|  |  |  |
| --- | --- | --- |
| logounikombw | **Diktat Mata Kuliah Pemrograman I** | IF |
| BAB XI  POINTER |

Pointer ?

Pointer adalah sebuah variabel yang isi datanya adalah alamat memori atau variabel lain. Sehingga pointer dapat juga disebut sebagai variabel alamat (address variable).

Deklarasi pointer

Untuk mendeklarasikan sebuah pointer, perintah dasarnya adalah :

Typedata \*namavariabel;

Untuk lebih jelasnya adalah :

int \*pint;

float \*pfloat;

Tmhs \*pmhs;

Pada contoh ke-1 kita mendeklarasikan sebuah pointer bernama pint yang menunjuk ke suatu alamat di memori yang menampung sebuah data bertipe integer. Contoh ke-2 mendeklarasikan sebuah variabel pointer bernama pfloat yang menunjuk ke suatu alamat yang menampung sebuah data bertipe float, begitu juga dengan contoh ke-3 yang mendeklarasikan suatu variabel bernama pmhs yang menunjuk ke suatu data yang bertipe Tmhs.

Pengisian data ke variabel pointer

Pengisian data ke variabel pointer bisa berarti pengisian alamat memori ke variabel tersebut atau pengisian data yang ditunjuk oleh pointer.

Untuk lebih jelasnya, perhatikan program dibawah ini :

|  |
| --- |
| 01: #include <stdio.h>  02: #include <conio.h>  03: #include <string.h>  04:  05: main()  06: {  07: char c,\*pc;  08: int i,\*pi;  09: float f,\*pf;  10: clrscr();  11: c='A';i=7;f=6.25;  12: printf("c : alamat=0x%p, isi=%c\n", &c, c);  13: printf("x : alamat=0x%p, isi=%d\n", &i, i);  14: printf("y : alamat=0x%p, isi=%5.2f\n", &f, f);  15: pc=&c;  16: pi=&i;  17: pf=&f;  18: printf("pc: alamat=0x%p, isi=%c\n", pc, \*pc);  19: printf("pi: alamat=0x%p, isi=%d\n", pi, \*pi);  20: printf("pf: alamat=0x%p, isi=%5.2f\n", pf, \*pf);  21: \*pc='B';  22: \*pi=125;  23: \*pf=512.56;  24: printf("c : isi=%c\n", c);  25: printf("x : isi=%d\n", i);  26: printf("y : isi=%5.2f\n", f);  27: getch();  28: return 0;  29: } |

Jika dieksekusi, maka akan menghasilkan sebuah tampilan sebagai berikut :

c : alamat=0xFFF5, isi=A

x : alamat=0xFFF2, isi=7

y : alamat=0xFFEE, isi= 6.25

pc: alamat=0xFFF5, isi=A

pi: alamat=0xFFF2, isi=7

pf: alamat=0xFFEE, isi= 6.25

c : isi=B

x : isi=125

y : isi=512.56

Keterangan Program :

* Pada baris 7 : Pendeklarasian sebuah variabel **c** dengan tipe char, dan sebuah pointer **pc** yang merupakan pointer char.
* Pada baris 8 : Pendeklarasian sebuah variabel **i** dengan tipe int, dan sebuah pointer **pi** yang merupakan pointer int.
* Pada baris 9 : Pendeklarasian sebuah variabel **f** dengan tipe int, dan sebuah pointer **pf** yang merupakan pointer float.
* Pada baris 11 : Pengisian variabel **c** dengan karakter ‘A’, variabel **i** dengan nilai 7, dan variabel **f** dengan nilai 6.25.
* Pada baris 12 : Menampilkan alamat variabel **c** dan isinya.
* Pada baris 13 : Menampilkan alamat variabel **i** dan isinya.
* Pada baris 14 : Menampilkan alamat variabel **f** dan isinya.
* Pada baris 15 : Variabel pointer **pc** diisi dengan alamat variabel **c** sehingga kedua variabel mengacu ke tempat yang sama sehingga ketika isi **pc** diubah maka berarti merubah isi variabel **c** dan begitu juga sebaliknya.
* Pada baris 15 : Variabel pointer **pi** diisi dengan alamat variabel **i** sehingga kedua variabel mengacu ke tempat yang sama sehingga ketika isi **pi** diubah maka berarti merubah isi variabel **i** dan begitu juga sebaliknya.
* Pada baris 15 : Variabel pointer **pf** diisi dengan alamat variabel **f** sehingga kedua variabel mengacu ke tempat yang sama sehingga ketika isi **pf** diubah maka berarti merubah isi variabel **f** dan begitu juga sebaliknya.
* Baris 16, 17 dan 18 : Menampilkan data yang ditunjuk oleh pointer **pc, pi,** dan **pf**, yang hasilnya pasti sama dengan hasil baris 12, 13, dan 14.
* Baris 19 : Mengisi data ke alamat yang ditunjuk oleh **pc** dengan nilai B, yang berarti juga mengganti nilai variabel **c**.
* Baris 20 : Mengisi data ke alamat yang ditunjuk oleh **pi** dengan nilai 125, yang berate juga mengganti nilai variabel **i**.
* Baris 21 : Mengisi data ke alamat yang ditunjuk oleh **pf** dengan nilai 512.56, yang berarti juga mengganti nilai variabel **f**.
* Baris 22,23 dan 24 : Menampilkan isi nilai variabel **c, i** dan **f**, yang telah diubah oleh variabel pointernya.

