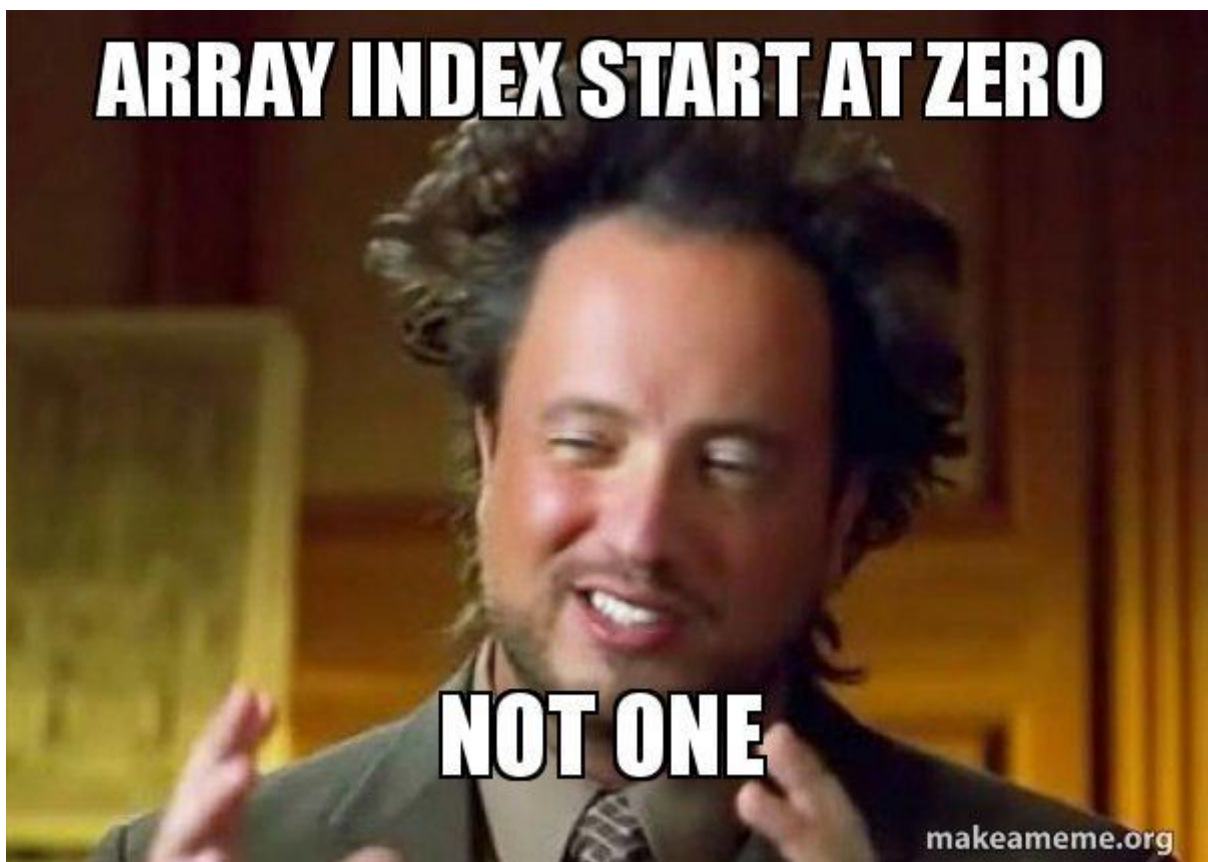


MODUL 10

ARRAY

Tujuan:

1. Praktikan dapat memahami konsep dasar array dan operasi pada array.
2. Praktikan dapat memahami cara penggunaan array serta mengaplikasikan modul sebelumnya dalam penggunaan array.
3. Praktikan dapat menyelesaikan permasalahan menggunakan konsep array.



