

## ARRAY DAN POINTER

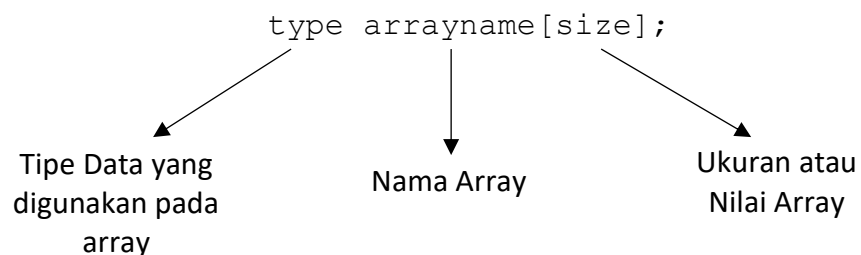
### OBJEKTIF :

1. Mahasiswa Mampu Memahami Tentang Array 1 Dimensi.
  2. Mahasiswa Mampu Memahami Tentang Array Multi Dimensi.
  3. Mahasiswa Mampu Memahami Tentang Passing Array.
  4. Mahasiswa Mampu Memahami Tentang Pointer.
  5. Mahasiswa Mampu Memahami Tentang String.
- 

### 1.1 ARRAY 1 DIMENSI

Array adalah kumpulan dari nilai – nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai – nilai data disuatu larik disebut dengan elemen – elemen larik. Letak urutan dari suatu elemen larik ditunjukkan oleh suatu *subscript* atau suatu *index*. Array 1 Dimensi adalah kumpulan elemen yang tersusun dalam suatu baris.

Sintaks dari bentuk pendeklarasian array adalah sebagai berikut :



Contoh :

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    //pendeklarasian variabel tanpa array
    int age = 5;
    int age2 = 6;
    int age3 = 7;
    int age4 = 10;

    //pendeklarasian variabel dengan array
    int ages[5]={5,6,7,10,8};

    /*index 0 base
    ages[0] = {5}
    ages[1] = {6}
    ages[2] = {7}
    ages[3] = {10}
    */

    //mengganti nilai atau elemen pada index 0
    ages[0] = 11;

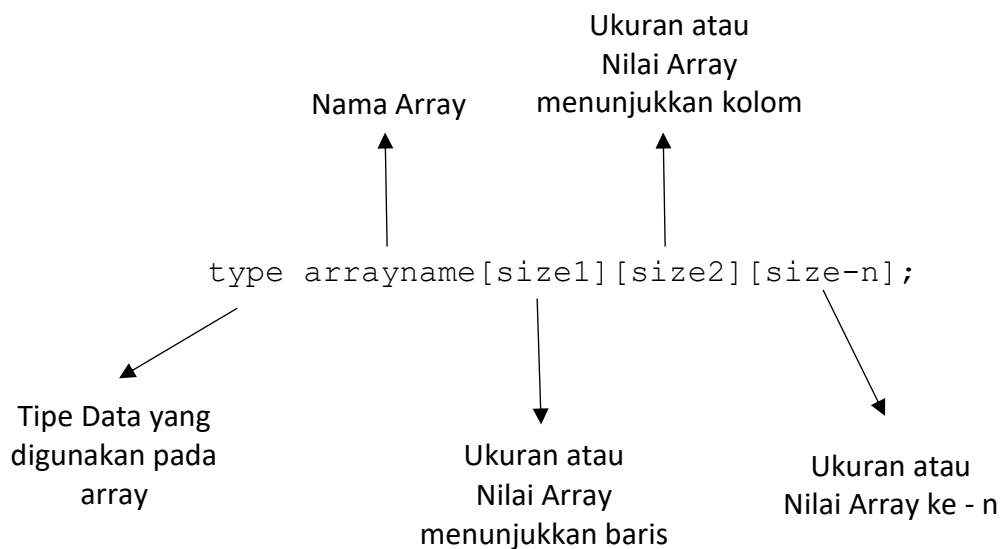
    for(int index = 0; index < 5; index++)
    {
        printf("index ke-%d sama dengan
        %d\n",index, ages[index]);
    }

    return 0;
}
```

## 1.2 ARRAY MULTI DIMENSI

Array Multi Dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang akan diakses melalui banyak indeks atau subskrip. Array seperti ini biasanya digunakan untuk matriks. Array 2 dimensi juga termasuk ke dalam array multi dimensi.

Sintaks dari bentuk pendeklarasian array multi dimensi adalah sebagai berikut :



Contoh :

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a[3][4] = {
        {0,1,2,3},
        {4,5,6,7},
        {8,9,10,11}
    };
    for(int row = 0; row < 3; row++){
        for(int column = 0; column < 4; column++){
```

```
                printf("a[%d][%d] = %d\n",
                        row, column, a[row][column]);
            }
        }

        return 0;
    }
```

### 1.3 PASSING ARRAY

Pada bagian ini akan dibahas mengenai bagaimana cara menggunakan Array sebagai parameter fungsi. Sebelumnya sudah diterapkan fungsi yang dapat mengambil beberapa parameter dan kemudian mengembalikan nilai berdasarkan parameter yang telah diberikan ke fungsi.

