

**MODUL PRAKTIKUM**  
**ALGORITMA PEMROGRAMAN 1**  
**SEMESTER GANJIL 2018/2019**



Institut Teknologi Sumatera

2018

## PETUNJUK PRAKTIKUM

### Tahap Pelaksanaan Praktikum

Praktikum dilaksanakan dalam 2 tahap yang dilaksanakan secara berturut-turut, yaitu:

1. Latihan (opsional)
2. Tugas Praktikum

#### Latihan

Tahap ini bersifat **opsional!** Bisa dilakukan secara formal di lab, bisa juga dilakukan di luar lab. Dilakukan sebelum Tugas Praktikum yang sebenarnya dikerjakan.

<b>Tujuan</b>	Mahasiswa berlatih mengetikkan contoh-contoh program yang sesuai dengan topik yang diberikan dan mengamati hasil eksekusi sekaligus untuk membiasakan dengan lingkungan praktikum
<b>Durasi</b>	±60 menit
<b>Sifat</b>	individual (boleh berdiskusi dengan teman dan asisten)
<b>Pelaksanaan</b>	Mahasiswa mengerjakan latihan-latihan soal yang terdapat pada bab LATIHAN: <ol style="list-style-type: none"><li>1. Baca petunjuk pada soal-soal latihan.</li><li>2. Salin contoh-contoh program yang ada.</li><li>3. <i>Compile</i> dan buatlah <i>executable file</i>, lalu eksekusi/jalankan program (jika kompilasi berhasil).</li><li>4. Amati hasil eksekusi dan bandingkan dengan petunjuknya.</li><li>5. Bahan latihan dapat dipilih sesuai dengan waktu yang tersedia.</li></ol>

#### Tugas Praktikum

**WAJIB** dilakukan dilakukan di lab!

<b>Tujuan</b>	Mahasiswa mengerjakan tugas-tugas praktikum secara mandiri sesuai dengan topik yang diberikan dan mengumpulkan hasilnya untuk penilaian
<b>Durasi</b>	±100 menit
<b>Sifat</b>	individual (tidak diperkenankan bekerja sama dengan pihak mana pun), tetapi masih boleh membuka material kuliah miliknya sendiri
<b>Pelaksanaan</b>	Mahasiswa mengerjakan soal-soal praktikum untuk modul terkait yang terdapat pada bab TUGAS PRAKTIKUM sesuai dengan petunjuk yang ada dalam durasi yang ditetapkan.

## Petunjuk Penamaan dan Penulisan File Program (Untuk Peserta)

1. Pada setiap soal baik latihan maupun tugas praktikum, perhatikan petunjuk penamaan file.
2. Pada petunjuk penamaan file, gantilah <NIM> dengan NIM Anda masing-masing dan XX dengan nomor soal dalam 2 digit.

Contoh:

Jika file harus disimpan dengan format: **LP1\_Hello1\_<NIM>.cpp** dan NIM Anda adalah **19999888**, maka nama file Anda adalah: **LP1\_Hello1\_19999888.cpp**

Jika file harus disimpan dengan format: **P1\_<NIM>\_XX.cpp** dan NIM Anda adalah **19999888** serta nomor soal yang sedang dikerjakan adalah **2**, maka nama file Anda adalah: **P1\_19999888\_02.cpp**

3. Untuk setiap file source code program berikan identitas, minimum:

```
// NIM>Nama :  
// Nama file :  
// Tanggal :  
// Deskripsi :
```

4. Simpan dan upload file source code hasil latihan dan praktikum pada direktori yang ditentukan asisten.

## Petunjuk Kompilasi Menggunakan GNU C++

Untuk petunjuk membuka lingkungan *compiler*, silakan berkonsultasi dengan asisten.

Contoh nama file source : **hello.cpp**

*Compile* program dan membentuk *executable file*, misalnya **hello.exe**:

```
>> g++ -c hello.cpp  
>> g++ -c hello.exe hello.o
```

*Compile* program sekaligus membentuk *executable file*, misalnya **hello.exe**:

```
>> g++ -o hello.exe hello.cpp
```

Pemanggilan program, misalnya **hello.exe**:

```
>> hello
```

## Petunjuk untuk Pengelola Praktikum

1. Untuk penyerahan materi praktikum ke mahasiswa, bagian LATIHAN dan bagian TUGAS PRAKTIKUM harus diletakkan dalam file terpisah.
2. LATIHAN bisa diletakkan dan dibuka kapan saja sepanjang semester sementara TUGAS PRAKTIKUM hanya dibuka pada saat pelaksanaan praktikum saja.
3. Satu sesi praktikum (100 menit) harap diberikan hanya satu versi TUGAS PRAKTIKUM.
4. Form penilaian praktikum untuk satu mahasiswa dapat dilihat di akhir dokumen ini.

