



## **Pendahuluan Pointer**

Secara konsep pointer adalah sebuah variabel yang menyimpan suatu alamat memori yang menyimpan suatu nilai. Sebenarnya konsep dari pointer itu sendiri tidak sesulit yang dibayangkan. Salah satu kelebihan bahasa C/C++ adalah karena bahasa-bahasa ini mendukung sepenuhnya untuk memanipulasi memori menggunakan pointer.

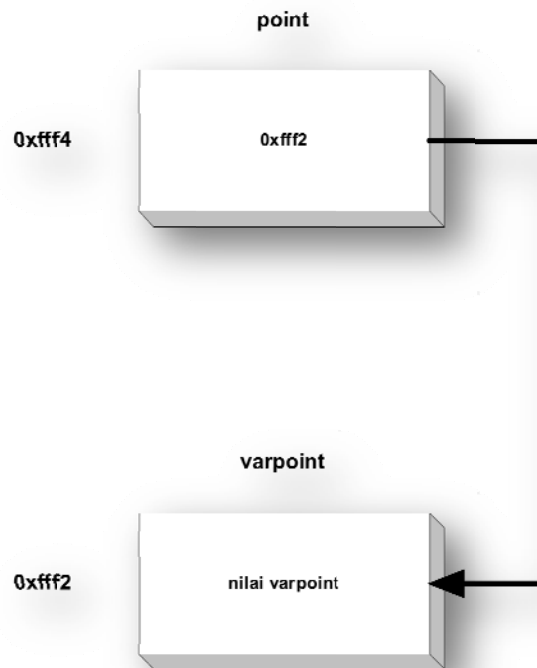
Ada satu hal yang harus diperhatikan dalam mempelajari pointer yaitu apabila seorang programmer menggunakan pointer dengan cara yang salah maka akan menyebabkan sistem operasi menjadi crash (rusak). Oleh sebab itu di dalam bab ini akan dipelajari cara menggunakan pointer dari yang paling dasar sampai kepada pengaplikasiannya di dalam program.

## **Alamat dan Pointer**

Seperti yang diketahui, satu byte di dalam memori akan memiliki sebuah alamat. Di dalam alamat memori inilah sebuah variabel akan disimpan. Tentunya seorang programmer tidak perlu menyebutkan alamat memori yang digunakan untuk menyimpan variabel karena compiler secara otomatis akan memesankan alamat memori untuk setiap variabelnya ketika program dijalankan.

Pointer di dalam program dimaksudkan untuk menunjuk suatu alamat memori variabel. Sebagai contoh ada sebuah variabel bernama `point` yang bertindak sebagai pointer dan sebuah variabel `varpoint` yang akan bertindak sebagai variabel yang alamatnya akan ditunjuk oleh `point` maka gambarannya akan sebagai berikut:

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.



Gambar 11.1 Gambaran variabel yang ditunjuk oleh pointer

## Mengetahui Alamat Suatu Variabel

Alamat suatu variabel dapat diketahui dengan mudah. Caranya gunakan operator alamat yaitu simbol & (and) di depan suatu variabel. Sebagai contoh suatu variabel bertipe integer bernama x maka untuk menampilkan alamatnya adalah &x. Untuk lebih jelasnya buatlah program berikut ini:

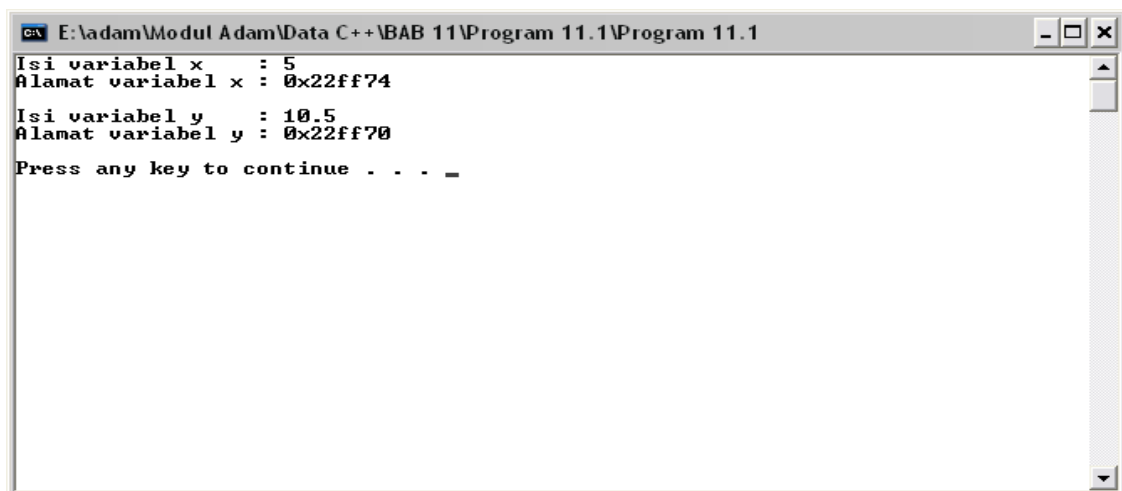
```
1  /*
2   Program 11.1
3   Nama File   : Lat-11.1.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int x=5;
13     float y=10.5;
14     printf("Isi variabel x      : %i\n",x);
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
15 printf("Alamat variabel x : 0x%p\n\n",&x);
16 printf("Isi variabel y      : %f\n",y);
17 printf("Alamat variabel y : 0x%p\n\n",&y);
18 system("PAUSE");
19 return 0;
20 }
```

```
1  /*
2  Program 11.1
3  Nama File   : Lat-11.1.cpp
4  Programmer  : Adam Mukharil Bachtiar, S.Kom.
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=5;
15     float y=10.5;
16     cout<<"Isi variabel x      : "<<x<<endl;
17     cout<<"Alamat variabel x  : "<<&x<<endl<<endl;
18     cout<<"Isi variabel y      : "<<y<<endl;
19     cout<<"Alamat variabel y  : "<<&y<<endl<<endl;
20     system("PAUSE");
21     return EXIT_SUCCESS;
22 }
```

