

STRUKTUR



SASARAN

Setelah menyelesaikan bab ini,
anda diharapkan dapat:

- Menjelaskan cara mendeklarasikan struktur
- Menjelaskan cara menginisialisasi struktur
- Menjelaskan cara mengakses elemen struktur
- Menjelaskan pembentukan dan cara mengakses array dari struktur (*array of struct*)



Dasar Struktur

- Struktur adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama.
- 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 kesatuan
- Variabel-variabel yang membentuk suatu struktur, selanjutnya disebut sebagai elemen dari struktur atau *field*.
- Suatu struktur dapat berisi elemen-elemen data berbeda tipe seperti *char*, *int*, *float*, *double*, dan lain-lain



Deklarasi Struktur

- Pendefinisian tipe data struktur

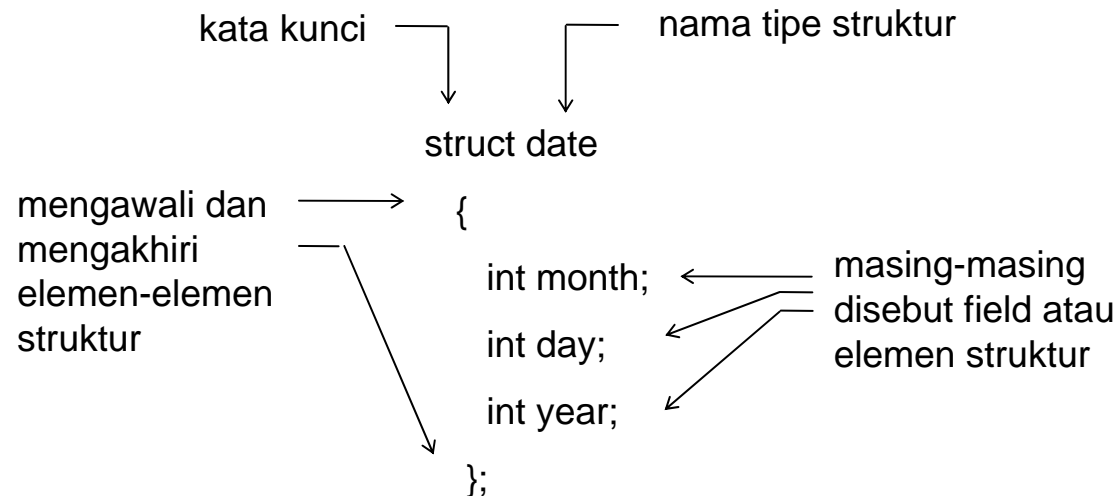
```
struct date {  
    int  month;  
    int  day;  
    int  year;  
};
```

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

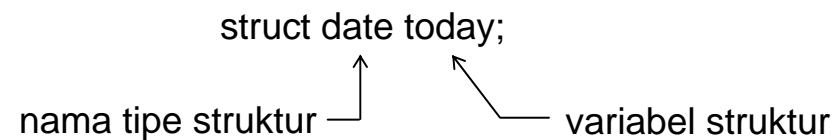
- **day**
- **month**
- **year**

Deklarasi Struktur – Cont. 1

Pendefinisian Tipe Struktur :



Pendeklarasian variabel bertipe struktur



Deklarasi Struktur – Cont.2

Contoh Pendefinisian & Pendeklarasian Struktur

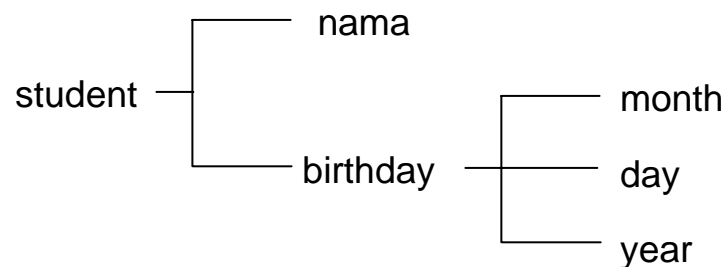
```
struct date {  
    int month, day, year;  
};
```

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

```
struct person student;
```

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

Diagram struktur data dari variabel **student**





Mengakses Elemen Struktur

Cara Penulisan :

```
variabel_struktur.nama_field
```


Contoh :

```
strcpy(student.name, "MUHAMMAD FARUQ");  
student.birthday.day = 10;
```

- Untuk mendapatkan isi suatu *field* dari variabel struktur

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

Contoh Program Struktur



```
#include <stdio.h>
#include <string.h>
struct date {                /* definisi global dari tipe date */
    int month;
    int day;
    int year;
};
struct person {              /* definisi global dari tipe person */
    char name[30];
    struct date birthday;
};
/* deklarasi global dari variabel student*/
struct person student;
main() {
    /* memberikan nilai kepada field dari struktur student */
    strcpy(student.name, "MUHAMMAD FARUQ");
    student.birthday.month = 8;
    student.birthday.day = 10;
    student.birthday.year = 1970;
    /* menampilkan isi semua field dari struktur student */
    printf("Name          : %s\n", student.name);
    printf("Birthday : %d-%d-%d\n", student.birthday.month,
           student.birthday.day, student.birthday.year );
}
```

Name : MUHAMMAD FARUQ Birthday : 8-10-1970

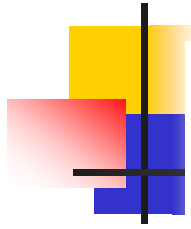


Inisialisasi Struktur

- Struktur juga bisa diinisialisasi pada saat dideklarasikan
 - Serupa dengan inisialisasi array, yaitu elemen-elemennya dituliskan di dalam sepasang kurung kurawal ('{ }') dengan masing-masing dipisahkan dengan koma.
- Contoh inisialisasi struktur :

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

Contoh Program Inisialisasi Struktur



```
/* File program : zodiak.c
Menentukan zodiak berdasarkan data tanggal lahir masukan*/
#include <stdio.h>
main() {
    struct zodiak {
        char nama[11];
        int tgl_awal;
        int bln_awal;
        int tgl_akhir;
        int bln_akhir;
    };
    static struct zodiak bintang = {"Sagitarium", 22, 11, 21, 12};
    int tgl_lhr, bln_lhr, thn_lhr;
    printf("Masukkan tgl lahir Anda (XX-XX-XXXX): ");
        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);
}
```

Masukkan tgl lahir Anda (XX-XX-XXXX): 23-11-1972
Bintang Anda adalah Sagitarium