

Bab 3

Dasar-Dasar C++

Sub CPMK:

Setelah mempelajari materi dalam Bab 3 anda akan mengetahui tentang pengertian peubah dan operator dan menggunakannya dalam pembuatan program.

3.1 Peubah dan Operator

Komputer merupakan suatu alat yang dapat digunakan untuk memproses data. Untuk melakukan pemrosesan data, maka dibutuhkan data harus disimpan dalam peubah atau variabel. Ada beberapa jenis variabel sederhana yaitu :

- (1) Tipe karakter
- (2) Tipe integer.
- (3) Tipe real.

Tipe karakter

Untuk mendeklarasikan peubah karakter kita gunakan kata kunci char di depan nama variabel. Sebagai contoh:

```
char karakter;  
char kar1,kar2,kar3;  
char kar4='A';  
char kar5=65;
```

Tipe karakter mempunyai jangkauan nilai dari 0 sampai 255 atau karakter ASCII ke 0 sampai karakter ASCII 255. Tipe karakter bisa ditampilkan sebagai suatu karakter atau sebagai suatu bilangan. Hal ini tergantung dari bagaimana penulisannya apakah dianggap sebagai karakter atau sebagai bilangan.

Untuk menampilkan isi dari data bertipe karakter kelayar monitor digunakan perintah printf dengan format penulisannya menggunakan tanda %c kalau ingin ditampilkan sebagai suatu karakter atau dengan %i jika ingin ditampilkan sebagai suatu angka. Sebagai contoh:

```
printf("Karakter dilihat dalam bentuk karakter = %c.\n",karakter);
```

```
printf("Karakter dilihat dalam bentuk angka = %i.\n",karakter);
```

Pemberian nilai (*assignment*) kepada suatu karakter digunakan perintah sebagai berikut :

```
karakter='A';
```

atau

```
karakter=65;
```

Kedua cara tersebut menghasilkan suatu efek yang sama yaitu memberikan nilai 65 atau karakter A ke variabel karakter. Kalau pengisian variable ingin menggunakan karakter maka karakter yang akan dimasukan harus diapit dengan tanda petik tunggal (*single quote*).

Contoh program 3.1:

```
#include <stdio.h>
#include <conio.h>
main()
{
    char k1,k2;
    k1='A';
    k2=k1;
    printf("Nilai variable K1 adalah %c\n",k1);
    printf("Nilai variable K2 dalam bentuk angka = %i\n",k2);
    getch();
    return 0;
}
```

Contoh Output Program:

Nilai variable K1 adalah A

Nilai variable K2 dalam bentuk angka = 65

Tipe Integer

Ada beberapa tipe variabel yang termasuk dalam tipe integer/bilangan bulat, sebagaimana disajikan dalam table berikut ini:

Tipe Data	Memori	Format	Jangkauan Nilai
int	2 byte	%d/%i	-32.768 s/d 32.767
unsigned int	2 byte	%u	0 s/d 65.535
char	1 byte	%d/%i	-128 s/d 127
unsigned char	1 byte	%u	0 s/d 255
unsigned long	4 byte	%lu	0 s/d 4.294.967.295
long	4 byte	%ld/%li	-2.147.483.648 s/d 2.147.483.647

Cara pendeklarasian tipe data ini sama seperti pendeklarasian lainnya, yaitu :

```
int a;
unsigned int b;
char c;
long d;
```

Contoh Program :

```
#include <stdio.h>
#include <conio.h>

main()
{
    int a=1000,b=64000;
    unsigned int c=64000;
    printf("Nilai yang telah dimasukan\n a: %i dan b: %i\n",a,b);
    printf("Nilai yang telah dimasukan : %u\n",c);
    getch();
    return 0;
}
```

Output Program:

a: 1000 dan b: -1536
 Nilai yang telah dimasukan : 64000

Tipe Real

Ada beberapa tipe variabel yang termasuk dalam tipe real, sebagaimana disajikan dalam table berikut ini:

Tipe Data	Memori	Format	Jangkauan Nilai
float	4 byte	%f	$3.4 \times (10^{-38}) - 3.4 \times (10^{+38})$
double	8 byte	%f	$1.7 \times (10^{-308}) - 1.7 \times (10^{+308})$
long double	10 byte	%lf	$3.4 \times (10^{-4932}) - 1.1 \times (10^{+4932})$