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.1\Program 11.1
Isi variabel x      : 5
Alamat variabel x  : 0x22ff74

Isi variabel y      : 10.5
Alamat variabel y  : 0x22ff70

Press any key to continue . . . _
```

Gambar 11.2 Hasil eksekusi program Lat 11.1

## Mendeklarasikan Variabel Pointer

Sebenarnya variabel pointer tidak terlalu jauh berbeda dengan variabel pada umumnya. Perbedaannya hanya terletak dari apa yang disimpannya. Bentuk umum pendeklarasian variabel pointer adalah sebagai berikut:

```
tipe_data *nama_variabel;  
atau  
tipe_data * nama_variabel;
```

Bentuk umum di atas menggambarkan bahwa pointer yang dideklarasikan akan ditempati oleh data dengan tipe yang disebutkan ketika pendeklarasian. Sebagai contoh untuk mendeklarasikan variabel pointer yang akan menampung menunjuk tipe data integer digunakan perintah sebagai berikut:

```
int *nama_variabel;
```

Catatan



Tipe data pada pendeklarasian variabel pointer harus sama dengan tipe data variabel yang ditunjuk. Pemberian nama variabel pointer disarankan menggunakan awalan p untuk mempermudah membedakan antara variabel pointer dengan variabel biasa.

## Penugasan Variabel Pointer

Setelah dideklarasikan, variabel pointer harus ditugaskan untuk menunjuk variabel. Hal ini dilakukan supaya variabel pointer dapat menunjuk variabel lain yang bersangkutan. Bentuk umum penugasan variabel pointer adalah sebagai berikut:

```
variabel_pointer=&variabel_yang_ditunjuk;
```

Pernyataan di atas berarti variabel pointer diisi dengan alamat dari variabel yang ditunjuk. Untuk dapat mengakses alamat dari variabel yang ditunjuk pointer tinggal menyebutkan nama variabel pointernya saja sedangkan untuk mengakses nilai dari variabel yang ditunjuk oleh pointer harus menyebutkan nama variabel pointer diawali dengan tanda \*. Sebagai contoh buatlah program berikut ini:

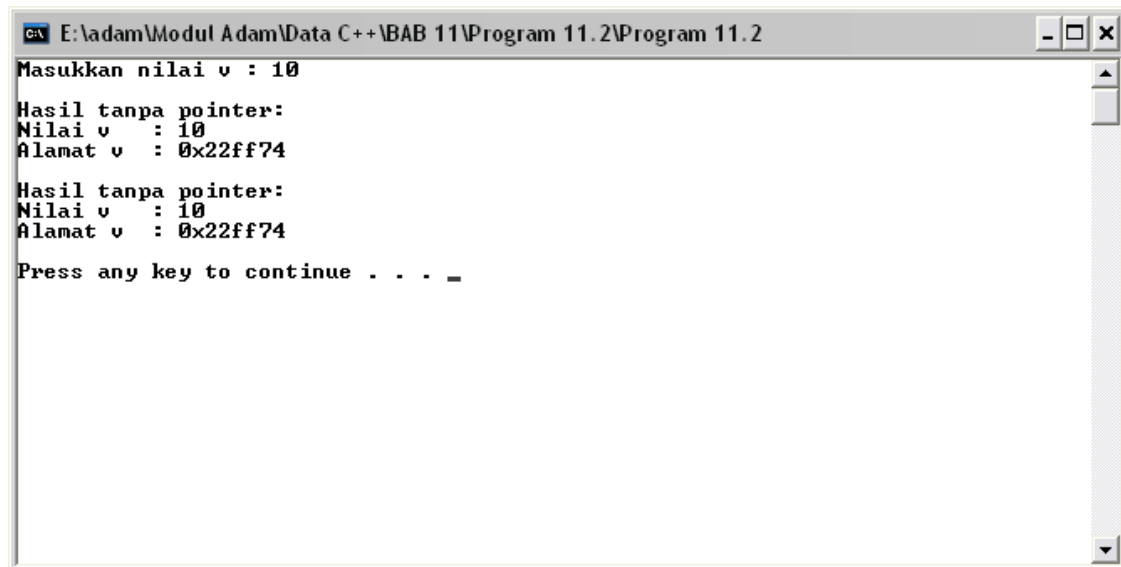
```
1  /*
2   Program 11.2
3   Nama File   : Lat-11.2.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int v;
13     int *pv;
14     printf("Masukkan nilai v : ");scanf("%i",&v);
15     printf("\n");
16     //penugasan pointer
17     pv=&v;
18     printf("Hasil tanpa pointer:\n");
19     printf("Nilai v   : %i\n",v);
20     printf("Alamat v  : 0x%p\n\n",&v);
21     printf("Hasil tanpa pointer:\n");
22     printf("Nilai v   : %i\n",*pv);
23     printf("Alamat v  : 0x%p\n\n",pv);
24     system("PAUSE");
25     return 0;
26 }
```

