

PRAKTIKUM

POINTER

Pointer adalah suatu variabel penunjuk yang menunjuk pada suatu alamat memori komputer tertentu. Pointer merupakan variabel level rendah yang dapat digunakan untuk menunjuk nilai integer, character, float, double, atau single, dan bahkan tipe-tipe data lain yang didukung oleh bahasa C. Variabel biasa, sifatnya statis dan sudah pasti, sedangkan pada pointer sifatnya dinamis dan dapat lebih fleksibel. Variabel pointer yang tidak menunjuk pada nilai apapun berarti memiliki nilai NULL, dan disebut sebagai dangling pointer karena nilainya tidak diinisialisasi dan tidak dapat diprediksi.

Pendeklarasian variabel pointer menggunakan tanda * sebelum nama variabelnya, sedangkan untuk menampilkan nilai yang ditunjuk oleh suatu variabel pointer, juga digunakan operator * (tanda asterisk). Jika diinginkan untuk menampilkan alamat tempat penyimpanan nilai yang ditunjuk oleh suatu variabel pointer, digunakan operator & (tanda ampersand).

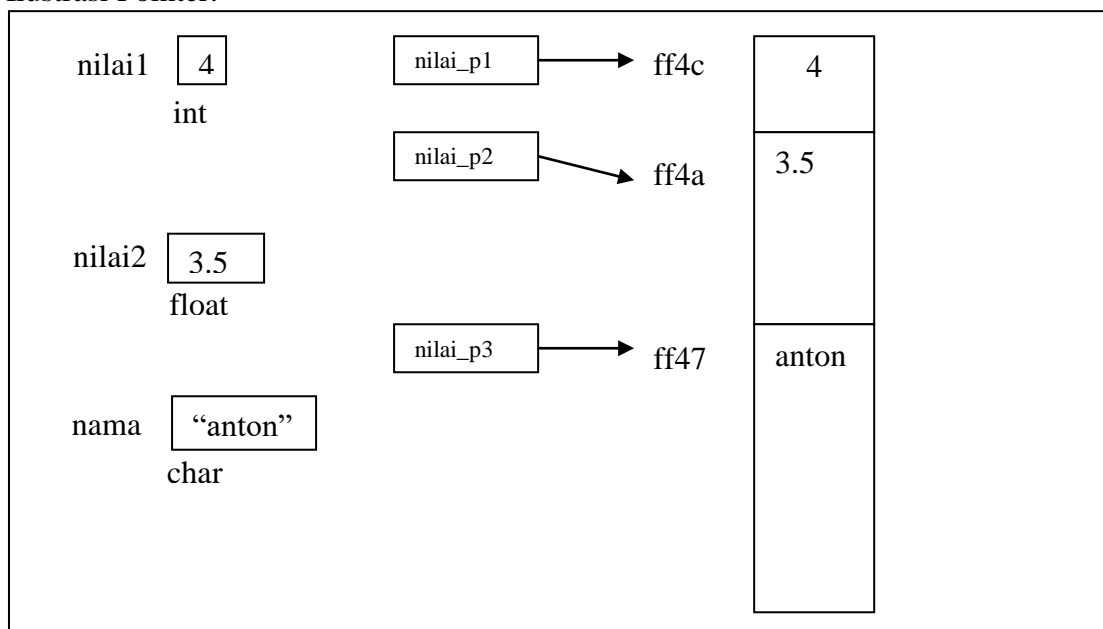
Pada suatu tipe data array, variabel pointer hanya perlu menunjuk pada nama variabel arraynya saja tanpa perlu menggunakan tanda ampersand, atau menunjuk pada nama variabel array pada indeks yang ke nol nya. Untuk lebih jelasnya, silahkan lihat contoh berikut:

Pendeklarasian variabel biasa dan pointer:

```
//variabel biasa
int nilai1 = 4;
float nilai2 = 3.5;
char nama[10] = "anton"; //array of char (string)

//variabel pointer
int *nilai_p1; //dangling pointer
int *nilai_p2 = &nilai1; //menunjuk ke tipe data int
char *nilai_p3 = nama; //menunjuk ke tipe data array of char
char *nilai_p4 = &nama[0];
```

