



Dasar Teori

Pointer adalah sebuah variabel yang isi datanya adalah alamat memori atau variabel lain. Sehingga pointer dapat juga disebut sebagai variabel alamat (address variable). Dengan kata lain, pointer berisi alamat dari variabel yang mempunyai nilai tertentu.

Deklarasi Pointer

Bentuk deklarasi dari pointer adalah sebagai berikut :

```
tipe_data *nama_variabel;
```

Keterangan dari deklarasi diatas adalah sebagai berikut :

tipe_data : tipe data dasar dari pointer
nama_variabel : Nama variabel pointer
*** : operator memori yang fungsinya untuk mengembalikan nilai variabel pada alamat

Agar suatu pointer menunjuk ke variabel lain, mula-mula pointer harus diisi dengan alamat dari variabel yang ditunjuk. Untuk menyatakan alamat dari suatu variabel, digunakan operator & didepan nama variabel.

*jika *x adalah variabel bertipe int, maka pointer px yang berisi alamat dari variabel x ditulis sebagai :*

```
int *x;  
px = &x
```

Untuk mengakses isi dari suatu variabel melalui pointer, perlu dinyatakan dulu penunjukkan alamat dari variabel tersebut. Dua pernyataan dibawah ini menunjukkan bahwa nilai y adalah sama dengan nilai x.

$px = \&x$	→	<i>pointer px berisi alamat dari variabel x</i>
$y = *px$	→	<i>variabel y berisi nilai yang ditunjuk oleh pointer px</i>

Antara tipe pointer dan tipe obyek yang akan ditunjuk oleh pointer haruslah sejenis. Jika tidak sejenis akan menimbulkan kesalahan. Contoh :

$int x;$	→	<i>x adalah obyek</i>
$int *px;$	→	<i>px adalah pointer yang menunjuk obyek x</i>

Contoh Kasus :

Anda diminta untuk membuat program yang akan menampilkan suatu nilai dalam suatu variabel tertentu dan menampilkan juga alamat suatu pointer yang isinya sama dengan yang ditunjuk oleh variabel tadi.

Penyelesaian :

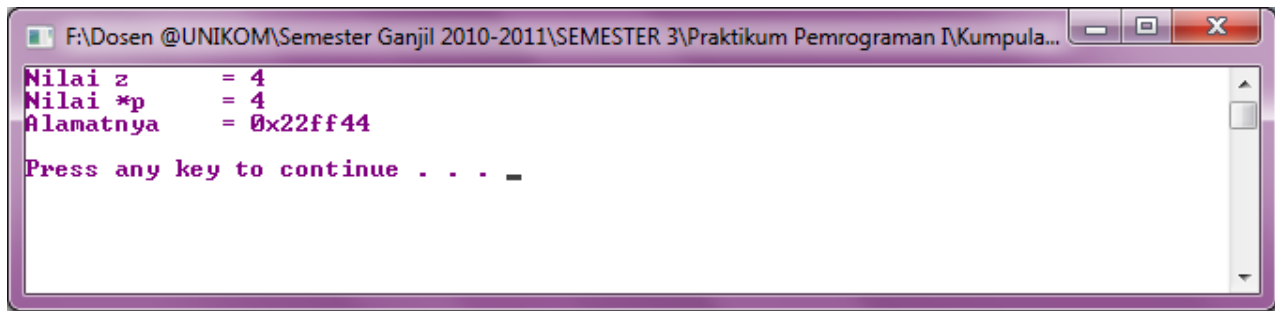
Algoritma yang digunakan untuk memecahkan masalah diatas adalah sebagai berikut :

1. Tentukan variabel yang menampung suatu nilai tertentu, misalkan nilai ini bertipe integer dan nama variabel nya adalah z. (*misalkan int z*)
2. Tentukan variabel pointer yang nantinya isinya akan menunjuk pada variabel pada langkah no.1. (*misalkan variabel *p*)
3. Tentukan variabel lain yang isinya menunjuk ke variabel pertama dalam langkah no.1. (*misalkan variabel p dan prosesnya adalah p = &z*)
4. Masukkan suatu nilai yang disimpan pada variabel z
5. Tampilkan nilai tersebut
6. Tampilkan memori yang tepat untuk menyimpan variabel pointer yang sesuai dengan langkah 2

