

POINTER

Salah satu kelebihan dari Bahasa C / C++ adalah Karena Bahasa ini mendukung sepenuhnya untuk pemanipulasian memori dengan menggunakan pointer.

Pointer yaitu variabel yang berisi (menyimpan) alamat memori dari sebuah variabel lain atau pointer dapat diartikan sebagai variabel yang menunjuk ke sebuah alamat memori dari sebuah variabel lain. Pointer ini digunakan dengan maksud untuk menunjukan ke suatu alamat memori sehingga kita dapat mengetahui dengan mudah alamat dari sebuah variabel. Pointer juga bisa diartikan sebagai tipe data yang nilainya mengarah pada nilai yang terdapat pada suatu alamat memori.

Dalam melakukan operasi pada Pointer, terdapat 2 macam operator pointer yang disediakan dalam bahasa C++ yaitu:

1. Operator dereference (&)
2. Operator reference (*).

Operator Dereference (&)

Operator Dereference (&) yaitu operator yang berfungsi mendeklarasikan sebuah variabel didalam penggantian memori. operator ini biasa disebut dengan "address of". Jika ingin mengetahui dimana variabel akan disimpan (untuk mengetahui letak / alamat penyimpanan variabel di memori), dapat memberikan tanda "&" (*ampersand sign*) di depan variabel, yang berarti "address of".

Penggunaan operator dereference (&) ini, sebuah variabel akan menghasilkan alamat lokasi penyimpanannya di memori. Sebagai contoh "int angka" ditempatkan pada memori dengan alamat 0x6dfec untuk mengetahuinya dapat menuliskan code seperti ini :

cout<<"\nAlamat variabel x (&x) : "<<&x; berikut code lengkapnya:

```
#include <iostream>
using namespace std;
int main()
{
    //deklarasi variabel x dengan tipe data int
    int x = 90;

    //deklarasi pointer ptrx dengan tipe data int
    int *ptrx;

    //ptrx sama dengan Address of x (0x6dfec)
    //sekarang pointer ptrx menunjuk ke alamat memori variabel x
    //atau pointer ptrx berisi (menunjuk) alamat memori variabel x
    ptrx = &x;

    cout<<"Nilai (isi) dari variabel x      = "<< x << endl;
```

```

cout<<"Alamat variabel x (&x)      = "<< &x << endl;
cout<<"Nilai yang ditunjuk pointer *x = "<< *ptrx << endl;

return 0;
}

```

Hasil Program :

```

Select D:\2020-PK2\Ptr1.exe
Nilai (isi) dari variabel x      = 90
Alamat variabel x (&x)          = 0x6dfee8
Nilai yang ditunjuk pointer *x = 90

Process returned 0 (0x0)   execution time : 0.028 s
Press any key to continue.

```

Nama Variabel	x		ptrx
Nilai Variabel	90		0x6dfeec (alamat memori yang ditunjuk)
Alamat Memory	0x6dfeec		0x6dfee8

Ilustrasi Pointer pada C++, pada program Contoh Penggunaan Operator Dereference (&)

Pada program di atas awalnya dibuat variabel dengan tipe data int dengan nama "x", disaat yang bersamaan juga dilakukan inisialisasi "int x" dengan nilai awal "90", kemudian dilakukan deklarasi sebuah pointer dengan tipe data int dan dengan nama "ptrx", berikut dituliskan code **"ptrx = &x;"** yang artinya memerintahkan pointer ptrx menunjuk ke alamat memori variabel x, sehingga nilai yang ditunjuk pointer ptrx sama dengan nilai dari variabel x yaitu 90 karena pointer ptrx telah menunjuk ke alamat memori variabel x.

Operator Reference (*)

Seandainya pada operator Dereference (&) sobat dapat mengeluarkan alamat memory dari sebuah variabel, sedangkan pada Operator Reference (*) dapat mengeluarkan (menampilkan) nilai dari alamat memori yang di tunjuk. untuk penggunaannya sendiri dapat menambahkan identifier asterisk (*) di depan variabel yang biasanya Operator reference disebut dengan "value pointed by".

Dari contoh program di atas, nilai *ptrx adalah "90", hal tersebut karena pointer ptrx menunjuk ke variabel x sehingga apabila dituliskan code cout << "Nilai yang ditunjuk pointer *ptrx : "<< ***ptrx**; maka akan tampil nilai 90 karena kita mengeluarkan nilai dari alamat memori yang ditunjuk oleh pointer ptrx (nilai/value yang berada di alamat memori 0x6dfee8).

