

PERTEMUAN 15

POINTER

POINTER A. Tujuan

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu :

- Mengetahui arti dan fungsi pointer dalam pemrograman C++.
- Memahami penggunaan pointer dalam listing program.
- Mengetahui keunggulan pointer pada suatu program.
- Membuat program sederhana dengan menerapkan konsep pointer.

B. Dasar Teori

1. 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 adalah dengan menambahkan tanda asterisk (*) didepan nama pointer, seperti berikut:

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

Keterangan:

Tipe_data dapat berupa sembarang tipe seperti halnya pendefinisian variabel.

Nama_pointer adalah nama variabel pointer yang akan digunakan. *

adalah operator memori untuk mengembalikan nilai variabel pada alamatnya yang ditentukan oleh operand.

Contoh:

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

2. Mengisi Variabel dan Mengakses Nilai Variabel Pointer

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

```

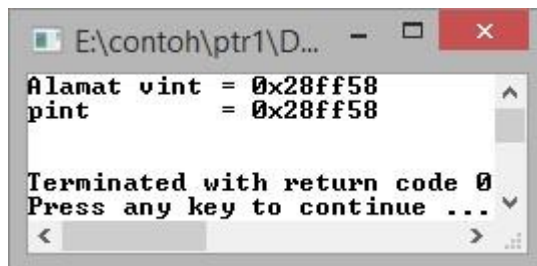
#include <iostream.h>
#include <conio.h>

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

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

```

Hasil dari program di atas adalah :



```

E:\contoh\ptr1\D...
Alamat vint = 0x28ff58
pint = 0x28ff58

Terminated with return code 0
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:

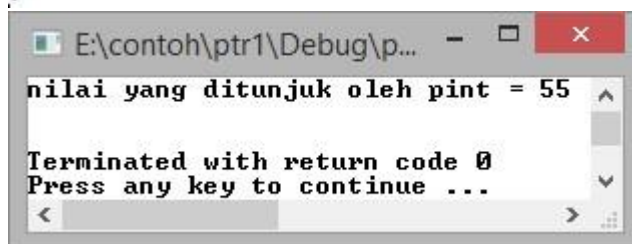
```

#include <iostream.h>
#include <conio.h>

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;
}

```



```

E:\contoh\ptr1\Debug\p...
nilai yang ditunjuk oleh pint = 55

Terminated with return code 0
Press any key to continue ...

```

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

- a. Operator & (Operator Diference)

Operator deference (&) 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”**.

3. 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.h>
#include <conio.h>

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

    int vint = 50;
    float vfl = 51.5;

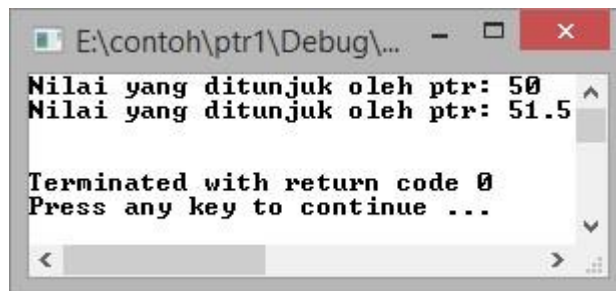
    ptr = &vint; // menunjuk ke int

    cout << "Nilai yang ditunjuk oleh ptr: "
         << *(int *)ptr << endl;

    ptr = &vfl; // menunjuk ke float

    cout << "Nilai yang ditunjuk oleh ptr: "
         << *(float *)ptr << endl;
}

```



4. Pointer dan Array

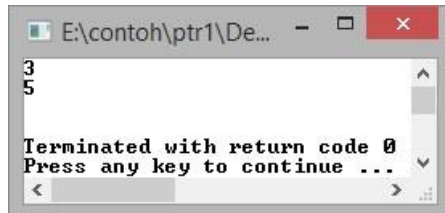
Pointer dan array merupakan hal yang hampir sama, karena sama-sama menunjuk pada suatu alamat. Karena hal tersebut pada pointer yang menunjuk ke array tidak menggunakan tambahan operator & di depan nama variabel array.

```
#include <iostream.h>
#include <conio.h>

int main()
{
    int angka[] = {3,5}; // array angka
    int *pang; // pointer pang

    pang = angka; // pointer menunjuk ke array

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

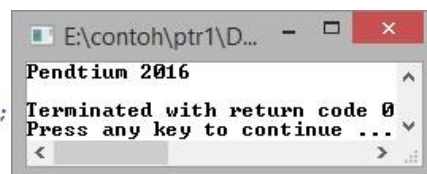


5. Pointer dan String

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