# Perulangan

Macam-macam struktur pengulangan :

## 1. Struktur FOR

Struktur pengulangan FOR digunakan untuk mengulang statemen atau satu blok statemen berulang kali. Jumlah pengulangan diketahui atau dapat ditentukan sebelum eksekusi. Untuk mencacah sudah jumlah pengulangan diperlukan sebuah variabel pencacah (counter). Variabel ini nilainya selalu bertambah satu setiap kali pengulangan dilakukan. Jika cacah pengulangan sudah mencapai jumlah yang dispesifikasikan, maka proses pengulangan berhenti. Pada struktur FOR, pencacah haruslah dari tipe data yang memiliki predecessor dan successor, yaitu integer atau karakter. Tipe riil tidak dapat digunakan sebagai pencacah. Aksi adalah satu atau lebih instruksi yang diulang. Bentuk struktur FOR ada dua macam:

### a. Menaik (ascending)

Pada struktur FOR menaik, *nilai\_awal* harus lebih kecil atau sama dengan *nilai\_akhir*. Jika *nilai\_awal* lebih besar dari *nilai\_akhir*, maka badan pengulangan tidak dimasuki. Pada awalnya, pencacah diinisialisasikan dengan *nilai\_awal*. Nilai pencacah secara otomatis bertambah satu setiap kali aksi pengulangan dimasuki, sampai akhirnya nilai pencacah sama dengan *nilai\_akhir*. Jumlah pengulangan yang terjadi =  $nilai\_akhir - nilai\_awal + 1$ .

```
for (pencacah = nilai_awal; pencacah <= nilai_akhir; pencacah++){  
    aksi  
}
```

### b. Menurun (descending)

Pada struktur FOR menurun, *nilai\_akhir* harus lebih besar atau sama dengan *nilai\_awal*. Jika *nilai\_akhir* lebih kecil dari *nilai\_awal*, maka badan pengulangan tidak dimasuki. Pada awalnya, pencacah diinisialisasikan dengan *nilai\_akhir*. Nilai pencacah secara otomatis berkurang satu setiap kali aksi diulangi, sampai akhirnya nilai pencacah sama dengan *nilai\_awal*. Jumlah pengulangan yang terjadi =  $nilai\_akhir - nilai\_awal + 1$ .

```
for (pencacah = nilai_awal; pencacah >= nilai_akhir; pencacah--){  
    aksi  
}
```

## 2. Struktur WHILE

Struktur pengulangan WHILE adalah pengulangan yang melakukan pemeriksaan di awal blok struktur. Aksi (atau runtunan aksi) akan dilaksanakan berulang kali selama kondisi yang didefinisikan di dalamnya terpenuhi (bernilai true). Jika kondisi tidak terpenuhi (bernilai false), badan pengulangan tidak akan dilaksanakan, yang berarti pengulangan selesai. Yang harus diperhatikan adalah pengulangan harus berhenti. Pengulangan yang tidak pernah berhenti menandakan bahwa logika pemrograman tersebut salah. Pengulangan berhenti apabila kondisi bernilai false. Agar kondisi suatu saat bernilai false, maka di dalam badan pengulangan harus ada instruksi yang mengubah nilai variabel kondisi.

```
inisiasi first element

while (kondisi){
    aksi;
    next element;
}
```

## 3. Struktur DO – WHILE

Struktur ini mendasarkan pengulangan pada kondisi boolean. Aksi di dalam badan pengulangan diulang sampai kondisi boolean bernilai true. Dengan kata lain, jika kondisi boolean masih false, pengulangan masih terus dilakukan. Karena proses pengulangan suatu saat harus berhenti, maka di dalam badan pengulangan harus ada aksi yang mengubah nilai variabel kondisi. Berbeda dengan struktur WHILE yang melakukan pemeriksaan kondisi di awal blok pengulangan, pada struktur DO-WHILE kondisi justru ditempatkan di bagian akhir.

```
inisiasi first element

do{
    aksi;
    next element;
} while (kondisi);
```

# Sub-Program Fungsi

Fungsi adalah sebuah blok program yang merupakan sekumpulan instruksi yang bertujuan untuk menyelesaikan suatu tugas khusus. Sebuah fungsi dibuat untuk membantu mengerjakan tugas yang kompleks secara efektif dan efisien.