Ilustrasi Pointer:



Contoh program untuk dicoba:

```
#include <stdio.h>
#include <conio.h>

int main(){
    int nilai1 = 4;
    int nilai2 = 5;
    float nilai3 = 3.5;
    char nama[11] = "abcdefghij";

    int *nilai_p1 = &nilai1;
    int *nilai_p2 = &nilai2;
    char *nilai_p4 = nama;
    float *nilai_p3= &nilai3;

    printf("nilai 1 = %d, alamat1 = %p, ukuran
%d\n",*nilai_p1,&nilai_p1,sizeof(nilai1));
    printf("nilai 2 = %d, alamat2 = %p, ukuran
%d\n",*nilai_p2,&nilai_p2,sizeof(nilai2));
    printf("nilai 3 = %f, alamat3 = %p, ukuran
%d\n",*nilai_p3,&nilai_p3,sizeof(nilai3));
    printf("nilai 4 = %s, alamat4 = %p, ukuran
%d\n",nama,&nilai_p4,sizeof(nama));
    getch();
    return 0;
}
```

Operasi pointer:

1. Operasi Pemberian nilai

Contoh 1:

```
#include <stdio.h>
#include <conio.h>

int main(){
    float nilai,*p1,*p2;
    nilai = 14.54;
    printf("nilai = %2.2f, alamatnya %p\n",nilai,&nilai);
    p1 = &nilai;
    printf("nilai p1 = %2.2f, p1 menunjuk alamat
%p\n",*p1,p1);
    //pada awalnya p2 masih dangling pointer
    printf("mula-mula nilai p2 = %2.2f, p2 menunjuk alamat
%p\n",*p2,p2);
    p2 = p1;    //operasi pemberian nilai, berarti alamat x2
sama dengan x1
    printf("sekarang nilai p2 = %2.2f, p2 menunjuk alamat
%p\n",*p2,p2);
    getch();
}
```

Contoh 2:

```
#include <stdio.h>
#include <conio.h>

int main(){
    int *p,a=25,b;
    //p masih dangling
    printf("nilai a = %d di alamat a = %p,\n",a,&a);
    printf("nilai p di alamat = %p\n",p);
    p = &a;
    printf("nilai p = %d di alamat %p\n",*p,p);
    //b diisi dengan nilai yang berasal dari nilai
//variabel a yang ditunjuk oleh pointer p
    b = *p;
    printf("nilai b = %d di alamat %p\n",b,&b);
    getch();
}
```

Contoh 3:

```
#include <conio.h>
#include <stdio.h>

int main(){
    int a=25,b=12,t;
    int *p,*q;
    p = &a;
    q = &b;
    printf("nilai yang ditunjuk p = %d di alamat %p\n",*p,p);
    printf("nilai yang ditunjuk q = %d di alamat %p\n",*q,q);
    //Contoh kasus, penukaran nilai 2 variabel dengan pointer

    t = *p;
    *p = *q;
    *q = t;
    printf("nilai yang ditunjuk p sekarang = %d di alamat
%p\n",*p,p);
}
```

```

        printf("nilai yang ditunjuk q sekarang = %d di alamat
%p\n",*q,q);
        getch();
    }

```

Contoh 4:

```

#include <stdio.h>
#include <conio.h>

int main(){
    int a,*p;
    p=&a;
    *p=25;
    printf("nilai a = %d",a);
}

```

2. Operasi Aritmatika

```

#include <stdio.h>
#include <conio.h>

int main(){
    int a,b=10,*p,*q;
    p=&a;
    *p=25;
    printf("nilai a = %d\n",a);
    printf("alamat p = %p\n",p);
    q=&b;
    printf("alamat q = %p\n",q);
    printf("nilai a + b = %d\n",(*p+*q));
    //posisi alamat p menjadi bergeser, nilai berubah
    p=p+1;
    printf("nilai p = %d, alamat = %p\n",*p,&p);
    q=q-1;
    printf("nilai q = %d, alamat = %p\n",*q,&q);
    getch();
}

