

MODUL 10

ARRAY

Tujuan:

1. Praktikkan dapat memahami konsep Array 1 dimensi
2. Praktikkan dapat memahami cara penggunaan Array serta mengaplikasikan modul sebelumnya dalam penggunaan Array
3. Praktikkan dapat menggunakan konsep Array untuk menyelesaikan permasalahan yang ada



PENGERTIAN ARRAY

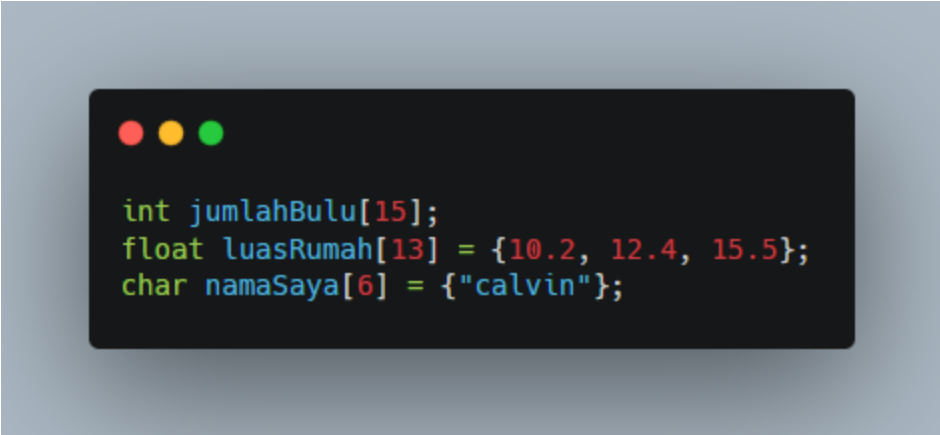
Jika pada variable kita hanya bisa menampung 1 data saja, Array sedikit berbeda. Array dapat menampung lebih dari 1 data yang bertipe data sama. Dengan ini, kita bisa memanfaatkan Array untuk menyelesaikan segala jenis permasalahan yang sekiranya memiliki data yang cukup banyak dengan cukup efisien karena tidak perlu membuat variable satu persatu. Untuk memahami Array, berikut adalah konsep dasar yang harus diperhatikan:

1. Array dapat menyimpan lebih dari 1 data dengan tipe data yang sama
2. Dalam pendeklarasiannya, size atau Panjang Array harus ditentukan dari pertama Array tersebut dibuat karena selanjutnya akan digunakan untuk memesan blok-blok memori
3. Array bersifat static
4. Array menggunakan index untuk mengakses data yang ada didalamnya
5. Index adalah bilangan bulat dan pada Array biasanya index dimulai dari 0 (nol)

PEMBUATAN ARRAY

Berikut adalah beberapa cara yang dapat digunakan untuk membuat Array:

1. `tipeData namaArray[panjangArray];`
2. `int jumlahBulu[15];`
3. `float luasRumah[3] = {10.2, 12.4, 15.5};`
4. `char namaSaya[6] = {"calvin"};`

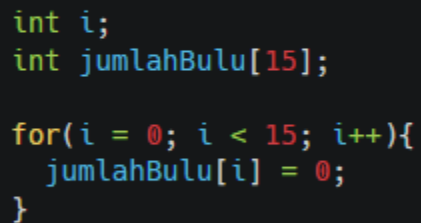


```
int jumlahBulu[15];  
float luasRumah[13] = {10.2, 12.4, 15.5};  
char namaSaya[6] = {"calvin"};
```

INISIALISASI ARRAY

Apabila Array yang telah kita buat masih belum memiliki isi, akan lebih baik jika kita menginisialisasi Array kita terlebih dahulu. Berikut adalah beberapa contoh inisialisasi Array untuk beberapa tipe data:

1. Array Numerik (int, float) = nol (0)
2. Array Karakter (char, string) = " " (spasi) atau "-" (hyphen)

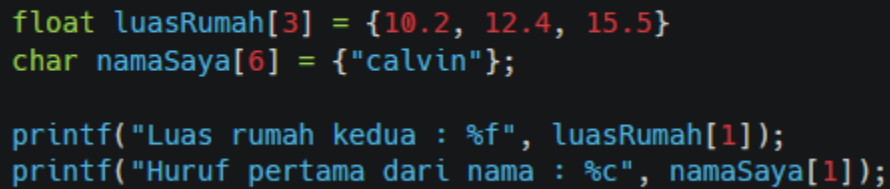


```
int i;  
int jumlahBulu[15];  
  
for(i = 0; i < 15; i++){  
    jumlahBulu[i] = 0;  
}
```

MENGAKSES ARRAY

Setelah deklarasi, Array juga bisa diakses dengan cara seperti berikut:

