Praktikum 5 (2/5)

FUNGSI

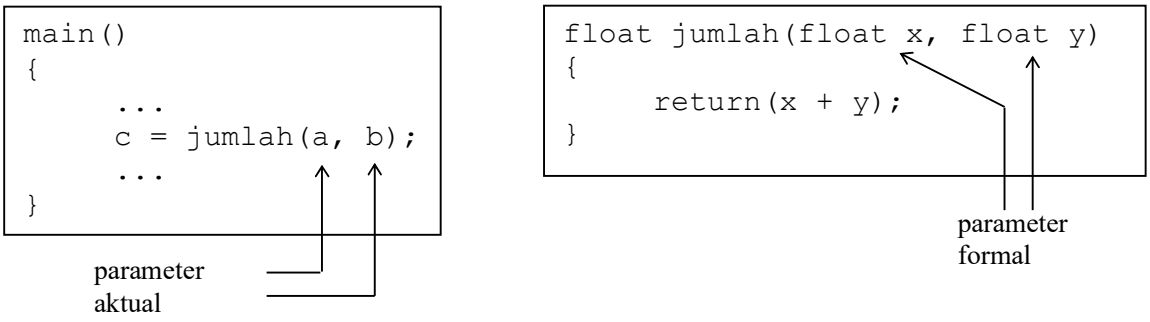
A. TUJUAN

1. Memecah program dalam fungsi fungsi yang sederhana.
2. Menjelaskan tentang pemrograman terstruktur.

B. DASAR TEORI

Parameter Formal dan Parameter Aktual

Parameter formal adalah variabel yang ada pada daftar parameter dalam definisi fungsi. Pada contoh program di atas misalnya, maka dalam fungsi **jumlah()** variabel **x** dan **y** dinamakan sebagai parameter formal. Adapun parameter aktual adalah parameter (tidak selalu berupa variabel) yang dipakai dalam pemanggilan fungsi.



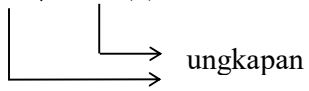
Gambar Paramater formal dan parameter aktual

Pada pernyataan :

```
x = jumlah(a, b);
y = jumlah(20.1, 45.6);
```

a dan **b** merupakan parameter aktual dari fungsi **jumlah()** dalam hal ini parameter berupa variabel. Demikian juga **20.1** dan **45.6** adalah parameter aktual, dalam hal ini berupa konstanta. Bahkan bisa juga parameter aktual berupa ungkapan yang melibatkan operator, misalnya :

```
printf("%g\n", jumlah(2+3, 3+6));
```



C. TUGAS PENDAHULUAN

Buatlah desain flowchart untuk setiap soal dalam percobaan

D. PERCOBAAN

1. a. Definisikan sebuah fungsi `ganjil()` yang memilliki sebuah parameter bilangan bulat dan mengembalikan nilai 1 jika parameter yang diberikan adalah bilangan ganjil dan mengembalikan nilai 0 jika parameter tsb bukan bilangan ganjil
   b. Tulislah prototipe fungsi untuk fungsi tersebut.
   c. Buat function main untuk memanggil function `ganjil()` yang menerima input sebuah bilangan bulat yang akan ditentukan ganjil/genapnya. Tampilkan pesannya (ganjil/genap) dalam `main()`.

2. Buatlah program untuk menghitung faktorial dengan menggunakan 2 fungsi (`main()` dan `faktorial()`). Fungsi `faktorial()` memberikan return value bertipe long int yang akan dicetak ke layar dalam fungsi `main()`.

3. Buatlah fungsi prima(), yang memberikan nilai balik 1 bila bilangan yang dimasukkan adalah prima, dan 0 bila bukan bilangan prima.

4. a. Definisikan sebuah fungsi `radian()` yang berfungsi untuk mengkonversi besaran sudut dari derajat ke radian dengan rumus sbb : rad = drjt / 180.0f * PI. Fungsi tersebut memiliki sebuah parameter yaitu derajat yang akan dikonversi, dan memiliki sebuah *return value* berupa hasil konversi dalam radian.
   b. Tulislah prototipe fungsi untuk fungsi tersebut.
   c. Buat function main untuk memanggil function `radian()`, setelah sebelumnya meminta masukan nilai derajat yang akan dikonversi.
   d. Definisikan PI sebagai sebuah konstanta yang bernilai : `3.14159f`

5. a. Definisikan sebuah fungsi `float konversi(suhu, asal, tuj)`, untuk mengkonversikan suhu dari Celsius ke Fahrenheit, Celsius ke Reamur, Fahrenheit ke Celsius, Fahrenheit ke Reamur, Reamur ke Celsius, dan Reamur ke Fahrenheit. Dimana suhu adalah suhu sumber, asal adalah satuan awal suhu yang akan dikonversi dan tuj adalah satuan hasil konversi

   b. Tulislah prototipe fungsi untuk fungsi tersebut.

   c. Buat function main() untuk memanggil function konversi`()`, setelah sebelumnya meminta masukan nilai suhu, satuan asal dan satuan tujuannya.

