

JOB SHEET 07

Struktur Data: Array

*"Everybody should learn to program a computer,
because it teaches you how to think."*

TUJUAN PEMBELAJARAN

1. Mampu menjelaskan dan mengimplementasikan *Data Structure: Array* dalam pemrograman menggunakan IDE.

POKOK MATERI

1. Pengertian Array
2. Penggunaan Array

URAIAN MATERI

A. Array

Untuk dapat menyimpan banyak data di program, maka kita perlu menggunakan array. Misalkan kita ingin menyimpan nama-nama teman untuk data kontak.

Mungkin saja, kita akan menyimpannya seperti ini:

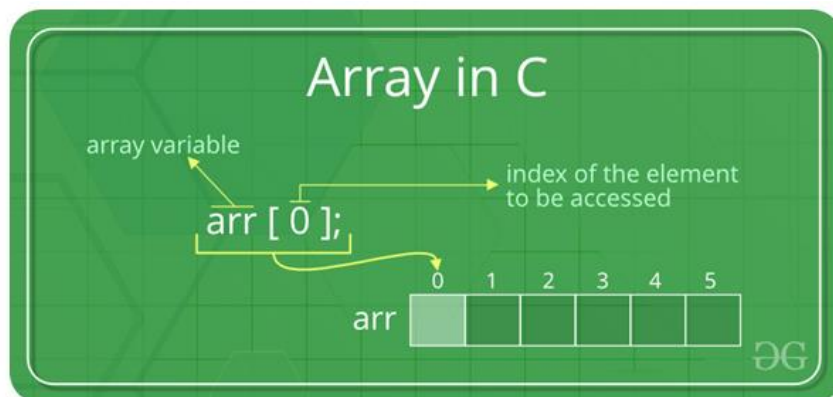
Misalnya, Anda ingin mencetak semua angka antara 1 hingga 10. Tanpa loop, mungkin Anda akan membuat program dengan banyak printf sampai 10 baris program bahkan lebih, atau mungkin saja menyimpannya seperti ini:

```
char[32] namaKontak1 = "Ayu";  
char[32] namaKontak2 = "Bunga";  
char[32] namaKontak3 = "Cyntia";  
char[32] namaKontak4 = "Deni";  
char[32] namaKontak5 = "Elisa";
```

Hal tersebut boleh saja dilakukan namun masalahnya bagaimana jika data itu dalam jumlah 100-1000? Pasti akan melelahkan untuk menulisnya perbaris seperti di atas. Karena itu kita membutuhkan array.

Pengertian Array

Array merupakan struktur data yang digunakan untuk menyimpan sekumpulan data dalam satu tempat, array ini juga disebut dengan istilah larik. Setiap data dalam Array memiliki indeks, sehingga kita akan mudah memprosesnya seperti mengakses atau mengganti datanya.

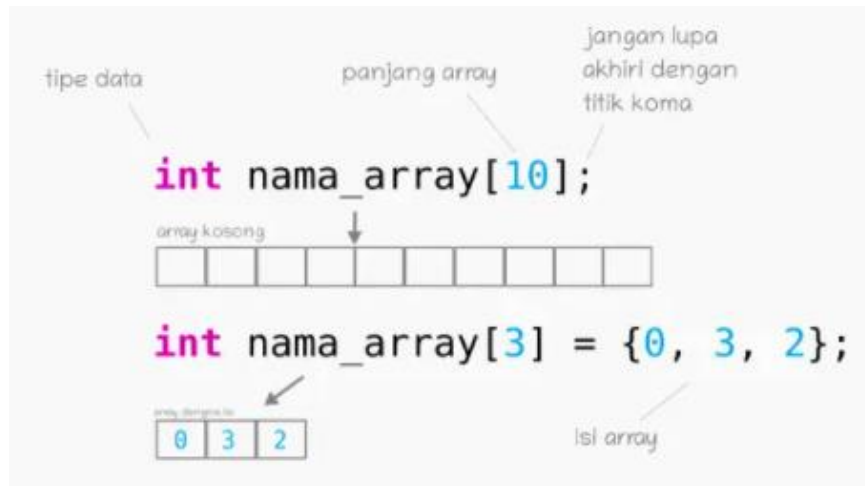


Seperti di atas, indeks array dimulai dari nol (0). Ukuran array akan bergantung dari banyaknya data yang ditampung di dalamnya.

