

# Bab 14. Struktur 1

Konsep Pemrograman  
Politeknik Elektronika Negeri Surabaya  
2006

# Overview

- Pendahuluan
- Mendefinisikan Struktur
- Mendeklarasikan Struktur
- Mengakses Elemen Struktur
- Menginisialisasi Struktur
- Array dan Struktur (*array of struct*)

# Pendahuluan

- Tipe Data ada 2 golongan
  - Tipe data standart (*standart data type*)
    - Tipe data yang telah didefinisikan oleh compiler, tinggal dipakai/dideklarasikan ketika dibutuhkan
    - *int, float, char, double, void*
  - Tipe data yang didefinisikan user (*user defined data type*)
    - Tipe data yang didefinisikan sendiri oleh user sesuai dengan kebutuhannya
    - Langkah pertama adalah mendefinisikan/membuat tipe data yang diinginkan terlebih dahulu, barulah mendeklarasikan variabel dengan tipe baru tsb
    - *struct, union*

# Pendahuluan

- Struktur (*struct*) adalah
  - Sekumpulan data
  - Tipe datanya BOLEH berbeda
  - Menggunakan nama yang sama
  - Dibedakan melalui nama field-nya
- Struktur biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah kesatuan (dalam bahasa PASCAL, struktur disebut dengan *record*)
- Variabel-variabel yang membentuk suatu struktur, selanjutnya disebut sebagai elemen dari struktur atau *field*.
- Dengan demikian dimungkinkan suatu struktur berisi elemen-elemen data berbeda tipe seperti *char*, *int*, *float*, *double*, dan lain-lain.
- Contoh sebuah struktur adalah informasi data tanggal (**date**) yang berisi :
  - **month**
  - **day**
  - **year**

# Mendefinisikan Struktur

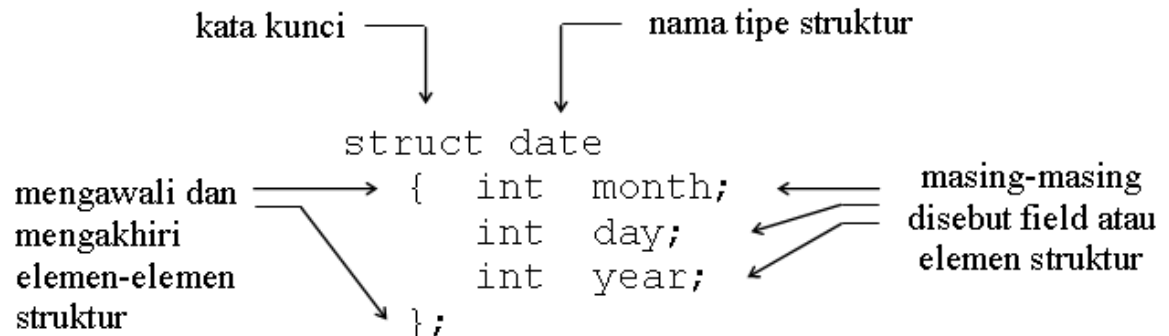
- Suatu struktur didefinisikan dengan menggunakan kata kunci *struct*.
- Dua Contoh pendefinisian sebuah tipe data struktur :

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

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

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

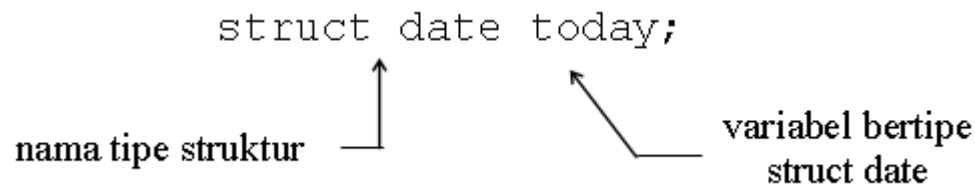
- month
- day
- year



# Mendeklarasikan Struktur

- Untuk mendeklarasikan sebuah variabel `today` yang bertipe `struct date` pernyataannya sbb:

```
struct date today;
```



nama tipe struktur      variabel bertipe  
                                 struct date

- Pernyataan di atas menyatakan bahwa variabel `today` bertipe `struct date`.

