

Sabtu, 23 Oktober 2021

LAPORAN RESMI  
PRAKTIKUM KONSEP PEMOGRAMAN



Entin Martiana Kusumaningtyas S.Kom, M.Kom

Nama : Muhammad Ilham Adi Pratama  
Kelas : D4 – Teknik Informatika  
NRP : 3121600014

## PRAKTIKUM 16 – FUNGSI PENGGOLONGAN VARIABEL DAN PEMOGRAMAN VARIABEL

Percobaan 1: Adakah sesuatu yang salah pada fungsi yang tidak memiliki return value? Jelaskan analisismu tentang fungsi yang tidak mempunyai return value?

Jawab : Tidak ada yang salah pada fungsi yang tidak mempunyai return value, karena fungsi bisa melakukan tugas khusus tanpa memakai return value, tetapi harus memakai tipe data void karena tipe data ini berfungsi mendeklarasikan fungsi yang tidak memiliki return value.

Percobaan 2: Apakah yang terjadi jika sebuah fungsi memberikan return value tetapi tidak diassign ke variabel apapun ?

Jawab : Tidak terjadi masalah atau error, Hasil return akan tetap muncul apabila tidak ada variabel yang meassign return value dan fungsi tersebut memiliki nilai yang sama dengan nilai return valuenya, meskipun begitu lebih baik memakai variabel penampung untuk menampung nilai variabel valuenya.

Percobaan 3: Apakah yang terjadi jika sebuah fungsi di assign ke sebuah variabel padahal fungsi tersebut tidak memiliki return value?

Jawab : Apabila suatu fungsi bertipe void dan tidak memiliki return value namun harus diassign pasti akan error karena fungsi bertipe void sama saja dengan fungsi yang tidak memiliki tipe keluaran. Tetapi, akan berbeda jika fungsi tersebut memiliki tipe keluaran selain void, karena fungsi tersebut memiliki tipe keluaran sehingga nilai dari fungsi tersebut bisa diassign ke suatu variable.

Percobaan 4: Trace secara manual semua program di bawah ini baris per barisnya, dan tampilkan nilai semua variabel pada setiap baris prosesnya. Selain itu, tebaklah tampilkan keluaran programnya

```
a)
int OddEvenTest(int);
main(){
    int a, hasil;
    a = 5;
    hasil = OddEvenTest(a);
    printf("a=%d; hasil=%d\n",a,hasil);
}

OddEvenTest(int b){
    int a;
    a = b % 2;
    return a;
}
```

a	hasil	b
93	71	-
5	69	-
-	-	5
23	-	5
-9	-	5

b)

```
void demo(void);
main () {
    int i=0;
    while(i < 3) {
        demo();
        i++;
    }
}
void demo(void) {
    auto int var_auto = 0;
    static int var_static = 0;
    printf("auto = %d, static = %d\n",
        var_auto, var_static);
    ++var_auto;
    ++var_static;
}
```

i	var_auto	var_static
0	-	-
1	-	-
2	-	-
3	-	-
-	0, 0, 0	0, 1, 2
-	1, 1, 1	1, 2, 3

c)

```
void fung_a(void);
void fung_b(void);
int x = 20;
main () {
    x += 2;
    fung_a();
    fung_a();
    printf("\nNilai x dalam main() = %d\n\n",x);
}
void fung_a (void) {
    static x = 5;
    x++;
    printf("Nilai x dalam fung_a() = %d\n", x);
    fung_b();
}
void fung_b (void) {
    x--;
    printf("Nilai x dalam fung_b() = %d\n", x);
}
```

X
20
22
5
6
21

Percobaan 05: Definisikanlah fungsi main(), masukan() dan average(), sebagai berikut :

