

FUNCTION

Bab 3

3.1 PENGERTIAN FUNGSI DI C++

Function/fungsi adalah satu blok kode yang melakukan tugas tertentu atau satu blok instruksi yang di eksekusi ketika dipanggil dari bagian lain dalam suatu program.

Bentuk umum deklarasi fungsi :

Tipe nilai balik fungsi (tipe para, ...) ;

Keterangan :

- **tipe nilai balik =**
tipe nilai yang dikembalikan dengan statemen 'return'. Tipe default nya : 'int'. Untuk menyatakan fungsi yang tidak mengembalikan nilai balik, dideklarasikan sebagai : 'void'
- **fungsi =**
nama fungsi tersebut
- **tipe para =**
tipe parameter, bila parameter lebih dari satu (1), masing-masing dipisahkan dengan tanda koma (,)
untuk menyatakan fungsi tanpa parameter dispesifikasikan : 'void'. Bila tipe parameter tidak dispesifikasikan, defaultnya : 'void'

Fungsi harus dideklarasikan terlebih dahulu sebelum didefinisikan. Maksudnya adalah memberitahu compiler jumlah dan tipe parameter yang diterima dan nilai balik fungsi (bila ada) agar compiler dapat memeriksa ketepatannya. **Definisi fungsi** itu sendiri adalah menspesifikasikan tugas fungsi tersebut.

Contoh 1 - program fungsi :

```
# include <iostream.h>
Int tambah (int a, int b )
{
    Int r ;
    r = a + b ;
    return 0 ;
}
Int main ( )
{
    Int z ;
    z = tambah ( 5,3 ) ;
    cout << "Hasilnya = " << z ;
    return 0 ;
}
```

Contoh 2 – program fungsi tanpa tipe :

```
# include <iostream.h>
Void contoh (void)
{
Cout << " ini adalah FUNGSI " ;
}
Int main ( )
{
Contoh ( ) ;
Return 0 ;
}
```

3.2 PARAMETER FUNGSI

Parameter fungsi ada 2, yaitu : parameter formal dan parameter actual.

1. Parameter Formal
Parameter yang dideklarasikan dibagian blok fungsi.
2. Parameter Aktual
Parameter yang sebenarnya / parameter yang dilewatkan saat fungsi dipanggil.

Contoh program yg menunjukkan parameter formal dan actual :

```
# include <iostream.h>
Int tambah (int a, int b)           // parameter formal
{
Int r ;
r = a + b ;
return 0 ;
}
Int main ( )
{
Int x, y, z ;
Cin >> x >> y ;
z = tambah ( x , y ) ;             //parameter aktual
cout << "Hasilnya = ' << z ;
return 0 ;
}
```

3.3 PARAMETER FUNGSI PASS BY VALUE

Parameter yang diberikan ke fungsi adalah 'Nilainya', tidak pernah menspesifikasikan variabelnya. Bila ada perubahan pada parameter formal, tidak akan mempengaruhi nilai pada parameter actual.

3.4 PARAMETER FUNGSI PASS BY REFERENCE

Memanipulasi nilai dari dalam fungsi. Setiap perubahan terhadap parameter formal akan memengaruhi nilai pada parameter actual. Parameter formal diberi simbol '&'. Cara ini adalah cara efektif yang memungkinkan sebuah fungsi mengembalikan lebih dari satu nilai.

Contoh program :

```
#include <iostream.h>
Void perkalian (int& a, int& b, int& c)
{
    a *= 2 ; b *= 2 ; c *= 2 ;
}
Int main ( )
{
    Int x = 1, y = 3, z = 7 ;
    Perkalian ( x, y, z ) ;
    Cout << " X = " << x << " Y = " << y << " Z = " << z ;
    Return 0 ;
}
```

Outputnya :

X = 2 Y = 6 Z = 14

Contoh program :

```
#include <iostream.h>
Void prevnext (int x, int& prev, int& next)
{
    prev = x - 1 ;
    Next = x + 1 ;
}
Int main ( )
{
    Int x = 100 , y, z ;
    Prevnex ( x, y, z ) ;
    Cout << " Previous = " << y << " , Next = " << z ;
    Return 0 ;
}
```

Outputnya :

Previuos = 99, Next = 101

Saat pendeklarasian fungsi dapat langsung diberikan nilai default untuk setiap parameter.

Contoh program :

```
# include <iostream.h>
Int pembagian (int a, int b = 2)
{
    Int r ;
    r = a / b ;
    Return ( r ) ;
}
Int main ( )
{
    Cout << pembagian ( 12 ) ;
    Cout << endl ;
    Cour << pembagian ( 20, 4 ) ;
    Return 0 ;
}
```

3.4 POLIMORPHISM

Polimorphisme adalah objek-objek yang berbeda tetapi berasal dari satu orang tua, dapat mempunyai metode yang sama tetapi cara pelaksanaannya berbeda-beda. Contohnya adalah objek kendaraan yang terdiri dari objek kendaraan bermotor dan kendaraan tidak bermotor.