```
1  /*
2   Program 11.2
3   Nama File   : Lat-11.2.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int v;
15     int *pv;
16     cout<<"Masukkan nilai v : ";cin>>v;
17     cout<<endl;
18     //penugasan pointer
19     pv=&v;
20     cout<<"Hasil tanpa pointer:"<<endl;
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
21     cout<<"Nilai v   : "<<v<<endl;
22     cout<<"Alamat v  : "<<&v<<endl<<endl;
23     cout<<"Hasil tanpa pointer:"<<endl;
24     cout<<"Nilai v   : "<<*pv<<endl;
25     cout<<"Alamat v  : "<<pv<<endl<<endl;
26     system("PAUSE");
27     return EXIT_SUCCESS;
28 }
```


Hasil eksekusi:



```

E:\adam\Modul Adam\Data C++\BAB 11\Program 11.2\Program 11.2
Masukkan nilai v : 10
Hasil tanpa pointer:
Nilai v   : 10
Alamat v  : 0x22ff74
Hasil tanpa pointer:
Nilai v   : 10
Alamat v  : 0x22ff74
Press any key to continue . . . _
```

Gambar 11.3 Hasil eksekusi program Lat 11.2

<p>Catatan</p> 	<p>Hasil eksekusi menggunakan bahasa C akan berbeda dengan hasil eksekusi menggunakan bahasa C++. Perbedaan terletak di dalam menampilkan alamat memori. Bahasa C akan menyebutkan alamat memori dengan dua nol di depan alamat memori (contoh: 0022ff44) sedangkan C++ tidak (contoh: 22ff44).</p>
--	---

## Pointer Tidak Bertipe (Pointer Void)

Pada sub bab sebelumnya telah disebutkan bahwa untuk mendeklarasikan variabel pointer dibutuhkan tipe data yang sama dengan tipe data variabel yang akan ditunjuk. Akan tetapi di dalam bahasa C/C++ bisa mendeklarasikan variabel pointer tanpa menyebutkan tipe datanya terlebih dahulu (tipe datanya diisi void). Hal

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

ini dilakukan agar satu pointer dapat menunjuk variabel yang berbeda tipe data. Bentuk umum pendeklarasian pointer tak bertipe adalah sebagai berikut: :

```
void *nama_pointer;
```

Tetapi tetap dalam pemanggilan alamat dan pemanggilan nilainya, variabel pointer harus disertai dengan tipe datanya. Bentuk umum pemanggilan pointer tak bertipe adalah sebagai berikut:

```
//pemanggilan alamat
(tipe_data *)nama_pointer;

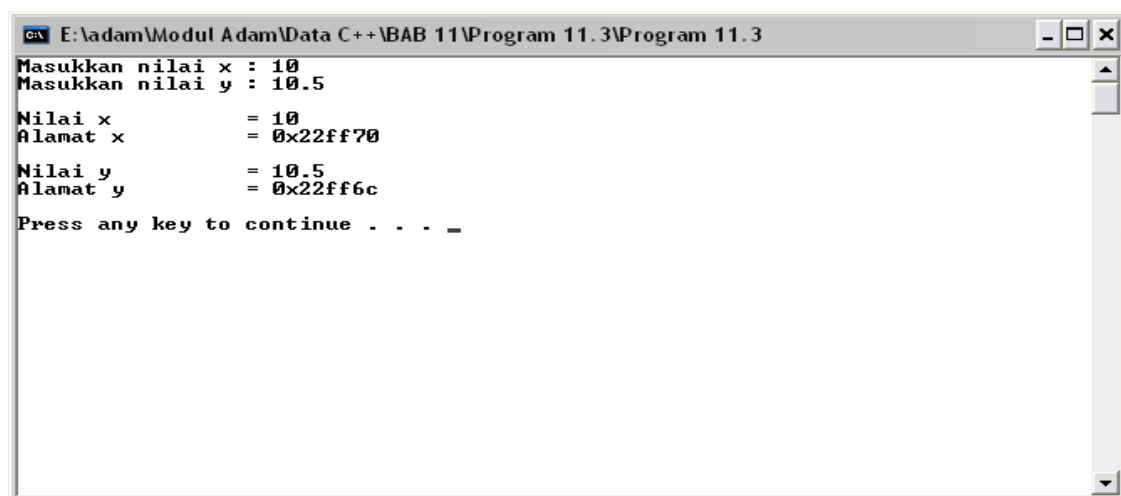
//pemanggilan nilai
*(tipe_data *)nama_pointer;
```

Untuk lebih jelasnya buatlah program berikut ini:

```
1  /*
2   Program 11.3
3   Nama File   : Lat-11.3.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     void *p;
13     int x;
14     float y;
15     printf("Masukkan nilai x : ");scanf("%i",&x);
16     printf("Masukkan nilai y : ");scanf("%f",&y);
17     p=&x;
18     printf("\n");
19     printf("Nilai x           = %i\n",*(int *)p);
20     printf("Alamat x          = 0x%p\n\n",(int *)p);
21     p=&y;
22     printf("Nilai y           = %f\n",*(float *)p);
23     printf("Alamat y          = 0x%p\n\n",(float *)p);
24     system("PAUSE");
25     return 0;
26 }
```

```
1  /*
2   Program 11.3
3   Nama File   : Lat-11.3.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     void *p;
15     int x;
16     float y;
17     cout<<"Masukkan nilai x : ";cin>>x;
18     cout<<"Masukkan nilai y : ";cin>>y;
19     p=&x;
20     cout<<endl;
21     cout<<"Nilai x           = "<<*(int *)p<<endl;
22     cout<<"Alamat x          = "<<(int *)p<<endl<<endl;
23     p=&y;
24     cout<<"Nilai y           = "<<*(float *)p<<endl;
25     cout<<"Alamat y          = "<<(float *)p
26         <<endl<<endl;
27     system("PAUSE");
28     return EXIT_SUCCESS;
29 }
```

Hasil eksekusi:



```
cmd E:\adam\Modul Adam\Data C++\BAB 11\Program 11.3\Program 11.3
Masukkan nilai x : 10
Masukkan nilai y : 10.5
Nilai x           = 10
Alamat x          = 0x22ff70
Nilai y           = 10.5
Alamat y          = 0x22ff6c
Press any key to continue . . . _
```

Gambar 11.4 Hasil eksekusi program Lat 11.3



## Pointer dan Array

Pointer dan array memiliki hubungan yang dekat. Secara internal array pun menyatakan alamat. Sebagai contoh didefinisikan sebuah array yang menyimpan lima bilangan bulat sebagai berikut:

```
int bil_bulat[]={0,1,2,3,4};
```

Dan sebuah pointer yang akan menunjuk array tersebut sebagai berikut:

```
int *pbil_bulat;
```

Maka penugasan pointer terhadap variabel arraynya adalah sebagai berikut:

```
pbil_bulat=bil_bulat;
atau
pbil_bulat=&bil_bulat[indeks];
```

Terlihat dalam penugasan pointer di atas terdapat pernyataan yang tidak menggunakan tanda **&**. Hal ini disebabkan karena nama array sebenarnya sudah menunjuk suatu alamat sehingga tanda **&** tidak lagi dibutuhkan. Untuk lebih jelasnya buatlah program berikut ini:

```
1  /*
2   Program 11.4
3   Nama File   : Lat-11.4.c
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int bil_bulat[]={0,1,2,3,4};
13     int *pbil,i;
14     //penugasan array
15     pbil=bil_bulat;
16     for(i=0;i<5;i++)
17     {
18         printf("Nilai yang ditunjuk [%i] : %i\n",
19                i+1,*pbil);
20         printf("Alamat yang ditunjuk [%i] : 0x%p\n\n",
21                i+1,pbil);
```

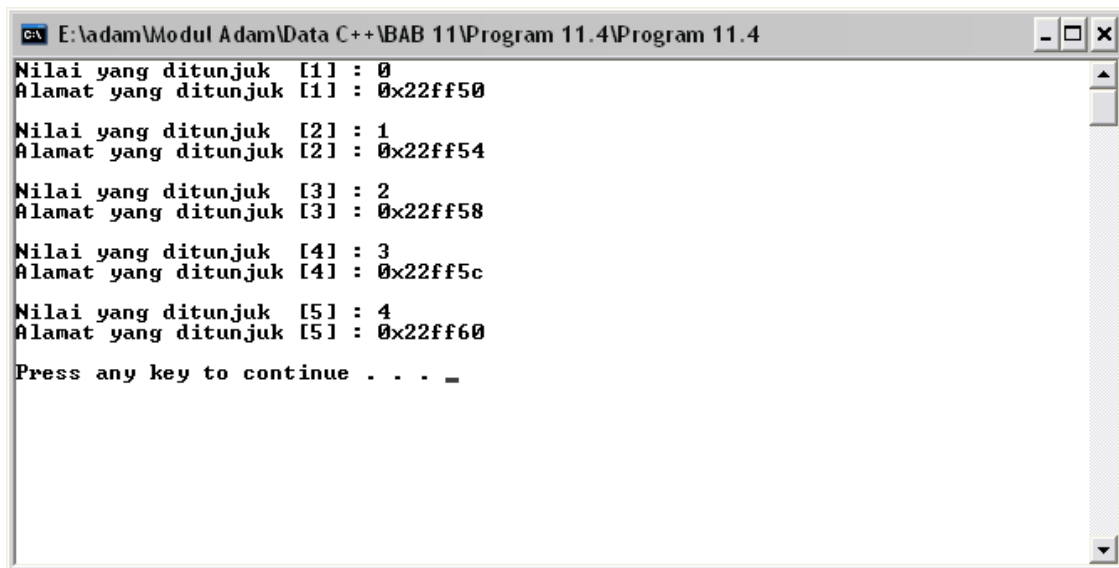
DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
20     pbil++;
21     }
22     system("PAUSE");
23     return 0;
24 }
```

```
1  /*
2   Program 11.4
3   Nama File   : Lat-11.4.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int bil_bulat[]={0,1,2,3,4};
15     int *pbil,i;
16     //penugasan array
17     pbil=bil_bulat;
18     for(i=0;i<5;i++)
19     {
20         cout<<"Nilai yang ditunjuk ["<<i+1<<" : "<<*pbil
21             <<endl;
22         cout<<"Alamat yang ditunjuk ["<<i+1<<" : "<<pbil
23             <<endl<<endl;
24         pbil++;
25     }
26     system("PAUSE");
27     return EXIT_SUCCESS;
28 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.4\Program 11.4
Nilai yang ditunjuk [1] : 0
Alamat yang ditunjuk [1] : 0x22ff50
Nilai yang ditunjuk [2] : 1
Alamat yang ditunjuk [2] : 0x22ff54
Nilai yang ditunjuk [3] : 2
Alamat yang ditunjuk [3] : 0x22ff58
Nilai yang ditunjuk [4] : 3
Alamat yang ditunjuk [4] : 0x22ff5c
Nilai yang ditunjuk [5] : 4
Alamat yang ditunjuk [5] : 0x22ff60
Press any key to continue . . . _
```

