a = a - 2$
*=	Pengalian bilangan	$a *= 2$ sama dengan $a = a * 2$
/=	Pembagian bilangan	$a /= 2$ sama dengan $a = a / 2$
%=	Pemerolehan sisa bagi	$a \% = 2$ sama dengan $a = a \% 2$

Operator	Keterangan	Contoh
&=	Pemberian nilai	$a \&= x$ sama dengan $a = a \& x$
^=	Penambahan bilangan	$a \hat{=} x$ sama dengan $a = a \hat{ } x$
=	Pengurangan bilangan	$a = x$ sama dengan $a = a x$
<<=	Pengalian bilangan	$a \ll = x$ sama dengan $a = a \ll x$
>>=	Pembagian bilangan	$a \gg = x$ sama dengan $a = a \gg x$

Tentukan nilai a dan b dari pernyataan berikut:

$a=b=6$

$a += 5$

$b -= 2$

$a \% = b$

TYPE CASTING

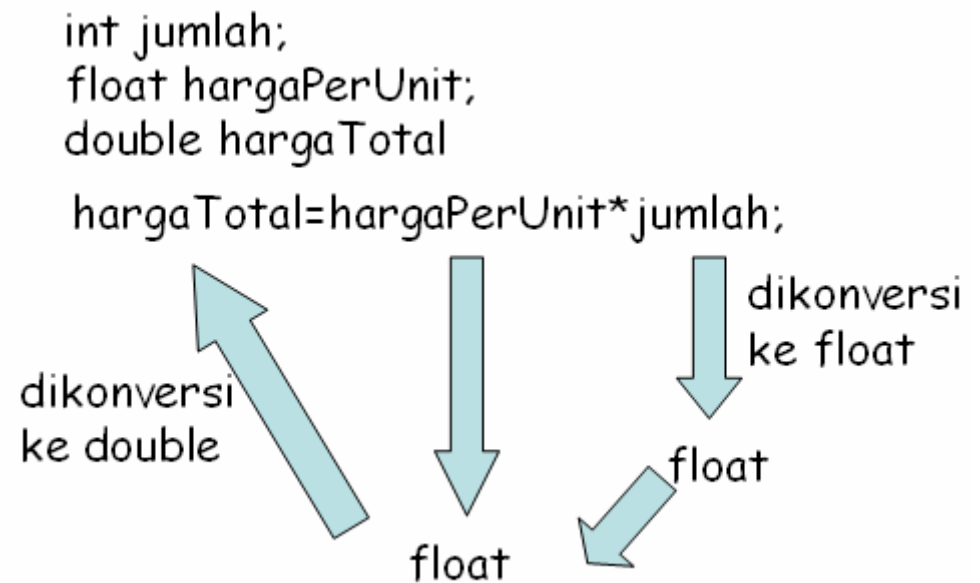
Operasi perhitungan pada C++ dilakukan dengan menggunakan tipe data yang sama. Jika dalam suatu ekspresi terdapat operand dengan tipe yang berbeda, C++ akan mengkonversikan salah satu tipe sehingga kedua tipe menjadi sama dengan aturan :

1. Jika salah satu operand bertipe *long double*, yg lain dikonversikan menjadi *long double*
2. Jika salah satu operand bertipe *double*, yg lain dikonversikan menjadi *double*
3. Jika salah satu operand bertipe *float*, yg lain dikonversikan menjadi *float*
4. Jika salah satu operand bertipe *char*, *signed char*, *unsigned char*, atau *unsigned short*, yg lain dikonversikan menjadi *int*
5. Tipe enumerasi akan dikonversikan ke *int*, *unsigned int*, *long*, atau *unsigned long* dengan mengakomodasikan jangkauan tipe enumerasi
6. Jika salah satu operand bertipe *unsigned long*, yg lain dikonversikan menjadi *unsigned long*
7. Jika salah satu operand bertipe *long* dan yang lain bertipe *unsigned int*, kedua operand dikonversikan menjadi *unsigned long*
8. Jika salah satu operand bertipe *long* , yg lain dikonversikan menjadi *long*

