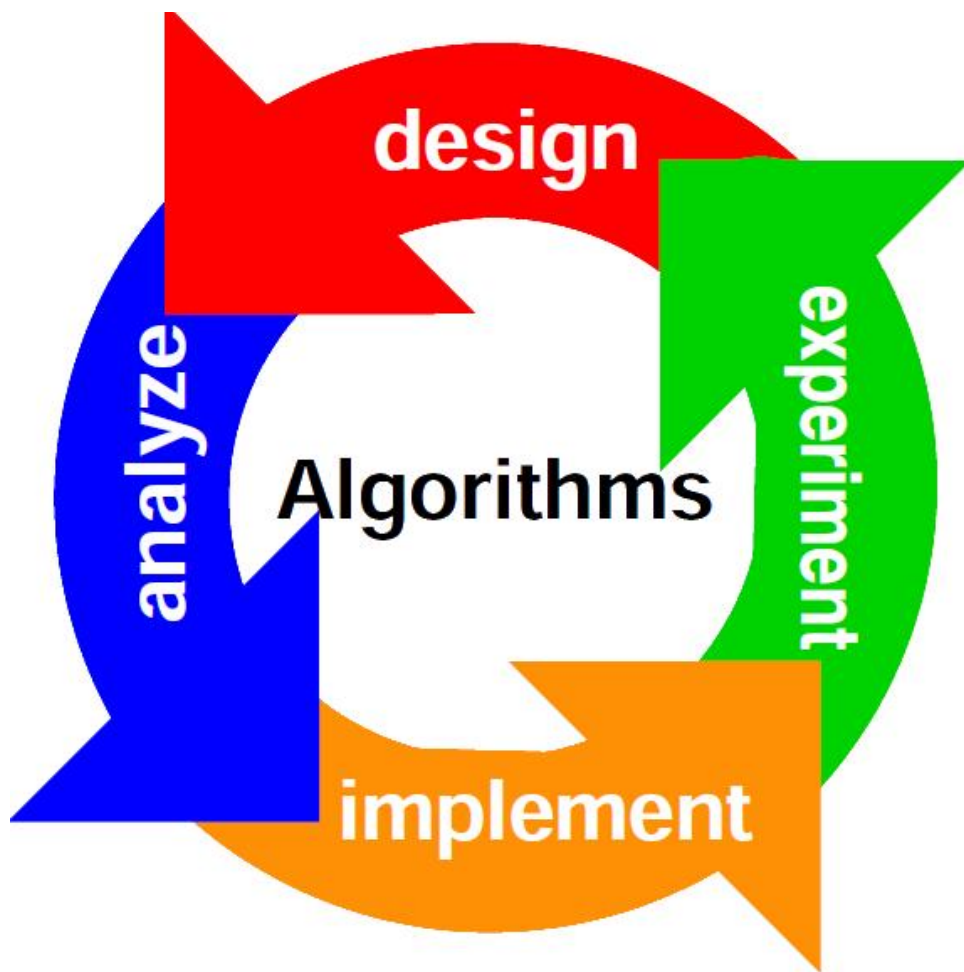


ALGORITMA DAN PEMROGRAMAN 3

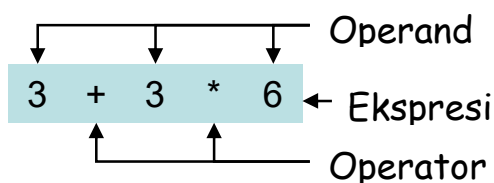


TEKNIK PERANGKAT LUNAK
FT - UNPAM

MODUL 3

(Operator dan ekspresi)

OPERATOR DAN EKSPRESI



Ekspresi/ungkapan : Suatu bentuk yang menghasilkan suatu nilai.