PENGERTIAN ARRAY

Array adalah stuktur data yang memungkinkan kita untuk menyimpan sekumpulan data dalam satu variabel yang sama. Dengan array, kita bisa menyelesaikan suatu permasalahan yang memiliki banyak data dengan efisien karena tidak perlu membuat variabel satu-persatu. Untuk mempermudah pemahaman tentang konsep array, berikut adalah hal-hal yang perlu dipahami tentang array:

1. Array dapat menyimpan lebih dari satu data dengan tipe data yang sama.
2. Ukuran dari array harus ditentukan saat pendeklarasiannya, sehingga perlu diketahui terlebih dahulu berapa maksimal data yang ingin ditampung di dalam variabel array.
3. Array bersifat static, artinya ukuran dari array tidak bisa berubah seiring berjalannya program.
4. Data pada array diakses menggunakan index, index adalah angka bilangan bulat dan index pada array dimulai dari 0 kemudian 1, 2 , 3 dan seterusnya.

Visualisasi array :

array bil[5]:	15	20	7	21	34
index ke -	0	1	2	3	4

Contoh di atas merupakan array bernama 'bil' bertipe data integer dan menampung 5 data bilangan bulat.

JENIS ARRAY

- Array satu dimensi

Array satu dimensi merupakan array yang hanya dideklarasikan satu dimensi saja. Array satu dimensi adalah array yang menyimpan satu deret nilai yang bertipe sama. Contoh penggunaan array satu dimensi yaitu untuk menyimpan daftar nama mahasiswa, daftar npm, daftar harga, dll.

- Array multidimensi

Array multidimensi adalah array yang memiliki ukuran dimensi lebih dari satu. Array ini bisa berukuran 2 dimensi, 3 dimensi dan seterusnya. Biasanya array multidimensi yang sering dipakai adalah array yang berukuran 2. Untuk array yang berukuran selebihnya jarang digunakan. Contoh penggunaan array berukuran 2 dimensi pada C adalah membuat peta game 2 dimensi atau membuat index posisi.

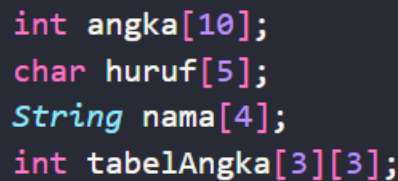
DEKLARASI ARRAY

Berikut struktur yang digunakan untuk mendeklarasikan variabel array :

```
tipeData namaVariabel[ukuran array];
```

atau

```
tipeData namaVariabel[ukuran dimensi ke-1][ukuran dimensi ke-n];
```



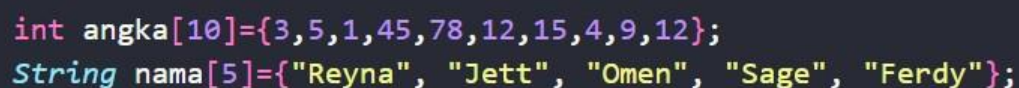
```
int angka[10];  
char huruf[5];  
String nama[4];  
int tabelAngka[3][3];
```

Gambar 1

Pada gambar 1, kita mendeklarasikan 4 variabel array. Variabel array pertama sampai variabel array ketiga adalah array 1 dimensi. Dan variabel array keempat adalah array 2 dimensi. Saat mendeklarasikan variabel array kita harus menentukan tipe datanya seperti saat mendeklarasikan variabel biasa. Namun, pada variabel array kita menambahkan kurung siku yang di dalamnya terdapat bilangan bulat. Bilangan bulat tersebut menandakan ukuran atau banyak data yang ingin ditampung pada variabel array tersebut.

INISIALISASI ARRAY

Terdapat dua cara untuk melakukan inisialisasi variabel array. Cara pertama yaitu cara manual:



```
int angka[10]={3,5,1,45,78,12,15,4,9,12};  
String nama[5]={\"Reyna\", \"Jett\", \"Omen\", \"Sage\", \"Ferdy\"};
```

Gambar 2

Cara yang pertama ini kita langsung mengisi data pada variabel array ketika dideklarasikan. Cara ini digunakan ketika kita sudah mengetahui nilai/data yang akan diisi. Sehingga kita dapat mengisi data yang kita inginkan sebelum program dijalankan. Dan data yang diisi harus sesuai dengan tipe datanya.

