

POINTER

- **Konsep Dasar**

Suatu pointer(variabel penunjuk) adalah suatu variabel yang berisi dengan alamat lokasi memori tertentu. Jadi suatu pointer bukan berisi dengan suatu nilai data tetapi berisi dengan suatu alamat.

Misalnya alamat dari suatu nilai X ini akan disimpan di suatu variabel, maka dapat dituliskan sebagai `Alamat_X = &X`. `Alamat_X` adalah variabel pointer, atau disebut pointer ke X, karena variabel ini menunjuk ke suatu lokasi dimana nilai X disimpan.

Contoh :

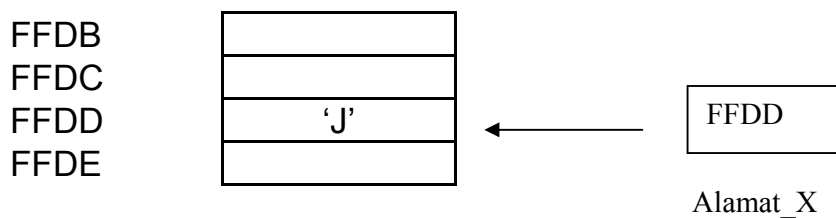
```
#include <stdio.h>
main()
{
    char*Alamat_X,X;

    X='J';
    Alamat_X = &X;
    printf("Nilai variabel X, yaitu 'J' berada di alamat %p\n", Alamat_X);
}
```

Jika program ini dijalankan, akan didapatkan hasil :

Nilai variabel X, yaitu 'J' berada di alamat FFDD

Alamat memori Nilai Disimpan



- **Operator POINTER**

Bahasa C menyediakan dua buah operator untuk operasi pointer yaitu operator '*' dan operator '&'. Operator alamat '&' digunakan untuk mendapatkan alamat memori dari operandnya. Operator '*' digunakan untuk mendapatkan nilai dari operandnya.

Contoh :

```
#include <stdio.h>
main()
{
    char*Alamat_X,X,Y,Z;

    X='J';
    Alamat_X = &X;
    Y = X;
    Z = *Alamat_X;
    printf("Nilai variabel X adalah %c\n", X);
    printf("Nilai variabel Y adalah %c\n", Y);
    printf("Nilai variabel Z adalah %c\n", Z);
    printf("Nilai variabel X adalah berada di alamat %p\n", Alamat_X);
}
```

Jika program ini dijalankan, akan didapatkan hasil :

```
Nilai variabel X adalah J
Nilai variabel Y adalah J
Nilai variabel Z adalah J
Nilai variabel X berada di alamat FFDB
```

- **DEKLARASI POINTER**

Bentuk umum deklarasi variabel pointer :

```
Tipe *nama-variabel-pointer;
```

```

Contoh :
#include <stdio.h>
main()
{
    float Nilai, *Alamat=&Nilai;

    Nilai = 25.55;

    printf("Nilai %7.2f berada di alamat memori %p\n", Nilai, Alamat);
}

```

Jika program ini dijalankan, akan didapatkan hasil :

Nilai 25.55 berada di alamat memori FFDA

• OPERASI POINTER

Variabel pointer dapat dioperasikan secara variabel biasa maupun secara aritmatika dan dapat dibandingkan secara operasi logika.

- Operasi Pengerjaan
Maksud dari operasi pengerjaan ini adalah nilai suatu variabel pointer dapat dipindahkan ke variabel pointer yang lainnya.

```

Contoh :
#include <stdio.h>
main()
{
    float Y, *X1, *X2;

    Y = 12.34;
    X1 = &Y;
    X2 = X1; /* pengerjaan variabel pointer*/

    printf("Nilai variabel Y ada di alamat %p\n", X1);
    printf("Nilai variabel Y ada di alamat %p\n", X2);
}

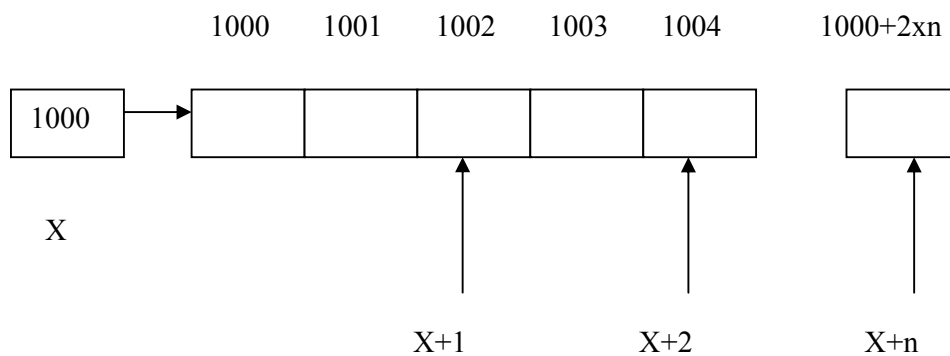
```

Jika program ini dijalankan, akan didapatkan hasil :

Nilai variabel Y ada di alamat FFDA
Nilai variabel Y ada di alamat FFDA

- Operasi Aritmatika

Suatu variabel pointer dapat ditambah atau dikurangi dengan suatu nilai integer saja. Operasi penambahan pointer dengan suatu nilai integer merupakan suatu peningkatan nilai pointer yang menunjukkan lokasi nilai data berikutnya (bukan byte berikutnya) di memori.



