

MODUL MATA KULIAH PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

Modul 2 : Pointer



Dosen Pengampu:
Herryawan Pujiharsono, S.T., M.Eng.
Slamet Indriyanto, S.T., M.T.
Prasetyo Yuliantoro, S.T., M.T.

**PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI
FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2019**

I. TUJUAN

1. Mengetahui arti dan fungsi pointer dalam pemrograman
2. Memahami penggunaan pointer dalam listing program
3. Mengetahui keunggulan pointer pada suatu program
4. Membuat program sederhana dengan menerapkan konsep pointer

II. DASAR TEORI

I. POINTER DAN PENDEFINISIAN

Pointer adalah sebuah variabel yang berisikan alamat memori (bukan nilai) atau dengan kata lain dapat dikatakan bahwa pointer adalah suatu variabel penunjuk ke alamat memori tertentu. Cara pendeklarasian pointer dalam bahasa C/C++ adalah dengan menambahkan tanda asterisk (*) didepan nama pointer, seperti contoh berikut:

```
Tipe_data *nama_pointer; atau Tipe_data* nama_pointer;
```

Keterangan:

Tipe_data : dapat berupa sembarang tipe seperti halnya pendefinisian variabel.

Nama_pointer : nama variabel pointer yang akan digunakan.

Tanda (*) : operator memori untuk mengembalikan nilai variabel pada alamatnya yang ditentukan oleh operand.

Contoh:

```
int *pint; atau int* pint;  
char *pchr; atau char* pchr;
```

II. MENGISI VARIABLE DAN MENGAKSES NILAI VARIABLE POINTER

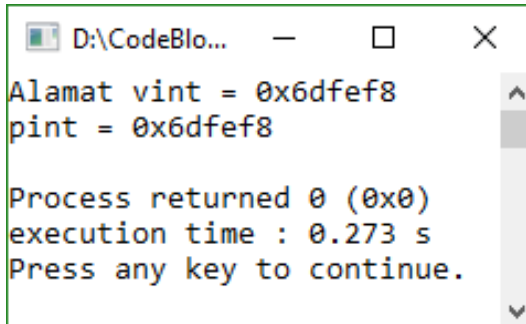
Suatu variabel pointer dapat menunjuk ke variabel lain. Contohnya adalah:

```
#include <iostream>
using namespace std;

int main()
{
    int vint = 55;        //variabel bukan pointer
    int *pint;            //variabel pointer

    pint = &vint;         // pointer menunjuk ke vint
    cout << "Alamat vint adalah = " <<&vint <<endl;
    cout << "pint = " << pint << endl;
}
```

Hasil program diatas adalah:



```
D:\CodeBlo...
Alamat vint = 0x6dfef8
pint = 0x6dfef8

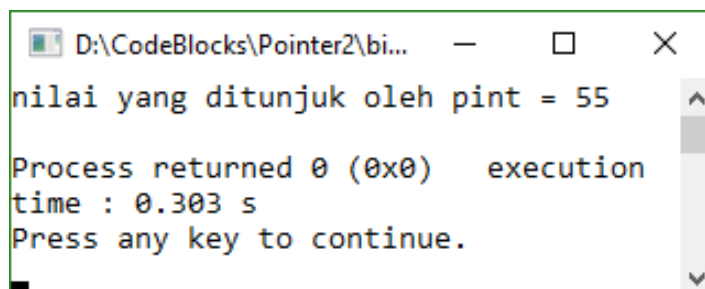
Process returned 0 (0x0)
execution time : 0.273 s
Press any key to continue.
```

Dari hasil tersebut membuktikan bahwa suatu variabel pointer dapat menunjuk alamat dari suatu variabel. Selain itu, variabel pointer juga dapat untuk mengakses isi dari suatu variabel. Berikut contohnya:

```
#include <iostream>
using namespace std;

int main()
{
    int vint = 55;        // variabel bukan pointer
    int *pint;            // variabel pointer

    pint =&vint;          // pointer menunjuk ke vint
    cout << "nilai yang ditunjuk oleh pint = " << *pint << endl;
}
```



```
D:\CodeBlocks\Pointer2\bi...
nilai yang ditunjuk oleh pint = 55

Process returned 0 (0x0)   execution
time : 0.303 s
Press any key to continue.
```

