

PANDUAN PRAKTIKUM  
***ALGORITMA DAN PEMROGRAMAN 1***

Edisi Revisi



Direvisi Oleh :

Femi Dwi Astuti, S.Kom.

**STMIK AKAKOM**

Yogyakarta  
2015

## KATA PENGANTAR

Puji Syukur kepada Allah SWT karena revisi modul Algoritma dan Pemrograman 1 ini dapat terselesaikan. Modul ini direvisi karena tuntutan kurikulum baru atau disebut sebagai kurikulum 2014 STMIK AKAKOM yang mulai diberlakukan pada tahun akademik 2014-2015.

Modul ini membahas tentang dasar-dasar pemrograman dengan menggunakan bahasa Pemrograman C++. Mulai dari pengantar algoritma, dasar-dasar pemrograman, sekuensial, percabangan, perulangan dan gabungan antara percabangan dan perulangan sampai kepada penerapannya dengan menggunakan variasi bentuk perintah yang ada.

Tentu saja dengan segala keterbatasan yang ada, modul ini jauh dari sempurna. Untuk itu masukan dan saran untuk perkembangan modul ini sangat diharapkan.

Akhirnya semoga modul ini berguna dalam membantu mahasiswa untuk mempelajari tentang Algoritma dan Pemrograman.

Yogyakarta, Agustus 2015

Penulis

## PERTEMUAN KE – 1 PENGANTAR ALGORITMA

### A. TUJUAN

Mahasiswa dapat menyusun algoritma untuk menyelesaikan suatu permasalahan

### B. TEORI SINGKAT

#### Algoritma

Algoritma adalah langkah – langkah logis tertentu untuk menyelesaikan suatu masalah. Guna algoritma adalah untuk membantu seseorang dalam menyelesaikan suatu masalah berdasarkan pada pola pikirnya masing-masing.

#### Ciri – ciri algoritma:

1. Ada masukan (*input*).
2. Ada proses.
3. Ada keluaran (*output*).
4. Memiliki perintah yang jelas dan tidak membingungkan (*ambigu*).
5. Harus mempunyai tanda akhir (*stopping role*)

#### Sifat algoritma :

1. Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
2. Tidak tergantung pada suatu bahasa pemrograman.
3. Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.

Algoritma dapat digunakan untuk merepresentasikan suatu urutan kejadian secara logis dan dapat diterapkan di semua kejadian sehari-hari.

Jadi algoritma adalah jembatan untuk mempermudah pemahaman alur kerja suatu proses.

### C. PRAKTIK

#### Praktik 1.

Contoh kasus dalam kehidupan sehari-hari:

##### 1. Algoritma memasak mie instan.

1. Rebus air hingga mendidih.
2. Masukkan mie instan ke dalam air mendidih tersebut.
3. Tunggu beberapa waktu hingga mie terlihat matang.
4. Jika mie sudah dirasa matang, angkat dan tiriskan.
5. Campurkan bumbu-bumbu, dan aduk hingga rata.
6. Mie siap disajikan/dimakan

##### 2. Algoritma pendaftaran Calon Mahasiswa Baru di STMIK AKAKOM.

1. Membeli Formulir di Bagian Penerimaan mahasiswa baru.
2. Mengisi dan menyerahkan formulir ke Bagian Penerimaan mahasiswa baru
3. Mendapatkan No test bagi yang lewat jalur test.
4. Melakukan Test
5. Jika Test sudah selesai, mendapatkan hasil pengumuman test.
6. Jika mahasiswa dinyatakan diterima diberi surat diterima beserta rincian pembayaran

##### 3. Algoritma menghitung luas persegi panjang.

1. Masukkan panjang
2. Masukkan lebar
3. Nilai luas adalah panjang dikalikan(\*) lebar
4. Tampilkan luas

##### 4. Algoritma menghitung sisi miring segitiga siku - siku.

1. Masukkan nilai sisi a
2. Masukkan nilai sisi b
3. Hitung nilai  $c^2 = a^2 + b^2$
4. Hitung nilai c

5. Tampilkan nilai c

### 5. Algoritma menghitung Luas Segi Tiga

1. Masukkan alas
2. Masukkan tinggi
3. Hitung luas dengan cara kalikan alas dengan tinggi lalu kalikan dengan 0.5 ( $0.5 \cdot \text{alas} \cdot \text{tinggi}$ )
4. Cetak luas

### Praktik 2.

Buatlah algoritma untuk menyelesaikan permasalahan berikut

1. Bidang Akademik  
Akan dibuat rerata nilai dari sejumlah siswa yang mengikuti Test, buatlah algoritma sehingga dihasilkan hasil akhir berupa :
  - 1). Nilai total semua siswa yang mengikuti test
  - 2). Rerata nilai dari siswa yang mengikuti Test.
2. Bidang Akuntansi  
Hitung jumlah yang harus dibayar oleh seorang pembeli yang membeli satu(1) atau beberapa barang dimana setiap barang yang dibeli memiliki harga sesuai dengan item barangnya, sehingga dihasilkan hasil akhir berupa TotalBayar yang harus dibayarkan oleh pembeli
3. Bidang Administrasi
  - a) Buat algoritma untuk menentukan criteria kelulusan administrasi jika disyaratkan seseorang dinyatakan LENGKAP administrasi jika telah melengkapi data-data sebagai berikut :
    - 1) Akte Kelahiran
    - 2) STTB
    - 3) Nilai UAN  $> 5$Jika salah satu atau ke-3 data tersebut tidak terpenuhi maka kelengkapan Administrasi dinyatakan KURANG LENGKAP.
  - b) Dalam perekrutan anggota kepolisian disyaratkan beberapa hal sebagai berikut :
    1. Umur Minimal 19 Tahun, Maksimal 20 Tahun
    2. Tinggi Badan Minimal 170 cm
    3. Berat Badan Ideal ( $\text{Berat Badan Ideal} = (\text{Tinggi Badan} - 100) - 10\% (\text{Tinggi Badan} - 100)$ )
    4. Hasil Akhir berupa LULUS jika ke-3 persyaratan tersebut terpenuhi, jika salah satu persyaratan tidak terpenuhi maka dinyatakan GAGAL

### D. LATIHAN

Diberikan oleh masing-masing dosen pengampu praktikum

### E. TUGAS

Diberikan oleh masing-masing dosen pengampu praktikum

## PERTEMUAN KE – 2 PENYAJIAN ALGORITMA

### A. TEORI SINGKAT

Berdasarkan definisi dari algoritma yaitu langkah-langkah dalam menyelesaikan masalah yang tersusun secara logis dan sistematis, maka penyajian algoritma dapat disajikan dalam bentuk:

1. Menggunakan Bahasa Formal (Inggris, Indonesia, dll)
2. Menggunakan **FlowChart** ( Diagram Alir )
3. Menggunakan **Pseudo-Code**

#### Penyajian Menggunakan Bahasa Formal

##### Aturan Penulisan Teks Algoritma

Setiap algoritma akan selalu terdiri dari :

- 1) Judul (header)
- 2) Deklarasi (kamus)
- 3) Deskripsi Algoritma

#### 1) JUDUL

**JUDUL ALGORITMA** → Algoritma NAMA ALGORITMA

{Penjelasan tentang algoritma, berisi uraian singkat cara kerja program, kondisi awal dan akhir dari program} → spesifikasi algoritma

Catatan, dalam menulis nama-nama dalam algoritma harus mempunyai makna yang mencerminkan proses, sifat atau identitas lainnya yang melekat dengan suatu proses, tipe, konstanta, variabel, sub-program dan lain-lainnya. Nama-nama yang bermakna disebut **mnemonic**.