Cara membuat variabel Array

```
// membuat array kosong dengan tipe data integer dan panjang 10  
int nama_array[10];  
  
// membuat array dengan langsung diisi  
int nama_arr[3] = {0, 3, 2}
```

Cara membuat array hampir sama dengan membuat variabel seperti biasanya. Bedanya saat menggunakan array maka perlu menentukan panjang array menggunakan kurung siku [].



Cara mengakses data Array

Array akan menyimpan sekumpulan data dan memberinya nomer indeks agar mudah diakses. Indeks array selalu dimulai dari nol 0.

Misalkan kita punya array seperti ini:

```
char huruf[5] = {'a', 'b', 'c', 'd', 'e'};
```

Bagaimana cara mengambil huruf c?

```
huruf[2];
```

Jawabannya adalah `huruf[2]`, kenapa bukan `huruf[3]`?

Dari program kita mengetahui indeks `huruf[5]` artinya huruf memiliki panjang tampungan data 5 yaitu 0, 1, 2, 3, 4, 5. Maka a berada pada indeks 0, dan e berada pada indeks 4, sementara c berada pada indeks 2.

Lakukan percobaan program berikut:

```
#include <stdio.h>

void main(){
    char huruf[5] = {'a', 'b', 'c', 'd', 'e'};

    printf("Huruf: %c\n", huruf[2]);
}
```

Merubah data pada array

Misalnya sebelumnya indeks 2 berisikan c, maka bisa kita ganti dengan huruf lain melalui indeksnya.

Caranya sebagai berikut:

```
huruf[2] = 'z';
```

Maka indeks 2 sekarang berisi z bukan lagi c.

Lakukan percobaan berikut:

```
#include <stdio.h>

void main(){
    // isi awal array
    char huruf[5] = {'a', 'b', 'c', 'd', 'e'};

    // mengubah isi array
    huruf[2] = 'z';

    // mencetak isi array
    printf("Huruf: %c\n", huruf[2]);
}
```

Percobaan lain:

```
#include <stdio.h>

void main(){
    // membuat array kosong
    int nilai[5];

    // mengisi array
    nilai[0] = 32;
    nilai[1] = 42;
    nilai[2] = 76;
    nilai[3] = 31;
    nilai[4] = 57;

    // mencetak isi array
    printf("Nilai ke-1: %d\n", nilai[0]);
    printf("Nilai ke-2: %d\n", nilai[1]);
    printf("Nilai ke-3: %d\n", nilai[2]);
    printf("Nilai ke-4: %d\n", nilai[3]);
    printf("Nilai ke-5: %d\n", nilai[4]);
}
```

Percobaan:

```
#include <stdio.h>
int main()
{
    int marks[10], i, n, sum = 0, average;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    for(i=0; i<n; ++i)
    {
        printf("Enter number%d: ", i+1);
        scanf("%d", &marks[i]);

        sum += marks[i];
    }
    average = sum/n;
    printf("Average = %d", average);
    return 0;
}
```

Percobaan:

```
#include <stdio.h>

int main {
    //deklarasi array of char
    char nama[20];

    printf("Masukkan nama Anda: ");
    gets(nama);

    printf("Hello, %s. Salam kenal.", nama);

    return 0;
}
```

Menggunakan perulangan dengan Array

Perhatikan contoh berikut tanpa perulangan:

```
printf("Nilai ke-1: %d\n", nilai[0]);
printf("Nilai ke-2: %d\n", nilai[1]);
printf("Nilai ke-3: %d\n", nilai[2]);
printf("Nilai ke-4: %d\n", nilai[3]);
printf("Nilai ke-5: %d\n", nilai[4]);
```

Kalau isi arraynya ada ribuan, apakah kita sanggup menuliskan sampai 1000? Tentu ini memakan waktu dan tenaga, makanya perlu penggunaan perulangan seperti yang sudah kita pelajari pada jobsheet sebelumnya.

