

# POINTER

Setiap byte di dalam memori komputer memiliki sebuah alamat. Alamat memori dimulai dari 0. Pada komputer yang memiliki memori 640Kb, alamat memori tertinggi yaitu 655.359. Didalam memori inilah variabel disimpan. Tetapi tentu saja pemogram tidak perlu menyebutkan alamat dari suatu variabel secara eksplisit. Pada saat program dimuat di dalam memori, variabel akan diletakkan dengan sendirinya pada alamat tertentu.

Pointer banyak dilibatkan dalam program C++, misalnya untuk melewati string dari suatu fungsi ke fungsi yang lain. Penerapan pointer yang paling umum yaitu untuk menciptakan variabel dinamis yang memungkinkan untuk memakai memori bebas (memori yang belum dipakai) selama eksekusi program.

Variabel pointer sering dikatakan sebagai variabel yang menunjuk ke obyek lain. Pada kenyataan yang sebenarnya, variabel pointer berisi alamat dari suatu objek lain yaitu objek yang dikatakan ditunjuk oleh pointer. Sebagai contoh, px adalah pointer dan x adalah variabel yang ditunjuk oleh px. Jika px berada pada alamat memori (alamat awal) 1000, maka px akan berisi 1000.

Agar suatu pointer menunjuk ke variabel lain, mula-mula pointer harus diisi dengan alamat dari variabel yang akan ditunjuk. Untuk menyatakan alamat dari suatu variabel, operator **&** (operator alamat, yang bersifat unary) bisa digunakan dengan cara menempatkan operator di depan nama variabel.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int alpha=11;
    int beta=22;
    int charlie=33;
    clrscr();
    cout <<"Isi variabel : " <<endl;
    cout <<"alpha  = " <<alpha <<endl;
    cout <<"beta   = " <<beta <<endl;
    cout <<"charlie = " <<charlie <<endl;
    cout <<"Alamat variabel : " <<endl;
    cout <<"alpha  = " <<&alpha <<endl;
    cout <<"beta   = " <<&beta <<endl;
    cout <<"charlie = " <<&charlie <<endl;
}
```

Hasil :

```
Isi variabel :
alpha   = 11
beta    = 22
charlie = 33
Alamat variabel :
alpha   = 0x23d7244c
beta    = 0x23d7244a
charlie = 0x23d72448
```

Suatu variabel pointer dideklarasikan dengan bentuk :

```
tipe_data *nama_variabel;
```

- tipe\_data berupa sembarang tipe
- nama\_variabel adalah nama dari variabel pointer.

Jika suatu variabel sudah ditunjuk pointer, variabel tersebut dapat diakses melalui variabel itu sendiri (dikatakan sebagai pengaksesan langsung) ataupun melalui pointer (dikatakan sebagai pengaksesan tidak langsung). Pengaksesan tak langsung dilakukan dengan menggunakan operator indirection berupa simbol \* (bersifat unary).

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
  int x, y;
  int *px; // px adalah pointer yang menunjuk objek bertipe int
  clrscr();

  x=44;
  px=&x;
  y=*px;
  cout <<"Nilai x = " <<x <<endl;
  cout <<"Nilai &x = " <<&x <<endl;
  cout <<"Nilai px=&x adalah " <<px <<endl;
  cout <<"Isi *px = " <<*px <<endl;
  cout <<"Nilai y=*px adalah " <<y;
}
```

Hasil :

```
Nilai x = 44
Nilai &x = 0x24072434
Nilai px=&x adalah 0x24072434
Isi *px = 44
Nilai y=*px adalah 44
```

Antara tipe pointer dan tipe objek yang akan ditunjuk oleh pointer haruslah sejenis.

Contoh :

```
/* Pemakaian pointer yang salah */
#include <iostream.h>
#include <conio.h>
void main()
{
    float *pu;
    float nu;
    int u = 1234;
    clrscr();
    pu=&u; //pernyataan ini salah karena tipe pu dan u berbeda
    nu = *pu;
    cout <<"Nilai u = " <<u <<endl;
    cout <<"Nilai nu = " <<nu;
}

```

Program di bawah ini memberikan gambaran tentang pengubahan isi suatu variabel secara tak langsung.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float d, *pd;
    clrscr();
    d=54.5;
    cout <<"Isi d semula = " <<d <<endl;
    pd=&d;
    *pd = *pd + 10; // identik dengan d = d + 10
    cout <<"Isi d kini = " <<d;
}

```

Hasil :

```
Isi d semula = 54.5
Isi d kini = 64.5

```

### POINTER dan ARRAY

Hubungan antara pointer dan array pada C++ sangatlah erat, sebab sesungguhnya array secara internal akan diterjemahkan dalam bentuk pointer.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
    static int tgl_lahir[]={24, 12, 1970};
    int i, *ptgl;
    clrscr();
}

```

```

ptgl = tgl_lahir; // ptgl berisi alamat array
for (i=0; i<3; i++)
    cout <<"Elemen subscript ke-" <<i <<" adalah " <<*(ptgl+i) <<endl;
}

```

Hasil :

```

Elemen subscript ke-0 adalah 24
Elemen subscript ke-1 adalah 12
Elemen subscript ke-2 adalah 1970

```

## POINTER dan STRING

Program di bawah ini menggambarkan pertukaran dua string yang dilakukan melalui pertukaran isi array melalui pointer. Perhatikan bahwa dengan pointer, penyalinan isi array tidak perlu dilakukan. Sebagai efeknya bisa menghemat waktu eksekusi.

