

MODUL 3

POINTER

DASAR TEORI

Tipe data pointer adalah tipe data yang digunakan untuk menunjukkan alamat memori dari suatu variabel yang ditunjuknya. Variabel yang dapat dirujuk oleh suatu pointer adalah variabel yang memiliki **tipe data yang sama** dengan tipe data pointer. Setiap variabel tersimpan di dalam suatu alamat memori yang berbeda-beda. Alamat memori disebut dengan *address* atau *reference*.

Misalkan:

```
int a = 10;  
int b = 10;  
string x = "pointer";
```

Alamat memori variabel a = 0x000b00

Alamat memori variabel b = 0x000c01

Alamat memori variabel x = 1xb00f01

Ketiga variabel di atas memiliki alamat memori yang berbeda-beda, sekalipun variabel a dan b memiliki nilai yang sama, namun mereka tidak memiliki hubungan di dalam alamat penyimpanan nilainya. Alamat memori bersifat random (acak) di setiap pemrosesan program.

Lagi, misalkan:

```
int a = 2;  
int b = a;
```

Alamat memori 'a' = 0x3g400f

Alamat memori 'b' = 1x00a2f1

Meskipun variabel b diinisialisasi dengan nilai variabel a, namun bukan berarti alamat penyimpanannya sama.

Pointer disimbolkan dengan tanda '*' (*asteris/asterik*) dan untuk reference disimbolkan dengan tanda '&' (*ampersand*) atau dalam hal ini biasa disebut dengan operator **address**.

Bentuk umum pointer:

```
tipe_data * nama_pointer;
```

Beberapa bentuk deklarasi pointer:

```
int *p_var;  
int* p_var;  
int * p_var;  
int*p_var;
```

Aturan: posisi tanda (*) selalu diantara tipe data dan nama variabel pointer.

Contoh:

```
int a = 100;  
int *b = &a;  
  
cout << b << endl;           //merujuk pada alamat memori 'a'  
cout << *b << endl;         //merujuk pada nilai 'a'
```

Output:

```
1x0g011f4  
100
```

Simbol pointer (*) digunakan untuk menyatakan nilai dari suatu variabel yang ditunjuk. Sedangkan simbol address (&) digunakan untuk menyatakan alamat memori suatu variabel yang ditunjuk.

GUIDED

Program Pointer 1

```
#include <iostream>

using namespace std;

int main()
{
    int v = 7;
    int *p;

    cout << "Nilai v    = " << v << endl;
    cout << "Nilai p    = " << *p << endl; //nilai akan random
    cout << "Alamat v   = " << &v << endl;
    cout << "Alamatnya = " << &p << endl;

    return 0;
}
```

Output Program:

```
Nilai v    = 7
Nilai p    = 1775249
Alamat v   = 0x61fe1c
Alamatnya = 0x61fe10
```

Program Pointer 2: Re-inisialisasi variabel dengan pointer

```
#include <iostream>

using namespace std;

int main()
{
    int value1 = 5, value2 = 15;
    int *mypointer;

    cout << "Nilai value1 = " << value1 << endl;
    cout << "Nilai value2 = " << value2 << endl;
    cout << endl;

    //Inisialisasi ulang dengan pointer
    mypointer = &value1;
    *mypointer = 10;
```

```
mypointer = &value2;
*mypointer = 20;

cout << "Nilai value1 = " << value1 << endl;
cout << "Nilai value2 = " << value2 << endl;

return 0;
}
```

Output Program:

```
Nilai value1 = 5
Nilai value2 = 15

Nilai value1 = 10
Nilai value2 = 20
```

Program Pointer 3: Array dengan pointer

```
#include <iostream>

using namespace std;

int main()
{
    int data[] = {1,2,3,4,5};
    int *pData = data;

    cout << pData[0] << endl;
    cout << pData[1] << endl;
    cout << pData[2] << endl;
    cout << pData[3] << endl;
    cout << pData[4] << endl;

    return 0;
}
```

Output Program:

```
1
2
3
4
5
```

TUGAS

Buatlah array yang menampung data bebas (minimal 10 data), kemudian buatlah variabel pointer yang menunjuk ke array tersebut.

1. Tampilkan semua nilai data/elemen dari array melalui pointer yang menunjuknya. **[Bobot 30]**
2. Tampilkan semua alamat memori dari tiap elemen array melalui pointer yang menunjuknya. **[Bobot 30]**
3. Lakukan perubahan (re-inisialisasi) beberapa elemen array (minimal 5 data yang diubah) melalui pointer yang menunjuknya. **[Bobot 40]**

~ SELAMAT MENGERJAKAN 😊 ~