#### 2) DEKLARASI

(Semua nama yang dipakai, meliputi nama file, nama variable, nama konstanta, nama prosedur serta nama fungsi )

#### 3) DESKRIPSI

(Semua langkah/aksi algoritma )

##### Contoh:

1). **judul algoritma:** Algoritma Luas\_Lingkaran

{ Menghitung luas lingkaran dengan ukuran jari-jari tertentu .Algoritma menerima masukan jejari lingkaran, menghitung luasnya, dan menyajikan hasilnya ke piranti keluaran }

2) **Deklarasi algoritma:**

DEKLARASI { nama konstanta }

const PHI = 3.14; { Nilai phi = 22/7 }

{ **nama peubah** }

var R : real; { input jejari lingkaran bilangan riil }

luas\_Ling : real; { luas lingkaran bilangan riil }

{ **nama sub program** }

procedure TUKAR (input/output A:integer, input/output B:integer)

{Mempertukarkan nilai A dan B.Parameter A dan B sudah terdefinisi

nilainya.Setelah pertukaran, A berisi nilai B dan B berisi nilai A }

3) **Deskripsi algoritma:**

{ Baca data jejari lingkaran R.Jika  $R \leq 0$  tulis pesan data salah, selain itu hitung luas ingkaran. Tampilkan luas lingkaran. }

baca(R);

jika  $R \leq 0$  then tulis ("Data salah !") selain itu luas\_Ling = PHI x R x R;

tulis(luas\_Ling);

Catatan : contoh penyajian algoritma dalam bentuk struktur bahasa ada di modul ke-1.

### Structure Chart (bagan terstruktur)

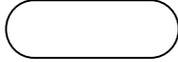
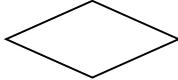
- Digunakan untuk mendefinisikan dan mengilustrasikan organisasi dari system secara berjenjang
- Berbentuk modul dan submodul
- Menunjukkan hubungan elemen data dan elemen control serta hubungan antar modul

Dapat disajikan dalam bentuk :

1. Flowchart
2. Activity Diagram (pendekatan Berorientasi Obyek)

### FLOWCHART

Flowchart adalah serangkaian bagan-bagan yang menggambarkan alir program. Flowchart atau diagram alir memiliki bagan-bagan yang melambangkan fungsi tertentu. Bagan, nama dan fungsinya seperti yang disajikan pada tabel berikut :

BAGAN	NAMA	FUNGSI
	<b>TERMINATOR</b>	Awal atau akhir program
	<b>FLOW</b>	Arah aliran program
	<b>PREPARATION</b>	inisialisasi/pemberian nilai awal
	<b>PROCES</b>	Proses/pengolahan data
	<b>INPUT/OUTPUT DATA</b>	input/output data
	<b>SUB PROGRAM</b>	sub program
	<b>DECISION</b>	Seleksi atau kondisi
	<b>ON CONNECTOR</b>	Penghubung bagian-bagian flowchart pada halaman yang sama
	<b>OFF CONNECTOR</b>	Penghubung bagian-bagian flowchart pada halaman yang berbeda
	<b>COMMENT</b>	Tempat komentar tentang suatu proses

Flowchart selalu diawali dan diakhiri oleh bagan terminator. Aliran selalu dari atas ke bawah, satu demi satu langkah. Tidak ada proses yang dikerjakan bersamaan, semua dikerjakan satu persatu.

Proses yang dilakukan komputer sebenarnya hanya ada 3 proses: input, proses data dan output. Dengan demikian, ketika ada suatu masalah yang akan diselesaikan dengan suatu software, maka hal yang perlu diidentifikasi adalah input, proses data dan output.

**Contoh 1:**

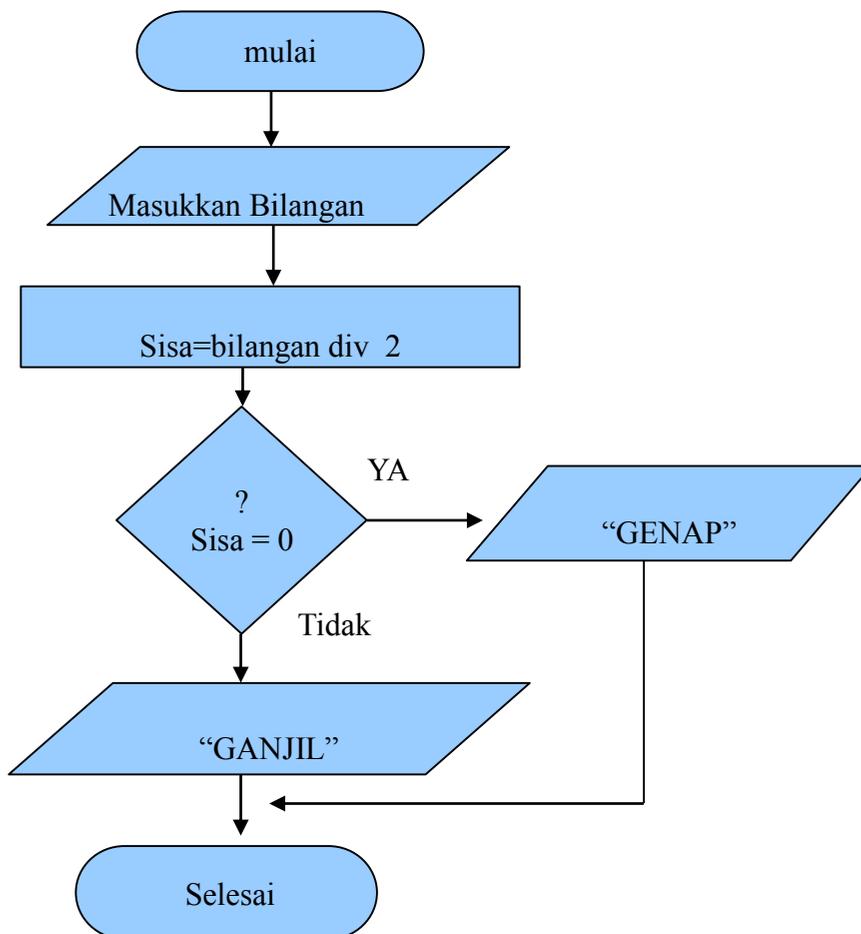
bagaimana menentukan bahwa suatu bilangan itu adalah bilangan genap atau ganjil?

Input : bilangan.

Proses : 1). menentukan bilangan ganjil atau genap dengan melakukan pembagian bilangan dengan bilangan  
2). Jika sisa pembagian NOL maka bilangan tersebut genap, sebaliknya jika sisanya 1, maka bilangan ganjil.