Contoh :

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

//deklarasi function
int addition(int *num1, int *num2);

int main()
{
    //array parameter
    int mynumberArray[6] = {23,30,21,18,17,25};
    int sum = totalValueofArray(mynumberArray,6);
    printf("total : %d\n",sum);

    return 0;
}
```

```
int totalValueofArray(int numberArray[], int size){
    int total = 0;
    for(int i=0; i<size; i++){
        //total = total + numberArray[i];
        total += numberArray[i];
    }
    return total;
}
```

## 1.4 POINTER

Setiap variabel memiliki alamat memori, dan alamat memori memiliki lokasinya sendiri sendiri yang bisa diakses menggunakan simbol ampersand(&). Pointer adalah sebuah variabel yang berisi alamat memori dari variabel yang lain. Pointer nantinya, bisa mengakses data yang ada di suatu alamat memori.

Sintaks dari bentuk pendeklarasian pointer adalah sebagai berikut :

```
tipe data *nama;
```

Untuk mendeklarasikan pointer pertama tentukan tipe datanya, misal int untuk integer. Setelah itu tambahkan sebuah symbol asterik (\*). Terakhir beri nama pointer tersebut.

Contoh :

1. `int *ip` adalah pointer yang bertipe integr yang memiliki nama ip
2. `double *db` adalah pointer yang bertipe double yang memiliki nama db
3. `float *fp` adalah pointer yang bertipe float dan memiliki nama fp

Nilai dari pointer akan berbentuk bilangan heksadesimal yang menunjukkan alamat memori. Jadi jika mendeklarasikan pointer bertipe integer, float atau apapun akan tetap bernilai heksadesimal.

Contoh :

```
//untuk mengetahui alamat memori dari suatu variabel

#include<stdio.h>
#include<stdlib.h>

int main()
{
    int var1;
    int var2[10];
    printf("alamat variabel var1 adalah %x\n",
           &var1);
    printf("alamat variabel var2 adalah %x\n",
           &var2);
    return 0;
}
```

Contoh :

```
//menggunakan pointer

#include<stdio.h>
#include<stdlib.h>

int main()
{
    int var1 = 20;
    int *ip;

    ip = &var1;

    printf("alamat dari variabel var1 adalah %x\n",
           &var1);

    printf("alamat dari pointer ip adalah %x\n",ip);
    printf("nilai dari pointer *ip adalah %d", *ip);

    return 0;
}
```

## 1.5 STRING

String adalah sebuah array 1 dimensi yang berisi kumpulan data huruf atau karakter dan diakhiri dengan karakter kosong (null). Bahasa pemrograman C memang tidak memiliki tipe data string, tapi kita tetap bisa membuat string dengan tipe data char.

Terdapat beberapa fungsi – fungsi untuk manipulasi string, diantaranya :

1. `strcpy( )` digunakan untuk mengcopy string dari sebuah variabel ke variabel lainnya.
2. `strcat( )` digunakan untuk menambahkan string dari belakang.
3. `strlen( )` digunakan untuk menghitung panjang string.
4. `strcmp( )` digunakan untuk membandingkan string dengan string yang lainnya.
5. `strchr( )` digunakan untuk mencari sebuah karakter di dalam string.
6. `strstr( )` digunakan untuk mencari sebuah teks (string) di dalam string.

Contoh :

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char name[6] = {'J','o','n','n','y','\0'};
    char lastname[] = "Fira";

    char strFullname[13];
    //menampilkan nilai string
    printf("Name %s \n",name);
    printf("Nama Belakang %s \n",lastname);
    //menggunakan fungsi pada string
    strcpy(strFullname,lastname);
```

```
        printf("Nama Lengkap %s \n", strFullname);  
        strcat(name, strFullname);  
        printf("Name = %s \n", name);  
  
        return 0;  
    }
```

## RANGKUMAN

1. Array 1 Dimensi adalah kumpulan elemen yang tersusun dalam suatu baris.
2. Array Multi Dimensi merupakan sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama dan elemen yang akan diakses melalui banyak indeks atau subskrip.
3. Pointer adalah sebuah variabel yang berisi alamat memori dari variabel yang lain.
4. String adalah sebuah array 1 dimensi yang berisi kumpulan data huruf atau karakter dan diakhiri dengan karakter kosong (null).

## TUNTUNAN LATIHAN

Berikut ini diberikan contoh program untuk mencetak nama – nama hari dengan menggunakan array tak berukuran dan perulangan for :

- (a) Mendeklarasikan pre-processor include diikuti dengan file judul atau header file. Caranya mengetikkan langsung pada lembar kerja, yaitu :

```
1  #include<stdio.h>  
2
```

- (b) Mendeklarasikan fungsi main. Caranya mengetikkan langsung pada lembar kerja, yaitu :

```
3  main()  
4  {
```

- (c) Mendeklarasikan variabel i dengan tipe data integer dan mendeklarasikan array hari. Caranya mengetikkan langsung pada lembar kerja, yaitu :



```

5      int i;
6      char hari[][10]={
7          "Senin",
8          "Selasa",
9          "Rabu",
10         "Kamis",
11         "Jum'at",
12         "Sabtu",
13         "Minggu"
14     };
15

```

- (d) Membuat perulangan for untuk menampilkan nama – nama hari dari array yang sudah dibuat. Caranya mengetikkan langsung pada lembar kerja, yaitu :

```

16     for(i=0;i<7;i++){
17         printf("%s\n",hari[i]);
18     }
19 }
20

```

## LATIHAN

1. Lengkapi dan perbaiki program berikut kemudian jalankan programnya !

```

#include <stdio.h>
#include <math.h>

main()
{
    int data,a,z,b;

    /*Buatlah format untuk input masukkan sejumlah data*/
    .....

    int nilai[data];
    /*Buatlah perulangan untuk memasukkan sejumlah data*/
    .....

    for(a=0;a<data;a++)
    {
        for(b=a+1;b<data;b++)
        {
            if(nilai[a]<nilai[b])
            {
                z=nilaib
                nilaib=nilaia
                nilaia=z
            }
        }
    }

    printf("Data urutannya dari terbesar adalah ");
    /*Buatlah perulangan for untuk menampilkan data dari
    terbesar*/
    .....

    getch();
}

```