Setelah satu fungsi dibuat, ia dapat dipanggil dibagian program manapun untuk menyelesaikan suatu tugas secara-berulang ulang. Selain itu fungsi dibuat untuk memecah program yang kompleks menjadi sub-sub program yang lebih kecil, hal ini diharapkan dapat memudahkan manajemen pembuatan program.

## Struktur Umum

```
TipeKeluaran functionName ( argumen ) {  
    ...  
    kumpulan instruksi  
    ...  
    return (sesuai tipe keluaran);  
}
```

Struktur umum dari fungsi diatas dapat dijelaskan sebagai berikut:

- Tipe keluaran menjelaskan mengenai jenis nilai yang diberikan oleh apabila fungsi tersebut dipanggil.
- Nama fungsi merupakan deklarasi nama fungsi yang akan kita buat. Dalam deklarasinya, fungsi harus memenuhi syarat-syarat sebagai berikut:
  - Hanya terdiri dari huruf, angka dan garis bawah (*underscore*).
  - Tidak boleh diawali dengan angka.
- Argumen adalah data atau nilai masukan yang akan diolah oleh fungsi., Argumen bersifat optional, namun tanda kurung harus tetap ditampilkan.
- Tubuh fungsi merupakan kumpulan instruksi yang akan dikerjakan oleh fungsi.

## LATIHAN

### Latihan 1. Perulangan FOR

Nama file: LP7\_<NIM>\_01.cpp

#### Deskripsi:

Program untuk menuliskan angka 1 sampai dengan 5 secara berurutan.

#### Program:

```
#include <iostream>
using namespace std;

int main(){
    int i;
    for (i=1; i<5; i++){
        .....
        cout << i << endl;
    }

    return 0;
}
```

#### Output:

```
1
2
3
4
5
```

## Latihan 2. Perulangan FOR Bersarang

Nama file: LP7\_<NIM>\_04.cpp

### Deskripsi:

Sama halnya seperti pada percabangan, pada struktur pengulangan juga dapat diterapkan pengulangan bersarang (nested looping). Konsepnya dalam sebuah pengulangan terdapat pengulangan yang lain..

### Program:

```
#include <iostream>
using namespace std;

int main(){
    for (int i=1; i<=10; i++){
        for (int j=1; j<=i; j++){
            cout << i*j << " ";
        }
        cout << endl;
    }
}
```

### Output :

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 73 81
10 20 30 40 50 60 70 80 90 100
```



### Latihan 3. Fungsi dengan satu buah argumen

Nama file: LP6\_Konversi\_<NIM>.cpp

#### Deskripsi:

Buatlah program yang digunakan untuk mengubah nilai suhu dari Fahrenheit ke Celcius. Input suhu di dapatkan dari pengguna.

#### Program:

```
#include <iostream>
using namespace std;

float konversi(float x){
    float hasil;
    hasil = ((x - 32) * 5) / 9;
    return hasil;
}

int main(){
    float celc, fahr;

    cout << "Masukkan nilai suhu dalam fahrenheit: ";
    cin >> fahr;

    celc = konversi(fahr);
    cout << "Nilai suhu dalam celcius adalah: " << celc;

    return 0;
}
```

#### Contoh Input dan Output:

Input	Output
Masukkan nilai suhu dalam fahrenheit: 122	Nilai suhu dalam celcius adalah: 50
Masukkan nilai suhu dalam fahrenheit: 50	Nilai suhu dalam celcius adalah: 10
Masukkan nilai suhu dalam fahrenheit: 5	Nilai suhu dalam celcius adalah: -15

#### Latihan 4. Fungsi dengan dua buah argumen

Nama file: LP6\_Perkalian\_<NIM>.cpp

#### Deskripsi:

Buatlah program yang digunakan untuk menghitung luas bangunan. Input nilai panjang dan lebar sisi persegi panjang didapatkan dari pengguna.

#### Program:

```
#include <iostream>
using namespace std;

void perkalian(int a, int b){
    int hasil;
    hasil = a*b;
    return hasil;
}

int main(){
    int bil1, bil2;

    cout << "Bilangan 1 : ";
    cin >> bil1;
    cout << "Bilangan 2 : ";
    cin >> bil2;

    cout << "Hasil = " << perkalian(bil1, bil2);

    return 0;
}
```

#### Contoh Input dan Output:

Input	Output
5 4	20
6 7	42