

# Konsep Pemrograman

## 13. Struktur - 1

Umi Sa'adah

Entin Martiana Kusumaningtyas

Tri Hadiah Muliawati

2020



Politeknik Elektronika Negeri Surabaya  
Departemen Teknik Informatika dan Komputer

# Overview

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

# Pendahuluan

- Tipe Data dapat dibagi ke dalam 2 golongan:
  1. Tipe data standar (*standart data type*)
    - Tipe data yang telah didefinisikan oleh compiler, tipe data tersebut bisa langsung dideklarasikan ketika dibutuhkan
    - Contoh: int, float, char, double
  2. Tipe data yang didefinisikan user (*user defined data type*)
    - Tipe data yang didefinisikan sendiri oleh user sesuai dengan kebutuhannya
    - Sebelum melakukan deklarasi, tipe data tersebut perlu didefinisikan terlebih dahulu
    - Contoh: struct

# Pendahuluan

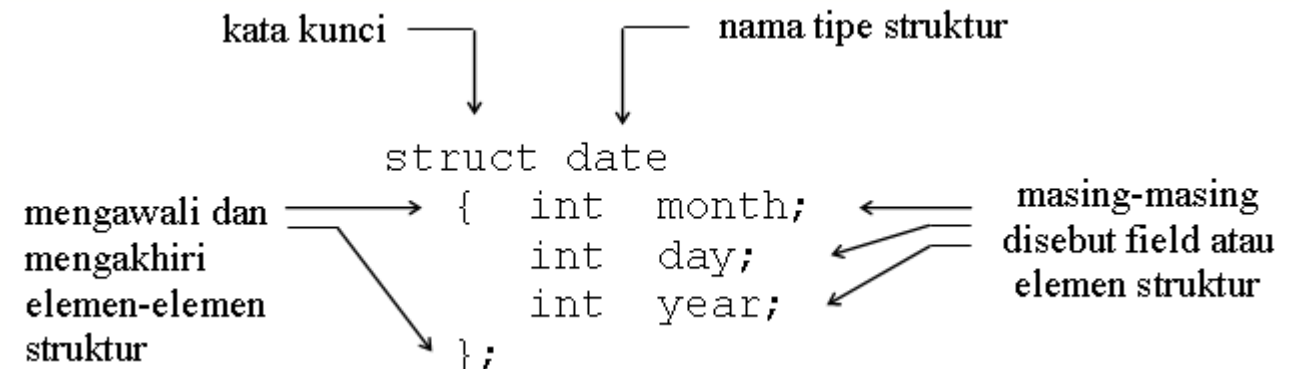
- Struktur (`struct`) biasa digunakan untuk mengelompokkan beberapa variabel yang berkaitan menjadi sebuah kesatuan
- Variabel-variabel yang membentuk `struct` disebut sebagai elemen dari `struct` atau `field`.
- Berikut merupakan karakteristik dari tipe data `struct`:
  - Terdiri atas sekumpulan variabel penyusun (`field`)
  - Tipe data yang digunakan oleh masing-masing `field` BOLEH berbeda
  - Tipe data yang digunakan oleh `field` bisa berupa tipe data standar (misal: `int`, `char`, `double`, `float`) maupun tipe data yang didefinisikan oleh user (misal: `struct`)
  - Pengaksesan nilai dari `field struct`, dapat dilakukan dengan menggunakan nama `field`-nya

# Mendefinisikan `struct`

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

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

- Potongan kode di atas berisi perintah untuk mendefinisikan sebuah tipe data struktur bernama `struct date` yang memiliki tiga buah elemen (field) penyusun berupa :
  - `month` dengan tipe data `int`
  - `day` dengan tipe data `int`
  - `year` dengan tipe data `int`



# Mendeklarasikan `struct`

- Untuk mendeklarasikan sebuah variabel `today` yang bertipe `struct date`, berikut pernyataan yang bisa digunakan:

```
struct date today;
```

nama tipe struktur      variabel bertipe  
                                 struct date

- Pernyataan di atas menyatakan bahwa variabel `today` memiliki tipe data `struct date`.

# Mendefinisikan dan Mendeklarasikan `struct`

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

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

Pendefinisian tipe data `struct date`.

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

Pendefinisian tipe data `struct student`.

```
struct student mhs;
```

# Mendefinisikan dan Mendeklarasikan `struct`

- 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;
```

Deklarasi variabel `birthday` dengan tipe data `struct date`.

