

PRAKTIKUM 13

STRUCT & POINTER

13.1 TUJUAN PRAKTIKUM

Tujuan Umum

Mahasiswa dapat memahami konsep struct dan pointer pada bahasa pemrograman C

Tujuan Khusus

Mahasiswa dapat :

1. Memahami Implementasi struct pada bahasa C
2. Memahami Implementasi pointer pada bahasa C

13.2 PELAKSANAAN PRAKTIKUM

13.2.1 Struct

Array adalah kumpulan elemen yang bertipe sama. Tetapi struct memudahkan Anda untuk mengumpulkan variabel dengan tipe yang berbeda di dalam satu nama. Fasilitas ini memungkinkan kita untuk melayani sekumpulan data yang rumit sebagai satuan tunggal.

Suatu struct dinyatakan dengan kata kunci struct, daftar pernyataan variabel yang disebut anggota, yang terlampir dalam tanda kurung { }. Tiap pernyataan anggota dan struct harus berakhir dengan semicolon (;).

Bentuk deklarasi struct:

```
struct nama_struct {  
    anggota_struct ;  
} ;
```

Apabila suatu struct telah dideklarasikan, struct ini dapat digunakan untuk mendefinisikan suatu variabel, misalnya :

```
nama_struct variabel_struct;
```

Deklarasi diatas adalah format pendefinisian variabel variabel_struct dengan tipe struct nama_struct.

Anggota struct dapat diakses dengan menggunakan bentuk :

```
variabel_struct.anggota_struct;
```

Praktikum 13.1

```

1  #include <stdio.h>
2
3  struct keluarga {
4      char suami[15];
5      char istri[15];
6      int jumlah_anak;
7  };
8
9  main() {
10     keluarga Andi,Budi;
11     printf("Masukan nama istri Andi : ");
12     scanf("%s", &Andi.istri);
13     printf("Masukan jumlah anak Andi : ");
14     scanf("%s", &Andi.jumlah_anak);
15     printf("Masukan nama istri Budi : ");
16     scanf("%s", &Budi.istri);
17     printf("Masukan jumlah anak Budi : ");
18     scanf("%s", &Budi.jumlah_anak);
19     printf("Nama istri Budi : %s", Budi.istri );
20 }

```

13.2.2 Array dari Struct

Array dari struct dapat dideklarasikan seperti halnya pendeklarasian array biasa.

Bentuk deklarasi Array dari Struct :

nama_array[index]. anggota_struct;

Praktikum 13.2

```

1  #include<stdio.h>
2  #include<string.h>
3
4  struct mahasiswa {
5      char nama[20];
6      char nim[10];
7      float kuis;
8      float mid;
9      float uas;
10     float Total_nilai;
11 };
12
13 main() {
14     int i,n;
15     mahasiswa nilai_mhs[10];
16     printf("Masukan banyak mahasiswa : "); scanf("%i", &n);
17     for (i=0;i<n;i++) {
18         printf("Mahasiswa ke-%i adalah ", i+1);
19         scanf("%s", &nilai_mhs[i].nama);
20         printf("NIM adalah ");scanf("%s", &nilai_mhs[i].nim);
21         printf("Nilai kuis = ");scanf("%f", &nilai_mhs[i].kuis);
22         printf("Nilai mid = ");scanf("%f", &nilai_mhs[i].mid);
23         printf("Nilai uas = ");scanf("%f", &nilai_mhs[i].uas);
24         nilai_mhs[i].Total_nilai
25             =0.25* nilai_mhs[i].kuis+0.3* nilai_mhs[i].mid+0.45* nilai_mhs[i].uas;
26     }
27     printf("Daftar Nilai \n");
28     for(i=0;i<n;i++) {
29         printf("%s \t %.0f \n", nilai_mhs[i].nama, nilai_mhs[i].Total_nilai);
30     }
31 }

```

13.2.3 Struct dari Struct

Struct juga dapat menjadi anggota struct lain. Contoh di atas juga salah satu contoh struct sebagai anggota struct. Untuk dapat lebih memahami di bawah ini diberikan sebuah contoh.

Praktikum 13.3

```
1  #include<stdio.h>
2
3  struct anak {
4      char pria[15];
5      char wanita[15];
6  };
7
8  struct keluarga {
9      char suami[15];
10     char istri[15];
11     struct anak data_anak;
12 };
13
14 main() {
15     struct anak Andi = {"Tedi","Lisa"};
16     struct keluarga Budi = {"Budi","Ana","Dika","Ika"};
17     printf("Anak laki-laki Andi adalah %s \n", Andi.pria);
18     printf("Putri bapak Budi adalah %s", Budi.data_anak.wanita);
19 }
```

13.2.4 Pointer

Beberapa tugas pemrograman C dilakukan lebih mudah dengan pointer, dan tugas lain, seperti alokasi memori dinamis, tidak dapat dilakukan tanpa menggunakan pointer. Jadi penting untuk mempelajari pointer untuk menjadi programmer C yang handal.

Seperti yang kita ketahui, setiap variabel adalah lokasi memori dan setiap lokasi memori memiliki alamatnya yang ditentukan yang dapat diakses menggunakan operator ampersand (&), yang menunjukkan alamat dalam memori. Perhatikan contoh berikut, yang mencetak alamat dari variabel yang ditentukan.

Praktikum 13.4

```
1  #include <stdio.h>
2
3  main () {
4      int var1;
5      char var2[10];
6
7      printf("Alamat variabel var1: %x\n", &var1 );
8      printf("Alamat variabel var2: %x", &var2 );
9  }
```

Apa itu Pointer?

Pointer adalah variabel yang nilainya adalah alamat variabel lain, mis., Alamat langsung dari lokasi memori. Seperti variabel atau konstanta apa pun, Anda harus mendeklarasikan pointer sebelum menggunakannya untuk menyimpan alamat variabel apa pun. Bentuk umum dari deklarasi variabel pointer adalah

```
tipe_data *nama_variabel;
```

Praktikum 13.5

```
1  #include <stdio.h>
2
3  int main () {
4      int var = 20;    /* deklarasi dan assign nilai variabel */
5      int *ip;         /* deklarasi variabel pointer*/
6      ip = &var;       /* menyimpan alamat var pada variabel pointer*/
7
8      printf("Alamat variabel var: %x\n", &var );
9      printf("Alamat var yang tersimpan di *ip : %x\n", ip );
10     printf("Nilai dari variabel *ip: %d\n", *ip );
11     printf("Alamat variabel ip: %x\n", &ip );
12 }
```

13.2.5 Pointer sebagai anggota Struct

Anggota dari struct dapat bertipe apapun, karakter, integer, float bahkan pointer dapat pula dijadikan sebagai tipe data anggota struktur

Praktikum 13.6

```
1  #include<stdio.h>
2
3  struct kuliah {
4      char mk1[30];
5      char mk2[30];
6      struct kuliah *ptr;
7  };
8
9  main() {
10     kuliah semester[3]
11     = {
12         {"Komputer 207", "Matematika 217", &semester[1]},
13         {"Electronic 210", "Sistem Kendali 303", &semester[2]},
14         {"Analisis Numerik 301", "Telekomunikasi 367", &semester[0]}
15     };
16     int n;
17     for(n=0; n<3; n++) {
18         printf("Isi sedang menunjuk ke : %s\n", semester[n].ptr->mk1);
19     }
20 }
```