





# Struktur Data

Saniati, S.ST., M.T.

**EPISODE 3**

Struct (Structure)



# Masalah pada Array ?

- Array mampu menyimpan data dalam jumlah tertentu namun dengan tipe data yang sama.
- Bagaimana jika dibutuhkan penyimpanan data secara terkelompok dengan tipe data yang berbeda?.
- Kita dapat menggunakan dan membuat tipe data baru yang kita sebut sebagai structure atau biasa disingkat dengan struct.



# Structure atau Struct ?

- Structure adalah kumpulan variabel dengan tipe data yang dapat berbeda-beda dengan satu nama.
- Structure mirip dengan class, bedanya jika class terdiri dari variabel/properti dan method/behavior, sedangkan structure hanya terdiri dari kumpulan variabel/property/data dengan beragam tipe data.
- Konsep dari structure mirip dengan array, perbedaannya hanya pada tipe datanya. Jika array hanya satu tipe data yang sama, sedangkan struct berbeda-beda tipe data.



# Structure, New Data Type ?

- Pada umumnya ketika membuat tempat penyimpanan data, kita akan menggunakan tipe data yang telah terdefinisi seperti int, float, char, bool, dan lainnya.
- Ternyata kita dapat membuat sendiri tipe data yang dibutuhkan.
- Tipe data baru yang didefinisikan sendiri bisa dalam bentuk structure, class, dan lainnya.
- Structure dapat berisi variabel, array, atau bahkan tipe data baru lainnya juga seperti structure.



# Deklarasi Structure ?

```
// Deklarasi Structure
/*
struct structName{
    // komponen / member
    dataTypeMember1 memberName1;
    dataTypeMember2 memberName2;
    dataTypeMember3 memberName3;
    . . . .
};
*/
```



# Deklarasi Var dengan Tipe Data Struct ?

```
// Cara 1 (Global)
/*
struct structName{
    // komponen / member
    dataTypeMember1 memberName1;
    dataTypeMember2 memberName2;
    dataTypeMember3 memberName3;
    . . . . .
}varName1, varName2, ...;
*/
```

```
// Cara 2 (Lokal)
/*
struct structName{
    // komponen / member
    dataTypeMember1 memberName1;
    dataTypeMember2 memberName2;
    dataTypeMember3 memberName3;
    . . . . .
};
*/
int main()
{
    // structName varName1, varName2, ... ;
}
```



# Akses Member Structure ?

- Untuk mengakses member dari structure, dapat digunakan operator akses member berupa period(.)

```
// structName varName1, varName2, ... ;  
  
// varName1.memberName1 = value;  
// cin >> varName1.memberName2;
```





# Inisialisasi Member Structure ?

```
// Cara 1
/*
varName1.memberName1 = value1;
varName1.memberName2 = value2;
...
*/

// Cara 2
/*
struct Name varName;
varName = {val1, val2, ...};
*/

// Cara 3
/*
structName varName = {val1, val2, ...};
*/
```



# Array dalam Member Structure ?

```
/*
struct structName{
    // komponen / member
    dataTypeMember1 memberName1[n];
    dataTypeMember2 memberName2;
    dataTypeMember3 memberName3;
    . . . . .
};
*/
```

```
// Cara 1
/*
varName1.memberName1[0] = value1;
varName1.memberName1[n] = value2;
...
*/

// Cara 2
/*
varName1.memberName1 = {val1, valN};
...
*/
```



# Array dari Structure ?

```
// Cara 1 (Global)
/*
structName varName[n];
*/

int main()
{
    // Cara 2 (Lokal)
    /*
    structName varName[n];
    */
}
```

```
// Inisialisasi
/*
varName[n].memberName1 = val1;
varName[n].memberName2 = val2;
. . .
*/
```



# Nested Structure ?

- Bisa dilakukan dengan dua cara :
  - Membuat dua structure berbeda
  - Membuat structure dalam structure

```
// Didalam Structure
/*
struct structName1{
    // komponen / member
    dataTypeMember1 memberName1[n];
    dataTypeMember2 memberName2;
    struct structName2{
        // komponen / member
        dataTypeMember1 memberName1;
        dataTypeMember2 memberName2;
    } varName;
};
*/
```

```
// Terpisah
/*
struct structName1{
    // komponen / member
    dataTypeMember1 memberName1;
    dataTypeMember2 memberName2;
};

struct structName2{
    // komponen / member
    dataTypeMember1 memberName1[n];
    dataTypeMember2 memberName2;
    structName1 varName;
    . . . .
};
*/
```

# Structure di Parameter Fungsi ?

```
/*  
void funcName(structName varName){  
    // mau diapain  
}  
  
dataType funcName(structName varName){  
    // mau diapain  
    return ... ;  
}  
*/
```



# Kesimpulan ?

- Structure adalah kumpulan variabel dengan tipe data yang dapat berbeda-beda dengan satu nama.
- Structure dapat berisi variabel, array, atau bahkan tipe data baru lainnya juga seperti structure.
- Untuk mengakses member dari structure, dapat digunakan operator akses member berupa period(.)
- Array structure merupakan kumpulan structure dengan panjang tertentu.
- Structure dalam structure disebut juga Nested Structure.
- Structure dapat digunakan sebagai parameter atau return dari fungsi.





**Video Selanjutnya**

Linked List





**Thank you**

**#KEEPLARNING  
#KEEPSPIRITS**