Output : Bilangan Ganjil atau Bilangan Genap

Gambaran penyelesaian dengan Flowchart :



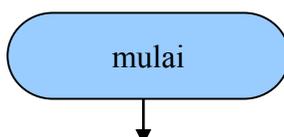
**Contoh 2:**

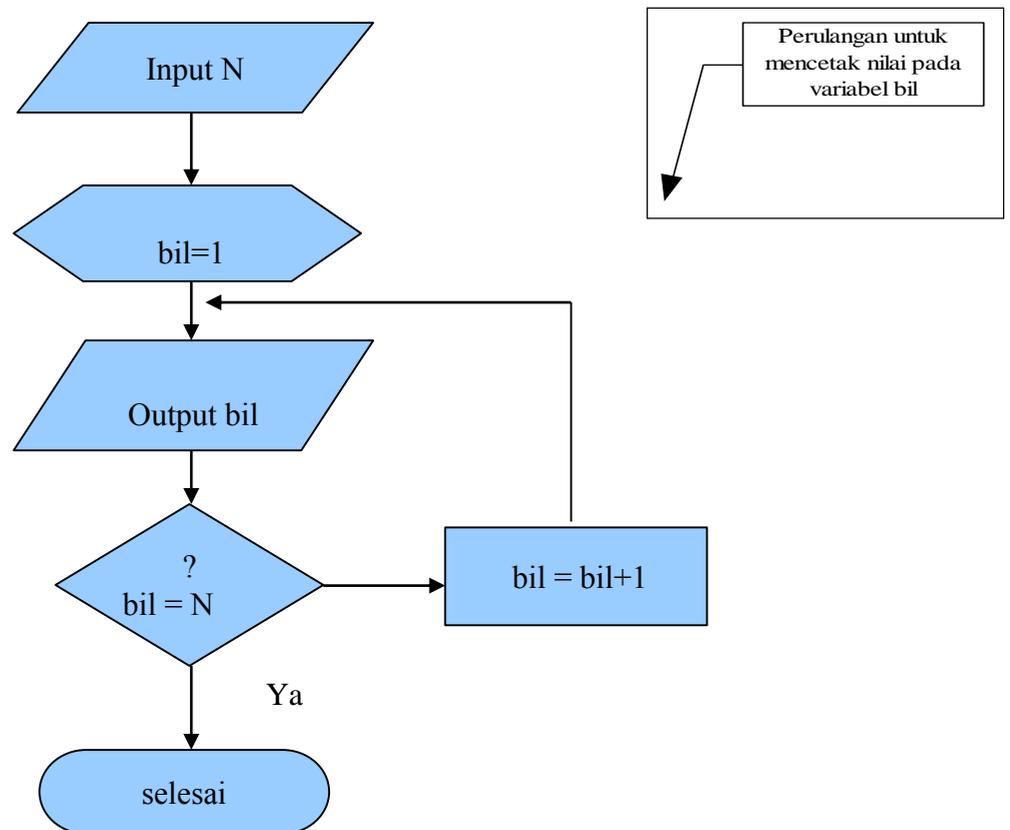
Bagaimana menampilkan sederet bilangan dari 1 sampai n dengan satu kali proses? N adalah batas bilangan yang ditampilkan.

Input : Batas bilangan

Proses : menampilkan bilangan dari 1 sampai batas bilangan

Output : bilangan 1 sampai batas bilangan.





Beberapa prinsip yang harus ditaati dalam membuat flowchart :

1. Tidak ada bagan yang menggantung
2. percabangan hanya ada 2 dengan indikasi Ya dan Tidak. Ya untuk menyatakan bahwa kondisi dipenuhi, Tidak untuk menyatakan sebaliknya
3. selalu diawali dengan Mulai dan Selesai atau Start dan Stop dengan bagan terminator
4. memanfaatkan konektor yang sesuai jika flowchart akan dibagi menjadi beberapa bagian
5. gunakan bahasa sederhana pada bagan yang digunakan

### ACTIVITY DIAGRAM

- Activity diagram merupakan cara lain dalam memodelkan aliran kejadian.
- Kita gunakan Activity Diagram untuk menggambarkan **workflow / aliran kerja** proses bisnis
- Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.
- Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi

#### 1.1 Workflow / Aliran Kerja

**Contoh :**

- 1) Use Case dimulai dengan melakukan pendaftaran bagi pasien yang belum terdaftar
- 2) Setelah dilakukan diagnosa oleh dokter maka hasil diagnosa akan dientri ke sistem
- 3) Dst..

Jika kita perhatikan workflow diatas dengan yang menggunakan teks memang bermanfaat, namun jika logika aliran kerja telah kompleks maka kita akan kesulitan. Oleh karena itu untuk lebih memudahkan dalam memahami aliran kerja tersebut kita gunakan **activity diagram**.

#### 1.2 Activity / Aktifitas

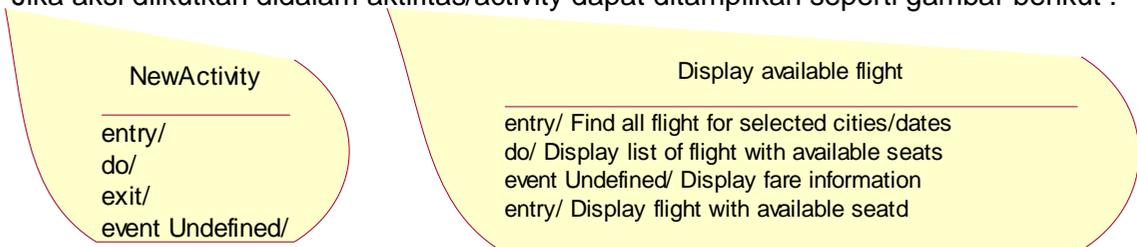
- Activity secara sederhana dapat diartikan sebagai langkah dalam. Pada UML disimbolkan sebagai berikut :



- Kita dapat menambahkan langkah-langkah yang lebih detail kedalam activity dengan menggunakan action.
- Action adalah langkah-langkah yang lebih kecil dalam suatu activity

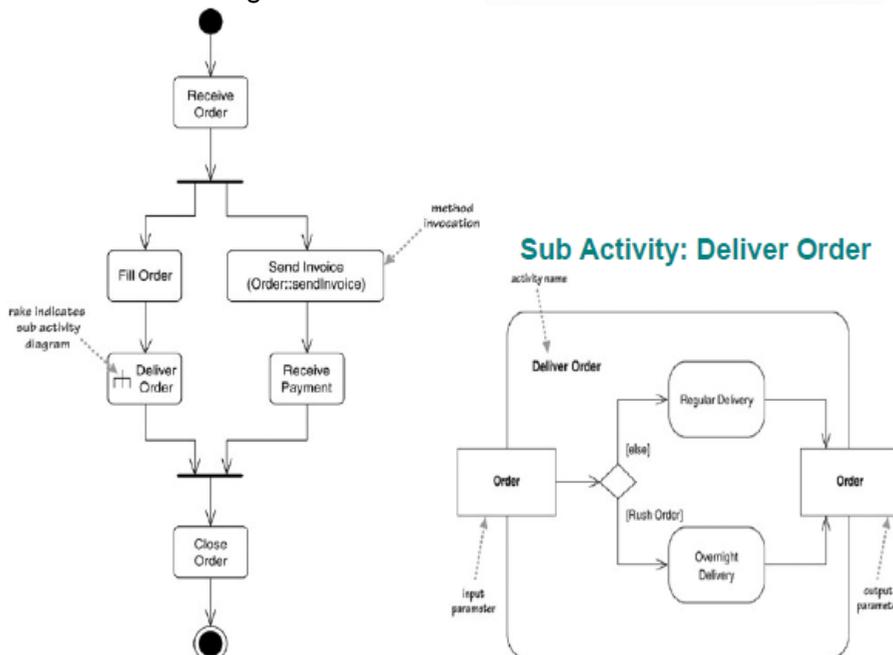
Action dapat berupa :