- Fungsi masukan() menerima satu parameter berupa jumlah data yang akan dimasukkan dan memberikan return value berupa nilai total dari seluruh data yang dimasukkan. Fungsi ini bertugas menerima masukan data sebanyak n kali dan sekaligus menghitung total nilai seluruh data.
- Fungsi average() menerima dua parameter berupa jumlah data yang telah dimasukkan dan nilai total seluruh data. Fungsi ini memberikan return value berupa nilai rata-rata dari seluruh data yang dimasukkan.
- Pada fungsi main() mintalah masukan jumlah data yang akan diinputkan. Selanjutnya lakukan pemanggilan fungsi masukan() dan average(), kemudian tampilkan nilai rata-rata dari seluruh datanya.

a. Listing Program

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

float masukan (float);
float average (float,float);
float jumlah=0;
void main () {
    float data;
    printf ("Masukkan jumlah data yang akan diinputkan = ");
    scanf ("%f", &data);
    printf ("Jumlah data yang diinputkan = %g\n", masukan(data));
    printf ("Rata rata dari semua data yang diinputkan = %g", average(data,
jumlah));
}

float masukan (float data) {
    int i;
    float nilai;
    for (i=1; i<=data; i++) {
        printf ("Nilai ke-%d = ", i);
        scanf ("%f", &nilai);
        jumlah+=nilai;
    }
    return jumlah;
}

float average (float data, float jumlah) {
    float ratarata=0;
    ratarata=jumlah/data;
    return ratarata;
}
```

b. Output

```
Masukkan jumlah data yang akan diinputkan = 3
Nilai ke-1 = 72
Nilai ke-2 = 65
Nilai ke-3 = 51
Jumlah data yang diinputkan = 188
Rata rata dari semua data yang diinputkan = 62.6667
Process returned 51 (0x33)   execution time : 17.623 s
Press any key to continue.
```

c. Analisis

Disini saya menggunakan 3 fungsi yaitu fungsi masukan, fungsi average bertipe float dan juga fungsi main, saya juga menggunakan variabel global dimana itu saya deklarasikan diluar fungsi main. Fungsi masukan digunakan untuk menerima masukan dari user sedangkan fungsi average digunakan untuk menghitung nilai rata rata dari sejumlah data yang dimasukkan oleh user.

Percobaan 06: Definiskanlah fungsi-fungsi sebagai berikut :

- Fungsi `f_to_i()` untuk mengubah ukuran dari satuan kaki (feet) ke inci
- Fungsi `i_to_cm()` untuk mengubah ukuran dari satuan inci ke centimeter
- Fungsi `c_to_m()` untuk mengubah ukuran dari satuan centimeter ke meter

Dalam `main()` mintalah masukan ukuran dalam satuan kaki (feet) kemudian lakukan konversi sampai mendapatkan keluaran berupa ukuran dalam meter. Tentukan jumlah dan tipe parameter dan return value yang dibutuhkan

Keterangan : 1 kaki = 12 inchi, 1 inchi = 2.54 cm, 100 cm = 1 meter

a. Listing Program

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

float f_to_i (float);
float i_to_cm (float);
float c_to_m (float);

void main () {
    float feet, inchi, cm, m;
    printf ("Masukkan ukuran (feet) = ");
    scanf ("%f", &feet);

    inchi=f_to_i(feet);
    cm=i_to_cm(inchi);
    m=c_to_m(cm);

    printf ("%g kaki = %g inchi\n", feet, inchi);
    printf ("%g inchi = %g cm\n", inchi, cm);
    printf ("%g cm = %g meter", cm, m);
}

float f_to_i (float feet) {
    float x=1;
    x=feet*12;
    return x;
}

float i_to_cm (float inchi) {
    float y=1;
    y=inchi*2.54;
    return y;
}

float c_to_m (float cm) {
    float z=1;
    z=cm*0.01;
    return z;
}
```

b. Output

```
Masukkan ukuran (feet) = 100
100 kaki = 1200 inchi
1200 inchi = 3048 cm
3048 cm = 30.48 meter
Process returned 21 (0x15)   execution time : 6.861 s
Press any key to continue.
```

c. Analisis

Disini saya menggunakan empat fungsi yaitu fungsi `f_to_i`, `i_to_cm`, `cm_to_m` bertipe float dan juga fungsi `main`, fungsi `f_to_i` digunakan untuk mendefinisikan fungsi konversi dari feet ke inchi dengan variabel aktual feet, fungsi `i_to_cm` digunakan untuk mendefinisikan fungsi konversi dari inchi ke

cm dengan variabel aktual inchi, kemudina fungsi cm\_to\_m digunakan untuk mendefinisikan fungsi konversi dari cm ke m dengan variabel aktual cm.