```
};
```

```
struct student mhs;
```

Deklarasi variabel `mhs` dengan tipe data `struct student`.



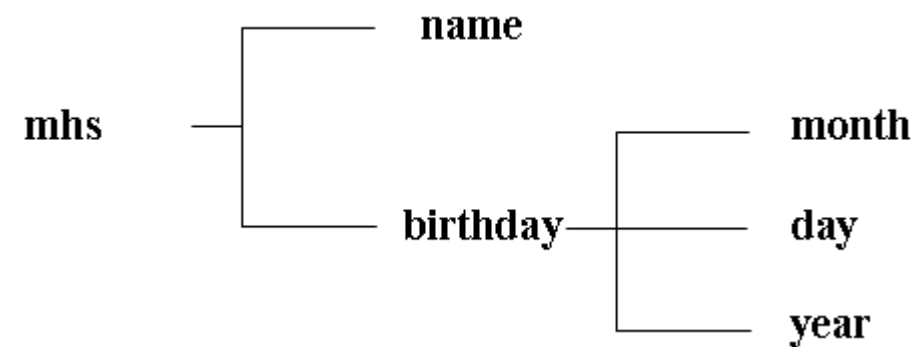


# Mendefinisikan dan Mendeklarasikan `struct`

- Contoh :

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

Diagram struktur data dari variabel `mhs`



# Mengakses elemen (field) pada `struct`

- Elemen dari `struct` dapat diakses dengan menyebutkan nama variabel `struct` diikuti dengan operator titik (.) dan nama dari elemen (field) strukturnya.
- Contoh :

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

- Misalnya, memberikan data tanggal, bulan, dan tahun ke field `day`, `month`, dan `year` dari variabel `today` di atas, maka pernyataan yang diperlukan adalah :

```
today.day = 12;  
today.month = 10;  
today.year = 2020;
```

- Pada pernyataan di atas, `today.day` dapat dibaca sebagai "field `day` dari variabel `today`".

# Mengakses elemen (field) pada struct

- Contoh :

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

- Apabila user ingin memberi nilai pada field month, day, dan year yang dimiliki oleh variabel birthday, maka pernyataan yang diperlukan adalah:

```
mhs.birthday.day = 12;  
mhs.birthday.month = 1;  
mhs.birthday.year = 2000;
```

# Mengakses elemen (field) pada `struct`

- Cara yang sama juga bisa digunakan untuk mendapatkan nilai dari suatu field yang dimiliki oleh variabel struktur.
- Contoh:

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

Pernyataan di atas merupakan instruksi untuk memberikan isi dari field `day` ke variabel `tgl`

```
printf("%d-%d-%d", mhs.birthday.day, mhs.birthday.month,  
mhs.birthday.year);
```

Pernyataan di atas merupakan instruksi untuk menampilkan isi dari field `day`, `month`, dan `year` secara berurutan ke layar

# Menginisialisasi `struct`

- Sebuah `struct` 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` dengan tipe data `struct student`, inisialisasi dapat dilakukan dengan menggunakan pernyataan berikut:

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

Urutan pengisian nilai  
sesuai dengan urutan field  
pada definisi `struct`

Mengisi field `name`

Mengisi field `month`

Mengisi field `day`

Mengisi field `year`

# 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 `struct`.
- Contoh:

```
struct date {  
    int month, day, year;  
};  
struct student {  
    char name[30];  
    struct date birthday;  
};  
struct student data_mhs[20];
```

Deklarasi variabel `data_mhs` dengan tipe data array of struct `student` dengan kapasitas maksimal 20 data.



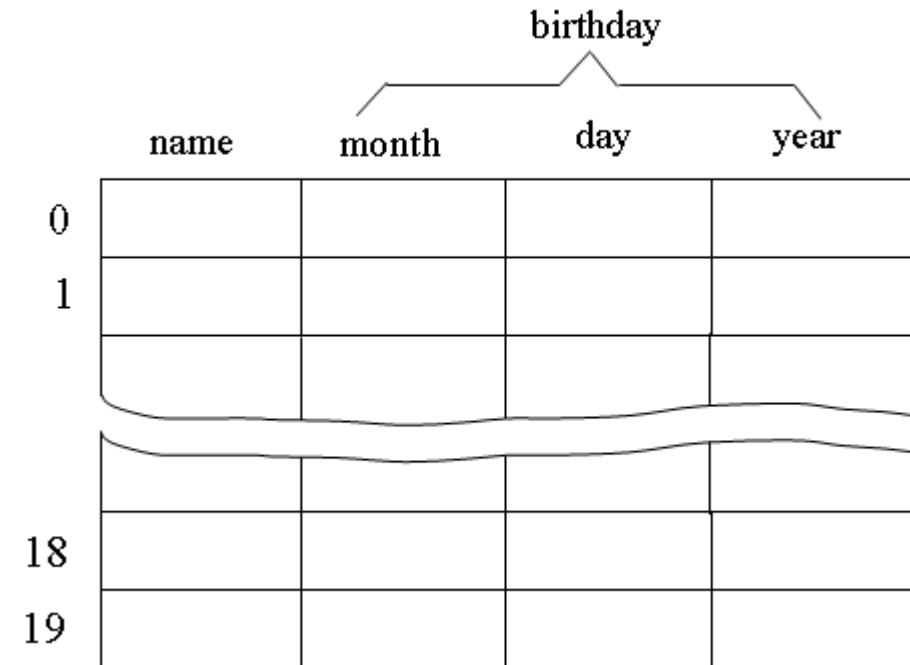
# array of struct

- Contoh:

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

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

struct student data_mhs[20];
```



