



Asia Cyber
University

Algoritma & Pemrograman 1

Struktur

14

Definisi dan Deklarasi

- **Structure**: Tipe data yang digunakan untuk menampung sekelompok data yang berbeda tipe.
- Komponen struktur disebut anggota, field atau elemen.
- Heterogen.
- Metode akses sekuensial.
- Disebut record di bahasa pemrograman lain.

Definisi dan Deklarasi

Sintaks

```
Struct tag {  
    Anggota 1;  
    Anggota 2;  
    ...  
    Anggota m;  
};  
Struct v1, ..., vn;
```

```
Struct tag {  
    Anggota 1;  
    Anggota 2;  
    ...  
    Anggota m;  
} Struct v1, ..., vn;
```

Struct adalah keyword
Tag adalah identifier structure.

Definisi dan Deklarasi

Contoh :

```
struct rekening {  
    int noRek;  
    char tipeRek;  
    char nama{31};  
    long saldo;  
};  
struct nasabah1, nasabah2;
```

Nested Structure

- ▶ Struktur yang salah satu anggotanya adalah struktur lain.
- ▶ Deklarasi struktur lain dilakukan sebelum deklarasi struktur yang memuatnya.

Contoh :

```
struct tanggal {  
    int tgl, bln, thn;  
};  
struct rekening {  
    int noRek;  
    char tipeRek;  
    char nama{31};  
    long saldo;  
    struct tanggal transAkhir;  
};  
struct rekening nasabah1, nasabah2;
```

Nilai Awal Variabel Struktur

- ▶ Hanya untuk static dan eksternal.
- ▶ Sintaks
 - ▶ Klas_penyimpanan tag variabel = {nilai_1, ..., nilai_m};
- ▶ Contoh
 - ▶ Static struct rekening nasabah = {1947, 'D', "Susi", 2000000, 19, 09, 04};

Array Of Structure

- ▶ Array dengan anggota struktur.
- ▶ Contoh

```
struct tanggal {  
    int tgl, bln, thn;  
};
```

```
struct rekening {  
    int noRek;  
    char tipeRek;  
    char nama{31};  
    long saldo;  
    struct tanggal transAkhir;  
};  
struct rekening nasabah[100];
```

Nilai Awal Array Of Structure

```
struct tanggal {  
    char nama[31];  
    int tgl, bln, thn;  
};  
static struct tanggal TglLahir[ ] = {  
    {"Susi", 9, 7, 1990},  
    {"Susy", 7, 9, 1992},  
    {"Suzy", 9, 9, 2002}  
};
```


Operator Titik

- ▶ Untuk mengakses anggota/subanggota struktur.
- ▶ Sintaks
 - ▶ `Var_structure.Anggota;`
 - ▶ `Var_structure.Anggota.Subanggota;`
- ▶ Contoh
 - ▶ `nasabah1.noRek;`
 - ▶ `ulTah[2].nama;`

Type Def

- ▶ Mendefinisikan tipe data baru yang ekui-valen dengan tipe data yang sudah ada.
- ▶ Sintaks
 - ▶ `Typedef tipe tipe_baru;`
- ▶ Di mana
 - ▶ `Typedef` adalah keyword.
 - ▶ `Tipe` adalah tipe data yang sudah ada .
- ▶ Contoh
 - ▶ `typedef int usia;`
 - ▶ `usia laki_laki, perempuan;`

Type Def

- ▶ Bermanfaat untuk mendefinisikan struktur karena dapat mengurangi pengulangan penulisan *struct tag*.

- ▶ Contoh :

```
typedef struct {  
    int tgl, bln, thn;  
} tanggal;
```

```
typedef struct {  
    int noRek;  
    char tipeRek;  
    char nama[31];  
    float saldo;  
    struct tanggal transAkhir;  
} record;  
record nasabah[100];
```

Pointer Dan Struktur

- ▶ Misalkan `var` adalah variabel tipe struktur maka `&var` adalah alamat variabel tersebut.
- ▶ Deklarasi variabel pointer ke struktur
 - ▶ `Type_of_struct *pt_var;`
- ▶ Arahkan variabel pointer ke struktur
 - ▶ `Pt_var = &var;`

Pointer Dan Struktur

```
typedef struct {  
    int noRek;  
    char tipeRek;  
    char nama[31];  
    long saldo;  
} rekening;  
rekening nasabah *rek;
```

```
struct {  
    int noRek;  
    char tipeRek;  
    char nama[31];  
    long saldo;  
} nasabah, *rek;  
rek = &nasabah;
```

Operator ->

- ▶ Nomor rekening nasabah diakses sbb:
 - (*rek). noRek;
 - rek -> noRek;
 - nasabah.noRek;

Passing Struktur Ke Fungsi

- ▶ Pass anggota struktur secara individual.
- ▶ Pass by value.

```
main() {  
    typedef struct {  
        int tgl, bln, thn;  
    } tanggal;  
    struct {  
        int noRek;  
        char tipeRek;
```

```
        char nama[31];  
        float saldo;  
        tanggal transAkhir;  
    } nasabah;  
    ...  
    nasabah.saldo =      adjust(  
        nasabah,noRek,  
        nasabah.saldo);  
    ...  
}
```



Passing Struktur Ke Fungsi

- ▶ Pass seluruh struktur.
- ▶ Pass by reference

```
typedef struct {  
    char *nama;  
    int noRek;  
    char tipeRek;  
    float saldo;  
} record;
```

```
int adjust(record *pt) {  
    pt->nama = "Susy";  
    pt->noRek = 9999;  
    pt->tipeRek = 'T';  
    pt->saldo = 99.99;  
    return;  
}
```


Passing Struktur Ke Fungsi

```
int main() {  
    static record nasabah = {"Susi", 3333, 'G', 33.33};  
    printf("\n%s %d %c %.2f", nasabah.nama,  
        nasabah.noRek, nasabah.tipeRek, nasabah.saldo);  
    adjust(&nasabah);  
    printf("\n%s %d %c %.2f", nasabah.nama,  
        nasabah.noRek, nasabah.tipeRek, nasabah.saldo);  
}
```

Array Vs Structure

► Array

- Hubungan antar elemen: linear
- Homogen
- Random akses
- Akses elemen dengan index
- Elemen di memori: contiguous

► Structure

- Hubungan antar elemen: linear
- Heterogen
- Sekuensial akses
- Akses elemen dengan field id.
- Elemen di memori: (mungkin) non-contiguous

Terimakasih