Contoh 2:

```
#include <iostream>
using namespace std;
int main()
{
    //deklarasi variabel x dengan tipe data int
    int x;

    //deklarasi pointer p dengan tipe data int
    int *p;

    x = 20;
    p = &x;

    cout<<"\t\t==== Program Contoh Pointer =====\n\n";

    cout<<"Nilai x \t\t= "<<x<<endl;
    cout<<"Alamat x \t\t= "<<&x<<endl<<endl;

    cout<<"Nilai yang ditunjuk *p\t="<<*p<<endl;
    cout<<"Alamat yang ditunjuk *p\t="<<p<<endl;
    cout<<"Alamat *p \t\t= "<<&p<<endl;

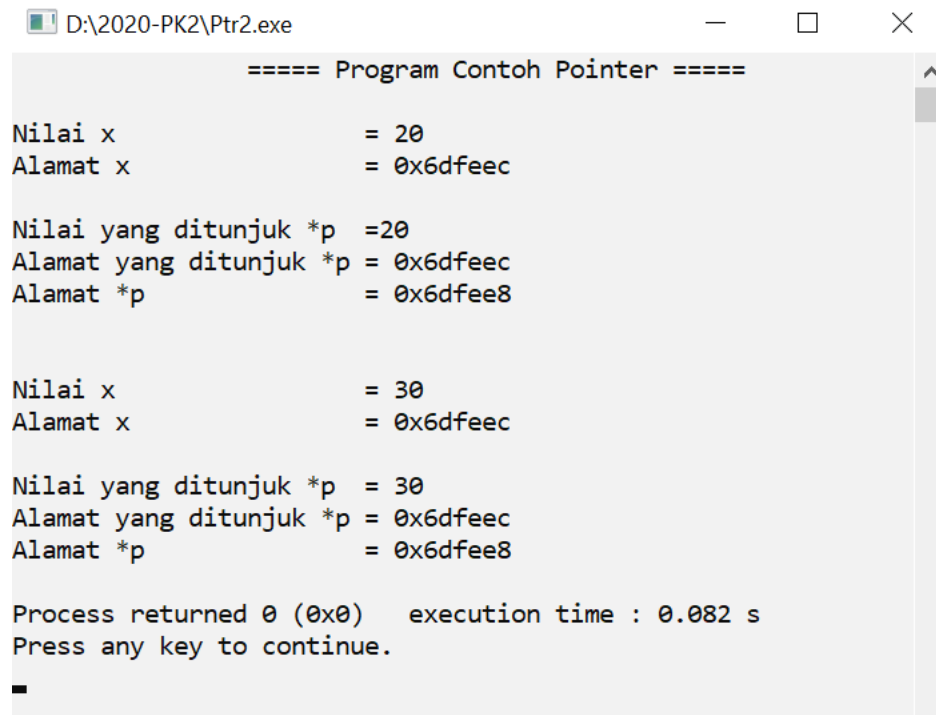
    // mengisi nilai 30 ke dalam *p / alamat yang ditunjuk pointer p
    // untuk megisikan nilai melalui alamat pointer gunakan *
    // karena alamat memori yang ditunjuk pointer p merupakan alamat memori variabel x
    // maka nilai dari variabel x juga ikut berubah
    *p = 30;

    cout<<"\n\nNilai x \t\t= "<<x<<endl;
    cout<<"Alamat x \t\t= "<<&x<<endl<<endl;

    cout<<"Nilai yang ditunjuk *p\t="<<*p<<endl;
    cout<<"Alamat yang ditunjuk *p\t="<<p<<endl;
    cout<<"Alamat *p \t\t= "<<&p<<endl;

    return 0;
}
```

Hasil Program :



```
===== Program Contoh Pointer =====

Nilai x          = 20
Alamat x         = 0x6dfeec

Nilai yang ditunjuk *p = 20
Alamat yang ditunjuk *p = 0x6dfeec
Alamat *p        = 0x6dfee8

Nilai x          = 30
Alamat x         = 0x6dfeec

Nilai yang ditunjuk *p = 30
Alamat yang ditunjuk *p = 0x6dfeec
Alamat *p        = 0x6dfee8

Process returned 0 (0x0)   execution time : 0.082 s
Press any key to continue.
■
```

Pada program di atas awalnya dibuat variabel dengan tipe data int dengan nama "x", kemudian sebuah pointer dengan tipe data int dengan nama "p". setelah itu memberikan nilai x dengan nilai "20" kemudian mengarahkan / mengisi / menunjukkan pointer p ke alamat memori variabel x. Setelah itu menampilkan data dari variabel x dan pointer p.

Kemudian code ***p = 30;** artinya mengisi nilai 30 ke dalam pointer p / alamat yang ditunjuk pointer p, karena alamat memori yang ditunjuk pointer p merupakan alamat memori variabel x maka nilai dari variabel x juga ikut berubah.