Contoh:

```
#include <stdio.h>

void main(){
    // membuat array kosong
    int nilai[5];

    // mengisi array
    nilai[0] = 32;
    nilai[1] = 42;
    nilai[2] = 76;
    nilai[3] = 31;
    nilai[4] = 57;

    // mencetak isi array dengan perulangan
    for(int i = 0; i < 5; i++){
        printf("Nilai ke-1: %d\n", nilai[i]);
    }
}
```

Dengan menggunakan perulangan for maka untuk mencetak keseluruhan nilai cukup dengan satu baris kode yang berada dalam blok perulangan for.

Namun perulangan di atas kita memberikan batas maksimal perulangan secara manual yaitu $i < 5$. Sebaiknya diganti dengan panjang isi array seperti:

```
int length = sizeof(nilai) / sizeof(*nilai);
for (int i = 0; i < length; i++){
    printf("Nilai ke-1: %d\n", nilai[i]);
}
```

Mengambil panjang Array

Pada contoh di atas, kita menggunakan fungsi `sizeof()` untuk mengambil panjang atau ukuran array. Fungsi `sizeof()` sebenarnya akan mengambil ukuran memori dari array. Misalkan kita punya array seperti ini:

```
int nilai[2] = {1,2};
```

Untuk mengambil panjang array maka dapat dilakukan dengan membagi besar array dan ukuran pointernya (pointer kita bahasa ada materi berikutnya).

Contoh:

```
int array[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

/* ukuran array dalam byte */
int size = sizeof(array);

/* banyaknya isi array `array` */
int length = sizeof(array) / sizeof(*array);
```

Lakukan percobaan berikut:

```
#include <stdio.h>

void main(){
    // membuat array
    int nilai[5] = {33, 22, 11, 44, 21};

    // mengambil banyaknya isi array
    int length = sizeof(nilai) / sizeof(*nilai);

    printf("Banyaknya isi array nilai: %d\n", length);
}
```

Contoh kasus:

Latihan: Program Hitung Rata-Rata

Anggap saja kita diminta untuk membuat program untuk menghitung tinggi rata-rata dari sepuluh orang. Pertama kita pasti akan membutuhkan sebuah array dengan panjang 10 yang berisi kumpulan tinggi badan.

```
int tinggi_badan [10] = {175, 165, 166, 157, 184, 156, 163, 176, 171, 169};
```

Berikutnya kita harus menghitung nilai rata-rata dari sekumpulan nilai tersebut.

Rumus nilai rata-rata:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}$$

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Kode program:

```
#include <stdio.h>

void main() {
    // membuat array dari tinggi badan
    int tinggi_badan[10] = {175, 165, 166, 157, 184, 156, 163, 176, 171, 169};

    // length itu sama dengan n
    int length = sizeof(tinggi_badan) / sizeof(*tinggi_badan);
    int sum = 0;

    for(int i = 0; i < length; i++){
        sum += tinggi_badan[i];
    }

    // rata-rata tinggi badan
    float rata_rata = (float)sum / (float)length;

    printf("Rata-rata tinggi badan: %.2f\n", rata_rata);
}
```

(Lakukan percobaan program di atas)

Array Multidimensi

Array yang kita buat pada contoh-contoh program di atas adalah array satu dimensi. Array bisa dibuat dua dimensi bahkan lebih. Array multidimensi biasanya digunakan untuk membuat matriks.

Contoh:

```
int matriks[3][3] = {  
    {1, 3, 5},  
    {5, 3, 1},  
    {6, 2, 3}  
};
```

Array dua dimensi biasanya digunakan untuk membuat matriks.

Lalu bagaimana cara mengambil data dari array dua dimensi?

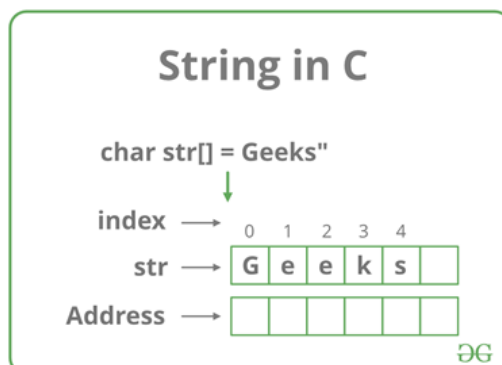
Begini caranya:

```
matriks[1][2];
```

Contoh program:

```
#include <stdio.h>  
  
void main(){  
    int matriks[3][3] = {  
        {1, 3, 5},  
        {5, 3, 1},  
        {6, 2, 3}  
    };  
  
    printf("Isi Data pada indeks ke-(1,0): %d\n", matriks[1][0]);  
}
```