## PRAKTIKUM 18 – ARRAY 1

Percobaan 01: Deklarasikan sebuah variabel array of int, selanjutnya isi array tsb kemudian tampilkan isi variabel tersebut menggunakan statement for(). Ingat apabila jumlah deklarasi index adalah n maka nilai index-nya adalah dimulai dari 0 sampai dengan n-1.

### a. Listing Program

```
#include <stdio.h>
#include <stdlib.h>
#define MAKS 30

void main () {
    int data [MAKS];
    int i, n;

    printf ("Berapa jumlah data yang akan diinputkan ? ");
    scanf ("%d", &n);
    for (i=0; i<n; i++) {
        fflush(stdin);
        printf ("Data ke-%d : ", i+1);
        scanf ("%d", &data[i]);
    }
}
```

### b. Output

```
Berapa jumlah data yang akan diinputkan ? 4
Data ke-1 : 72
Data ke-2 : 59
Data ke-3 : 81
Data ke-4 : 79

Process returned 4 (0x4)   execution time : 12.024 s
Press any key to continue.
```

### c. Analisis

Disini saya mendefinisikan array of int dengan indeks yang sudah saya deklarasikan dalam bentuk konstanta 25

Percobaan 02: Deret fibonacci adalah deret yang dimulai dengan dua angka yang bernilai 0 dan 1, kemudian deret ketiga ditentukan dari penjumlahan kedua angka tersebut, sedangkan deret keempat ditentukan dari dua angka sebelumnya begitu seterusnya. Sehingga didapatkan deret fibonacci sebagai berikut: 0 1 1 2 3 5 8 13 21 ...

Buatlah program untuk meminta input dari user berupa sebuah bilangan, kemudian tampilkan deret fibonacci mulai dari 1 sampai dengan bilangan tsb.

a. Listing Program

```
#include <stdio.h>
#include <stdlib.h>
#define MAKS 35

void main () {
    int i, n, data[MAKS];

    printf ("Masukkan bilangan maksimal: ");
    scanf ("%d", &n);

    for (i=0; i<n; i++){
        if (i<2)
            data[i]=i;
        else
            data[i]=data[i-1]+data[i-2];

        if (data[i]>n)
            break;

        printf ("%d ", data[i]);
    }
}
```

b. Output

```
Masukkan bilangan maksimal: 20
0 1 1 2 3 5 8 13
Process returned 20 (0x14)   execution time : 10.549 s
Press any key to continue.
```

c. Analisis

Disini saya mendefinisikan array of int dengan indeks yang sudah saya deklarasikan dalam bentuk konstanta 35, kemudian program meminta masukan dari user, lalu mengambil hasil masukan dari user dan diteruskan ke looping for, dimana angka yang dimasukkan adalah angka pembatasnya, apabila i kurang dari 2 maka data akan mencetak 0 dan 1, sedangkan jika sebaliknya maka program akan menghitung bilangan ke i, bilangan sebelumnya ditambah dua bilangan sebelumnya. , dan jika banyak data lebih dari batas maka program akan keluar paksa dari looping.