```

Pointer pada array 1 dimensi:

```

#include <stdio.h>
#include <conio.h>

int main(){
    char S[] = "anton";
    char *p;
    //cara 1
    //langsung menunjuk nama array.
    p=S;
    for(int i=0;i<5;i++){
        printf("%c",*p);
        p++;
    }
    printf("\n");

    //cara 2
    p=&S[0];
    for(int i=0;i<5;i++){

```

```

        printf("%c", *p);
        p++;
    }
    printf("\n");

    //Membalik kalimat
    p--;
    for(int i=0;i<5;i++){
        printf("%c", *p);
        p--;
    }
    printf("\n");

    getch();
}

```

Pointer pada Struct:

```

#include <stdio.h>
#include <conio.h>

typedef struct{
    int nim;
    int umur;
    float ipk;
} Mahasiswa;

Mahasiswa m;
Mahasiswa *p = &m;

int main(){
    //struct biasa
    m.nim=123;
    m.ipk=3.2;
    m.umur=23;
    printf("nim = %d\n",m.nim);
    printf("ipk = %f\n",m.ipk);
    printf("umur = %d\n",m.umur);

    //struct pointer
    p->ipk = 3.5;
    p->nim = 321;
    p->umur = 32;
    printf("nim = %d\n",p->nim);
    printf("ipk = %f\n",p->ipk);
    printf("umur = %d\n",p->umur);

    //mengacu pada variabel aslinya
    printf("nim = %d\n",m.nim);
    printf("ipk = %f\n",m.ipk);
    printf("umur = %d\n",m.umur);
    getch();
}

```

Pengembangan:

Buatlah sebuah program untuk mengecek apakah suatu kata palindrom atau bukan, tanpa memperhatikan spasi dan huruf besar/kecilnya. Program dibuat dengan menggunakan template struct sebagai berikut:

```
typedef struct{
    char elemen[30];
    int jml_kata;
} Kata;

Kata kata;
Kata *p_kata=&kata;
```

Lanjutkanlah program berikut agar hasilnya sesuai dengan soal di atas:

```
#include <stdio.h>
#include <conio.h>

typedef struct{
    char elemen[30];
    int jml_kata;
} Kata;

Kata kata;
Kata *p_kata=&kata;

int main(){
    char kalimat[30];
    p_kata->jml_kata=0;
    char *p = p_kata->elemen;
    printf("Masukkan kata : ");gets(kalimat);
    fflush(stdin);
    printf("Kalimat : %s\n",kalimat);
    for(int i=0;i<kalimat[i];i++){
        *p=kalimat[i];
        p_kata->jml_kata++;
        p++;
    }

    p=p_kata->elemen;
    //tampilkan kembali kalimat tersebut
    for(int i=0;i<=p_kata->jml_kata;i++){
        printf("%c ",*p);
        p++;
    }

    //kembangkan....

    getch();
}
```

Soal:

Buatlah program data KTP, dengan menggunakan pointer pada struct KTP sebagai berikut:

```
typedef struct
{
    int tgl;
    int bln;
    int th;
}Tanggal;

typedef struct
{
    char noID[5];
    char nama[30];
```

```
        char jenis_kelamin; //'L' atau 'P'  
        Tanggal t;  
}KTP;  
  
typedef struct  
{  
    KTP ktp;  
    int jml;  
}Data_KTP;  
  
Data_KTP data_ktp;  
Data_KTP *p;
```

Buatlah fungsi untuk:

1. Menambah data
2. Mencari data berdasarkan tahun kelahiran tertentu
3. Menampilkan data berdasarkan L dan P
4. Mengedit data

Semua pengaksesan dilakukan dengann menggunakan pointer.