String



char namavariabel [ukuran];

Percobaan 1:

```
#include <stdio.h>

int main() {

    char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
    printf("Greeting message: %s\n", greeting );
    return 0;
}
```

| | | | | | | |
|----------|---------|---------|---------|---------|---------|---------|
| Index | 0 | 1 | 2 | 3 | 4 | 5 |
| Variable | H | e | l | l | o | \0 |
| Address | 0x23451 | 0x23452 | 0x23453 | 0x23454 | 0x23455 | 0x23456 |

String.h merupakan library yang menyimpan fungsi-fungsi yang digunakan untuk menangani string ataupun substring.

1. strcpy()

Fungsi ini digunakan untuk mengkopi string pada variabel 2 ke variabel 1.

variabel sumber yang berisi string

strcpy(s1, s2);

Variabel tujuan

```
#include <stdio.h>
void main() {

    char title[] = "Belajar String.h pada Bahasa C";
    char title_copy[20];

    // copy string title ke title_copy
    strcpy(title_copy, title);

    // maka sekarang title_copy akan berisi:
    printf("isi title_copy\t: %s\n", title_copy);

}
```

2. strcat()

Fungsi strcat() digunakan untuk menambahkan string dari belakang.

variabel yang berisi string



strcat(s1, s2);



variabel yang berisi string
untuk ditambahkan ke s1

```
#include <stdio.h>

void main() {

    char nama[] = "Agariadne D. S.";
    char gelar[] = ", S.Kom., M.Pd.T.";

    strcat(nama, gelar);

    printf("Nama Lengkap\t: %s\n", nama);

}
```

3. strlen()

Fungsi strlen() digunakan untuk menghitung panjang string.

variabel berisi string yang
akan dihitung panjangnya

strlen(s1);

```
#include <stdio.h>

void main(){

    char nama[] = "_____";
    int panjang_nama;

    // menghitung panjang nama
    panjang_nama = strlen(nama);

    printf("Nama\t: %s\n", nama);
    printf("Panjang\t: %d Karakter\n", panjang_nama);
}
```

4. strcmp()

Fungsi strcmp() digunakan untuk membandingkan string dengan string yang lainnya.

variabel berisi string yang
akan dibandingkan

strcmp(s1, s2);

variabel berisi string yang
akan dibandingkan

```
#include <stdio.h>
void main(){
    char str1[] = "Agariadne";
    char str2[] = "Agariadne";
    int hasil;

    // membandingkan string
    hasil = strcmp(str1, str2);

    if(hasil == 0){
        printf("str1 dan str2 sama");
    } else {
        printf("str1 dan str2 berbeda");
    }
}
```

LATIHAN

1. Buatlah sebuah program untuk mencari rata-rata dari nilai dari 20 mahasiswa menggunakan array.
2. Buatlah sebuah program array untuk menampung input nama mahasiswa dari keyboard. Program pertama kali meminta banyak mahasiswa yang akan diinputkan melalui keyboard. Kemudian program meminta input nama mahasiswa ke-n (n =jumlah mahasiswa yang diinput). Kemudian tampilkan keseluruhan elemen dari mahasiswa tersebut. (Gunakan array dan perulangan).

Ilustrasi:

Input banyak siswa: 3

Input nama mahasiswa 1:

Input nama mahasiswa 2:

Input nama mahasiswa 3:

Nama mahasiswa ke-1:

...

Nama mahasiswa ke-n:

3. Buatlah program halaman login yang meminta pengguna untuk menginput username dan password, jika password yang digunakan benar maka muncul notifikasi “berhasil login”, sebaliknya kalau salah muncul notifikasi di layer “password salah”.

REFERENCES

1. The C Programming Language. 2nd Edition
2. [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
3. https://en.wikipedia.org/wiki/Imperative_programming
4. <https://www.petanikode.com/tutorial/c/>
5. <https://www.learn-c.org/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.programiz.com/>
8. <https://www.dicoding.com/>
9. <https://data-flair.training/blogs/c-tutorials-home/>