Contoh :

```
#include <stdio.h>
main()
{
    int Nilai1,Nilai2,Nilai3, *Penunjuk;

    Nilai1 = 123;
    Nilai1 = 123;
    Nilai1 = 123;
    Penunjuk = &Nilai1;

    printf("Nilai %d di alamat memori %p\n", *Penunjuk, Penunjuk);
    printf("Nilai %d di alamat memori %p\n", *(Penunjuk+1), Penunjuk+1);
    printf("Nilai %d di alamat memori %p\n", *(Penunjuk+2), Penunjuk+2);
}
```

Jika program ini dijalankan, akan didapatkan hasil :

Nilai 123 di alamat memori FFD8
Nilai 345 di alamat memori FFDA
Nilai 678 di alamat memori FFDC

- Operasi Logika

Dua buah variabel pointer dapat dibandingkan jika keduanya mempunyai tipe yang sama atau keduanya bernilai null.

Contoh :

```
#include <stdio.h>
main()
{
    int Nilai1=100,Nilai2=200, *PN1,*PN2,*PN3;

    PN1 = &Nilai1;
    PN2 = &Nilai2;
    PN3 = PN1;
    Penunjuk = &Nilai1;

    If(PN1<PN2) printf("PN1 menunjuk ke memori lebih rendah dari PN2\n");
    If(PN1= =PN2) printf("PN1 menunjuk ke memori yg sama dengan PN2\n");
    If(PN1>PN2) printf("PN1 menunjuk ke memori lebih rendah dari PN2\n");

    If(PN1<PN3) printf("PN1 menunjuk ke memori lebih rendah dari PN3\n");
    If(PN1= =PN3) printf("PN1 menunjuk ke memori yg sama dengan PN3\n");
    If(PN1>PN3) printf("PN1 menunjuk ke memori lebih rendah dari PN3\n");

}
```

Jika program ini dijalankan, akan didapatkan hasil :

```
PN1 menunjuk ke memori lebih rendah dari PN2
PN1 menunjuk ke memori yg sama dengan PN3
```

- **OPERASI POINTER UNTUK LARIK**

Pointer dan larik mempunyai hubungan yang sangat erat. Misalnya suatu larik dimensi satu dengan nama X dan variabel pointer dengan nama P. Alamat elemen-elemen larik dimensi satu ini mulai elemen pertama sampai dengan n dapat ditunjukkan sebagai berikut :

Elemen ke 1: &X[0] atau X atau X+0 atau P atau P+0
 Elemen ke 2: &X[1] atau X+1 atau P+1
 Elemen ke 3: &X[2] atau X +2 atau P+2
 Elemen ke n: &X[n-1] atau X +n-1 atau P+n-1

Sedangkan nilai-nilai dari elemen larik dimensi satu ini dapat diakses sebagai berikut ini.

Elemen ke 1: &X[0] atau *X atau *(X+0) atau *P atau *(P+0)
 Elemen ke 2: &X[1] atau *(X+1) atau *(P+1)
 Elemen ke 3: &X[2] atau *(X +2) atau *(P+2)
 Elemen ke n: &X[n-1] atau *(X +n-1) atau *(P+n-1)

Contoh :

```
#include <stdio.h>
main()
{
    int *P, X[7] = {10,75,50,12,15,55,25};
    int I;

    P = X;

    printf("Elemen      Alamat                      Nilai\n");
    printf("-----\n");

    for(I=0; I<7; I++)
        printf("%4d P+%1d=%p;X+%1d=%p;&X[%1d]=%p X[%d]=%2d; \"\
          \"*(P+%1d)=%2d;*(X+%1d)=%2d\n\", I,I,P+I,I,X+I,I,\
            &X[I], I, X[I], I, *(P+I), I, *(X+I));
}
```

Jika program ini dijalankan, akan didapatkan hasil

Elemen	Alamat	Nilai
0	P+0=FFCE;X+0=FFCE; &X[0]=FFCE	X[0]=10;*(P+0)=10;*(X+0)=10
1	P+1=FFD0;X+1= FFD0; &X[1]=FFD0	X[1]=75;*(P+1)=75;*(X+1)=75
2	P+2=FFD2;X+2= FFD2; &X[2]=FFD2	X[2]=50;*(P+2)=50;*(X+2)=50
3	P+3=FFD4;X+3= FFD4; &X[3]=FFD4	X[3]=12;*(P+3)=12;*(X+3)=12
4	P+4=FFD6;X+4= FFD6; &X[4]=FFD6	X[4]=15;*(P+4)=15;*(X+4)=15
5	P+5=FFD8;X+5= FFD8; &X[5]=FFD8	X[5]=55;*(P+5)=55;*(X+5)=55
6	P+6=FFDA;X+6=FFDA; &X[6]=FFDA	X[6]=25;*(P+6)=25;*(X+6)=25