**NAMA : IMAM ALIF RAMADHAN**

**NIM : 5180711067**

Pengertian Pointer

Pointer adalah penunjuk suatu variabel. Karena menunjuk suatu variabel, maka pointer wajib memiliki alamat dari variabel yang ditunjuknya. Kadang kala dalam program yang besar, penghematan memori wajib untuk dilakukan. Dengan mekanisme *copy* dan *paste* nilai variabel satu kedalam variabel lain, akan sangat memboroskan memori. Dengan mekanisme pointer, suatu variabel dalam suatu fungsi dapat diakses oleh fungsi yang lain.

#### Deference (&)

Deference (&) merupakan suatu operator yang berfungsi untuk menanyakan alamat dari suatu variabel. Apabila kamu memberikan simbol & pada awal variabel dan mencetak hasilnya pada jendela CLI, maka yang akan tercetak adalah alamat dari variabel tersebut bukan nilai yang ditampung oleh variabel tersebut.

Berikut contoh programnya:

#include <iostream>

using namespace std;

int main()

{ int a = 5;

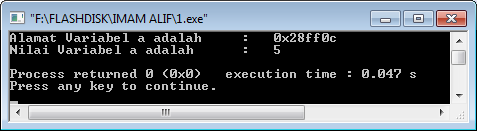
cout<<"Alamat Variabel a adalah : "<<&a<<endl;

cout<<"Nilai Variabel a adalah : "<<a<<endl;

return 0;

}

Hasil programnya



#### Reference (\*)

Reference (\*) merupakan suatu operator yang berfungsi menyatakan suatu variabel adalah variabel pointer. Sama halnya dengan operator deference, peletakan simbol operator reference diletakan diawal variabel. Operator reference ini akan membuat suatu variabel pointer untuk menampung alamat.

Berikut contoh programnya:

#include <iostream>

using namespace std;

int main()

{ int a=5;

int \*b;

b = &a ;

cout<<"Nilai variabel a adalah : "<<a<<endl;

cout<<"Alamat variabel a adalah : "<<&a<<endl;

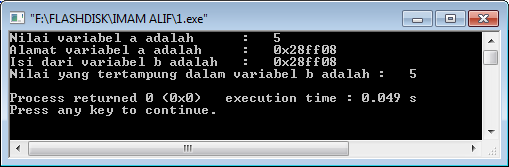
cout<<"Isi dari variabel b adalah : "<<b<<endl;

cout<<"Nilai yang tertampung dalam variabel b adalah : "<<\*b<<endl;

return 0;

}

Hasil programnya



#### Pointer dan Array

Jika kamu menuliskan sebuah array tanpa tanda kurung kotak ([]) maupun indeksnya, maka array tersebut menunjuk atau bersi alamat elem pertama dari array tersebut. Misalkan kamu membuat sebuah pointer bertipe int dengan nama pointerArray dan sebuah array dengan tipe yang sama yaitu int dengan nama nilaiArray, maka pemberian nilai berikut sah dan dapat dilakukan:

int \*pointerArray;

int nilaiArray[5];

pointerArray = nilaiArray;

Dari sintaks di atas, variabel pointer dengan nama pointrArray sekarang menunjuk pada alamat memori elemen pertama array dengan nama nilaiArray. Nah secara tidak langsung kita dapat mengakses nilai elemen array dengan menggunakan variabel pointer. Berikut Mimin contohkan sintaks pemrogramannya:

#include <iostream>

using namespace std;

int main()

{ int nilaiArray[5] = {2,5,4,6,7};

int \*pointerArray;

pointerArray = nilaiArray;

cout<<"Nilai pada elemen 0 nilaiArray adalah = "<<nilaiArray[0]<<endl;

cout<<"Nilai pada elemen 0 nilaiArray (Pointer) adalah = "<<\*pointerArray<<endl;

pointerArray++;

\*pointerArray = 10;

cout<<"Nilai pada elemen 1 nilaiArray adalah = "<<nilaiArray[1]<<endl;

cout<<"Nilai pada elemen 1 nilaiArray (Pointer) adalah = "<<\*pointerArray<<endl;

pointerArray = nilaiArray;  cout<<endl;

cout<<"Akses nilai semua array dengan pointer dan looping for"<<endl;

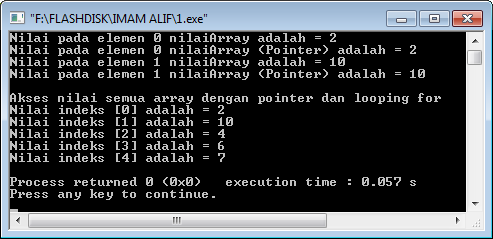
for (int i=0; i<5; i++)

{ cout<<"Nilai indeks ["<<i<<"] adalah = "<<\*(pointerArray+i)<<endl;}

return 0;

}

Hasil program



#### Pointer Sebagai Parameter Suatu Fungsi

Seperti halnya dengan array, pointer dapat digunakan sebagai parameter suatu fungsi. Karena sifat pointer yang hanya sebagai penunjuk, maka setiap perubahan yang terjadi pada parameter, sebenarnya terjadi pada variabel yang ditunjuk bukan pada variabel pointer.

Berikut contoh program :

#include <iostream>

using namespace std;

void tambah (int \*angka)

{ \*angka +=20; }

int main ()

{ int nilai = 10;

cout<<"Nilai variabel nilai adalah = "<<nilai<<endl;

tambah(&nilai);

cout<<"Nilai variabel nilai adalah = "<<nilai<<endl;

return 0;

}

Hasil program