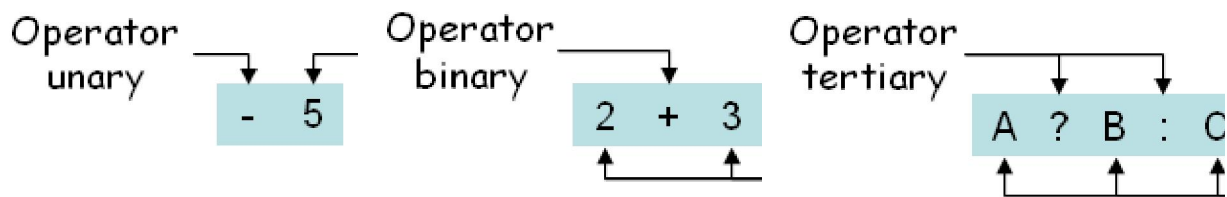
Operator : simbol atau karakter khusus (mis. +,*) yang digunakan dalam suatu ekspresi untuk menghasilkan suatu nilai.

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 = 5;
```

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

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

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

```
    x = 5;
```

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

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

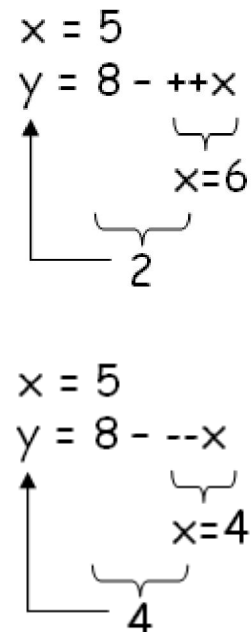
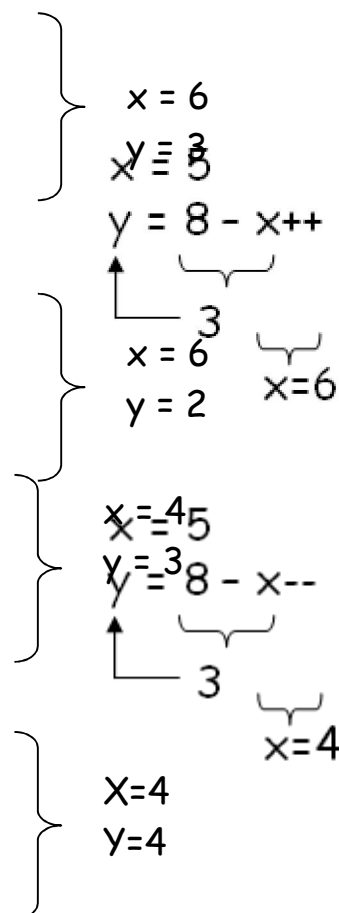
```
    cout<<y<<endl;
```

```
    x = 5;
```

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

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

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



```

return 0;

}

```

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
<hr/>					
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
<hr/>					
1	0	1	1	→	11

Operator ^

Bit 1	Bit 2	Hasil
0	0	0
0	1	1

Contoh :
 9 ^ 10 memberikan
 hasil 3

1	0	0	1	→	9
1	0	1	0	→	10
<hr/>					
0	0	1	1	→	3

1	0	1
1	1	0

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 << 1 memberikan hasil 58

29 = 0 0 0 1 1 1 0 1

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

58 = 0 0 1 1 1 0 1 0 ← 0 disisp 0 disispkan

Operator >>

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

29 >> 1 memberikan hasil 14

29 = 0 0 0 1 1 1 0 1

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

14 = 0 0 0 0 1 1 0 1

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	Out put
<pre>//Type casting #include<iostream.h> void main(void) { double fa,fb,fc; fa=3.14; fb=5.23;</pre>	

fc=10.25;	
cout<<fa<<endl;	3.14
cout<<fb<<endl;	5.23
cout<<fc<<endl;	10.2
cout<<(int)fa<<endl;	5
cout<<int(fb)<<endl;	3
cout<<static_cast<i	5
nt>(fc)<<endl;	10
}	

```
int jumlah;
float hargaPerUnit;
double hargaTotal
hargaTotal=hargaPerUnit*ju
```

dikonversi ke double

float

Soal :

Coba kalian buat algoritma dasar dengan menggunakan fungsi aritmatika, dikumpulkan paling lambat sampai jam

23:00