Ilustrasi ruang penyimpanan data variabel data\_mhs

# Contoh array of struct

```

1  #include <stdio.h>
2  #define MAKS 20
3  struct date { //definisi struct date
4      int month, day, year;
5  };
6  struct student { //definisi struct student
7      char name[30];
8      struct date birthday;
9  };

```

```

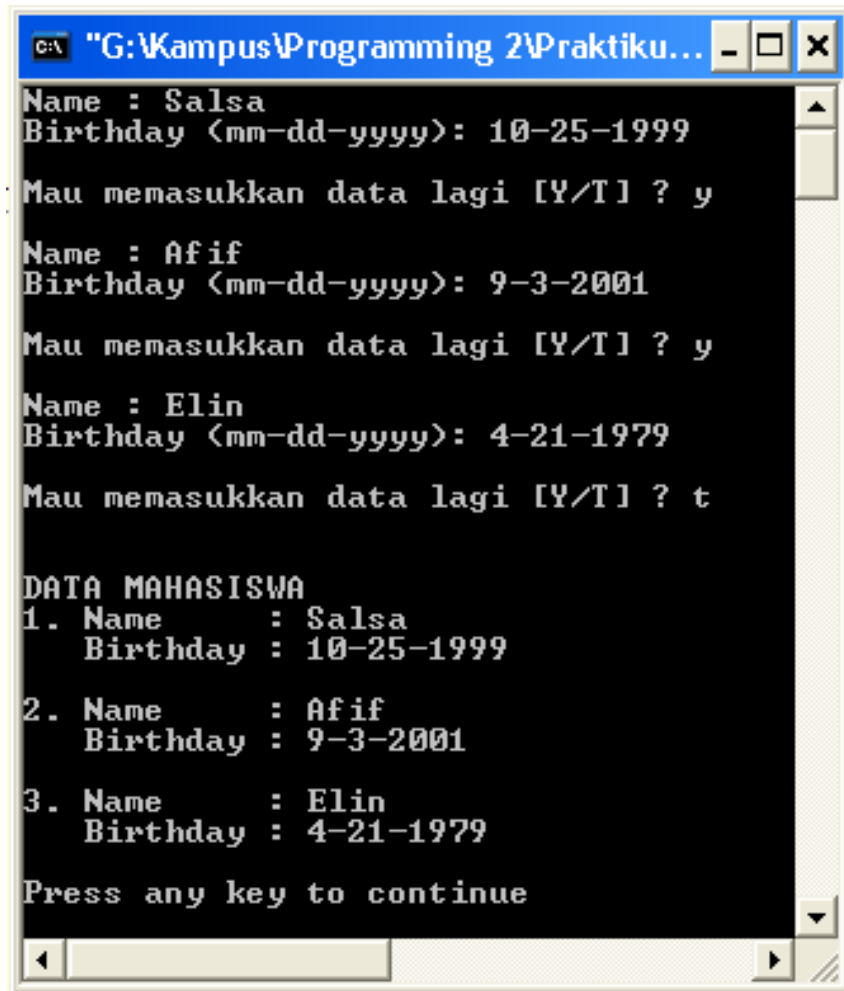
10 main() {
11     struct student data_mhs[MAKS];
12     int i=0, jml;
13     char lagi;
14     do
15     {
16         printf("Name : ");
17         gets(data_mhs[i].name);
18         printf("Birthday (mm-dd-yyyy): ");
19         scanf("%d-%d-%d", &data_mhs[i].birthday.month,
20             &data_mhs[i].birthday.day, &data_mhs[i].birthday.year);
21         i++;
22         fflush(stdin); //hapus sisa data dlm buffer keyboard
23         printf("\nMau memasukkan data lagi [Y/T] ? ");
24         lagi = getchar(); //baca tombol
25         fflush(stdin); //hapus sisa data dlm buffer keyboard
26     } while(lagi == 'Y' || lagi == 'y');
27     jml = i;
28     printf("\nDATA MAHASISWA\n");
29     for (i=0; i<jml; i++)
30     {
31         printf("%d. Name : %s", i+1, data_mhs[i].name);
32         printf("\n Birthday : %d-%d-%d\n\n",
33             data_mhs[i].birthday.month,
34             data_mhs[i].birthday.day,
35             data_mhs[i].birthday.year);
36     }
37     return 0;
38 }

```





# Contoh array of struct



```
"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
```

# Mendefinisikan dan Mendeklarasikan `struct`

- Definisi dan deklarasi `struct` juga bisa dilakukan secara bersamaan, sbb:

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

- Selain itu, inisialisasi `struct` juga bisa dilakukan bersamaan dengan mendefinisikan dan mendeklarasikan `struct` tersebut:

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



**bridge to the future**

<http://www.eepis-its.edu>