Implementasi dari kasus diatas adalah :

```
1  /*
2  Program 11-1
3  Nama File : Program 11-1.c
4  Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int z=4,*p;
13     p=&z; // penugasan
14     printf("Nilai z = %d \n",z);
15     printf("Nilai *p = %d \n", *p);
16     printf("Alamatnya = %p \n", p);
17     system("PAUSE");
18     return 0;
19 }
```

```
1  /*
2  Program 11-1
3  Nama File : Program 11-1.cpp
4  Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int z=4;
15     int *p;
16     p = &z;
17     cout<<"Nilai z    = "<<z<<endl;
18     cout<<"Nilai *p  = "<<*p<<endl;
19     cout<<"Alamatnya = "<<p<<endl<<endl;
20     system("PAUSE");
21     return EXIT_SUCCESS;
22 }
```



```
F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai z      = 4
Nilai *p     = 4
Alamatnya   = 0x22ff44
Press any key to continue . . . _
```

Gambar 11.1 Tampilan Program 11-1

Pointer dan Array

Sebagai contoh didefinisikan sebuah array untuk menyimpan data tanggal lahir sebagai berikut :

```
int tgl_lahir [3] = {16, 10, 1993};
int *ptgl_lahir ;// pointer yang akan menunjuk array tsb
```

Maka pointer yang ditugaskan terhadap variabel array tersebut adalah sebagai berikut :

```
ptgl_lahir = tgl_lahir;
atau
ptgl_lahir = &ptgl_lahir [indeks];
ptgl_lahir = &tgl_lahir [0];
```

Penjelasannya adalah : Pointer `ptgl_lahir` berisikan alamat dari array `tgl_lahir` yang ke 0 (alamat pertama dari array `tgl_lahir`). Jika ingin menampilkan seluruh elemen array `tgl_lahir`, instruksi yang biasa dilakukan adalah berupa :

```
for (i=0; i<3;i++)
printf ("%d", tgl_lahir[i]);
```

Instruksi diatas, jika menggunakan pointer dapat dirubah menjadi :

```
ptgl_lahir = tgl_lahir;
for (i=0; i<3;i++)
printf ("%d", *(ptgl_lahir+i));
```

Untuk lebih jelasnya, dapat dilihat contoh dibawah ini :

```
1  /*
2   Program 11-2
3   Nama File : Program 10-2.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int tgl_lahir[]={31,12,1999};
13     int *ptgl_lahir, i;
14     ptgl_lahir=tgl_lahir;
15     for (i=0; i<3; i++)
16     {
17         printf("Nilai Ke   - %d   = %d\n", i+1, *ptgl_lahir);
18         printf("Alamat Ke  - %d   = %p\n", i+1, ptgl_lahir);
19         ptgl_lahir++;
20         // bisa juga seperti ini
21         // printf("Nilai Ke   - %d   = %d\n", i+1, *(ptgl_lahir+i));
22         // printf("Nilai Ke   - %d   = %d\n", i+1, (ptgl_lahir+i));
23         printf("\n");
24     }
25     system("PAUSE");
26     return 0;
27 }
```

```
1  /*
2   Program 11-2
3   Nama File : Program 11-2.cpp
4   Programmer : Eko Budi Setiawan
5  */
6  #include <cstdlib>
7  #include <iostream>
8
9  using namespace std;
10
11 int main(int argc, char *argv[])
12 {
13     int tgl_lahir[]={31,12,1999};
14     int *ptgl_lahir, i;
15     ptgl_lahir=tgl_lahir;
16     for (i=0; i<3; i++)
17     {
18         cout<<"Nilai Ke   - "<<i+1<<" = "<<*ptgl_lahir<<endl;
19         cout<<"Alamat Ke  - "<<i+1<<" = "<<ptgl_lahir<<endl;
```

```

20     ptgl_lahir++;
21     cout<<"\n";
22 }
23     system("PAUSE");
24     return EXIT_SUCCESS;
25 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Nilai Ke - 1 = 31
Alamat Ke - 1 = 0x22ff30
Nilai Ke - 2 = 12
Alamat Ke - 2 = 0x22ff34
Nilai Ke - 3 = 1999
Alamat Ke - 3 = 0x22ff38
Press any key to continue . . . _