Gambar 11.5 Hasil eksekusi program Lat 11.4

Di dalam program ini ada pernyataan **pbil++**. Pernyataan ini digunakan agar pointer array tersebut bergerak untuk menunjuk data berikutnya di dalam array. Sebenarnya ada cara lain untuk mengakses seluruh isi seluruh elemen array. Caranya adalah sebagai berikut:

```
//pengaksesan nilai array
for (var=0;var<maks_array;var++)
{
    //perintah output diikuti *(nama_array+var);
}

//pengaksesan alamat
for (var=0;var<maks_array;var++)
{
    perintah output diikuti (nama_array+var);
}
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Untuk lebih jelasnya buatlah program berikut:

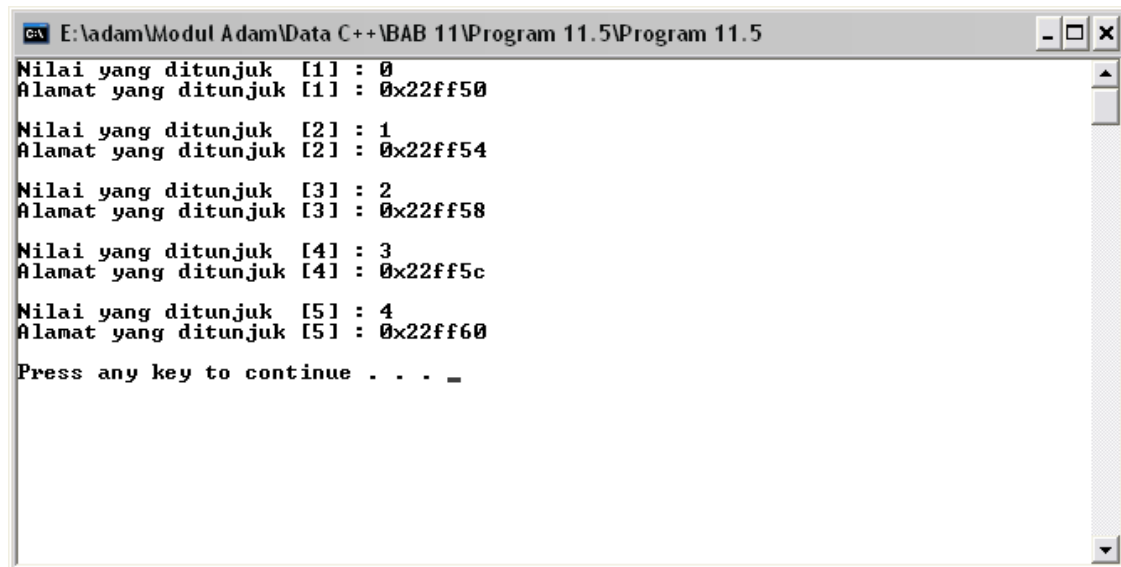
```
1  /*
2     Program 11.5
3     Nama File   : Lat-11.5.c
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int bil_bulat[]={0,1,2,3,4};
13     int *pbil,i;
14     //penugasan array
15     pbil=bil_bulat;
16     for(i=0;i<5;i++)
17     {
18         printf("Nilai yang ditunjuk [%i] : %i\n",i+1,
19                *(pbil+i));
20         printf("Alamat yang ditunjuk [%i] : 0x%p\n\n",i+1,
21                (pbil+i));
22     }
23     system("PAUSE");
24     return 0;
25 }
```

```
1  /*
2     Program 11.5
3     Nama File   : Lat-11.5.cpp
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int bil_bulat[]={0,1,2,3,4};
15     int *pbil,i;
16     //penugasan array
17     pbil=bil_bulat;
18     for(i=0;i<5;i++)
19     {
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
20     cout<<"Nilai yang ditunjuk ["<<i+1<<"] : "  
      <<*(pbil+i)<<endl;  
21     cout<<"Alamat yang ditunjuk ["<<i+1<<"] : "  
      <<(pbil+i)<<endl<<endl;  
22     }  
23     system("PAUSE");  
24     return EXIT_SUCCESS;  
25 }
```

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.5\Program 11.5  
Nilai yang ditunjuk [1] : 0  
Alamat yang ditunjuk [1] : 0x22ff50  
Nilai yang ditunjuk [2] : 1  
Alamat yang ditunjuk [2] : 0x22ff54  
Nilai yang ditunjuk [3] : 2  
Alamat yang ditunjuk [3] : 0x22ff58  
Nilai yang ditunjuk [4] : 3  
Alamat yang ditunjuk [4] : 0x22ff5c  
Nilai yang ditunjuk [5] : 4  
Alamat yang ditunjuk [5] : 0x22ff60  
Press any key to continue . . . _
```

Gambar 11.6 Hasil eksekusi program Lat 11.5

## Pointer dan String

Pointer bisa digunakan untuk menunjuk variabel bertipe string. Untuk menggambarkan hubungan pointer dengan string buatlah program berikut ini:

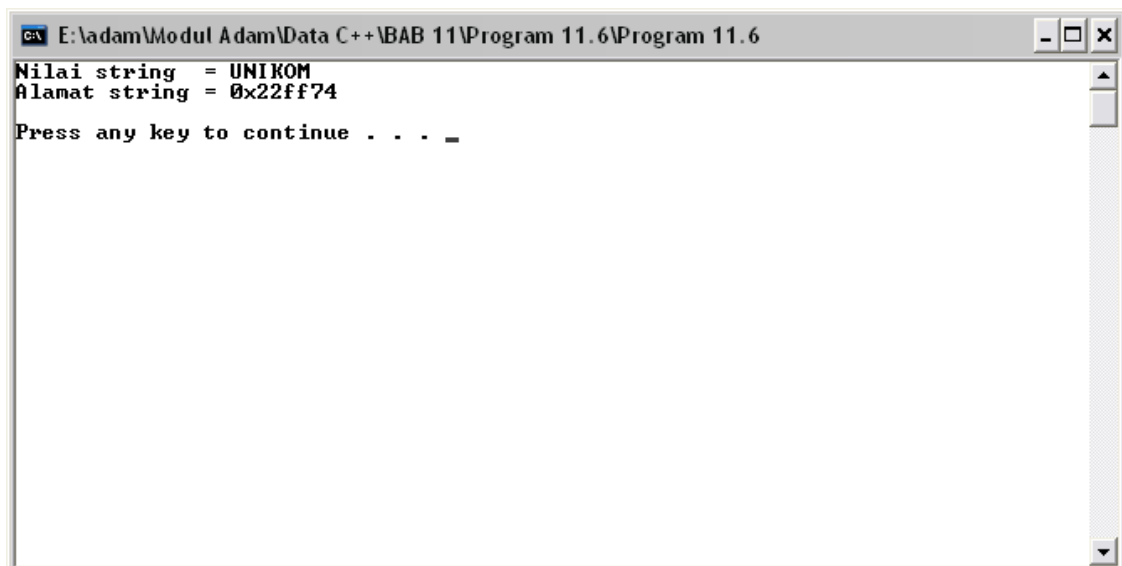
```
1  /*  
2  Program 11.6  
3  Nama File   : Lat-11.6.c  
4  Programmer  : Adam Mukharil Bachtiar, S.Kom.  
5  */  
6  
7  #include <stdio.h>  
8  #include <stdlib.h>  
9  
10 int main(int argc, char *argv[])  
11 {
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

```
12 char *pstring="UNIKOM";
13 printf("Nilai string = %s\n",pstring);
14 printf("Alamat string = 0x%p\n\n",&pstring);
15 system("PAUSE");
16 return 0;
17 }
```

```
1  /*
2   Program 11.6
3   Nama File   : Lat-11.6.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     char *pstring="UNIKOM";
15     cout<<"Nilai string = "<<pstring<<endl;
16     cout<<"Alamat string = "<<&pstring<<endl<<endl;
17     system("PAUSE");
18     return EXIT_SUCCESS;
19 }
```

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.6\Program 11.6
Nilai string = UNIKOM
Alamat string = 0x22ff74
Press any key to continue . . . _
```

Gambar 11.7 Hasil eksekusi program Lat 11.6

## Pointer Menunjuk Pointer

Selain untuk menunjuk variabel biasa, pointer juga bisa digunakan untuk menunjuk variabel pointer yang lain. Mungkin untuk beberapa programmer pemula, kasus ini membingungkan. Tetapi apabila sudah mempelajari struktur data (bab linked list) maka sub bab ini menjadi berguna. Bentuk umum pointer menunjuk pointer adalah sebagai berikut:

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

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

//pemanggilan nilai
perintah output diikuti **pointer2

//pemanggilan alamat
perintah output diikuti *pointer2
```

Untuk lebih jelasnya buatlah program berikut:

```
1  /*
2     Program 11.7
3     Nama File   : Lat-11.7.c
4     Programmer : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int x=10;
13     int *px1;
14     int **px2;
15     px1=&x;
16     px2=&px1;
```

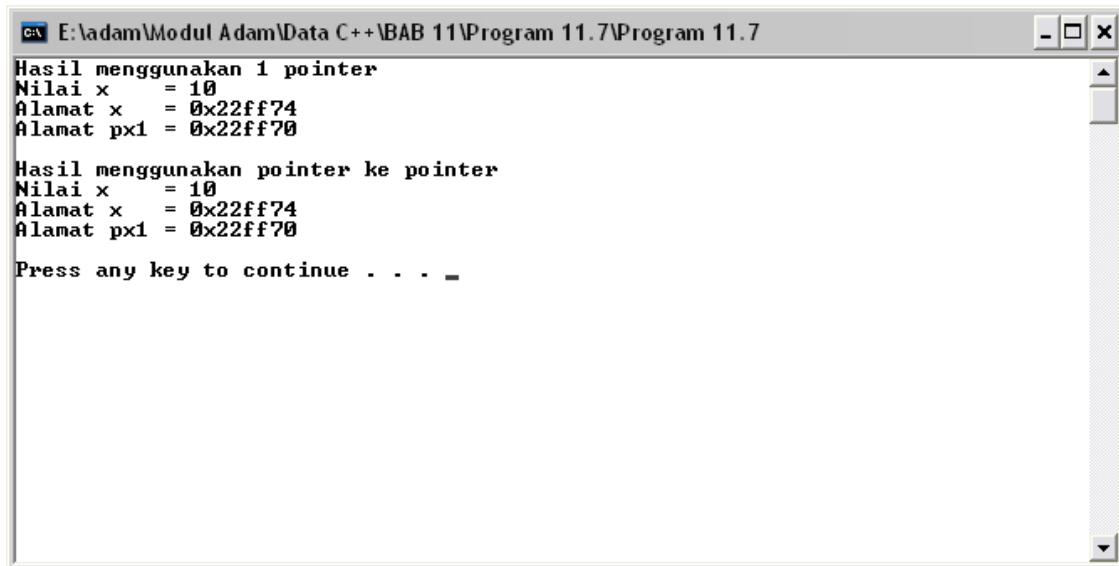
```
17 printf("Hasil menggunakan 1 pointer\n");
18 printf("Nilai x      = %i\n",*px1);
19 printf("Alamat x    = %p\n",px1);
20 printf("Alamat px1 = %p\n\n",&px1);
21 printf("Hasil menggunakan pointer ke pointer\n");
22 printf("Nilai x      = %i\n",**px2);
23 printf("Alamat x    = %p\n",*px2);
24 printf("Alamat px1 = %p\n\n",px2);
25 system("PAUSE");
26 return 0;
27 }
```