Type casting : proses mengubah suatu tipe ke tipe yang lain. Bentuk pengarah tipe sbb :

```
(tipe_data)data  
tipe_data(data)  
static_cast<tipe_data>(data)
```

Code Program	Output
<pre>//Type casting #include<iostream.h> void main(void) { double fa,fb,fc; fa=3.14; fb=5.23; fc=10.25; cout<<fa<<endl; cout<<fb<<endl; cout<<fc<<endl; cout<<(int)fa<<endl; cout<<int(fb)<<endl; cout<<static_cast<int>(fc)<<endl; }</pre>	<pre>3.14 5.23 10.25 3 5 10</pre>



FILE I/O

Untuk dapat mengakses file menggunakan C, dapat dilakukan dengan menggunakan pointer FILE.

Contoh :

```
FILE *fp;
```

Untuk membuka sebuah file :

```
FILE *fopen(const char *filename, const char *mode);
```

mode :

- r : open for reading
- w : open for writing (file need not exist)
- a : open for appending(file need not exist)
- r+ : open for reading and writing, start at beginning
- w+ : open for reading and writing (overwrite file)
- a+ : open for reading and writing (overwrite file)

Contoh sederhana :

```
FILE *fp;
fp=fopen("c:\\test.txt", "w");
fprintf(fp, "Testing...\n");
fclose(fp);
```

Code Program	Hasil
<pre>//Create a sequential file #include<stdio.h> void main(void) { int akun; char nama[30]; FILE *fp; fp=fopen("d:\\kampus\\program\\c++\\data1.dat","w"); printf("Masukan akun, nama\n"); scanf("%d%s",&akun,nama); fprintf(fp,"%d %s\n",akun,nama); fclose(fp); }</pre>	<p>Input :</p> <p>100 Heri [Enter]</p> <p>Output :</p> <p>File yang bernama data1.dat akan berisi data :</p> <p>100 Heri</p>

Code Program	Hasil
<pre>#include<stdio.h> void main(void) { int akun; char nama[30]; FILE *fp; fp=fopen("d:\\kampus\\program\\c++\\data1.dat","r"); fscanf(fp,"%d %s\n",&akun,nama); fclose(fp); printf("%d %s\n",akun,nama); }</pre>	<p>Akan tercetak di layar :</p> <p>100 Heri</p>

FORMAT KELUARAN PADA C++

Manipulator dalam C++

Manipulator	Fungsi
endl	end line. Kursos berpindah ke posisi awal baris berikutnya sama dg \n
ends	Menyisipkan karakter NULL
flush	Mencetak isi buffer
dec	Mengkonversi ke bilangan desimal
hex	Mengkonversi ke bilangan hexadesimal
oct	Mengkonversi ke bilangan oktal
setbase(n)	Mengkonversi ke bilangan berbasis n
setw(n)	Mengatur lebar cetakan selebar n
setfill(n)	Mengisi leading field dengan karakter n

setprecision(n)	Membuat lebar desimal point=n
setiosflags(lf)	Menset format yang diatur dengan tanda format ios:
resetiosflags(lf)	Merest format yang diset oleh setioflags()

Setiap menggunakan manipulator setxxxxx() atau resetxxxxxx() harus menggunakan :
#include<iomanip.h>

Code Program	Hasil
<pre>//Penggunaan setw #include<iostream.h> #include<iomanip.h> void main(void) { int x; x=64; cout<<123456789<<endl; cout<<setw(9)<<123<<endl; cout<<123<<endl; cout<<hex<<x<<endl; cout<<oct<<x<<endl; cout<<dec<<x<<endl; cout<<setfill('.'); cout<<setw(10)<<x<<endl; cout<<x<<endl; }</pre>	<pre>123456789 123 123 40 100 6464 64</pre>

setw() hanya berlaku untuk satu elemen cetak berikutnya

Code Program	Hasil
<pre>//Penggunaan setprecision() #include<iostream.h> #include<iomanip.h> void main(void) { double x=123.456789; cout<<x<<endl; cout<<setprecision(0)<<x<<endl; cout<<setprecision(1)<<x<<endl; cout<<setprecision(2)<<x<<endl; cout<<setprecision(3)<<x<<endl; cout<<setprecision(4)<<x<<endl; cout<<setprecision(5)<<x<<endl; cout<<setprecision(6)<<x<<endl; cout<<setprecision(7)<<x<<endl; cout<<setprecision(8)<<x<<endl; cout<<setprecision(9)<<x<<endl; cout<<setprecision(10)<<x<<endl; }</pre>	<pre>123.457 1e+002 1e+002 1.2e+002 123 123.5 123.46 123.457 123.4568 123.45679 123.456789 123.456789</pre>