Cara kedua adalah menggunakan perulangan :

```
float desimal[10];
int i;
for(i=0;i<10;i++){
    desimal[i]=0.0;
}
```

Gambar 3

Pada contoh gambar 3, kita mendeklarasikan variabel array bertipe data float dengan ukuran 10 yang diberi nama desimal. Kemudian di bawahnya, dilakukan perulangan dari 0 sampai 9 (array ukuran 10 berarti memiliki index dari 0 sampai 9) dan mengisi variabel array dengan data 0.0.

Cara seperti ini kita gunakan ketika kita belum mengetahui dengan pasti data apa yang akan diisi ke dalam variabel array atau data berasal dari inputan pengguna.

Cara di atas dinamakan inisialisasi array, inisialisasi ini berguna agar data yang ditampung di dalam array tidak acak atau undefined behavior.

MENGAKSES ARRAY

Setelah mendeklarasikan variabel array, kita akan mengakses data pada variabel array agar datanya bisa diubah atau ditampilkan. Untuk mengakses data pada variabel array kita membutuhkan index dari array tersebut.

```
char nama[5]={"riksi"}; //deklarasi array nama
int angka[5];           //deklarasi array angka

angka[0]=1; //
angka[1]=3; //
angka[2]=18; // } mengakses array angka
angka[3]=9; //
angka[4]=12; //

//di bwh ini menampilkan/akses array nama dan angka
printf("\nHuruf ketiga dari nama saya : %c",nama[2]);
printf("\nAngka index ke-2 : %d",angka[2]);
```

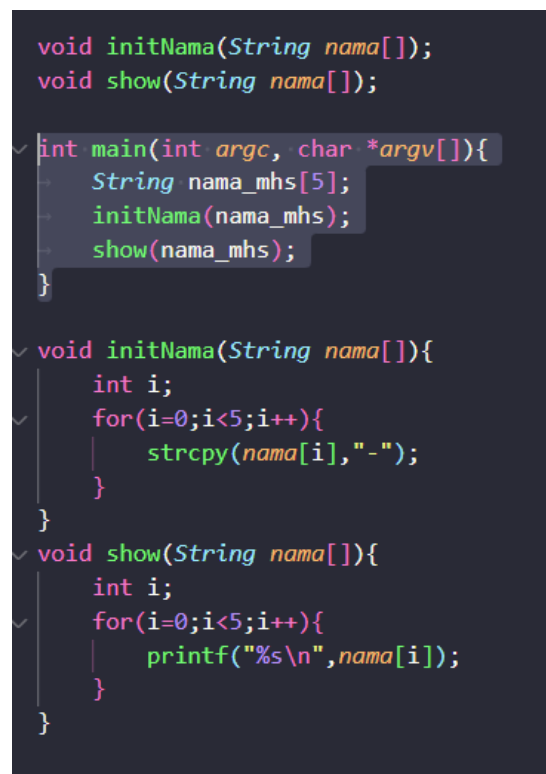
Gambar 4

Perlu diperhatikan bahwa saat mengakses data pada variabel array, angka di dalam kurung siku menyatakan index. Sedangkan saat deklarasi variabel array, angka tersebut menyatakan ukuran variabel array.

ARRAY PADA PROSEDUR ATAU FUNGSI

Ketika sebuah variabel array dipassing ke dalam sebuah prosedur atau fungsi, sifatnya sudah berbentuk *passing by reference*, serupa dengan parameter input output dan tidak memerlukan (*) karena bentuk dari array sebenarnya adalah sebuah pointer (*). Oleh karena itu, perubahan data pada variabel array di dalam fungsi/prosedur juga akan mengubah data pada variabel array di main() program.

