

**Pertemuan 03**

# **Structure (Record)**

# **Learning Outcomes**

**Pada akhir pertemuan ini,  
diharapkan mahasiswa akan  
mampu :**

- mendemonstrasikan Tipe Data Structure.**
- melakukan perhitungan dg SMF.**
- menerapkan Structure pada program aplikasi komputer.**

# Outline Materi

- **Karakteristik**
- **Deklarasi**
- **Akses Data**
- **SMF**
- **Passing Parameter**

# Karakteristik

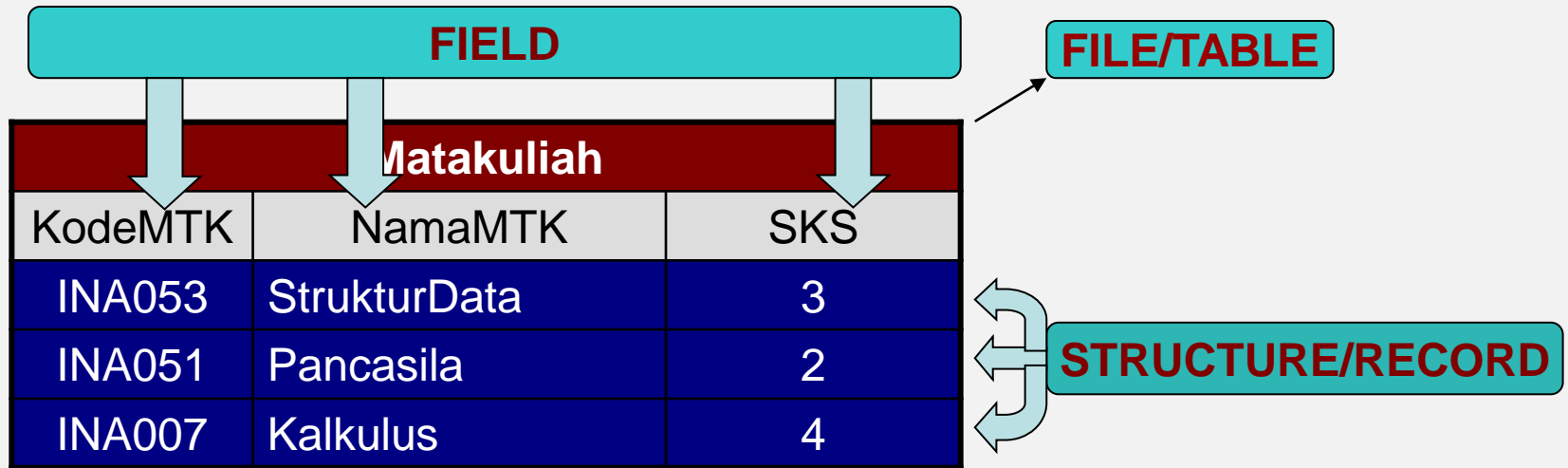
## Karakteristik dari structure :

- Hubungan antara elemen linier
- Penempatan di memori secara fisik maupun secara logik sama.
- Hubungan antara *Field Identifier* dengan Komponen adalah *one-to-one*.
- Tipe data Heterogen
- Cara access Random

## Elemen :

sering disebut dengan field, component, atau member.

# Hubungan file-structure-field



# Deklarasi

BU pada C :

```
struct <struct_name>{  
    <type> <elemen_name1>;  
    <type> <elemen_name2>;  
    ...  
} <structure_variable>;
```

Contoh :

```
struct Matakuliah{    char    KodeMTK[7];  
                    char    NamaMTK[40]  
                    int     SKS; };  
Variabel X dg tipe struct Matakuliah didefinisikan :  
    struct Matakuliah X;
```

```
struct Matakuliah{    char    KodeMTK[6];  
                    char    NamaMTK[40]  
                    int     SKS; } X, Y;
```

```
struct {              char    KodeMTK[6];  
    char    NamaMTK[40]  
    int     SKS; } X;
```

# Akses Data

**BU :**

```
<structure_variabel>.<field_name>  
<structure_pointer> → <field_name>
```

**Contoh :**

```
strcpy(X.KodeMTK,"INA034");  
X.SKS = 2;
```

```
Matakuliah *ptr = &X;  
ptr→SKS = 4;
```

```
Membaca data dari keyboard :  
scanf("%s", X.KodeMTK);  
scanf("%d", &X.SKS);
```

Besar memori yg diperlukan oleh structure  
sama dg jumlah memori yg diperlukan oleh setiap fieldnya.