```
1  /*
2   Program 11.7
3   Nama File   : Lat-11.7.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
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=10;
15     int *px1;
16     int **px2;
17     px1=&x;
18     px2=&px1;
19     cout<<"Hasil menggunakan 1 pointer"<<endl;
20     cout<<"Nilai x      = "<<*px1<<endl;
21     cout<<"Alamat x    = "<<px1<<endl;
22     cout<<"Alamat px1 = "<<&px1<<endl<<endl;
23     cout<<"Hasil menggunakan pointer ke pointer"<<endl;
24     cout<<"Nilai x      = "<<**px2<<endl;
25     cout<<"Alamat x    = "<<*px2<<endl;
26     cout<<"Alamat px1 = "<<px2<<endl<<endl;
27     system("PAUSE");
28     return EXIT_SUCCESS;
29 }
```



DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.7\Program 11.7
Hasil menggunakan 1 pointer
Nilai x = 10
Alamat x = 0x22ff74
Alamat px1 = 0x22ff70

Hasil menggunakan pointer ke pointer
Nilai x = 10
Alamat x = 0x22ff74
Alamat px1 = 0x22ff70

Press any key to continue . . . _
```

Gambar 11.8 Hasil eksekusi program Lat 11.7

## Manajemen Memori Dinamis

Salah satu pengaplikasian pointer adalah untuk manajemen memori dinamis. Maksudnya seorang programmer bisa memesan memori untuk variabelnya ketika program sedang dijalankan. Ada dua pasang perintah yang bisa digunakan yaitu:

1. Malloc dan free
2. New dan delete.

## Malloc dan Free

Malloc adalah perintah untuk mengalokasikan memori ketika program sedang berjalan sedangkan free adalah perintah untuk mendealokasikan memori ketika program sedang berjalan. Bentuk umum dari malloc dan free adalah sebagai berikut:

```
//malloc
pointer=(tipe_data *)malloc(sizeof(tipe_data)*jumlah_data;

//free
free(pointer);
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

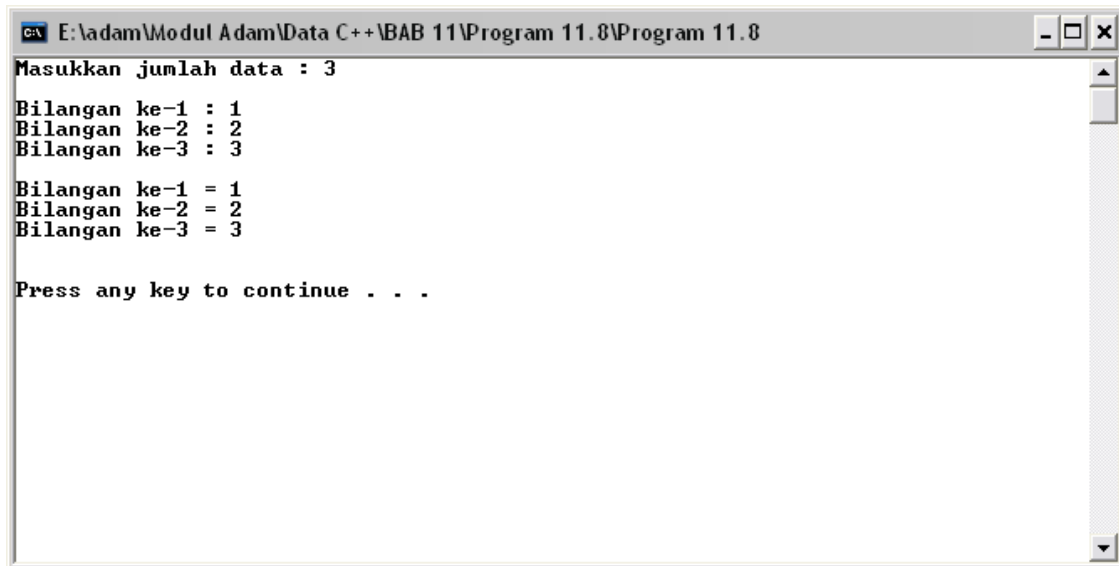
Untuk lebih jelasnya buatlah program berikut ini:

```
1  /*
2     Program 11.8
3     Nama File   : Lat-11.8.c
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(int argc, char *argv[])
11 {
12     int *bil;
13     int i, jumlah_bil;
14     printf("Masukkan jumlah data : ");
15     scanf("%i",&jumlah_bil);
16
17     //penugasan pointer dan alokasi memori
18     bil=(int *)malloc(sizeof(int)*jumlah_bil);
19
20     //input data
21     printf("\n");
22     for(i=0;i<jumlah_bil;i++)
23     {
24         printf("Bilangan ke-%i : ",i+1);scanf("%i",(bil+i));
25     }
26
27     //output data
28     printf("\n");
29     for(i=0;i<jumlah_bil;i++)
30     {
31         printf("Bilangan ke-%i = %i\n",i+1,* (bil+i));
32     }
33     printf("\n\n");
34
35     //dealokasi memori
36     free(bil);
37
38     system("PAUSE");
39     return 0;
40 }
```

