

PRAKTIKUM 13

STRUKTUR 1

A. Tujuan

1. Menjelaskan cara mendeklarasikan struktur
2. Menjelaskan cara menginisialisasi struktur
3. Menjelaskan cara mengakses elemen struktur
4. Menjelaskan pembentukan dan cara mengakses array dari struktur (*array of struct*)

B. DASAR TEORI

Struktur adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama. Berbeda dengan array yang berisi kumpulan variabel-variabel yang bertipe sama dalam satu nama, maka suatu struktur dapat terdiri atas variabel-variabel yang berbeda tipenya dalam satu nama struktur. Struktur biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah.

Variabel-variabel yang membentuk suatu struktur, selanjutnya disebut sebagai elemen dari struktur atau field. Dengan demikian dimungkinkan suatu struktur dapat berisi elemen-elemen data berbeda tipe seperti char, int, float, double, dan lain-lain. Contoh sebuah struktur adalah informasi data tanggal (date) yang berisi:

- **day**
- **month**, dan
- **year**

Mendefinisikan & Mendeklarasikan Struktur

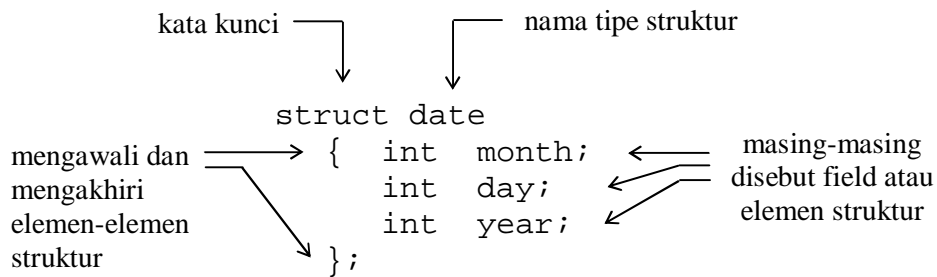
Suatu struktur didefinisikan dengan menggunakan kata kunci struct. Contoh pendefinisian sebuah tipe data struktur :

```
struct date {
    int  month;
    int  day;
    int  year;
};

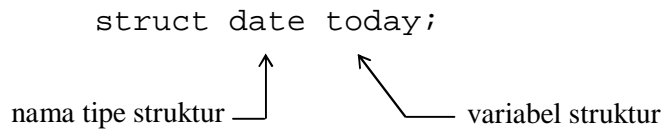
struct date {
    int  month, day, year;
};
```

yang mendefinisikan sebuah tipe data struktur bernama date yang memiliki tiga buah elemen (field) berupa :

- **day**
- **month**
- **year**



Untuk mendeklarasikan sebuah variabel **today** yang bertipe struktur **date** pernyataan yang diperlukan adalah sebagai berikut:



Pernyataan di atas menyatakan bahwa variabel **today** bertipe struktur **date**.

Dalam mendefinisikan sebuah struktur, elemen yang terkandung di dalamnya bisa juga berupa sebuah struktur,

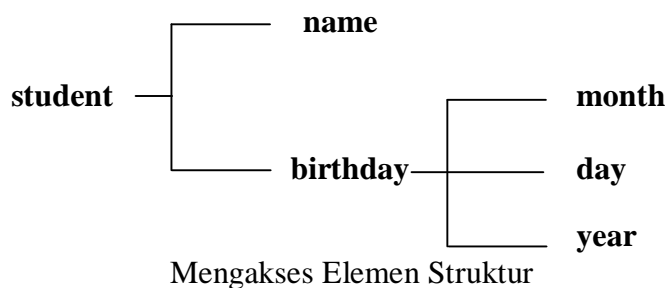
contoh :

```
struct date {
    int month, day, year;
};

struct person {
    char name[30];
    struct date birthday;
};

struct person student; //deklarasi var student
```

Diagram struktur data dari variabel student dapat digambarkan sbb :



Elemen dari suatu variabel struktur dapat diakses dengan menyebutkan nama variabel struktur diikuti dengan operator titik (‘.’) dan nama dari elemen strukturnya. Cara penulisannya sebagai berikut

variabel_struktur.nama_field

Untuk memberikan data nama ke field name dari variabel student di atas, maka pernyataan yang diperlukan misalnya adalah :

```
strcpy(student.name, "MUHAMMAD IHSAN");
```

Pada pernyataan di atas, student.name dapat dibaca sebagai "field name dari student". Contoh berikut merupakan instruksi untuk mengisi data pada field birthday :

```
student.birthday.day = 10;
```

Sedangkan untuk mendapatkan isi suatu field dari variabel struktur, contohnya :

- `tgl = student.birthday.day;`
- `puts(student.name);`

Contoh pertama merupakan instruksi untuk memberikan isi dari field day ke variabel tgl. Sedangkan contoh kedua merupakan instruksi untuk menampilkan isi dari field name.

