

Unit 4

Array

kita memiliki tiga variabel dengan tipe data **int** yang memiliki **identifier** yang berbeda untuk tiap variabel.

```
int number1;  
int number2;  
int number3;
```

```
number1 = 1;  
number2 = 2;  
number3 = 3;
```

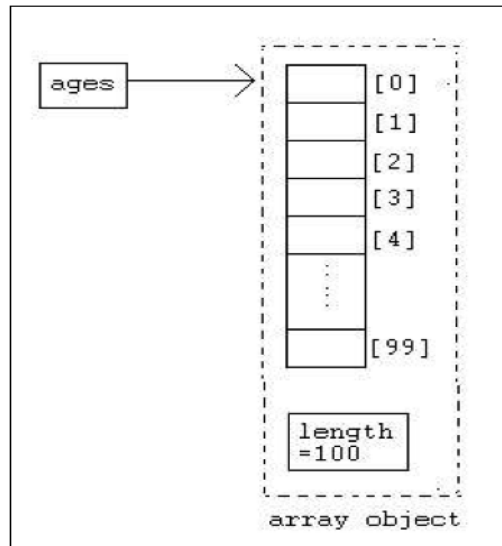
Code di atas bukanlah contoh yang baik untuk memperlihatkan proses penginisialisasian dan penggunaan variabel, terutama jika variabel tersebut digunakan untuk tujuan yang sama.

baik C maupun bahasa pemrograman lain, memiliki kemampuan untuk menggunakan satu variabel yang dapat menyimpan sebuah data list, kemudian memanipulasinya dengan lebih efektif. Tipe variabel inilah yang disebut sebagai array. Sebuah array akan menyimpan beberapa item data dengan tipe data yang sama di dalam sebuah blok memori yang berdekatan yang kemudian dibagi menjadi beberapa slot.

Untuk meng-*instantiate* (atau membuat) sebuah array,

- Tulislah **new** keyword,
- Diikuti oleh tanda kurung [] yang berisi angka dari elemen yang Anda inginkan pada array tersebut.
- Contoh,

```
int ages[];
```



```
int arr[] = {1, 2, 3, 4, 5};
```

Pernyataan ini mendeklarasikan dan meng-*instantiate* sebuah array dari integer dengan lima elemen (diinisialisasi dengan nilai 1, 2, 3, 4, dan 5).

Untuk mengakses sebuah elemen dalam array , atau mengakses sebagian elemen dari array , Anda menggunakan penomoran yang disebut sebagai **index** atau **subscript**.

- Diberikan kepada tiap anggota array, agar program dapat mengakses anggota dari array.
- Dimulai dari nol dan kemudian akan terus bertambah sampai list value dari array tersebut berakhir.
- CATATAN: elemen di dalam array anda dimulai dari 0 sampai (sizeofArray-1)

Contoh, pada array yang telah kita deklarasikan tadi, kita mempunyai

```
//memberikan nilai 10 kepada elemen pertama array
```

```
int ages[0] = 10;
```

```
//mencetak elemen array
```

```
printf(“%d” ,ages[99]);
```

8.1 Buatlah file latihan Array.c tulislah kode sumber berikut

```
int ages[100] ;
void main() {

    //isi data ke array
    for( i=0; i<100; i++ ){
        ages[i]=i;
    }
    //cetak isi data
    for( j=0; j<100; j++ ){
        Printf("%d \n", ages[j] );
    }
}
```

- a. Uji program diatas dan perbaiki kesalahan pada kode sumber tersebut

Kesalahan pada bagian:

Koreksi yang seharusnya:

8.2. Buatlah file operasimatrik.c jelaskan fungsi fungsi yang digunakan serta perbaiki kesalahan yang mungkin terjadi

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define MAXROWS 30
#define MAXCOLS 30
void readinput(int a[][MAXCOLS],int m,int n)
{
    int row,col;
    for (row=0;row<m;++row)
    {
        printf(" Masukan data untuk baris ke. %2d \n",row+1);
        for (col=0;col<n;++col)
            scanf("%d",&a[row][col])
        }
    printf("TABLE 1 \n");
    for (row=0;row<m;++row)
    {
        printf("\n");
        for (col=0;col<n;++col)
            printf("      %d%c",a[row][col],' ')
        }
    }
void largest(int a[][MAXCOLS],int m,int n)
{
    int i,j,largest;
    largest = a[0][0];
    for (i=0;i<m;++i)
    {
        for (j=0;j<n;++j)

            if (a[i][j]>largest)
                largest=a[i][j];
        }
    }
    printf("\n Elemen terbesar matrik adalah %d",largest);
}
```