   Contoh tampilan:

```
Masukkan suhu sumber = 100
Masukkan satuan asal = C
Masukkan satuan tujuan = R

Hasil konversi suhu 100 C = 80 R
```

E. LAPORAN RESMI

1. Tulis listing program dari semua percobaan yang dilakukan.

2. Kemudian tuliskan outputnya. Terangkan mengapa demikian.

3. Apa hasil eksekusi dari program berikut :

```
#include <stdio.h>

void ubah(int);

main()
{
   int x;

   printf("Masukkan nilai x : ");
   scanf("%d", &x);
   ubah(x);
   printf("x = %d\n", x);
}

void ubah(int y)
{
   y = 85;
}
```

Praktikum 5 (4/5)

FUNGSI

A. TUJUAN

1. Mengetahui perbedaan antara variabel lokal, eksternal, statis dan register

B. DASAR TEORI

Penggolongan Variabel berdasarkan Kelas Penyimpanan

Suatu variabel, di samping dapat digolongkan berdasarkan jenis/tipe data juga dapat diklasifikasikan berdasarkan kelas penyimpanan *(storage class).* Penggolongan berdasarkan kelas penyimpanan berupa :

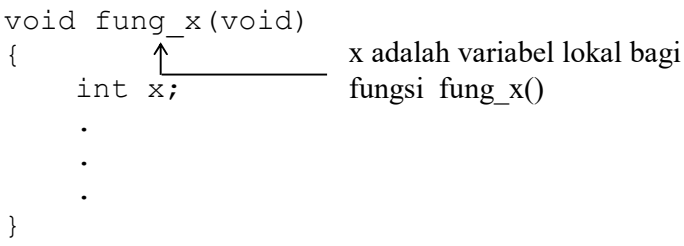
- variabel lokal
- variabel eksternal
- variabel statis
- variabel register

Variabel Lokal

Variabel lokal adalah variabel yang dideklarasikan dalam fungsi, dengan sifat :

- secara otomatis diciptakan ketika fungsi dipanggil dan akan sirna (lenyap) ketika eksekusi terhadap fungsi berakhir.
- Hanya dikenal oleh fungsi tempat variabel tersebut dideklarasikan
- Tidak ada inisialisasi secara otomatis (saat variabel diciptakan, nilainya tak menentu).

Dalam banyak literatur, variabel lokal disebut juga dengan variabel otomatis. Variabel yang termasuk dalam golongan ini bisa dideklarasikan dengan menambahkan kata kuci *auto* di depan tipe-data variabel. Kata kunci ini bersifat opsional, biasanya disertakan sebagai penjelas saja. Contoh variabel lokal ditunjukkan pada gambar 5.8.



Gambar 5.8 Variabel lokal

Pada **fung_x()**, deklarasi

```
int x;
```

dapat ditulis menjadi

```
auto int x;
```

Penerapan variabel lokal yaitu bila variabel hanya dipakai oleh suatu fungsi (tidak dimaksudkan untuk dipakai oleh fungsi yang lain). Pada contoh berikut, antara variabel **i** dalam fungsi **main()** dan **fung_1()** tidak ada kaitannya, sebab masing-masing merupakan variabel lokal.

Variabel Eksternal

Variabel eksternal merupakan variabel yang dideklarasikan di luar fungsi, dengan sifat :

- dapat diakses oleh semua fungsi
- kalau tak diberi nilai, secara otomatis diinisialisasi dengan nilai sama dengan nol.

Variabel eksternal haruslah dideklarasikan sebelum definisi fungsi yang akan mempergunakannya. Untuk memperjelas bahwa suatu variabel dalam fungsi merupakan variabel eksternal, di dalam fungsi yang menggunakannya dapat mendeklarasikan variabel itu kembali dengan menambahkan kata kunci *extern* di depan tipe data variabel.

Kalau dalam suatu program terdapat suatu variabel eksternal, suatu fungsi bisa saja menggunakan nama variabel yang sama dengan variabel eksternal, namun diperlakukan sebagai variabel lokal.

Variabel Statis

Variabel statis dapat berupa variabel internal (didefinisikan di dalam fungsi) maupun variabel eksternal. Sifat variabel ini :

- Kalau variabel statis bersifat internal, maka variabel hanya dikenal oleh fungsi tempat variabel dideklarasikan
- Kalau variabel statis bersifat eksternal, maka variabel dapat dipergunakan oleh semua fungsi yang terletak pada file yang sama, tempat variabel statis dideklarasikan
- Berbeda dengan variabel lokal, variabel statis tidak akan hilang sekeluarnya dari fungsi (nilai pada variabel akan tetap diingat).
- Inisialisasi akan dilakukan hanya sekali, yaitu saat fungsi dipanggil yang pertama kali. Kalau tak ada inisialisasi oleh pemrogram secara otomatis akan diberi nilai awal nol

Variabel statis diperoleh dengan menambahkan kata kunci *static* di depan tipe data variabel.

Variabel Register

Variabel register adalah variabel yang nilainya disimpan dalam register dan bukan dalam memori RAM. Variabel yang seperti ini hanya bisa diterapkan pada variabel yang lokal atau parameter formal, yang bertipe *char* atau *int*. Variabel register biasa diterapkan pada variabel yang digunakan sebagai pengendali *loop*. Tujuannya untuk mempercepat proses dalam *loop*. Sebab variabel yang dioperasikan pada register memiliki kecepatan yang jauh lebih tinggi daripada variabel yang diletakkan pada RAM.

C. PERCOBAAN

Lakukan percobaan-percobaan untuk bisa menjawab semua pertanyaan di bawah ini, analisislah dan tuliskan alasannya

1. Adakah sesuatu yang salah pada sebuah fungsi yang tidak mempunyai return value ? Jelaskan analisismu tentang sebuah fungsi yang tidak memiliki return value!

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

3. Apakah yang terjadi jika sebuah fungsi diassign ke sebuah variabel padahal fungsi tersebut tidak memiliki return value ?

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

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

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

D. LAPORAN RESMI

1. Tulis listing program dari semua percobaan yang dilakukan.

2. Kemudian tuliskan outputnya. Terangkan mengapa demikian.