Cara pendeklarasian tipe data ini sama seperti pendeklarasian lainnya, yaitu :

```
float f;
double d;
long double l;
```

Contoh Program :

```
#include <conio.h>

main()
{
    float a=1234567890123456789;
    double d=1234567890123456789;
    printf("Nilai a adalah : %30.20f\n",a);
    printf("Nilai d adalah : %30.20f\n",d);
    getch();
    return 0;
}
```

Output Program :

Nilai a adalah : 1234567939550609410.00
 Nilai d adalah : 1234567890123456770.00

Operator Aritmatika

Pemrosesan data akan melibatkan operator-operator. Operator-operator yang terdapat dalam bahasa C++ antara lain seperti yang tersaji pada tabel berikut ini:

Operator	Contoh	Arti
+	c=a+b	Variable c diisi dari isi variable a ditambah isi variable b
-	c=a-b	Variable c diisi dari isi variable a dikurangi isi variable b
*	c=a*b	Variable c diisi dari isi variable a dikali dengan isi variable b
/	c=a/b	Variable c diisi dari isi variable a dibagi oleh isi variable b
++	a++	Isi variable a ditambah 1. Perintah ini sama dengan a=a+1 atau a+=1
--	b--	Isi variable a dikurang. Perintah ini sama dengan a=a-1 atau a-=1
%	c=a % b	Variable c diisi dari sisa pembagian variable a dibagi variable b
+=	c+=a	Variable c ditambah dengan isi variable a. Sama dengan c=c+a
/=	c/=a	Variable c dibagi dengan isi variable a. Sama dengan c=c/a
-=	c-=a	Variable c dikurangi dengan isi variable a. Sama dengan c=c-a
=	c=a	Variable c dikali dengan isi variable a. Sama dengan c=c*a
%=	c%=a	Variable c diisi dari sisa pembagian c dibagi isi variable a. Sama dengan c=c%a

Operator Logika

Operator-operator logika yang terdapat dalam bahasa C++ dapat dilihat pada tabel berikut ini:

Operator	Contoh	Arti
==	A==B	Apakah Isi Variabel A sama dengan Isi Variabel B
!=	A!=B	Apakah Isi Variabel A Tidak Sama Dengan Isi Variabel B
>	A>B	Apakah Isi Variabel A lebih besar dari Isi Variabel B
<	A<B	Apakah Isi Variabel A lebih kecil dari Isi Variabel B
>=	A>=B	Apakah Isi Variabel A lebih besar atau sama dengan Isi Variabel B
<=	A<=B	Apakah Isi Variabel A lebih kecil atau sama dengan Isi Variabel B
&&	(A<=100) &&(A>=80)	Apakah A lebih kecil atau sama dengan dari 100 dan A lebih besar atau sama dengan 80
 	(A>100) (A<0)	Apakah A lebih besar dari 100 atau A lebih kecil dari 0
!	!(A==B)	Apakah A Tidak Sama dengan B

Untuk memantapkan pemahaman anda tentang penggunaan operator logika, perhatikan dua contoh program berikut ini:

```
#include <conio.h>

main()
{
    int Nilai;
    printf("Nilai : ");scanf("%i",&Nilai);
    if(Nilai>=50)
        printf("Selamat Anda Lulus.");
    else
        printf("Maaf. Anda Tidak Lulus.");
    getch();
    return 0;
}
```

Perintah di atas hanya mempunyai 2 kemungkinan yaitu keterangan “Selamat Anda Lulus” jika nilai lebih besar dari atau sama dengan 50 dan keterangan “Maaf. Anda Tidak Lulus”, ketika nilai lebih kecil dari 50. Jika perintah yang akan dieksekusi ketika kondisi tercapai lebih dari 1 perintah, maka perintah-perintah tersebut harus diblok dengan tanda kurawal {}.

```
#include <stdio.h>
#include <conio.h>

main()
{
    int a,b;
```

```

printf("Masukan A : ");scanf("%i",&a);
printf("Masukan B : ");scanf("%i",&b);
if(a==b)
    printf("Isi Variabel A Sama Dengan B");
else if(a>b)
    printf("Isi Variabel A lebih besar dari B");
else if(a<b)
    printf("Isi Variabel A lebih kecil dari B");
getch();
return 0;
}

```

Program di atas akan meminta anda untuk memasukan nilai variabel A dan B, kemudian program akan memeriksa apakah variabel A sama dengan B, atau A lebih besar dari B, dan A lebih kecil dari B.