```

Gambar 11.2 Tampilan Program 11-2

Pointer dan String

Sebagai contoh kasus penggunaan pointer dan string, perhatikan masalah berikut ini. Misalkan kita diminta untuk menampilkan teks dengan pointer. Maka algoritma untuk menjawab permasalahan diatas adalah sebagai berikut :

1. Tentukan variabel pointer yang menampung teks dengan tipenya adalah char (misalkan *teks)
2. Tentukan kalimat yang akan ditunjuk oleh pointer
3. Tampilkan variabel pointer teks, yang isinya sudah menunjuk ke teks
4. Tampilkan alamat teks tersebut
5. Selesai

```

1  /*
2  Program 11-3
3  Nama File : Program 11-3.c
4  Programmer : Eko Budi Setiawan
5  */
6  #include <stdio.h>
7  #include <stdlib.h>
8
9  int main(int argc, char *argv[])
10 {
11     char *teks;

```

```

12     teks="Pointer";
13     printf("Isi teks = %s\n", teks);
14     printf("Alamat teks = %p\n", &teks);
15     printf("\n");
16     system("PAUSE");
17     return 0;
18 }

```

```

1  /*
2   Program 11-3
3   Nama File : Program 11-3.cpp
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     char *teks;
15     teks="Pointer";
16     cout<<"Isi teks = "<<teks<<endl;
17     cout<<"Alamat teks = "<<&teks<<endl;//%p\n", &teks);
18     cout<<"\n";
19     system("PAUSE");
20     return EXIT_SUCCESS;
21 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman I\Kumpula...
Isi teks = Pointer
Alamat teks = 0x22ff44
Press any key to continue . . . -

```

Gambar 11.3 Tampilan Program 11-3

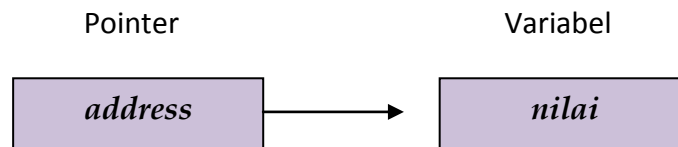
Pointer dalam Pointer

C dan C++ juga memperbolehkan penggunaan pointer didalam pointer yang masing-masing berisi data yang sama atau berbeda. Bentuk umum dari pointer dalam pointer adalah sebagai berikut :

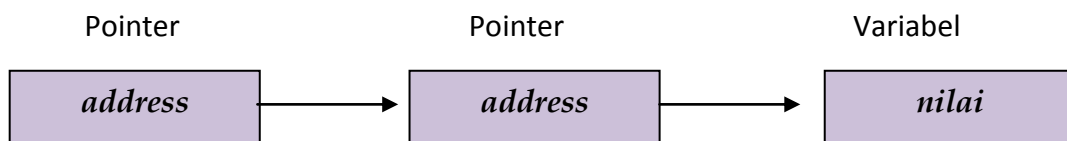
```
// deklarasi
tipe_data nama_variabel;
tipe_data *pointer1; // menunjuk ke variabel
tipe_data **pointer2; // menunjuk ke pointer

// penugasan
pointer1 = &nama_variabel;
pointer2 = &pointer1;
```

Pada kondisi pointer biasa atau pointer tunggal, diagramnya adalah sebagai berikut :



Sedangkan untuk kondisi pointer dalam pointer, diagramnya adalah sebagai berikut :



```
1  /*
2   Program 11-4
3   Nama File : Program 10-4.c
4   Programmer : Eko Budi Setiawan
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int x, *y, **z;
```



```

13     x=5;
14     y=&x;
15     z=&y;
16     printf("Nilai X   = %i\n",**z);
17     printf("Alamat X = %p\n",z);
18     system("PAUSE");
19     return 0;
20 }

```

```

1  /*
2  Program 11-4
3  Nama File : Program 10-4.c
4  Programmer : Eko Budi Setiawan
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int x, *y, **z;
15     x=5;
16     y=&x;
17     z=&y;
18     cout<<"Nilai X   = "<<**z<<endl;
19     cout<<"Alamat X = "<<z<<endl;
20     system("PAUSE");
21     return EXIT_SUCCESS;
22 }

```

```

F:\Dosen @UNIKOM\Semester Ganjil 2010-2011\SEMESTER 3\Praktikum Pemrograman \Kumpula...
Nilai X   = 5
Alamat X = 0x22ff40
Press any key to continue . . . -

```

Gambar 11.4 Tampilan Program 11-4