```
#include <iostream.h>
#include <conio.h>

int main()
{
    char *pstring = "Pendtium 2016";
    cout << pstring;
}
```



6. Pointer menunjuk Pointer

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

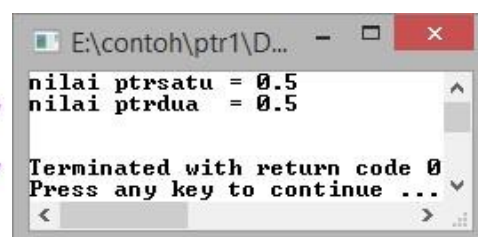
Berikut contohnya:

```
#include <iostream.h>
#include <conio.h>

int main()
{
    float a = 0.5;
    float *ptrsatu, **ptrdua;

    ptrsatu = &a;
    ptrdua = &ptrsatu;

    cout << "nilai ptrsatu = "
    << *ptrsatu << endl;
    cout << "nilai ptrdua = "
    << **ptrdua << endl;
}
```



7. Pointer dan Fungsi

Pointer juga dapat digunakan sebagai argumen maupun keluaran bagi fungsi. Pointer sebagai argumen maka nilai argumen dapat diubah di dalam fungsi. Suatu fungsi dapat dibuat agar keluarannya berupa pointer. Dalam hal tersebut terkadang difungsikan juga array di dalam fungsi tersebut.

8. Keuntungan menggunakan pointer

- a. Untuk menciptakan data struktur yang kompleks.
- b. Memungkinkan suatu fungsi untuk menghasilkan lebih dari satu nilai.
- c. Memiliki kemampuan untuk mengirimkan alamat suatu fungsi ke fungsi yang lain.
- d. Penanganan terhadap array dan string akan lebih singkat dan efisien.
- e. Memungkinkan untuk berhubungan langsung ke hardware.
- f. Memungkinkan untuk berhubungan langsung dengan parameter command line.
- g. Kemampuan bekerja dengan memori yang telah dialokasikan secara dinamik.

C. Latihan Praktikum 1. Latihan 1

```
// header file yang melibatkan cout
#include <iostream.h>
// deklarasi fungsi utama
int main ()
{
    // deklarasi pointer *pBil
    int *pBil;
    // deklarasi variabel bilX=77
    int bilX=77;
    // pBil menunjuk nilai pada bilX
    pBil=&bilX;
    // output nilai awal
    cout << "Isi bilX      = " << bilX << endl;
    cout << "Nilai yang ditunjuk pBil = " << *pBil << endl;
    // pointer *pBil menunjuk *pBil+3
    *pBil = *pBil + 3;
    // output nilai sekarang
    cout << "Isi bilX sekarang = " << bilX << endl;
    cout << "Nilai yang ditunjuk pBil = " << *pBil << endl;
    return 0;
}
```

2. Latihan 2

```
#include <iostream_h>

int main ()
{
    long *p, x;

    p = &x;
    x = 40;
    |
    cout<<"nilai X  = "<<x<<endl;
    cout<<"nilai *P = "<<*p<<endl;
    cout<<"nilai P  = "<<p<<endl;
    cout<<"nilai &X = "<<&x<<endl<<endl;

    *p = 25;

    cout<<"nilai X  = "<<x<<endl;
    cout<<"nilai *P = "<<p<<endl;
    coin<<"nilai P  = "<<*p<<endl;
    cout<<"nilai &X = "<<&x<<endl<<endl;

    return 0;
}
```

3. Latihan 3

```
#include <iostream.h>

int main()
{
    int a = 12;

    cout << "isi nilai semula = " << a << endl;

    ubah(&b);

    cout << "isi nilai sekarang = " << a << endl;
}

void ubah(int *b)
{
    *b = *b+22;
}
```

4. Latihan 4

```
#include <iostream.h>
#include <conio.h>

int main()
{
    void tunjuk;

    double n_double = 25.5;
    char n_char[1] = "D";

    tunjuk = &n_double;
    cout << "Nilai yang ditunjuk adalah : "
    << *(double *)tunjuk << endl;

    tunjuk = n_char;
    cout << "Nilai yang ditunjuk adalah : "
    << *(char *)tunjuk << endl;
}
```

5. Latihan 5