- Saat sedang memasuki aktifitas. Aksi Entry (entri action). Terjadi segera saat aktifitas dimulai. Ditandai dengan kata “entry”
- Ketika sedang meninggalkan aktifitas. Aksi exit (exit action). Terjadi ketika meninggalkan aktifitas dan ditandai dengan kata “exit”
- Ketika sedang berada dalam aktifitas. Aksi ini terjadi saat didalam aktifitas. Ditandai dengan kata “do”
- Kejadian Spesifik. Terjadi jika dan hanya jika kejadian spesifik terjadi. Ditandai dengan kata “event”
- Aksi-aksi diatas adalah opsional, tetapi dapat memberikan informasi rinci yang membantu pengembang
- Jika aksi diikuti dalam aktifitas/activity dapat ditampilkan seperti gambar berikut :



### 1.2 1 Activity Decomposition

Untuk menyederhanakan diagram, seringkali dilakukan dekomposisi aktivitas  
 Contoh: Proses Pengiriman



### 1.3 Start dan End State

- Start → memulai aliran
- End → mengakhiri aliran
- Setiap activity diagram harus mempunyai start state untuk memulai dan diakhiri dengan end state.
- End state adalah opsional dan bisa lebih dari satu dalam satu diagram



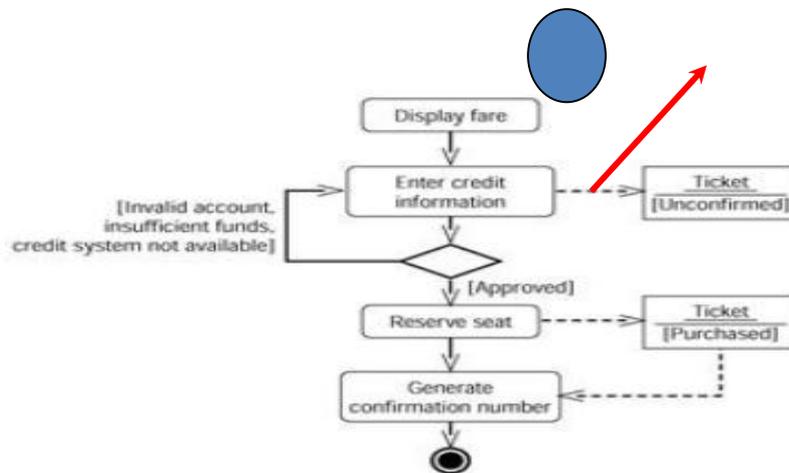
Start State



End State

### 1.4 Objek dan Aliran Objek

- Objek → Entitas yang dipergunakan dalam aliran.
- Objek dapat digunakan atau dirubah dalam aliran.
- Pada diagram, objek dihubungkan ke aktifitas dengan menggunakan aliran objek(object now)
- Aliran objek digambarkan dengan garis panah putus-putus.
- Jika garis panah mempunyai arah dari aktifitas ke objek berarti ia memperbaharui nilai objek tersebut. Jika arah panah dari objek ke aktifitas berarti ia sedang menggunakan objek



### 1.5 Transition

- Menunjukkan bagaimana aliran kontrol bergerak dari aktifitas ke aktifitas lainnya.
- Digambarkan dengan anak panah dengan nama yang menunjukkan nama kejadiannya.

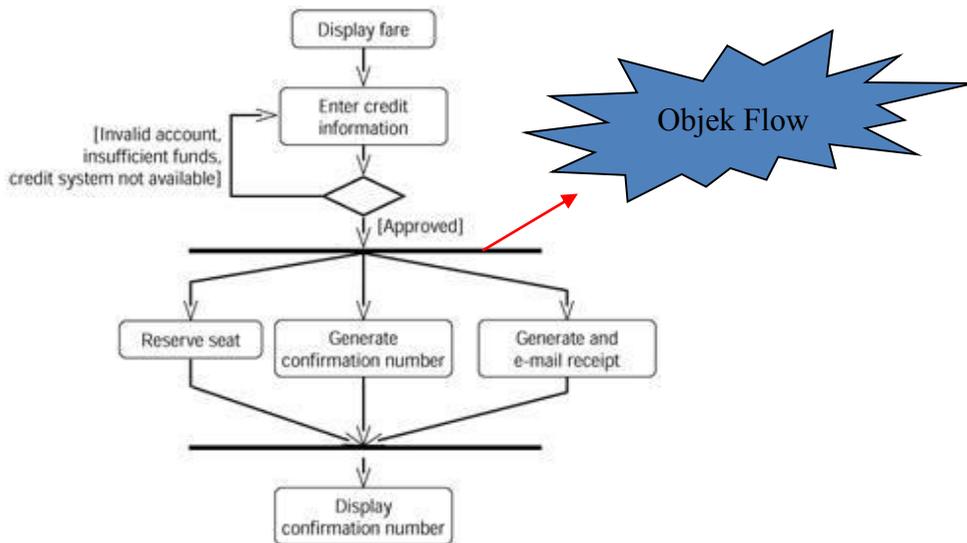
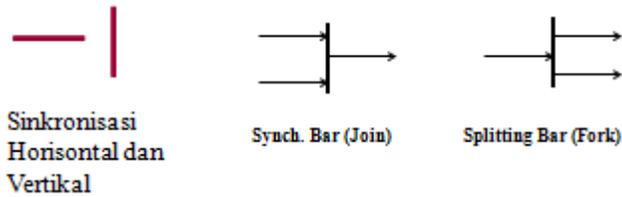


- Kejadian (event) memicu terjadinya transisi
- kondisi (guard condition) mengontrol atau tidak ada transisi yang terjadi