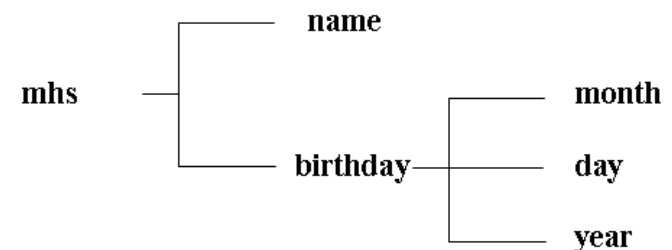
# Mendefinisikan & Mendeklarasikan Struktur

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

```
struct date {
    int  month, day, year;
};
struct student {
    char name[30];
    struct date birthday;
};
struct student mhs;
```

//deklarasi var mhs

- Diagram struktur data dari variabel mhs dapat digambarkan sbb :



# Mengakses Elemen Struktur

- 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`
- Misalnya, memberikan data nama ke *field* name dari variabel `student` di atas, maka pernyataan yang diperlukan misalnya adalah :

```
strcpy(mhs.name, "Budi" );
```

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

```
mhs.birthday.day = 10;
```



# Mengakses Elemen Struktur

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

```
tgl = mhs.birthday.day;
```

merupakan instruksi untuk memberikan isi dari *field* day ke variabel tgl

```
puts (mhs.name) ;
```

merupakan instruksi untuk menampilkan di layar 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.
- Contoh untuk variabel mhs di atas :  

```
struct student mhs =  
    { "Muhammad" , 2 , 28 , 1970 } ;
```

# Array dan Struktur (*array of struct*)

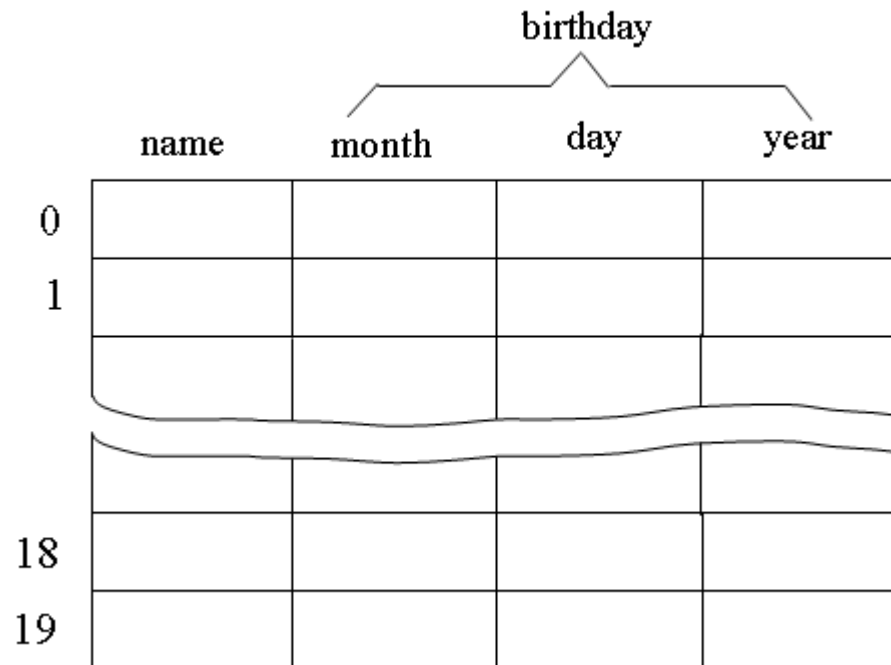
- Elemen-elemen dari suatu array juga dapat berbentuk sebuah struktur (*array of struct*), misalnya array yang dipakai untuk menyimpan sejumlah data siswa (*struct student*).
- Array struktur berdimensi satu ini membentuk suatu tabel, dengan barisnya menunjukkan elemen dari array-nya dan kolomnya menunjukkan elemen/field dari struktur.

```
#define MAKS 2
struct date {                                //definisi dari tipe struct date
    int month, day, year;
};
struct student {                             //definisi dari tipe struct student
    char name[5];
    struct date birthday;
};
struct student data_mhs[MAKS];               //deklarasi variabel array mhs
```

yang artinya, mendeklarasikan array `data_mhs` yang memiliki elemen yang bertipe `struct student` sebanyak `MAKS`.

# Array dan Struktur (*array of struct*)

- Setelah array `data_mhs` dideklarasikan, maka ruang yang disediakan ditunjukkan dalam gambar di bawah ini.