Contoh penggunaan array pada prosedur :

The image is a screenshot of a code editor with a dark background and light-colored text. It contains C code demonstrating array passing by reference. The code is as follows:

```
void initNama(String nama[]);
void show(String nama[]);

int main(int argc, char *argv[]){
    String nama_mhs[5];
    initNama(nama_mhs);
    show(nama_mhs);
}

void initNama(String nama[]){
    int i;
    for(i=0;i<5;i++){
        strcpy(nama[i],"-");
    }
}

void show(String nama[]){
    int i;
    for(i=0;i<5;i++){
        printf("%s\n",nama[i]);
    }
}
```

Gambar 5

Pada code di gambar 5, kita membuat prosedur yang bernama `initNama(string nama[])`. Pada parameter prosedur atau fungsi, ukuran array tidak perlu diisi nilainya (Kecuali untuk array 2 dimensi atau lebih). Pada prosedur ini kita menginisialisasi array string yang diisi “-“ dengan menggunakan perulangan for yang dimulai dari 0 sampai 4. Dalam perulangan tersebut, setiap elemen/data array diakses menggunakan index ke-i, Mengapa pakai ke-i? Karena nilai i ini akan dimulai dari 0 s/d 4 seperti yang dituliskan pada perulangan tersebut. Sehingga sesuai dengan nilai index yang dimiliki oleh variabel array tersebut. Kemudian pada pemanggilan prosedur di main(), parameternya hanya perlu nama variabel arraynya saja.

GUIDED

*Mohon dibaca ketentuannya terlebih dahulu

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h> // Library ini digunakan untuk memanipulasi string
#include <conio.h> // library untuk getch()
// [Tidak perlu memanggil library ini karena sudah otomatis diisi]

typedef char string[100];

void initData(string namaMhs[], int npmMhs[]);
int cekKosong(int npmMhs[]);
int getEmptyIndex(int npmMhs[]);
void tampilMhs(string namaMhs[], int npmMhs[]);
int findMhs(int npmMhs[], int npm);

void sortingAscending(string namaMhs[], int npmMhs[]);

int main(int argc, char *argv[]) {
    string namaMhs[5], nama; // pendeklarasian variabel dan pastikan selalu berada di paling atas
    int npmMhs[5], npm;
    int menu, index;

    initData(namaMhs, npmMhs); // pemanggilan prosedur init data atau inisialisasi data

    do{
        system("cls");
        printf("\n\t>> [GUIDED ARRAY] <<");
        printf("\n\t>> [PROJECT CRUD] <<\n");
        printf("\n[1] Create");
        printf("\n[2] Read");
        printf("\n[3] Update");
        printf("\n[4] Delete");
        printf("\n[5] Sorting [ TUGAS ]");
        printf("\n[0] Exit");
        printf("\n>>> ");scanf("%d", &menu);
        switch(menu){

            case 1:

                if(getEmptyIndex(npmMhs) != -1){ // untuk mencari index array yang datanya masih kosong
                    index = getEmptyIndex(npmMhs); // kemudian index array tersebut akan disimpan
                                                    // kedalam variabel "index"

                    printf("\n\t[Create Data]\n");
                    do{
                        printf("\nNama Mahasiswa : ");fflush(stdin);gets(nama);
                        if(strlen(nama) == 0){
                            printf("[!] Nama tidak boleh kosong");
                        }
                    }while(strlen(nama) == 0);
                    do{
                        printf("\nNPM Mahasiswa : ");scanf("%d", &npm);
                        if(npm < 1){
                            printf("[!] NPM tidak boleh kurang dari 1");
                        }else if(findMhs(npmMhs, npm) != -1){ // untuk mengecek apakah npm inputan ada yang sama
                            printf("[!] NPM tidak boleh sama"); // atau tidak dengan yang ada dalam array
                        }
                    }while(npm < 1 || findMhs(npmMhs, npm) != -1);