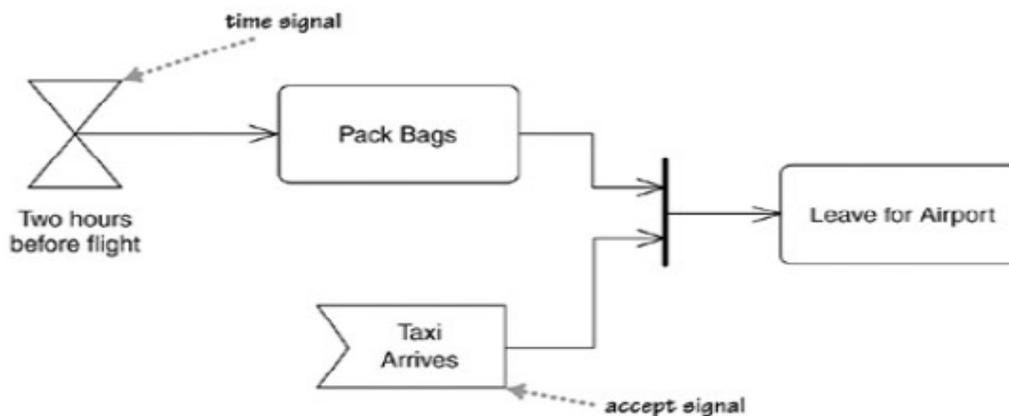
### 1.6 Sinkronisasi

Digunakan untuk menunjukkan bahwa dua atau lebih cabang dari aliran terjadi secara paralel



### 1.7 Signal

- Tanda untuk memulai sebuah aktivitas tidak harus melalui pemanggilan fungsi, tetapi juga bisa melalui sinyal, misalnya sinyal waktu atau kejadian lain yang bisa menjadi trigger.
- Sebuah time event tanpa aliran masuk, memberikan artiaktifasi secara berkala.



### 1. Pseudocode

**Pseudocode** adalah deskripsi dari algoritma pemrograman computer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman agar dapat dikomunikasikan dengan pemrogram. Secara umum, pseudocode mengekspresikan ide-ide secara informal dalam proses penyusunan Algoritma. Biasanya yang ditulis dari pseudocode adalah variabel dan fungsi. Dalam pseudocode, tidak ada syntax standar yang resmi. Karena itu kita dapat menggunakan pseudocode dalam berbagai bahasa pemrograman. Tujuan dari pseudocode adalah untuk membantu manusia dalam memahami suatu Algoritma.

- Pseudo= imitasi atau mirip / menyerupai
- Code=program
- Kode yang mirip dengan kode pemrograman yang sebenarnya
- Menggambarkan Algoritma yang akan dikomunikasikan kepada programmer
- Lebih rinci dari English Structure (mis: dalam menyatakan tipe data yang digunakan)

Contoh Pseudocode :

Algoritma	Pseudo-code
Masukkan panjang	Input panjang
Masukkan lebar	Input lebar
Nilai luas adalah panjang x lebar	Luas $\leftarrow$ panjang x lebar
Tampilkan luas	Print luas

### Latihan :

Dari contoh 1 buatlah flowchart, activity diagram dan pseudocode dimana Contoh 1 dapat juga dikembangkan menjadi :

1. Ditambahkan informasi apakah bilangan yang dicek bilangan **Genap Positif** atau bilangan **Genap Negatif**
2. Dilakukan pemeriksaan awal bahwa input yang diberikan harus berada antara nilai 10 sampai 30.

Jika kemudian Contoh 2 berkembang menjadi :

1. mencetak bilangan tidak mulai dari 1 tapi dari nilai M sampai batas N, M dan N ditentukan oleh user.
2. Mencetak bilangan ganjil mulai dari M sampai batas N, M dan N ditentukan oleh user
3. Permasalahan nomor 2 dilengkapi dengan pemeriksaan apakah batas M bukan NOL dan kurang dari 5
4. Permasalahan nomor 3 dilengkapi dengan pemeriksaan apakah batas N tidak kurang dari 20.
5. Pada Masalah 2, bagaimana jika nilai bil=0 pada preparation, seperti apa penyelesaiannya?

### Tugas

1. Buat algoritma dan flowchart (activity diagram) untuk menghitung nilai Total yang didapat dari 10 buah soal, jika setiap nilai benar adalah 10 dan setiap nilai salah adalah 0 (jika semua benar maka nilai total=100)

## PERTEMUAN KE – 3 DASAR PEMROGRAMAN C++

### A. TUJUAN :

1. Mahasiswa mampu membuat program dengan bahasa pemrograman C++
2. Mahasiswa dapat mengimplementasikan algoritma yang telah dibuat pada modul 1 dan 2 menjadi program untuk diuji kebenarannya
3. Mahasiswa dapat membuat variabel beserta tipe data yang sesuai dan mengimplementasikannya dalam program

### B. TEORI SINGKAT

#### 1. Pengenalan C++

**Bahasa C merupakan pengembangan dari bahasa B yang ditulis oleh Ken Thompson** pada tahun 1970. Bahasa C untuk pertama kali ditulis oleh Brian W. Kernighan dan Denies M. Ritchie pada tahun 1972. Bahasa C, pada awalnya dioperasikan diatas sistem operasi UNIX. Bahasa C adalah merupakan bahasa pemrograman tingkat menengah yaitu diantara bahasa tingkat rendah dan tingkat tinggi yang biasa disebut dengan Bahasa Tingkat Tinggi dengan Perintah Assambly. Bahasa C mempunyai banyak kemampuan yang sering digunakan diantaranya kemampuan untuk membuat perangkat lunak, misalnya dBASE, Word Star dan lain-lain. Pada tahun 1980 seorang ahli yang bernama Bjarne Stroustrup mengembangkan beberapa hal dari bahasa C yang dinamakan "C with Classes" yang berganti nama pada tahun 1983 menjadi C++. Penambahan yang terdapat pada C++ ini adalah Object Oriented Programming (OOP), yang mempunyai tujuan utamanya adalah membantu membuat dan mengelola program yang besar dan kompleks.

#### 2. Struktur Program C++

Struktur program C++ terdiri sejumlah blok fungsi, setiap fungsi terdiri dari satu atau beberapa pernyataan yang melaksanakan tugas tertentu

#### **#include<file-include>**

Merupakan salah satu pengarah *Preprocessor Directive* yang tersedia pada C++. *Preprocessor* selalu dijalankan terlebih dahulu pada saat proses kompilasi terjadi.

Bentuk umumnya:

`# include <nama file>`

Baris tersebut mengintruksikan kepada kompiler yang menyisipkan file lain dalam hal ini file yang berakhiran .h (**file header**) yaitu file yang berisi sebagai deklarasi.

Contonya: #include <iostream.h>, #include<conio.h>, #include <stdio.h>

#### **File Header**

(#include) adalah salah satu jenis dari preprocessor directive yg kegunaannya adalah untuk menyisipkan file-file header (contoh:**iostream.h**) ke dalam program.

Pada contoh diatas **iostream.h,conio.h,stdio.h** disebut file header. File header tersebut diperlukan agar perintah cout atau cin bisa dijalankan. Apabila file heder tersebut dihapus maka akan terjadi error. File Header adalah kumpulan variable, fungsi, konstanta, yang tersimpan dalam file berekstensi.h (contoh:iostream,conio,stdio,iomanip,dll)

Masing-masing file header mempunyai anak (fungsi), contoh:

1. iostream = cout,cin
2. stdio = scanf,printf,gets,puts,
3. conio = getch,clrscr,endl
4. ctime = header waktu
5. cmath = header untuk operasi matematika seperti sin, cos, sqrt, dan tan
6. cstring = header untuk memanipulasi string seperti strcpy, strcat.