Contoh :

```

#include <iostream.h>
#include <conio.h>
void main()
{
    char *nama1 = "CAT WOMAN";
    char *nama2 = "BATMAN";
    char *namax;
    clrscr();

    cout <<"Kata Semula :" <<endl;
    cout <<"Nama-1 = " <<nama1 <<endl;
    cout <<"Nama-2 = " <<nama2 <<endl;

    namax=nama1; /* penukaran string dengan pointer */
    nama1=nama2;
    nama2=namax;

    cout <<"Setelah ditukar :" <<endl;
    cout <<"Nama-1 = " <<nama1 <<endl;
    cout <<"Nama-2 = " <<nama2;
}

```

Hasil :

```

Kata Semula :
Nama-1 = CAT WOMAN
Nama-2 = BATMAN
Setelah ditukar :
Nama-1 = BATMAN
Nama-2 = CAT WOMAN

```

## POINTER MENUNJUK POINTER

Suatu pointer bisa saja menunjuk ke pointer lain. Definisi untuk membentuk rantai pointer adalah sebagai berikut :

```
int x;
int *px1;
int **px2;
```

dimana:

- x adalah variabel bertipe int
- px1 adalah variabel pointer yang menunjuk ke x
- px2 adalah variabel pointer yang menunjuk ke pointer px1 dengan menggunakan dua tanda \*

Agar px1 menunjuk ke variabel x digunakan perintah : `px1 = &x;`

Agar px2 menunjuk ke px1 digunakan perintah : `px2=&px1;`

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int x = 44;
    int *px1;
    int **px2;
    px1=&x;
    px2=&px1;
    cout <<"Nilai x=44" <<endl;
    cout <<"Nilai &x = " <<&x <<endl;
    cout <<"Nilai &px1 = " <<&px1 <<endl;
    cout <<"Nilai *px1 = " <<*px1 <<endl;
    cout <<"Nilai **px2 adalah : " <<**px2 <<endl;
}
```

Hasil :

```
Nilai x=44
Nilai &x = 0x1bd72430
Nilai &px1 = 0x1bd7242c
Nilai *px1 = 44
Nilai **px2 adalah : 44
```

## POINTER dan FUNGSI

Pointer dan kaitannya dengan fungsi meliputi :

- Pointer sebagai parameter fungsi  
Penerapan pointer sebagai parameter yaitu jika diinginkan agar nilai suatu variabel internal dapat diubah oleh fungsi yang dipanggil.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void rubah(int *x, int *y);
void main()
```

```

{
int a=3; int b=7;
clrscr();
cout <<"Nilai semula : " <<endl;
cout <<"a = " <<a <<endl;
cout <<"b = " <<b <<endl;
rubah(&a,&b);
cout <<"Nilai setelah fungsi : " <<endl;
cout <<"a = " <<a <<endl;
cout <<"b = " <<b <<endl;
}

```

```

void rubah(int *x, int *y)
{
*x = *x + 2;
*y = *y + 2;
}

```

Hasil :

```

Nilai semula :
a = 3
b = 7
Nilai setelah fungsi :
a = 5
b = 9

```

- Pointer sebagai keluaran fungsi  
Suatu fungsi dapat dibuat agar keluarannya berupa pointer. Misalnya suatu fungsi menghasilkan keluaran berupa pointer yang menunjuk ke string.

Contoh :

```

#include <iostream.h>
#include <conio.h>
char *nama_bulan(int n);
void main()
{
int bl;
char lagi;

do
{
clrscr();
cout <<"Masukkan Kode Bulan [1..12] : ";
cin >> bl;
cout <<"Kode bulan : " <<bl <<" , nama bulan = " <<nama_bulan(bl) <<endl;
cout <<"Input lagi [y/t] : ";
lagi=getch();
}
while (lagi == 'y' || lagi == 'Y');
}

```

```

char *nama_bulan(int n)
{
    static char *bulan[] =
    {
        "Kode bulan salah", "Januari", "Februari", "Maret", "April",
        "Mei", "Juni", "Juli", "Agustus", "September", "Oktober",
        "November", "Desember"
    };
    return ((n<1 || n>12) ? bulan[0] : bulan[n]);
}

```

Hasil :

```

Masukkan Kode Bulan [1..12] : 4
Kode bulan : 4, nama bulan = April
Input lagi [y/t] :

```

### LATIHAN

1. Buat program dengan menggunakan pointer sehingga menghasilkan keluaran sebagai berikut :

```

D
ND
AND
LAND
RLAND
ORLAND
BORLAND

```