# Structure Mapping Function


- **Parameter**
  - **Lokasi Basis / Base Location (b)**
  - **Field List**
  - **Field Length**
- **Perhitungan Memory :**  
**Alamat Field = Base location + Offset**



# Contoh SMF

**Jika base location = 500, dan  
component length tipe integer=2.**

```
struct Tgl {  
    int Tanggal;  
    int Bulan;  
    int Tahun;  
};  
  
struct Peg {  
    int NIP;  
    struct Tgl Mulai_Kerja;  
    struct Tgl Berhenti;  
};
```



# Perhitungan Memory Structure

Field List	Type	Length	Offset
NIP	int	2	0
Mulai_Kerja	Tgl	6	2
Tanggal	int	2	2
Bulan	int	2	4
Tahun	Int	2	6
Berhenti	Tgl	6	8
Tanggal	int	2	8
Bulan	int	2	10
Tahun	int	2	12

**Lokasi field Berhenti = base location + offset**  
**= 500 + 8 = 508**

**Lokasi field Berhenti.Tahun = 500 + 12 = 512**

**Memory yg diperlukan per record = 2 + 6 + 6 = 14 Byte**

# Passing Parameter

- **Passing Field ke Fungsi**

**Contoh :**

```
void Cetak( int SKS ){  
    printf("Satuan Kredit Semester =%d\n",SKS); }
```

```
void main(){  
    Matakuliah MTK;  
    MTK.SKs=4;  
    Cetak(MTK.SKs); }
```

- **Passing Structure ke Fungsi**

**Contoh :**

```
void Cetak(struct Matakuliah X){  
    printf("Nama Matakuliah  
    =%s\n",X>NamaMTK);  
    printf("Satuan Kredit Semester          =%d\n",X.SKs); }
```

```
void main(){  
    Matakuliah X;  
    Cetak(X); }
```

# Inisialisasi nilai awal

**Structure dapat diinisialisasi pada saat sebuah variabel didefinisikan.**

**Contoh :**

```
struct StrPoint{
```

```
    int x;
```

```
    int y;
```

```
};
```

```
struct StrPoint Point = { 40, 12};
```

# **PEMANFAATAN DARI STRUCTURE.**

**Contoh :**

```
struct MHS{  
    char NIM[10];  
    char NAMA[25];  
    int SEM;  
    float IPK;  
};
```

```
struct MHS DAF_MHS[20];
```

**Jika : Base Location untuk DAF\_MHS =  
1000**

**Tipe int memerlukan 2 Byte**

**Tipe char butuh 1 Byte**

**Tipe float butuh 4 Byte untuk  
penyimpanannya**

## **Pertanyaan :**

**1. Hitung berapa Byte yang diperlukan untuk menyimpan**

**satu Record MHS ?**

$$\text{Jwb : } 10 + 25 + 2 + 4 = 41 \text{ Byte}$$

**2. Tentukan alamat dari Record dengan indeks 4**

**Base Location + Offset**

$$1000 + 4 * 41 = 1164$$

**3. Tentukan alamat dari Record dengan indeks 9**

$$1000 + 9 * 41 = 1369$$

**4. Tentukan alamat dari Record dengan indeks 14**

$$1000 + 14 * 41 = 1574$$

**5. Tentukan alamat dari Record dengan indeks 17**

$$1000 + 17 * 41 = 1697$$

**6. Tentukan alamat dari DAF\_MHS[4].NAMA**

**Base Location + Offset**

$$1000 + (4 * 41 + 10) = 1174$$

**7. Tentukan alamat dari DAF\_MHS[9].SEM**

$$1000 + (9 * 41 + 10 + 25) = 1404$$

**8. Tentukan alamat dari DAF\_MHS[14].IPK**

$$1000 + (14 * 41 + 10 + 25 + 2) = 1611$$

**9. Tentukan alamat dari DAF\_MHS[17].NIM**

$$1000 + (17 * 41 + 0) = 1697$$

**10. Tentukan alamat dari DAF\_MHS[25].NAMA**

**→ TIDAK PUNYA ALAMAT**

# Selesai