\* Masih banyak jenis2 file header yang lain. File-file Header dapat anda lihat di folder instalasi C++ yang otomatis ada, jika anda menginstal program c++ di komputer anda.

## Main()

Merupakan fungsi yang menjadi inti dari program dan merupakan awal dan akhir eksekusi. Fungsi ini harus ada dalam setiap program. Tanda kurawal buka { merupakan tanda awal fungsi main dan tanda kurawal tutup } merupakan tanda berakhirnya fungsi main.

## Statement (Pernyataan)

Pernyataan merupakan bagian program yang berupa instruksi kepada computer untuk melakukan sesuatu. Menampilkan tulisan dilayar, menghitung operasi aritmatika, membaca inputan dan lain sebagainya. Setiap pernyataan harus diakhir dengan tanda titik koma “;”.

## Komentar

Komentar merupakan bagian program yang tidak ikut dieksekusi oleh program. Bagian ini berfungsi sebagai keterangan atas program yang telah ditulis, komentar sangat penting karena dapat mempermudah pembaca program dalam memahami program yang dibacanya. Pada C++ suatu komentar diawali dengan dua garis miring ( // ). Selain menggunakan //, komentar pada C++ juga dapat ditulis dengan ( /\* komentar \*/ ) untuk lebih dari satu baris.

contoh:

```
main() // main adalah fungsi utama

main() /* main adalah fungsi utama yang
        mengawali sebuah listing program*/
```

## Tanda Semicolon

Tanda semicolon “ ; ” digunakan untuk mengakhiri sebuah pernyataan. Setiap pernyataan harus diakhiri dengan sebuah tanda semicolon

## 3. Aturan umum penulisan bahasa C++

1. Bahasa C++ membedakan penulisan huruf besar dan kecil (case sensitive).
2. Untuk memberi komentar pada suatu statement (keadaan), dapat menggunakan /\* di awal dan \*/ di akhir atau // dalam satu baris.
3. Awal dan akhir subroutine atau fungsi harus diapit kurung kurawal.
4. Setiap statement harus diakhiri tanda titik koma.
5. Setiap variabel yang digunakan wajib dideklarasikan terlebih dahulu.

## Identifiers

Identifier adalah untaian satu atau lebih huruf, angka, atau garis bawah ( \_ ). Panjang dari identifier, tidak terbatas, walaupun untuk beberapa kompilator hanya 32 karakter pertama saja yang dibaca sebagai identifier (sisanya diabaikan). Identifier harus selalu diawali dengan huruf atau garis bawah ( \_ ).

Ketentuan lainnya yang harus diperhatikan dalam menentukan identifier adalah tidak boleh menggunakan **key word** dari bahasa C++. Dibawah ini adalah **key word** dalam C++ :

asm	Auto	bool	break	case
catch	Char	class	const	const_cast
continue	default	delete	do	double
dynamic_cast	Else	enum	explicit	extern
false	Float	for	friend	goto
if	Inline	int	long	mutable
namespace	New	operator	private	protected
public	register	reinterpret_cast	return	short
signed	Sizeof	static	static_cast	struct
switch	template	this	throw	true
try	typedef	typeid	typename	union
unsigned	Using	virtual	void	volatile
wchar_t				

Sebagai tambahan, representasi alternatif dari operator, tidak dapat digunakan sebagai identifier.

Contoh:

**and, and\_eq, bitand, bitor, compl, not, not\_eq, or, or\_eq, xor, xor\_eq**

**Catatan:** Bahasa C++ adalah bahasa yang "case sensitive", ini berarti identifier yang dituliskan dengan huruf kapital akan dianggap berbeda dengan identifier yang sama tetapi dituliskan dengan huruf kecil, sabagai contoh: variabel **RESULT** tidak sama dengan variable **result** ataupun variabel **Result**.

### 1. Kata Kunci

Pengenal sistem yang mempunyai makna khusus bagi kompilier. Kegunaan dari golongan ini tidak dapat diubah dan C menyediakan 32 kata kunci, seperti di bawah ini :

Tabel 1. Kata Kunci

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	Short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

### 2. Tipe Data

Tipe data adalah bagian dari program yang penting karena tipe data mempengaruhi setiap instruksi yang akan dilakukan oleh komputer. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Tabel 2. Tipe Data

Tipe Data	Ukuran memori	Jangkauan nilai
char	1 byte	-128 s.d. +127
int	2 byte	-32768 s.d. +32767
long	4 byte	-2.147.438.648 s.d. 2.147.438.647
float	4 byte	$3.4 \times 10^{-38}$ s.d. $3.4 \times 10^{+38}$
double	8 byte	$1.7 \times 10^{-308}$ s.d. $1.7 \times 10^{+308}$
long double	10 byte	$3.4 \times 10^{-4932}$ s.d. $3.4 \times 10^{+4932}$

Tipe data yang berhubungan dengan bilangan bulat adalah char, int, short dan long, sedangkan yang lainnya berhubungan dengan bilangan pecahan.

Untuk mengetahui ukuran memori yang diperlukan semua tipe data, bisa menggunakan sizeof, dimana sizeof berguna untuk mendapatkan ukuran memori dari suatu tipe data, seperti program berikut ini :

### 3. Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Variabel yang digunakan dalam program haruslah didifisikan terlebih dahulu, bentuknya adalah :

**tipe daftar\_variabel**

pada pendefinisian variabel, daftar\_variabel dapat berupa sebuah variabel atau beberapa variabel yang dipisahkan dengan koma.

Contoh :

**int jumlah;**

**float bagi, rata;**

Nama suatu variabel dapat ditentukan sendiri oleh programmer, dengan ketentuan sebagai berikut :

- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Yang harus diingat adalah bahasa C mempunyai sifat case-sensitive artinya huruf besar dan kecil dianggap berbeda, misalnya menulis NOMHS, Nomhs dan nomhs dianggap berbeda
- Tidak boleh mengandung spasi
- Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscope).
- Panjangnya bebas, tetapi hanya 32 karakter pertama yang akan terpakai.

## Inisialisasi Variabel

Ketika mendeklarasikan variabel local, kita dapat memberikan nilai tertentu. Sintaks penulisan sbb:

```
type identifier = initial_value ;
```

Misalkan kita akan mendeklarasikan variabel **int** dengan nama **a** yang bernilai **0**, maka dapat dituliskan :

```
int a = 0;
```

Atau dengan cara lainnya, yaitu menyertakan nilai yang akan diberikan dalam tanda **()**:

```
type identifier (initial_value) ;
```

Contoh :

```
int a (0);
```

## Lingkup Variabel

Pada C++, kita dapat mendeklarasikan variable dibagian mana saja dari program, bahkan diantara 2 kalimat perintah.

**variabel Global** dapat digunakan untuk setiap bagian dari program, maupun fungsi, walaupun dideklarasikan diakhir program.

Lingkup dari **variable local** terbatas. Hanya berlaku dimana variable tersebut dideklarasikan. Jika dideklarasikan diawal fungsi (seperti dalam **main**) maka lingkup dari variable tersebut adalah untuk seluruh fungsi **main**. Seperti contoh diatas, jika terdapat fungsi lain yang ditambahkan pada main(), maka variable local yang dideklarasikan dalam **main** tidak dapat digunakan pada fungsi lainnya dan sebaliknya.