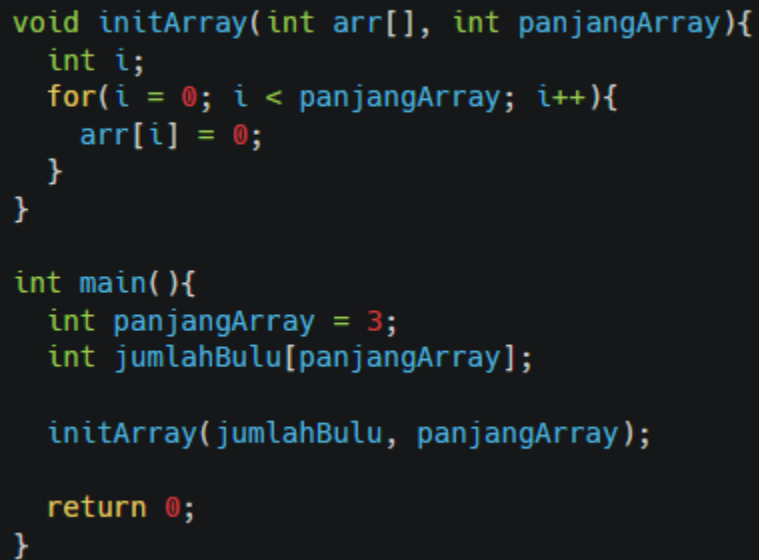
1. namaArray[index];
2. luasRumah[1]; → Akan didapatkan data berupa 12.4
3. namaSaya[0]; → Akan didapatkan data berupa 'c'



```
float luasRumah[3] = {10.2, 12.4, 15.5}  
char namaSaya[6] = {"calvin"};  
  
printf("Luas rumah kedua : %f", luasRumah[1]);  
printf("Huruf pertama dari nama : %c", namaSaya[1]);
```

PENGUNAAN PADA PROSEDUR DAN FUNGSI

Array juga bisa dimanfaatkan pada Prosedur dan Fungsi yakni sebagai berikut:



```
void initArray(int arr[], int panjangArray){
    int i;
    for(i = 0; i < panjangArray; i++){
        arr[i] = 0;
    }
}

int main(){
    int panjangArray = 3;
    int jumlahBulu[panjangArray];

    initArray(jumlahBulu, panjangArray);

    return 0;
}
```

GUIDED

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define MAX 5

void tampilArray(int arr[]){
    int i;
    for(i = 0; i < MAX; i++){
        printf("Array index-%d : %d\n", i, arr[i]);
    }
}

int cariData(int arr[], int cariNilai){
    int i;
    for(i = 0; i < MAX; i++){
        if(arr[i] == cariNilai){
            return i;
        }
    }
    return -1;
}
```

Pada code diatas terdapat prosedur tampilArray berparameter arr[] yang berfungsi untuk menampilkan seluruh isi dari suatu Array.

```
void editArray(int arr[], int cariNilai, int nilaiBaru){
    int i;
    int indexDiCari = cariData(arr, cariNilai);
    if(indexDiCari != -1){
        arr[indexDiCari] = nilaiBaru;
    } else{
        printf("Data tidak ditemukan!");
    }
}
```

Pada potongan kode diatas, terlihat bahwa prosedur editArray yang berfungsi untuk mengubah suatu data pada Array yang dibawa pada parameter (dalam kasus ini adalah arr[]) tidak menggunakan asterisk (*). Ini dikarenakan Array sendiri sudah memesan blok-blok memori dan apabila merubahnya akan otomatis merubah data pada memori tertentu yang telah dipesan.

Pada prosedur editArray, kita akan membutuhkan prosedur lain seperti cariData karena untuk mengedit sebuah Array kita harus mencari data yang ada pada Array yang biasa disebut dengan **traversal**. Prosedur cariData disini digunakan untuk mempersingkat code yang ada pada prosedur editData dikarenakan prosedur editData akan selalu mencari data terlebih dahulu.

```

int main(){
    int arrA[MAX];
    int arrB[MAX];
    int arrC[MAX];

    int i, j;

    printf("\nArray A:");
    for(i = 0; i < MAX; i++){
        printf("\nMasukkan data Array A index-%d : ", i); scanf("%d", &arrA[i]);
    }

    printf("\n\nArray B :");
    for(i = 0; i < MAX; i++){
        printf("\nMasukkan data Array B index-%d : ", i); scanf("%d", &arrB[i]);
    }

    printf("\n\nIsi seluruh Array A : \n");
    tampilArray(arrA);

    printf("\n\nIsi seluruh Array B : \n");
    tampilArray(arrB);

    int cari, index, nilaiBaru;
    printf("\n\nMasukkan nilai yang ingin di cari pada Array A : "); scanf("%d", &cari);
    index = cariData(arrA, cari);
    if(index == -1){
        printf("Data tidak ditemukan!");
    } else{
        printf("Data %d terdapat pada index ke-%d Array A", cari, index);
    }

    printf("\n\nMasukkan nilai yang ingin diubah pada Array B : "); scanf("%d", &cari);
    printf("Masukkan nilai yang baru : "); scanf("%d", &nilaiBaru);
    editArray(arrB, cari, nilaiBaru);
    printf("\n\nArray B yang sudah di-edit : \n");
    tampilArray(arrB);

    return 0;
}

```

Format Pengumpulan Guided:

1. Kerjakan guided dalam Bahasa C dan file .c
2. Beri nama file dengan GD10_X_YYYYY
3. Keterangan:
 - X = Kelas
 - YYYYY = 5 digit terakhir NPM

Referensi Pembelajaran dan Latihan Array:

<http://www.eecs.qmul.ac.uk/~pc/teaching/introprogramming/week7/exercises7.html>

<https://www.geeksforgeeks.org/introduction-to-arrays/>

<https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=arrays>