3.2 Console Masukan/Keluaran

Console Masukan

Pada umumnya suatu program mempunyai proses pemasukan data. Pada sebuah program yang ditulis dengan menggunakan bahasa C++, pemasukan data dapat dilakukan dengan perintah scanf. Fungsi scanf merupakan fungsi yang dapat digunakan untuk memasukan berbagai jenis data, tergantung dengan format penentunya. Format-format tipe variabel yang terdapat dalam bahasa C++ antara lain:

Format	Kegunaan
%c	Digunakan untuk pemasukan data bertipe char
%i atau %d	Digunakan untuk pemasukan data bertipe int, char.
%u	Digunakan untuk pemasukan data berupa unsigned int atau unsigned char.
%f	Digunakan untuk pemasukan data berupa bilangan pecahan (float)
%o	Digunakan untuk pemasukan data angka berbasis oktal
%x	Digunakan untuk pemasukan data angka berbasis hexadesimal
%s	Digunakan untuk pemasukan data berupa string.

Sintaks untuk fungsi scanf adalah:

```
scanf("format",&namavariabel);
```

Contoh :

```

int i,jam,menit,detik;
unsigned int j;
float f;
char nama[60];

```

```
scanf("%i",&i);
scanf("%u",&j);
scanf("%f",&f);
scanf("%i %i %i",&jam,&menit,&detik);
scanf("%s",nama);
```

Fungsi scanf memiliki suatu kelemahan yaitu data yang akan dibaca hanya sampai spasi pertama yang ditemuinya, jadi apabila kita mengisikan 2 buah kata dengan pemisah spasi, maka kata yang terbaca hanyalah kata yang pertama. Oleh karena itu, pembacaan data bertipe string sebaiknya menggunakan perintah gets() dengan sintaks:

```
gets(namavariabel);
```

Contoh Penggunaan:

```
gets(nama);
gets(alamat);
```

Untuk pembacaan data bertipe char, selain dengan menggunakan scanf dengan format %c, bisa juga dengan menggunakan fungsi getch atau getche. Perbedaan antara getch dan getche adalah getch digunakan untuk membaca data bertipe char tanpa menampilkannya di layar, dan getche digunakan untuk membaca data bertipe karakter dengan menampilkan data yang dimasukkan di layar.

Contoh :

```
#include <stdio.h>
#include <conio.h>

main()
{
    char a,b;
    printf("Masukan Huruf pertama : ");
    a=getch();
    printf("Masukan Huruf kedua : ");
    b=getche();
    printf("Data yang dimasukan adalah %c dan %d\n",a,b);
    getch();
    return 0;
}
```

Console Keluaran

Untuk menampilkan data ke layar monitor, perintah yang bisa dipakai adalah cout atau printf. Ketika menampilkan data dengan fungsi printf, kita harus mengatur format tampilannya, dengan format-format penentu. Untuk lebih jelas perhatikan program di bawah ini.

```
#include <stdio.h>
#include <conio.h>

main()
{
    int a=25000;
    unsigned int b=50000;
    float c=12345.678;
    char nama[50]="Institut Teknologi Telkom ";
    char alamat[10]="Purwokerto";
    clrscr();
    printf("Penampilan data tanpa di format\n");
    printf("Nilai a : %d\n",a);
    printf("Nilai b : %u\n",b);
    printf("Nilai c : %f\n",c);
    printf("String : %s %s\n",nama,alamat);//rata kanan
    printf("Penampilan data setelah di format\n");
    printf("Nilai a : %8d\n",a);
    printf("Nilai b : %8u\n",b);
    printf("Nilai c : %11.2f\n",c);
    printf("String 1: %40s %10s\n",nama,alamat);//rata kanan
    printf("String 2: %-40s %-10s\n",nama,alamat);//rata kanan
    getch();
    return 0;
}
```

Program di atas akan menampilkan hasil eksekusi seperti di bawah ini :

```
Penampilan data tanpa di format
Nilai a : 25000
Nilai b : 50000
Nilai c : 12345.677734
String : Institut Teknologi Telkom Purwokerto
Penampilan data setelah di format
Nilai a : 25000
Nilai b : 50000
Nilai c : 12345.68
String 1: Institut Teknologi Telkom Purwokerto
String 2: Institut Teknologi Telkom Purwokerto
```