```
#include <iostream.h>
#include <conio.h>

void main()
{
    void *p_void;

    float var_f = 3.9;
    p_void = &var_f;

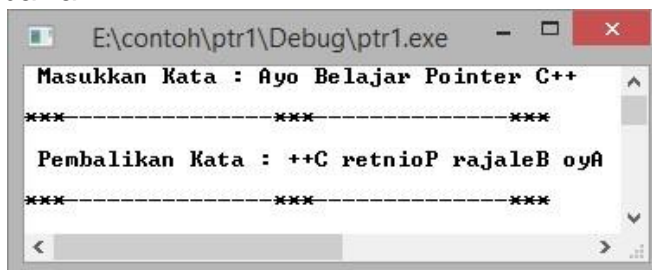
    cout << "Nilai variabel = "
    << var_f << endl;

    *(float *)p_void = 7.8;

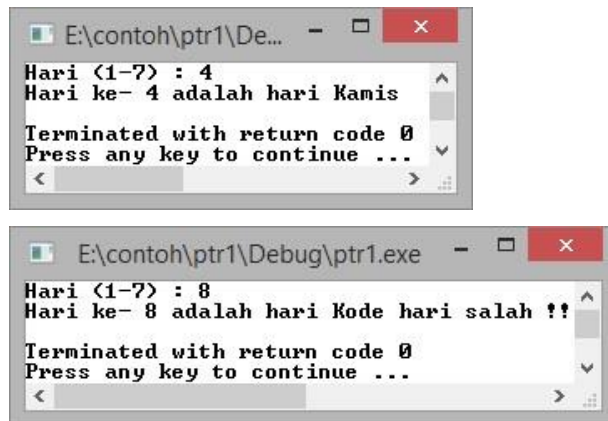
    cout << "Nilai ubah variabel = "
    << *var_f;
}
```

D. Tugas Praktikum

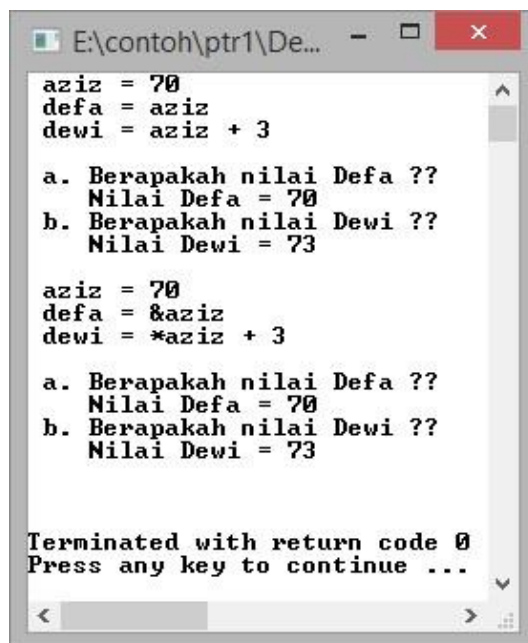
1. Buatlah program untuk membalik kata dalam suatu kalimat seperti di bawah ini:



2. Buatlah function yang akan menampilkan nama hari sesuai dengan angka yang dimasukkan menggunakan fasilitas pointer, seperti tampilan berikut:



3. Buatlah program untuk menjawab pertanyaan di bawah ini dengan menggunakan fungsi pointer.



E. Tugas Rumah

1. Buatlah sebuah function untuk membuat setiap huruf pertama dari string yang diberikan menjadi huruf besar dengan fasilitas pointer.
2. Buatlah program pointer yang melewati 4 variabel, yaitu variabel w, x, y, dan z ke fungsi yang tidak memiliki nilai balikan. Lalu buat keluaran nilai variabel setelahnya menjadi 3 kali lipat dari nilai sebelumnya.

3. Gunakan fasilitas pointer untuk membuat fungsi yang dapat digunakan untuk menghitung jumlah kata dari masukan string oleh user.
4. Gunakan fasilitas array pointer untuk membuat tampilan nama-nama bulan secara urut dari bulan Januari sampai dengan Desember.