Tanda Format

Untuk mengatur format cetakan diperlukan tanda format yang diset dengan menggunakan manipulator `setiosflags()` dan direset dengan menggunakan manipulator `resetiosflags()`.

Tanda Format	Fungsi
<code>ios::left</code>	Keluaran yang diatur oleh <code>setw()</code> dicetak rata kiri
<code>ios::right</code>	Keluaran yang diatur oleh <code>setw()</code> dicetak rata kanan
<code>ios::scientific</code>	Keluaran dicetak dalam bentuk scientific
<code>ios::fixed</code>	Keluaran dicetak dalam bentuk fixed point
<code>ios::dec</code>	Keluaran dicetak dalam bentuk desimal
<code>ios::hex</code>	Keluaran dicetak dalam bentuk hexadesimal
<code>ios::oct</code>	Keluaran dicetak dalam bentuk oktal
<code>ios::uppercase</code>	Keluaran yang dicetak dalam bentuk hexadesimal dicetak dengan huruf besar
<code>ios::showbase</code>	Menambahkan 0x diawal hasil cetakan yang berbentuk hexademal atau 0 (nol) pada cetakan yang berbentuk oktal
<code>ios::showpoint</code>	Menampilkan desimal point pada hasil cetakan yang mengandung pecahan
<code>ios::showpos</code>	Menambahkan tanda + pada hasil cetakan yang bernilai positif

Code Program	Hasil
<pre>//Penggunaan setiosflags() #include<iostream.h> #include<iomanip.h> void main(void) { int x=255; double y=123.44; cout<<setiosflags(ios::showbase); cout<<setiosflags(ios::left)<<setw(10)<<x<<endl; cout<<resetiosflags(ios::left); cout<<setiosflags(ios::right)<<setw(10)<<x<<endl; cout<<setiosflags(ios::hex)<<x<<endl; cout<<resetiosflags(ios::hex); cout<<setiosflags(ios::oct)<<x<<endl; cout<<resetiosflags(ios::oct); cout<<setiosflags(ios::dec)<<x<<endl; cout<<setiosflags(ios::fixed)<<setprecision(5)<<y<<endl; cout<<resetiosflags(ios::fixed); cout<<setiosflags(ios::scientific)<<setprecision(5)<<y<<endl; }</pre>	<pre> 255 255 0xff 0377 255 123.44000 1.23440e+00 2</pre>

Soal :

1. Carilah nilai j dan terangkan bagaimana nilai j diperoleh, jika diketahui :

int m=24, k=21;

- | | | | | |
|------------------|------------------|-----------------|--------------------|------------------|
| a. $j = m k$ | b. $j = m \% k$ | c. $j = m \& k$ | d. $j = m + ++k$ | e. $j = m - --k$ |
| f. $j = m - --k$ | g. $j = m \gg 2$ | h. $j = k += m$ | i. $j = --m + k++$ | j. $j = --m * 5$ |

2. Buatlah program untuk konversi suhu dengan rumus :

Fahrenheit=(9/5)C+32

Reamur=(4/9)C+32

C dalam derajat celcius, Hasilnya disimpan dalam file suhu.dat dengan bentuk :

C F R

...

3. Pak Budi menyimpan uangnya di BCA sebesar M rupiah dengan bunga B% per tahun selama 5 tahun. Buatlah program dan simpan hasilnya dalam file bank.dat untuk menghitung uang Pak Buda pada akhir tahun ke 5.

(Inputnya M, B)

Rumus :

$$U = M(1+B)^n$$

U : uang pada akhir tahun ke-n

M : uang pada awal tahun

B : bunga per tahun