Dari kedua contoh tersebut terdapat kesimpulan bahwa operator pointer ada dua, yaitu:

- a. Operator & (Operator Diference)

Operator difference (&) yaitu mendeklarasikan suatu variabel didalam penggantian memori. Operator ini biasa disebut dengan “*address of*”.

b. Operator * (Operator Reference)

Operator Reference (*) yaitu dapat mengakses secara langsung nilai yang terdapat didalam variabel yang berpointer, hal ini dapat dilakukan dengan menambahkan identifier asterisk (*). Operator ini biasa disebut dengan “*value pointed by*”.

III. POINTER VOID

Pointer void adalah pointer yang dapat menunjuk ke sembarang tipe data. Jadi satu pointer dapat menunjuk ke tipe data integer, tipe data float, dan lain sebagainya. Berikut ini contoh dari pointer void:

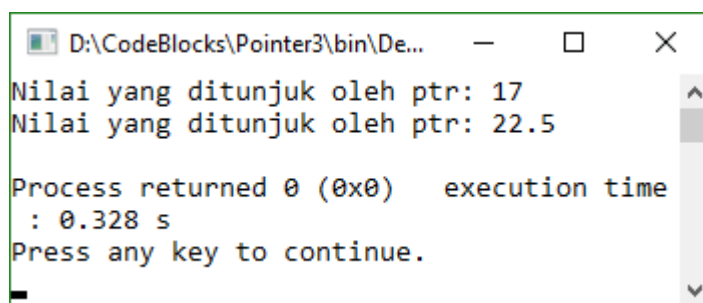
```
#include <iostream>
using namespace std;

int main()
{
    void *ptr;                // pointer tak bertipe

    int vint = 17;
    float vfl = 22.5;

    ptr = &vint;              // menunjuk ke int
    cout << "Nilai yang ditunjuk oleh ptr: " << *(int *)ptr << endl;

    ptr = &vfl;               // menunjuk ke float
    cout << "Nilai yang ditunjuk oleh ptr: " << *(float *)ptr << endl;
}
```



IV. POINTER DAN ARRAY

Pointer dan Array merupakan hal yang hampir sama, karena sama-sama menunjuk pada suatu alamat. Karena hal tersebut pada pointer yang menunjuk ke array tidak menggunakan tambahan operator & di depan nama variabel array.

```
#include <iostream>
using namespace std;

int main()
{
    int angka[] = {2,4};           // array angka
    int *pang;                     // pointer angka

    pang = angka;                  // pointer menunjuk ke array

    for (int i = 0; i<2; i++)
        cout << *(pang + i) << endl;
}
```

```
D:\CodeBlocks\Pointer4\bin\Debug...
2
4
Process returned 0 (0x0) execution time :
0.066 s
Press any key to continue.
```

V. POINTER DAN STRING

Pointer juga dapat menunjuk ke sebuah String. Contoh hubungan antara pointer dengan String adalah:

```
#include <iostream>
using namespace std;

int main()
{
    char *pstring = "ITTelkom Purwokerto";
    cout << pstring << endl;
}
```

```
D:\CodeBlocks\Pointer5\bin\Debug...
ITTelkom Purwokerto
Process returned 0 (0x0) execution time :
0.046 s
Press any key to continue.
```

VI. POINTER MENUNJUK POINTER

Pointer dapat menunjuk ke pointer lain. Hal tersebut ditandai dengan adanya dua tanda asterisk (**) sebelum nama variabel pointer. Berikut contohnya:

```

#include <iostream>
using namespace std;

int main()
{
    float a = 0.7;
    float *ptrsatu, **ptrdua;

    ptrsatu = &a;
    ptrdua = &ptrsatu;

    cout << "Nilai ptrsatu= " << *ptrsatu << endl;
    cout << "Nilai ptrdua= " << **ptrdua << endl;
}

```

```

D:\CodeBlocks\Pointer6\bin\Deb...
Nilai ptrsatu= 0.7
Nilai ptrdua= 0.7

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