Percobaan 03: Buat penjumlahan dua matriks A[2][2] dengan B[2][2]. Hasil penjumlahan tersebut disimpan dalam matrik C. Tampilkan di layar masing-masing isi dari matriks A, B dan C

Catatan modifikasi (gunakan nested loop) :

- indeks MAKS dibuat konstanta
- variabel ordo matriks diinputkan
- memungkinkan pengulangan proses dengan menginputkan ordo yang lain.

a. Listing Program

```
#include <stdio.h>
#include <stdlib.h>
#define MAKS 2

void main () {
    int i, j, a[MAKS][MAKS], b[MAKS][MAKS];

    for(i=0; i<MAKS; i++){
        for(j=0; j<MAKS; j++){
            printf ("Matriks A ordo [%d] [%d] = ", i+1, j+1);
            scanf ("%d", &a[i][j]);
        }
    }
    printf ("\n");
    for(i=0; i<MAKS; i++){
        for(j=0; j<MAKS; j++){
            printf ("Matriks B ordo [%d] [%d] = ", i+1, j+1);
            scanf ("%d", &b[i][j]);
        }
    }
    printf ("\nMatriks A\n");
    for(i=0; i<MAKS; i++){
        for(j=0; j<MAKS; j++){
            printf ("%d ", a[i][j]);
        }
        printf("\n");
    }
    printf ("\nMatriks B\n");
    for(i=0; i<MAKS; i++){
        for(j=0; j<MAKS; j++){
            printf ("%d ", b[i][j]);
        }
        printf("\n");
    }
    printf ("\nHasil Penjumlahan Matriks A dan Matriks B : \n");
    for(i=0; i<MAKS; i++){
        for(j=0; j<MAKS; j++){
            printf ("%d ", a[i][j]+b[i][j]);
        }
        printf("\n");
    }
}
```

b. Output

```
Matriks A ordo [1] [1] = 1
Matriks A ordo [1] [2] = 2
Matriks A ordo [2] [1] = 3
Matriks A ordo [2] [2] = 4

Matriks B ordo [1] [1] = 5
Matriks B ordo [1] [2] = 6
Matriks B ordo [2] [1] = 7
Matriks B ordo [2] [2] = 8

Matriks A
1 2
3 4

Matriks B
5 6
7 8

Hasil Penjumlahan Matriks A dan Matriks B :
6 8
10 12

Process returned 10 (0xA)   execution time : 7.353 s
Press any key to continue.
```

c. Analisis

Disini saya mendefinisikan array of int dengan indeks yang sudah saya deklarasikan dalam bentuk konstanta 2, ordo adalah baris dikali dengan kolom, variabel I dan j sebagai counter dari looping for

Percobaan 04: Buatlah sebuah variabel array of int yang memiliki ukuran MAKS. Dengan menggunakan looping for(), masukkan sebanyak n data ke dalam array tsb kemudian carilah nilai terbesar yang ada dalam array tsb.

a. Listing Program

```
#include <stdio.h>
#include <stdlib.h>
#define MAKS 30

void main () {
    int i, n, nilai[MAKS], nilai_maks=0;

    printf ("Masukkan banyak nilai : ");
    scanf ("%d", &n);
    for (i=0; i<n; i++){
        printf ("Nilai ke-%d : ", i+1);
        scanf ("%d", &nilai[i]);

        if (nilai[i]>nilai_maks || nilai_maks==0)
            nilai_maks=nilai[i];
    }
    printf ("Nilai Maksimal dari nilai yang diinputkan adalah %d",
    nilai_maks);
}
```

b. Output

```
Masukkan banyak nilai : 5
Nilai ke-1 : 46
Nilai ke-2 : 78
Nilai ke-3 : 43
Nilai ke-4 : 79
Nilai ke-5 : 81
Nilai Maksimal dari nilai yang diinputkan adalah 81
Process returned 51 (0x33)   execution time : 23.880 s
Press any key to continue.
```

c. Analisis

Disini saya mendefinisikan array of int dengan indeks yang sudah saya deklarasikan dalam bentuk konstanta 30, untuk mencari nilai maksimal kita beri kondisi if dengan nilai[i]>nilai maks, dengan inisialisasi awal nilai maks adalah 0