# Array dan Struktur

```
#include <stdio.h>
#define MAKS 20
struct date {                                //definisi global dr tipe date
    int month, day, year;
};
struct student {                             //definisi global dr tipe student
    char name[30];
    struct date birthday;
};

main() {
    struct student data_mhs[MAKS];
    int i=0, jml;
    char lagi;

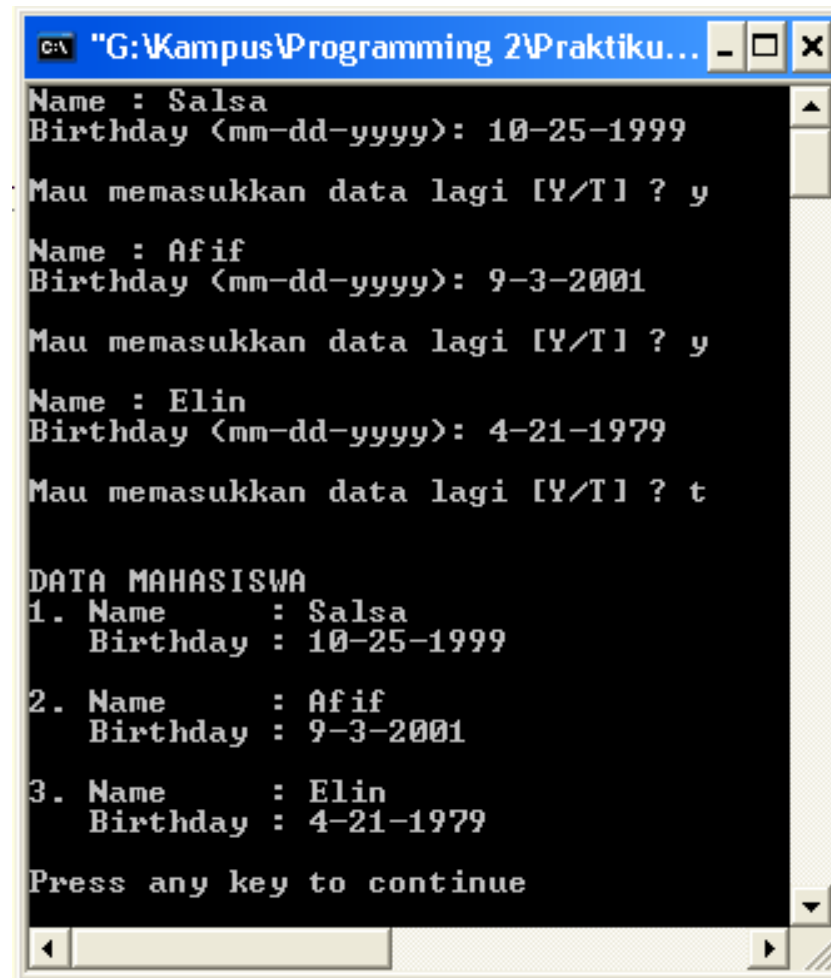
    do                                        //memasukkan data
    {
        printf("Name : ");
        gets(data_mhs[i].name);
        printf("Birthday (mm-dd-yyyy): ");
        scanf("%d-%d-%d", &data_mhs[i].birthday.month,
            &data_mhs[i].birthday.day, &data_mhs[i].birthday.year);
        i++;
    }
```

# Array dan Struktur

```
fflush(stdin); //hapus sisa data dlm buffer keyboard
printf("\nMau memasukkan data lagi [Y/T] ? ");
lagi = getchar( ); //baca tombol
fflush(stdin); //hapus sisa data dlm buffer keyboard
} while(lagi == 'Y' || lagi == 'y');

jml = i;
printf("\nDATA MAHASISWA\n");
for (i=0; i<jml; i++) //menampilkan data
{
    printf("%d. Name : %s", i+1, data_mhs[i].name);
    printf("\n Birthday : %d-%d-%d\n\n",
        data_mhs[i].birthday.month,
        data_mhs[i].birthday.day,
        data_mhs[i].birthday.year);
}
}
```

# Hasilnya....



```
C:\ "G:\Kampus\Programming 2\Praktiku..."
Name : Salsa
Birthday <mm-dd-yyyy>: 10-25-1999
Mau memasukkan data lagi [Y/T] ? y
Name : Afif
Birthday <mm-dd-yyyy>: 9-3-2001
Mau memasukkan data lagi [Y/T] ? y
Name : Elin
Birthday <mm-dd-yyyy>: 4-21-1979
Mau memasukkan data lagi [Y/T] ? t

DATA MAHASISWA
1. Name      : Salsa
   Birthday  : 10-25-1999
2. Name      : Afif
   Birthday  : 9-3-2001
3. Name      : Elin
   Birthday  : 4-21-1979

Press any key to continue
```

# Variasi Pendeklarasian Struktur

- Selain cara yang sudah dibahas sebelumnya, struktur juga dapat dideklarasikan dalam berbagai bentuk yang lain sbb:

```
struct date {  
    int month, day, year;  
} today, tomorrow;  
struct student {  
    char name[30];  
    struct date birthday;  
} data_mhs[MAKS];
```

yaitu mendefinisikan `struct date`, sekaligus mendeklarasikan variabel `today` dan `tomorrow` dengan tipe `struct date`.

Demikian juga mendefinisikan `struct student`, sekaligus mendeklarasikan variabel array `data_mhs` sebanyak `MAKS` elemen dengan tipe `struct student`.

- Cara lain : mendefinisikan, mendeklarasikan sekaligus menginisialisasi struktur sbb:

```
struct date {  
    int month, day, year;  
} today = {12, 25, 2006};
```