                    strcpy(namaMhs[index], nama); // mengcopy data hasil inputan user ke dalam array, variabel
                    npmMhs[index] = npm; // index didalam kurung kotak menandakan lokasi array yang kosong

                    printf("\n\t[+] Berhasil menginputkan Data [+]");
                }else{
                    printf("\n\t[!] Data pada array mahasiswa sudah penuh [!]);
                }
            break;

            case 2:

                printf("\n\t[READ DATA]\n");
                if(cekKosong(npmMhs) != -1){ // untuk mengecek apakah data kosong atau tidak
                    printf("\n\t[!] Data Masih Kosong [!]);
                }
                tampilMhs(namaMhs, npmMhs); // memanggil prosedur untuk menampilkan data dalam array

            break;

            case 3:

                printf("\n\t[Update Data]\n");
                printf("\nMasukkan NPM yang dicari : ");scanf("%d", &npm);
                if(findMhs(npmMhs, npm) != -1 && npm != 0){ // npm yang sudah diinput akan dicari didalam array
                                                            // menggunakan fungsi findMhs. Alasan npm != 0 dikarenakan
                                                            // pada array, semua variabel npm di inisialisasikan menjadi 0
                                                            // maka dari itu 0 dibuat menjadi sebuah exception dimana 0
                                                            // dianggap kosong atau belum ada datanya
                    index=findMhs(npmMhs, npm);
                }
            break;
        }
    }while(menu != 0);
}
```

```

        do{
            printf("\nNama Mahasiswa : ");fflush(stdin);gets(nama);
            if(strlen(nama) ==0){
                printf("\t[!] Nama tidak boleh kosong [!]\n");
            }
        }while(strlen(nama) == 0);

        do{
            printf("\nNPM Mahasiswa : ");scanf("%d", &npm);
            if(npm<1){
                printf("\t[!] NPM TIDAK BOLEH KURANG DARI 1 [!]\n");
            }else if(findMhs(npmMhs, npm) != -1){
                printf("\t[!] NPM TIDAK BOLEH SAMA [!]\n");
            }
        }while(npm<1 || findMhs(npmMhs, npm) != -1);
        strcpy(namaMhs[index], nama);
        npmMhs[index]= npm;
        printf("\n\t[+] Berhasil mengubah data [!]\n");

    }else{
        printf("\n\t[!] Mahasiswa tidak ditemukan [!]\n");
    }

    break;

    case 4:

        printf("\n\t[DELETE DATA]\n");
        printf("\nMasukkan NPM yang akan dihapus : ");scanf("%d", &npm);
        if(findMhs(npmMhs, npm) != -1 && npm != 0){
            index = findMhs(npmMhs, npm);
            strcpy(namaMhs[index], ""); // merubah nama dan npm sesuai dengan yang dilakukan pada
            npmMhs[index] = 0; // inisialisasi dikarenakan pada array, nama dianggap kosong
                                // jika berisi "" dan npm dianggap kosong jika berisi 0
            printf("\n\t[+] Berhasil menghapus data [!]\n");
        }else{
            printf("\n\t[!] Mahasiswa tidak ditemukan [!]\n");
        }

        break;

    case 5:

        // Tugas GD
        // buatlah sebuah prosedur yang dapat mensortir NPM yang sudah diinputkan secara Ascending (kecil ke besar)
        // Contoh ada 3 data, [220711697, 220711695, 220711696]
        // ketika disortir maka hasilnya akan [220711695, 220711696, 220711697]
        sortingAscending(namaMhs, npmMhs);

        // Pelajarin logika sorting tersebut dikarenakan nantinya akan digunakan pada UGD

        break;

    case 0:

        printf("\n\t[!] Keluar dari Program [!]\n");
        printf("\n\n\t[!] NAMA - NPM - KELAS [!]\n");

        break;

    default:
        printf("\n[!] Menu tidak tersedia");
        break;