Menginisialisasi Struktur

Sebuah struktur juga bisa diinisialisasi pada saat dideklarasikan. Hal ini serupa dengan inisialisasi array, yaitu elemen-elemennya dituliskan di dalam sepasang kurung kurawal (‘{ }’) dengan masing-masing dipisahkan dengan koma. Deklarasi struktur didahului dengan kata kunci *static*, contoh

```
static struct zodiak bintang =
    {"Sagitarious", 22, 11, 21, 12};
```

Array dan Struktur

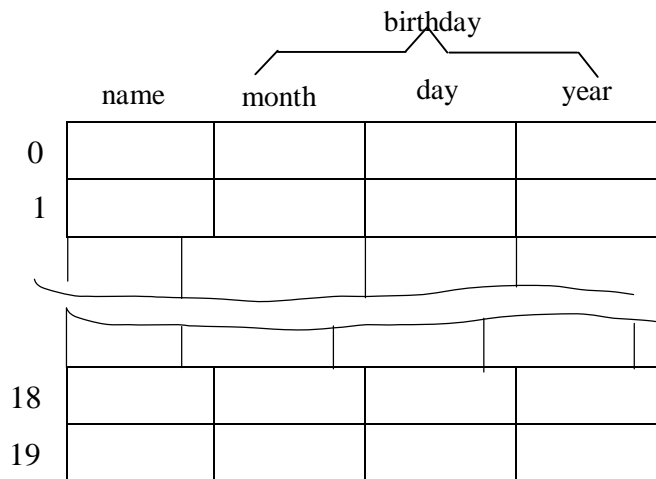
Elemen-elemen dari suatu array juga dapat berbentuk sebuah struktur. Misalnya array yang dipakai untuk menyimpan sejumlah data siswa (struct student). Array struktur berdimensi satu ini membentuk suatu tabel, dengan barisnya menunjukkan elemen dari array-nya dan kolomnya menunjukkan elemen dari struktur. Dalam hal ini maka deklarasi yang dibutuhkan adalah sebagai berikut :

```
#define MAKS 20
.
.
.
struct date {      /* definisi dari tipe date */
    int month;
    int day;
    int year;
};

struct person {   /* definisi dari tipe person */
    char name[30];
    struct date birthday;
};

/* deklarasi dari variabel array student */
struct person student[MAKS];
```

yang artinya, mendeklarasikan array **student** yang memiliki elemen yang bertipe *struct person* sebanyak **MAKS**. Setelah array **student** dideklarasikan, maka ruang yang disediakan ditunjukkan dalam Gambar 13.1.



Gambar 13.1 Array student

Elemen-elemen dari array stuktur tersebut bisa diakses dengan cara sebagai berikut :

```

for (i=0; i<MAKS; i++)
{
    printf("Name          : ");
    fgets(student[i].name, sizeof student[i].name, stdin);
    printf("Birthday (mm-dd-yyyy): ");
    scanf("%d-%d-%d", &student[i].birthday.month,
          &student[i].birthday.day,
          &student[i].birthday.year);
    printf("\n");

    /* hapus sisa data dalam penampung keyboard */
    fflush(stdin);
};

```

Di samping cara pendeklarasian di atas, struktur juga dapat dideklarasikan dalam berbagai bentuk yang lain, di antaranya sbb :

```

struct date {
    int month, day, year;
} today, tomorrow;

struct person {
    char name[30];
    struct date birthday;
} student[MAKS];

```

yaitu mendefinisikan struktur date, sekaligus mendeklarasikan variabel today dan tomorrow dengan tipe struktur date. Demikian juga mendefinisikan struktur person,

sekaligus mendeklarasikan variabel array student sebanyak MAKS elemen dengan tipe struktur person. Atau cara lainnya mendefinisikan, mendeklarasikan sekaligus menginisialisasi struktur, sebagai berikut :

```
struct date {
    int month, day, year;
} today = {5,14,2001};
```

C. TUGAS PENDAHULUAN

1. Definisikan sebuah struktur (misalkan namanya=karyawan) yang memiliki 3 buah *field* berupa sebuah array karakter dengan 10 elemen (misalkan namanya = nama), sebuah array karakter dengan 10 elemen (misalkan namanya = alamat), sebuah long integer (misalkan namanya = gaji).