```

void main(){
    int nrows,ncols;
        int a[MAXROWS][MAXCOLS];
printf("Berapa banyaknya baris matrix? ");
    scanf("%d",&nrows);
    printf("Berapa banyaknya kolom the matrix? ");
    scanf("%d",&ncols);
    printf("Table");
    readinput(a,nrows,ncols);
    largest(a,nrows,ncols);}

```

Kesalahan pada bagian:

Koreksi yang seharusnya:

Luaran uji program

Telah diperiksa	Tanggal :	Modul 8.1, 8.2
Dengan ini praktikan telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada		
Assisten	Praktikan	Nilai

Searching & Sorting

Sorting adalah proses menyusun elemen – elemen dengan tata urut tertentu dan proses tersebut terimplementasi dalam bermacam aplikasi. Kita ambil contoh pada aplikasi perbankan. Aplikasi tersebut mampu menampilkan daftar *account* yang aktif. Hampir seluruh pengguna pada sistem akan memilih tampilan daftar berurutan secara *ascending* demi kenyamanan dalam penelusuran data. Beberapa macam algoritma *sorting* telah dibuat karena proses tersebut sangat mendasar dan sering digunakan. Oleh karena itu, pemahaman atas algoritma –algoritma yang ada sangatlah berguna.

Binary search

9.1. Buatlah Aloritma berikut pada file BinarySearch.c lalu lakukan pengujian

```
#include <stdio.h>

#define TRUE 0
#define FALSE 1

int main(void) {
    int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int left = 0;
    int right = 10;
    int middle = 0;
    int number = 0;
    int bsearch = FALSE;
    int i = 0;

    printf("ARRAY: ");
    for(i = 1; i <= 10; i++)
        printf("[%d] ", i);

    printf("\nNomor yang akan dicari: ");
    scanf("%d", &number);

    while(bsearch == FALSE && left <= right) {
        middle = (left + right) / 2;

        if(number == array[middle]) {
            bsearch = TRUE;
            printf("*** Ketemu **\n");
        } else {
            if(number < array[middle]) right = middle - 1;
            if(number > array[middle]) left = middle + 1;
        }
    }

    if(bsearch == FALSE)
        printf("-- Nomor Tidak ditemukan --\n");
    return 0;
}
```

Luaran uji program

Jelaskan maksud jalannya program beserta pengujian tiap variable pada setiap iterasi yang terjadi

Lanjutan

Bubble Sort

9.2. Buatlah Algoritma berikut pada file BubbleSort.c lalu lakukan pengujian

```
#include <stdio.h>

void bubble_sort(int a[], int size);

int main(void) {
    int arr[10] = {10, 2, 4, 1, 6, 5, 8, 7, 3, 9};
    int i = 0;

    printf("before:\n");
    for(i = 0; i < 10; i++) printf("%d ", arr[i]);
    printf("\n");

    bubble_sort(arr, 10);

    printf("after:\n");
    for(i = 0; i < 10; i++) printf("%d ", arr[i]);
    printf("\n");

    return 0;
}

void bubble_sort(int a[], int size) {
    int switched = 1;
    int hold = 0;
    int i = 0;
    int j = 0;

    size -= 1;

    for(i = 0; i < size && switched; i++) {
        switched = 0;
        for(j = 0; j < size - i; j++)
            if(a[j] > a[j+1]) {
                switched = 1;
                hold = a[j];
                a[j] = a[j + 1];
                a[j + 1] = hold;
            }
    }
}
```

Luaran uji program

Jelaskan maksud jalannya program beserta pengujian tiap variable pada setiap iterasi yang terjadi

Lanjutan

Telah diperiksa	Tanggal :	Modul 8.1, 8.2
Dengan ini praktikan telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada		
Assisten	Praktikan	Nilai