Contoh Aplikasi Pointer

Salah satu penggunaan pointer adalah untuk membuat suatu array yang dinamis (banyaknya data yang bisa ditampung sesuai keperluan).

Sebagaimana kita ketahui jika kita membuat suatu program yang dapat menampung data nilai sebanyak 5 buah maka kita akan membuat suatu variabel array yang bertipe int dengan perintah **int data[5].** Dengan cara begitu maka program hanya akan berjalan dengan baik jika data yang diinputkan banyaknya di kurang atau sama dengan 5 buah. Apa yang terjadi ketika data yang akan diinputkan ternyata 10 buah, maka langkah yang dilakukan adalah harus mengubah programnya dan mengganti **int data[5]** menjadi **int data[10].**

Cara lain untuk membuat program tersebut adalah dengan menggunakan suatu variabel array yang dinamis dimana pemesanan tempat yang diperlukan untuk menyimpan data tidak dideklarasikan dalam program tapi dilakukan secara runtime (ketika program berjalan).

|  |
| --- |
| 01: #include <stdio.h>  02: #include <conio.h>  03: #include <stdlib.h>  04: main()  05: {  06: int \*data;  07: int i,banyakdata;  08: printf("Banyak data yang akan diinputkan : ");scanf("%i",&banyakdata);  09: data=(int \*)malloc(sizeof(int)\*banyakdata);  10: for(i=0;i<banyakdata;i++)  11: {  12: printf("Pemasukan data ke-%i :",i+1);scanf("%i",(data+i));  13: }  14: printf("Data yang telah diinputkan adalah : \n");  15: for(i=0;i<banyakdata;i++)  16: printf("Data ke-%i : %i\n",i+1,\*(data+i));  17: getch();  18: return 0;  19: } |

Tampilan yang dihasilkan oleh program diatas adalah sebagai berikut :

Banyak data yang akan diinputkan : 5

Pemasukan data ke-1 :12

Pemasukan data ke-2 :3

Pemasukan data ke-3 :4

Pemasukan data ke-4 :5

Pemasukan data ke-5 :67

Data yang telah diinputkan adalah :

Data ke-1 : 12

Data ke-2 : 3

Data ke-3 : 4

Data ke-4 : 5

Data ke-5 : 67

Keterangan program :

* Baris 6 : Pendeklarasian sebuah variabel pointer **int** yang bernama **data**.
* Baris 7 : Pendeklarasian variabel **i** sebagai counter dan **banyakdata** untuk menyimpan banyaknya data.
* Baris 8 : Pengisian data banyak data.
* Baris 9 : Pemesanan alokasi di memori untuk variabel pointer data sebesar besarnya **int** (**sizeof(int)**) dikali dengan **banyakdata**.
* Baris 10 : Perulangan untuk membaca data dari data ke-0 sampai ke **banyakdata**-1.
* Baris 12 : Membaca data dari keyboard dan dimasukan ke alamat data pada urutan ke-**i**.
* Baris 15 – 16 : Menampilkan isi data yang ditunjuk oleh pointer