D. PERCOBAAN

1. Buatlah program untuk menghitung gaji dari seorang karyawan, dengan menggunakan struktur karyawan yang telah dibuat pada tugas pendahuluan. Gaji bersih yang diterima adalah gaji kotor yang dikurangi 15%.

Tampilan :

Masukkan nama karyawan : andi

Masukkan alamat karyawan : Surabaya

Masukkan gaji kotor karyawan : 500000

Gaji bersih yang diterima : 425000

2. Definisikan sebuah struktur (misalkan namanya = data_toko) yang memiliki 3 buah *field* berupa : sebuah array karakter dengan 5 elemen (misalkan namanya = nama), sebuah integer (misalkan namanya = data), dan sebuah *float* (misalkan namanya = harga). Inputkan data & tampilkan hasilnya !

Tampilan :

Masukkan nama barang : sabun

Masukkan jumlah barang : 5

Masukkan harga per barang : 20000

Jadi barang yang dimasukkan adalah sabun, jumlah barang 5 dan total harganya adalah 100000

3. Buat program untuk menghitung luas segitiga, dengan variabel-variabelnya didefinisikan dalam sebuah struktur (ada 3 field yaitu integer alas; integer tinggi & float luas).

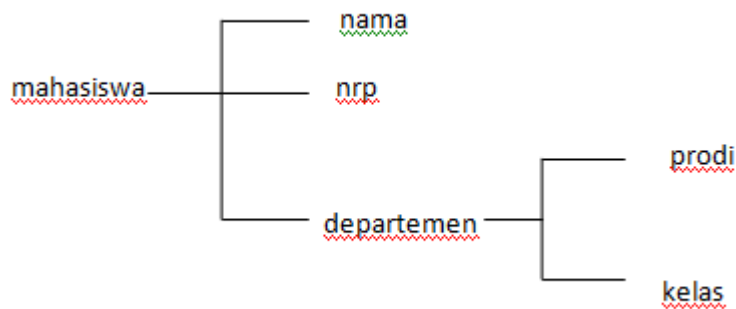
Tampilan :

Masukkan alas : 10

Masukkan tinggi : 5

- Jadi luasnya adalah : 25

4. Buatlah definisi struktur dan programnya dari diagram struktur data berikut ini :



Lakukan pengisian langsung pada field yang ada, sehingga didapat tampilan seperti dibawah ini :

Tampilan :

Nama mahasiswa andri nrp 1203, prodi teknik telekomunikasi kelas 2

E. LAPORAN RESMI

1. Menampilkan zodiak dari tanggal lahir yang dimasukkan

```
/* File program : zodiak1.c */
#include <stdio.h>

main()
{
    struct zodiak {
```

```

    char nama[11];
    int tgl_awal;
    int bln_awal;
    int tgl_akhir;
    int bln_akhir;
};

struct zodiak bintang =
{"Sagitararius", 22, 11, 21, 12};

int tgl_lhr, bln_lhr, thn_lhr;

printf("Masukkan tgl lahir Anda (tgl/bln/thn) : ");
scanf("%d/%d/%d",&tgl_lhr, &bln_lhr, &thn_lhr);

if((tgl_lhr >= bintang.tgl_awal && bln_lhr ==
    bintang.bln_awal) || (tgl_lhr <= bintang.tgl_akhir &&
    bln_lhr == bintang.bln_akhir))
    printf("Bintang Anda adalah %s\n", bintang.nama);
else
    printf("Bintang Anda bukan %s\n", bintang.nama);
}

```

Kembangkanlah program **zodiak1.c** di atas sehingga dapat dipakai untuk menentukan bintang sesuai tanggal kelahiran. Sebagai acuan, gunakan data berikut :

- Aries : 21 Maret – 19 April
- Taurus : 20 April – 20 Mei
- Gemini : 21 Mei – 20 Juni
- Cancer : 21 Juni – 22 Juli
- Leo : 23 Juli – 22 Agustus
- Virgo : 23 Agustus – 22 September
- Libra : 23 September – 22 Oktober
- Scorpio : 23 Oktober – 21 November
- Sagitararius : 22 November – 21 Desember
- Capricorn : 22 Desember – 19 Januari
- Aquarius : 20 Januari – 18 Februari
- Pisces : 19 Februari – 20 Maret

Gunakan array untuk mengimplementasikan programnya.