    }
    getch();
}while(menu != 0);

return 0;
}

void initData(string namaMhs[], int npmMhs[]){ // prosedur ini akan men-set/inisialisasi data array
    int i; // agar data pada array tidak undefined behavior
    for(i = 0; i<5; i++){

        strcpy(namaMhs[i], ""); // nama mahasiswa di set ke "" yang artinya kosong
        npmMhs[i] = 0; // npm di-set ke 0/nol
    }
}

int cekKosong(int npmMhs[]){ // fungsi ini untuk mengecek apakah data dalam array kosong atau tidak
    // kosong atau tidak dengan menggunakan npm sebagai tumpuan untuk pengecekan
    int i;
    for(i = 0; i<5; i++){
        if(npmMhs[i] != 0){ // jika pada array terdapat npm yang != 0, maka terdapat data dalam array tersebut
            return -1;
        }
    }
    return 1; // jika setelah semua pengecekan diatas tidak ada npm yang bukan 0, maka benar kosong.
}

int getEmptyIndex(int npmMhs[]){ // fungsi ini akan mencari index array yang belum ada datanya
    int i;
    for(i = 0; i<5; i++){
        if(npmMhs[i] == 0){ // dengan menggunakan kondisi jika npm index ke-i = 0, maka akan mereturnkan i
            return i; // i disini menunjuk ke index dari array
        }
    }
    return -1;
}

void tampilMhs(string namaMhs[], int npmMhs[]){ // prosedur ini untuk menampilkan data mahasiswa

```



```

void tampilMhs(string namaMhs[], int npmMhs[]){ // prosedur ini untuk menampilkan data mahasiswa
    int i; // yang ada pada dalam array
    for(i = 0; i<5; i++){
        if(npmMhs[i] > 0){

            printf("\n[Mahasiswa Ke- %d]", i+1);
            printf("\nNama : %s", namaMhs[i]);
            printf("\nNPM : %d", npmMhs[i]);
            printf("\n");

        }
    }
}

int findMhs(int npmMhs[], int npm){ // fungsi ini akan mencari npm yang diinput
    // kemudian akan mereturnkan indexnya jika
    // npm tersebut ditemukan pada data array,
    // fungsi ini juga bisa digunakan untuk
    // pengecekan npm jika npm yang diinput
    // oleh pengguna sudah ada pada array

    for(i = 0; i<5; i++){
        if(npmMhs[i] == npm){
            return i;
        }
    }

    return -1;
}

void sortingAscending(string namaMhs[], int npmMhs[]){

    // Buat kodenya disini

}

/*
*TAMBAHAN
Mengapa pada fungsi dan prosedur diatas selalu menggunakan perulangan?
Hal ini dikarenakan untuk mengakses semua data pada array kita perlu memakai indexnya,
dan kita perlu mengakses data array satu persatu. Oleh karena itu,
agar menjadi efisien maka kita dapat menggunakan perulangan,
perulangan dimulai dari 0 sampai index terakhir array,
disini index awal array dimulai dari 0 dan index terakhir adalah 4
sehingga pada kondisi perulangan bisa ditulis i<5 (kurang dari 5)
*/

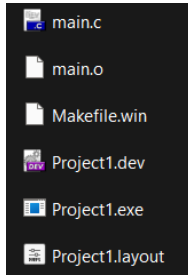
```

Programming is a skill best acquired by practice and example rather than from books

- Alan Turing

Ketentuan Guided:

1. **Tidak perlu** membuat *comment* pada guided, buat *code*-nya saja
2. **Jangan hanya** mengumpulkan file .c saja



Kurang lebih harus ada file di atas, terutama yang berekstensi .c dan .dev, **semua file di atas** dimasukkan ke dalam **satu folder**

3. Format penamaan folder : **GD10_X_YYYYY** lalu di-zip
4. Keterangan:
 - X = Kelas
 - Y = 5 digit NPM terakhir

Selamat Mengerjakan dan Semangat! ^_^

Referensi Pembelajaran dan Latihan Array :

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

<https://www.petanikode.com/c-array/>

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

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