```

VII. KEUNTUNGAN MENGGUNAKAN POINTER

1. Untuk menciptakan data struktur yang kompleks
2. Memungkinkan suatu fungsi untuk menghasilkan lebih dari satu nilai
3. Memiliki kemampuan untuk mengirimkan alamat suatu fungsi ke fungsi yang lain
4. Penanganan terhadap array dan string akan lebih singkat dan efisien
5. Memungkinkan untuk berhubungan langsung ke hardware
6. Memungkinkan untuk berhubungan langsung dengan parameter command line
7. Kemampuan bekerja dengan memori yang telah dialokasikan secara dinamik

III. PERCOBAAN

1. Latihan 1

```

#include <iostream>
using namespace std;

int main()
{
    int *pBil;

    int bilX=17;

    pBil = &bilX;

    cout << "isi bilX   = " << bilX << endl;
    cout << "Nilai yang ditunjuk pBil = " << *pBil << endl;

    *pBil = *pBil + 3;

    cout << "isi bil X sekarang   = " << bilX << endl;
    cout << "Nilai yang ditunjuk pBil = " << *pBil << endl;
    return 0;
}

```

2. Latihan 2

```

#include <iostream>
using namespace std;

int main()
{
    int x, y;
    int *px;
    x = 17;
    y=x;
    px = &x;

    cout << "Nilai x = " << x << endl;
    cout << "Nilai y = " << y << endl;
    cout << "Alamat px = " << px << endl;
    cout << "Nilai px = " << *px << endl;
}

```

3. Latihan 3

```

#include <iostream>
using namespace std;

int main()
{
    int x;
    int *px;           // pointer ke variabel
    int **ppx;         // pointer ke pointer

    x = 1717;
    px = &x;
    ppx = &px;

    cout << "Nilai x = " << x << endl;
    cout << "Nilai px = " << *px << endl;
    cout << "Nilai ppx = " << **ppx << endl;

}

```

4. Latihan 4

```

#include <iostream>
using namespace std;

int main()
{
    long *p, x;

    p = &x;
    x = 17;

    cout << "Nilai X    = " << x << endl;
    cout << "Nilai *p    = " << *p << endl;
    cout << "Nilai p      = " << p << endl;
    cout << "Nilai &x    = " << &x << endl << endl;

    *p = 20;

    cout << "Nilai X    = " << x << endl;
    cout << "Nilai *p    = " << *p << endl;
    cout << "Nilai p      = " << p << endl;
    cout << "Nilai &x    = " << &x << endl << endl;
    return 0;
}

```

5. Latihan 5


```

#include <iostream>
#define MAX 5
using namespace std;

int main()
{
    int a[MAX];
    int *pa; pa = a;          // atau pa = &a[0]

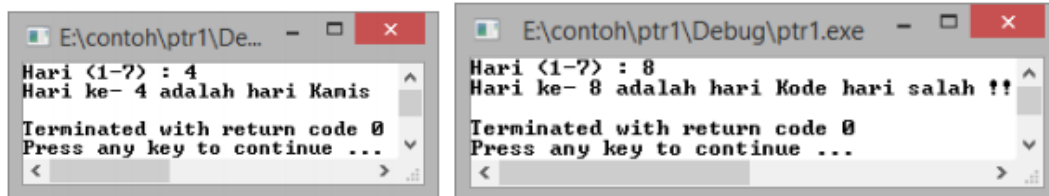
    for (int i = 0; i<MAX; i++)
    {
        cout << "Masukkan Nilai " << i+1 << " : ";
        cin >> a[i];
    }
    cout << endl;

    for (int i = 0; i<MAX; i++)
    {
        cout << "Nilai a[" << i << "] = " << *pa << endl;
        pa++;
    }
}

```

IV. TUGAS

1. Buatlah program yang menampilkan nama hari sesuai dengan angka yang dimasukkan menggunakan fasilitas pointer, seperti tampilan berikut:



2. Buatlah program untuk menjawab pertanyaan dibawah ini dengan menggunakan fungsi pointer.

