

Pelatihan Pemrograman Bahasa C Array & Struct

29 Oktober 2022

Learning Outcome

1. Peserta diharapkan mampu memahami Array pada Bahasa Pemrograman C (C2).
2. Peserta diharapkan mampu memahami Struct pada Bahasa Pemrograman C (C2)



Pembicara: **Hendra, S.Kom., MT.**

Kode Dosen: **D1828**

Faculty: **Computer Science**

Binus University - Jakarta

Topic

- **Pengenalan Bahasa Pemrograman C**
- **Pengenalan Array dalam Bahasa C**
- **Pengenalan Struct dalam Bahasa C**

Pengenalan Bahasa Pemrograman C

- C adalah Bahasa pemrograman untuk berbagai keperluan, yang dibuat oleh Dennis Ritchie di Bell Laboratories pada 1972.
- Bahasa C merupakan bahasa pemrograman populer sampai sekarang walau sudah cukup lama.
- Bahasa C sangat berkaitan dengan sistem operasi UNIX, karena bahasa C digunakan untuk membuat sistem operasi UNIX.

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

Baris 1: #include <stdio.h> adalah header file library yang diperlukan untuk menjalankan fungsi input dan output, seperti: printf() (di baris 4).

Baris 2: Baris kosong. Bahasa C mengabaikan white space. Baris kosong ini diberikan supaya coding terlihat rapi atau mudah dibaca.

Baris 3: Fungsi main() ini pasti ada di dalam program C. Fungsi main ini akan dipanggil pertama kali Ketika program dijalankan.

Baris 4: printf() adalah function untuk output / menampilkan text di layar. Pada contoh program ini akan ditampilkan "Hello World".

Line 5: return 0 untuk mengakhiri main() function.

Array dalam Bahasa C

- Arrays digunakan untuk menyimpan deretan nilai dalam single variable
- Untuk membuat array, definisikan tipe data type (misal int) dan tuliskan nama arraynya diikuti kurung siku [].
- Untuk memasukan values ke dalam array, gunakan comma-separated list, dalam kurung siku:

```
int myNumbers[] = {25, 50, 75, 100};
```

Mengakses Elemen Array

- Untuk mengakses elemen array menggunakan nomor indexnya.
- Index Array dimulai dari 0:
 - [0] adalah elemen pertama.
 - [1] adalah elemen kedua, dst.
- Berikut ini contoh cara mengakses elemen array:

```
int myNumbers[] =  
{25, 50, 75, 100};  
printf("%d", myNumbers[0]);
```

```
// Outputs 25
```


Mengubah Elemen Array

Untuk mengubah nilai elemen array, bisa menunjuk nomor index arraynya:

```
int myNumbers[] =  
{25, 50, 75, 100};  
myNumbers[0] = 33;
```

```
printf("%d", myNumbers[0]);
```

```
// Now outputs 33 instead of 25
```

Mengakses array dengan Loop

- Kita dapat menggunakan loop untuk mengakses elemen.
- Berikut ini contoh menampilkan semua isi elemen dalam array myNumbers:

```
int myNumbers[] = {25, 50, 75, 100};  
int i;  
  
for (i = 0; i < 4; i++) {  
    printf("%d\n", myNumbers[i]);  
}
```

Men-set ukuran array

```
// Declare an array of four integers:  
int myNumbers[4];
```

```
// Add elements  
myNumbers[0] = 25;  
myNumbers[1] = 50;  
myNumbers[2] = 75;  
myNumbers[3] = 100;
```

Struct

- Structure (disebut juga struct) adalah cara mengelompokkan beberapa variables ke dalam satu tempat. Tiap variable dalam structure di sebut member structure.
- Tidak seperti array, structure dapat berisi tipe data berbeda (int, float, char, dst.).

Membuat Struct

Kita dapat membuat structure dengan menggunakan keyword struct dan mendeklarasikan tiap membernya di dalam kurung kurawal:

```
struct MyStructure {    // Structure declaration
    int myNum;           // Member (int variable)
    char myLetter;       // Member (char variable)
}; // End the structure with a semicolon
```

Mengakses Struct

- Untuk mengakses structure, dideklarasikan variable dari struct tersebut.
- Gunakan keyword struct dalam fungsi main(), diikuti nama structure dan nama variable structure-nya:

```
struct myStructure {  
    int myNum;  
    char myLetter;  
};  
  
int main() {  
    struct myStructure s1;  
    return 0;  
}
```

Mengakses Member Struct

- Untuk mengakses member structure, gunakan operator dot (.)

```
// Create a structure called myStructure
struct myStructure {
    int myNum;
    char myLetter;
};

int main() {
    // Create a structure variable of myStructure called s1
    struct myStructure s1;

    // Assign values to members of s1
    s1.myNum = 13;
    s1.myLetter = 'B';

    // Print values
    printf("My number: %d\n", s1.myNum);
    printf("My letter: %c\n", s1.myLetter);

    return 0;
}
```


Copy Structures

- Kita bisa menduplikat / copy satu structure ke structure lainnya.
- Berikut ini contoh cara meng-copy structure:

```
struct myStructure s1 = {13, 'B', "Some text"};  
struct myStructure s2;
```

```
s2 = s1;
```

- Untuk mengubah value dalam member Strut, gunakan dot operator (.)
- Untuk mengubah string value, gunakan function strcpy()

```
struct myStructure {
    int myNum;
    char myLetter;
    char myString[30];
};

int main() {
    // Create a structure variable and assign values to it
    struct myStructure s1 = {13, 'B', "Some text"};

    // Modify values
    s1.myNum = 30;
    s1.myLetter = 'C';
    strcpy(s1.myString, "Something else");

    // Print values
    printf("%d %c %s", s1.myNum, s1.myLetter, s1.myString);

    return 0;
}
```

Contoh Kasus dengan Struct

- Program C untuk menyimpan informasi tentang Mobil dengan menggunakan Struct.

```
struct Car {  
    char brand[50];  
    char model[50];  
    int year;  
};  
  
int main() {  
    struct Car car1 = {"BMW", "X5", 1999};  
    struct Car car2 = {"Ford", "Mustang", 1969};  
    struct Car car3 = {"Toyota", "Corolla", 2011};  
  
    printf("%s %s %d\n", car1.brand, car1.model, car1.year);  
    printf("%s %s %d\n", car2.brand, car2.model, car2.year);  
    printf("%s %s %d\n", car3.brand, car3.model, car3.year);  
  
    return 0;  
}
```



CDA

Enhancing individuals & enriching the community

***Thankyou
very much***

Reference

- https://www.w3schools.com/c/c_intro.php
- https://www.w3schools.com/c/c_arrays.php
- https://www.w3schools.com/c/c_structs.php