```
1  /*
2   Program 11.8
3   Nama File   : Lat-11.8.cpp
4   Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int *bil;
15     int i, jumlah_bil;
16     cout<<"Masukkan jumlah data : ";cin>>jumlah_bil;
17
18     //penugasan pointer dan alokasi memori
19     bil=(int *)malloc(sizeof(int)*jumlah_bil);
20
21     //input data
22     cout<<endl;
23     for(i=0;i<jumlah_bil;i++)
24     {
25         cout<<"Bilangan ke-"<<i+1<<" : ";cin>>*(bil+i);
26     }
27
28     //output data
29     cout<<endl;
30     for(i=0;i<jumlah_bil;i++)
31     {
32         cout<<"Bilangan ke-"<<i+1<<" = "<<*(bil+i)<<endl;
33     }
34     cout<<endl<<endl;
35
36     //dealokasi memori
37     free(bil);
38
39     system("PAUSE");
40     return EXIT_SUCCESS;
41 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:




```

E:\adam\Modul Adam\Data C++\BAB 11\Program 11.8\Program 11.8
Masukkan jumlah data : 3
Bilangan ke-1 : 1
Bilangan ke-2 : 2
Bilangan ke-3 : 3
Bilangan ke-1 = 1
Bilangan ke-2 = 2
Bilangan ke-3 = 3
Press any key to continue . . .

```

Gambar 11.9 Hasil eksekusi program Lat 11.8

<p>Catatan</p> 	<p>Ada satu perbedaan di dalam bahasa c dan c++ yaitu terletak di perintah input data dinamis. Di dalam bahasa c++ perintah cin&gt;&gt; harus diikuti dengan *nama_pointer sedangkan dalam bahasa c tidak perlu.</p>
--	--

## New dan Delete

Sama seperti malloc dan free, new digunakan untuk mengalokasikan memori sedangkan delete untuk dealokasi memori. Tetapi perintah ini punya satu kelemahan yaitu harus menyertakan file header **cstdlib** yang hanya ada di bahasa C++. Bentuk umum new dan delete adalah sebagai berikut :

```

//new
pointer=new tipe_data_pointer;//satu memori
atau
pointer=new tipe_data_pointer[jumlah_data];

//delete
delete pointer;

```

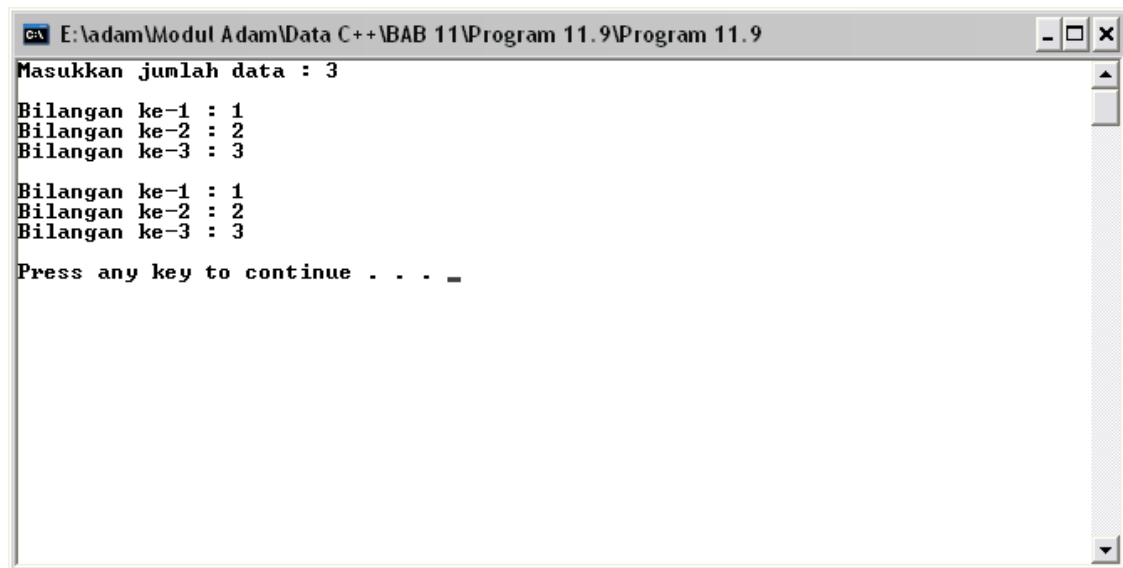
DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Untuk lebih jelasnya buatlah program berikut ini:

```
1  /*
2     Program 11.9
3     Nama File   : Lat-11.9.cpp
4     Programmer  : Adam Mukharil Bachtiar, S.Kom.
5  */
6
7  #include <cstdlib>
8  #include <iostream>
9
10 using namespace std;
11
12 int main(int argc, char *argv[])
13 {
14     int *bil;
15     int i,jumlah_bil;
16     cout<<"Masukkan jumlah data : ";cin>>jumlah_bil;
17
18     //alokasi memori
19     bil=new int[jumlah_bil];
20
21     //input data
22     cout<<endl;
23     for(i=0;i<jumlah_bil;i++)
24     {
25         cout<<"Bilangan ke-"<<i+1<<" : ";cin>>*(bil+i);
26     }
27
28     //output data
29     cout<<endl;
30     for(i=0;i<jumlah_bil;i++)
31     {
32         cout<<"Bilangan ke-"<<i+1<<" : "<<*(bil+i)<<endl;
33     }
34
35     cout<<endl;
36
37     //dealokasi memori
38     delete bil;
39     system("PAUSE");
40     return EXIT_SUCCESS;
41 }
```

DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Hasil eksekusi:



```
E:\adam\Modul Adam\Data C++\BAB 11\Program 11.9\Program 11.9
Masukkan jumlah data : 3
Bilangan ke-1 : 1
Bilangan ke-2 : 2
Bilangan ke-3 : 3
Bilangan ke-1 : 1
Bilangan ke-2 : 2
Bilangan ke-3 : 3
Press any key to continue . . . _
```

Gambar 11.10 Hasil eksekusi program Lat 11.9