*Pada C++, lingkup variable local ditandai dengan blok dimana variable tersebut dideklarasikan ( blok tersebut adalah sekumpulan instruksi dalam kurung kurawal {} ). Jika dideklarasikan dalam fungsi tersebut, maka akan berlaku sebagai variable dalam fungsi tersebut, jika dideklarasikan dalam sebuah perulangan, maka hanya berlaku dalam perulangan tersebut, dan seterusnya.*

## 4. Konstanta

Konstanta adalah nilai yang tetap, yang tidak dapat diubah selama proses program sedang berlangsung. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string.

Contoh :

```
float jumlah;  
jumlah = 10;
```

## 5. Konstanta String

Merupakan deretan karakter yang diawali dan diakhiri dengan tanda petik ganda ("),

## 6. Konstanta bernama

Hal ini dapat dilakukan dengan menggunakan kata kunci const.

Contoh :

```
const float PHI = 3.14;
```

## 7. Tipe signed dan unsigned

Tipe signed dan unsigned hanya diterapkan pada data bertipe bilangan bulat (char, int dan long). Variabel yang dimodifikasi tipe unsigned akan menyebabkan nilai yang ada didalamnya selalu bernilai positif, sedangkan pemodifikasian tipe signed merupakan default dari tipe-tipe dasar, yang menyatakan data dapat bernilai positif dan negatif.

Tabel 3. Pemodifikasian Tipe signed dan unsigned

Pemodifikasi tipe	Persamaan	Jangkauan Nilai
signed char	char	-128 s/d 127
signed int	int	-32.768 s/d 32.767
signed short int	short, signed short	-32.768 s/d 32.767
signed long int	long, long int, signed long	-2.147.483.648 s/d 2.147.483.647

unsigned char	Tidak ada	0 s/d 255
unsigned int	unsigned	0 s/d 65.535
unsigned short int	unsigned short	0 s/d 65.535
unsigned long int	unsigned long	0 s/d 4.294.967.295

## 8. Operator Dan Ungkapan

Operator-operator yang disediakan C++ berupa *keyword* atau karakter khusus. Operator-operator ini cukup penting untuk diketahui karena merupakan salah satu dasar bahasa C++.

### Assignment (=).

Operator *assignment* digunakan untuk memberikan nilai ke suatu variable.

```
a = 5;
```

Memberikan nilai integer **5** ke variabel **a**. Sisi kiri dari operator disebut *lvalue* (left value) dan sisi kanan disebut *rvalue* (right value). *lvalue* harus selalu berupa variabel dan sisi kanan dapat berupa konstanta, variabel, hasil dari suatu operasi atau kombinasi dari semuanya.

Contoh :

```
int a, b; // a:? b:?
a = 10; // a:10 b:?
b = 4; // a:10 b:4
a = b; // a:4 b:4
b = 7; // a:4 b:7
```

Hasil dari contoh diatas, **a** bernilai **4** dan **b** bernilai **7**.

Contoh :

```
a = 2 + (b = 5);
```

equivalen dengan :

```
b = 5;
```

```
a = 2 + b;
```

Operator merupakan simbol yang bisa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi, misalnya digunakan untuk menjumlahkan dua buah bilangan, memberikan nilai ke suatu variabel dan membandingkan kesamaan dua buah nilai.

C++ memiliki operator jauh lebih banyak daripada bahasa pemrograman lainnya. Sebagian operator C++ tergolong sebagai operator binary, yaitu operator yang digunakan terhadap dua buah nilai (operand).

Operator C++ ada yang bersifat binary, unary dan ada pula operator ternary, berikut adalah tabel dari sifat operator.

Tabel 4. Sifat operator

Sifat	Keterangan	Contoh
Unary	Operator yang hanya melibatkan 1 operand	-1
Binary	Operator yang hanya melibatkan 2 operand	1 + 2
Tenary	Operator yang hanya melibatkan 3 operand	1 + 2 * 2

Operator aritmatika yang tergolong sebagai operator binary, operator seperti penjumlahan, pengurangan, perkalian dan pembagian merupakan operator yang umum, sedangkan operator sisa pembagian ada baiknya diterangkan lebih lanjut karena akan sangat berguna. Operator ini diterapkan pada operand bertipe integer, operator-operator tersebut dapat dilihat pada tabel berikut ini :

Tabel 5. Operator Aritmatika

Operator	Keterangan	Contoh
*	Perkalian	7 * 2 = 14
/	Pembagian	7 / 2 = 3
%	Sisa pembagian (modulus)	7 % 2 = 1
+	Penjumlahan	7 + 2 = 9
-	Pengurangan	7 - 2 = 5

Adapun operator yang tergolong sebagai operator unary, dapat di lihat pada tabel berikut ini :

Tabel 6. Operator Unary Aritmatika

Operator	Keterangan	Contoh
-	Minus	- 4
+	Plus	+ 4

### Prioritas Operator

Operator aritmatika mempunyai prioritas, operator yang mempunyai perioritas tertinggi akan dikerjakan terlebih dahulu, beriku adalah tabel prioritas operator :

Tabel 7. Prioritas Operator

Operator	Prioritas
+ --	Tertinggi
- (unary)	
*, /, %	
+ -	Terendah

### Operator Penaikan dan Penurunan

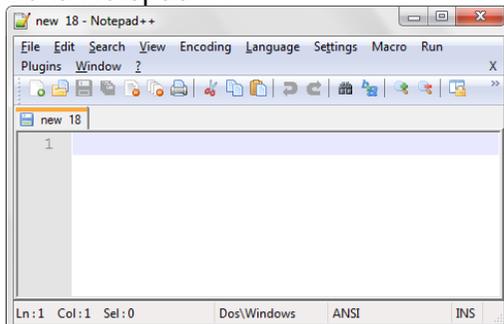
C++ menyediakan operator yang dinamakan sebagai operator penaikan (increment) dan operator penurunan (decrement), kedua opeartpr ini digunakan pada operand yang bertipe bilangan bulat.

