

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.

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

Tugas Praktikum

WAJIB dilakukan dilakukan di lab!

Tujuan	Mahasiswa mengerjakan tugas-tugas praktikum secara mandiri sesuai dengan topik yang diberikan dan mengumpulkan hasilnya untuk penilaian
Durasi	±100 menit
Sifat	individual (tidak diperkenankan bekerja sama dengan pihak mana pun), tetapi masih boleh membuka material kuliah miliknya sendiri
Pelaksanaan	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);
```

LATIHAN

Latihan 1. Perulangan FOR

Nama file: LP3_<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 WHILE

Nama file: LP3_<NIM>_02.cpp

Deskripsi:

Program untuk menuliskan angka 1 sampai dengan 5 secara berurutan.

Program:

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

int main(){
    int i = 1;

    while (i<=5){
        cout << i << endl;
        i++;
    }
    return 0;
}
```

Output:

```
1
2
3
4
5
```

Latihan 3. Perulangan DO-WHILE

Nama file: LP3_<NIM>_03.cpp

Deskripsi:

Program untuk menuliskan angka 1 sampai dengan 5 secara berurutan.

Program:

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

int main(){
    int i = 1;

    do{
        cout << i << endl;
        i++;
    } while (i<=5);

    return 0;
}
```

Output :

```
1
2
3
4
5
```


Latihan 4. Perulangan FOR Bersarang

Nama file: LP3_<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
```