Tabel 8. Operator Penaikan dan Penurunan

Operator	Keterangan
++	Penaikan
--	Penurunan

## C. PRAKTIK

### 1. Buka Notepad ++



Ketikkan Program Berikut:

```

1. #include <conio.h>
2. #include <iostream.h>
3. void main ()
4. {
5. cout <<"selamat Datang di Borland C++\n";
6. getch();
7. }
    
```

- Simpan dengan nama **program1.cpp**
- Kompilasi dan jalankan program tersebut dengan menekan tombol F6
- Hilangkan tanda semicolon (;) pada baris ke-5, simpan dan jalankan kembali, amati yang terjadi
- Hapus sintak **#include<iostream.h>** (baris ke 2), simpan dan jalankan kembali
- Amati hasilnya, mengapa bisa demikian
- Tuliskan di laporan

### 2. Modifikasi **program1.cpp** menjadi bentuk C++ standar seperti berikut

```

#include <conio>
#include <iostream>
using namespace std;
void main ()
{
    
```

```
    cout <<"selamat Datang di Borland C++\n";
    getch();
}
```

3. Modifikasi **program2.cpp** menjadi seperti berikut

```
#include <conio.h>
#include <iostream.h>
main()
{
float a, b, jumlah;
a=7.5;
b=8.4;
jumlah=a+b;
cout<<"Nilai A : "<<a<<endl;
cout<<"Nilai B : "<<b<<endl;
cout<<"Jumlah : "<<jumlah<<endl;
getch();
}
```

- Jalankan kembali dan amati hasilnya
- Tuliskan dalam laporan

4. Ketikkan program berikut

```
/******
nama file      : program3.cpp
dibuat        : 1 Oktober 2014
oleh          : femi
*****/

#include<iostream.h>
int main()
{
    cout<<"Hello World . . . . ."<<endl;
    return 0; //ini untuk mengembalikan nilai
}
```

- Simpan dengan nama **program3.cpp**
- Jalankan dan amati hasilnya
- Tuliskan dalam laporan

5. Buat program untuk menghasilkan keluaran sebagai berikut :

Nim : .....  
Nama : .....  
Jurusan : .....

Dimana isi dari Nim, nama dan jurusan adalah data saudara

6. Buatlah program seperti dibawah, amati hasilnya

```
#include <iostream.h>
int main()
{
    cout<<"Ukuran char   = "<<sizeof(char)<<endl;
    cout<<"Ukuran short  = "<<sizeof(short)<<endl;
    cout<<"Ukuran int     = "<<sizeof(int)<<endl;
    cout<<"Ukuran long    = "<<sizeof(long)<<endl;
    cout<<"Ukuran float    = "<<sizeof(float)<<endl;
    cout<<"Ukuran double=  "<<sizeof(double)<<endl;
    return 0;
}
```

7. Buatlah program seperti dibawah, amati hasilnya

```
#include<iostream.h>
void main()
{
    float jumlah, total;
    jumlah = 10;
    float harga = 250;
    cout<<"jumlah = "<<jumlah<<endl;
    cout<<"harga = "<<harga<<endl;
    cout<<"Total = "<<jumlah*harga<<endl;
    getch();
}
```

8. Buatlah program seperti dibawah, amati hasilnya

```
#include <iostream.h>
void main()
{
    int bil1, bil2;
    float kali, bagi, mod, jumlah, kurang;
    bil1=6;
    bil2=3;
    kali = bil1 * bil2;
    bagi = bil1/bil2;
    mod = bil1 % bil2;
    jumlah = bil1 + bil2;
    kurang = bil1 - bil2;
    cout<<"Hasil perkalian = "<<kali<<endl;
    cout<<"Hasil pembagian = "<<bagi<<endl;
    cout<<"Hasil modulus = "<<mod<<endl;
    cout<<"Hasil penjumlahan = "<<jumlah<<endl;
    cout<<"Hasil pengurangan= "<<kurang<<endl;
    getch();
}
```

9. Buatlah program seperti dibawah

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int z;
    z = 3 + 5 * 6;
    cout<<"nilai Z = "<<z<<endl;
    z = (3 + 5) * 6;
    cout<<"nilai Z = "<<z<<endl;
    getch();
}
```

amati hasilnya, mengapa nilai z dapat berbeda, jelaskan

10. Buatlah program seperti dibawah

<pre>#include&lt;iostream.h&gt; main() {     int b = 3;     int a;     a = ++b;     cout&lt;&lt;"Nilai A = "&lt;&lt;a&lt;&lt;endl;     cout&lt;&lt;"Nilai B = "&lt;&lt;b&lt;&lt;endl;     return 0; }</pre>	<pre>#include&lt;iostream.h&gt; main() {     int b = 3;     int a;     a = b++;     cout&lt;&lt;"Nilai A = "&lt;&lt;a&lt;&lt;endl;     cout&lt;&lt;"Nilai B = "&lt;&lt;b&lt;&lt;endl;     return 0; }</pre>
---	---

```
} | }
```

Bandingkan kedua output program tersebut

11. Buatlah program seperti dibawah, perhatikan nilai a yang digunakan

```
#include <iostream.h>
main () {
    int a,b,c,d;
    a = 3;
    b = 2;
    c = a * b;
    a = 5;
    d = a + b;
    cout<<"Nilai C ="<<c<<endl;
    cout<<"Nilai D ="<<d<<endl;
}
```

#### D. LATIHAN

Diberikan oleh dosen pengampu pada saat praktikum.

#### E. TUGAS

1. Buatlah program untuk kasus pada praktik Modul 1 untuk praktik 2 no.1 dan 2.
2. Jelaskan perbedaan variabel dan konstanta
3. Perhatikan ekspresi berikut ini :  
Buatlah program untuk menghitung nilai berikut :  
 $4 + 7 * 10 - 5 \text{ mod } 13$   
Apakah output dari ekspresi di atas, jelaskan mengapa bisa demikian
4. Perhatikan ekspresi berikut ini :  
Buatlah program untuk menghitung nilai berikut :  
 $x * 3 + (7 \text{ mod } 5) * y$   
dengan  $x \leftarrow 5$  dan  
 $y \leftarrow 3$   
Apakah output dari ekspresi di atas, jelaskan mengapa bisa demikian

## PERTEMUAN KE – 4 SEKUENSIAL

### A. TUJUAN

1. Menggunakan pernyataan masukan dan keluaran
2. Membuat algoritma untuk memecahkan permasalahan tertentu yang harus diselesaikan menggunakan pernyataan masukan dan keluaran
3. Membuat program lengkap dengan menggunakan pernyataan masukan – keluaran.
4. Mahasiswa dapat membuat algoritma dan program untuk menyelesaikan masalah sederhana

### B. TEORI SINGKAT

#### Komunikasi melalui *console*

*Console* merupakan interface dasar pada computers, biasanya berupa keyboard dan monitor. Keyboard merupakan alat *input* standar dan monitor adalah alat *output* standar. Dalam library *iostream* C++ , standard operasi *input* dan *output* untuk pemrograman didukung oleh 2 data streams: **cin** untuk input dan **cout** untuk output. Juga, **cerr** dan **clog** sebagai tambahan untuk output streams yang di desain khusus untuk menampilkan *error messages*. Dapat diarahkan langsung ke standard output maupun ke log file.

Biasanya **cout** (standard output stream) ditujukan untuk monitor dan **cin** (standard input stream) ditujukan untuk keyboard. Dengan menggunakan dua streams ini, maka kita dapat berinteraksi dengan user dengan menampilkan messages pada monitor dan menerima input dari keyboard.

Pernyataan keluaran digunakan untuk menampilkan output (keluaran) ke standart output (biasanya layar monitor). Pernyataan yang digunakan adalah **cout**, dan dituliskan sebagai berikut :

```
cout << "yang akan ditampilkan " ;
```

Jika diinginkan untuk pindah baris, gunakan karakter khusus `\n`.

Pernyataan masukan digunakan untuk memasukkan data masukan secara interaktif ke dalam program. Cara yang paling banyak digunakan adalah memasukkan data dengan mengetikannya dari keyboard. Dalam C++ pernyataan yang digunakan adalah **cin**. Bentuk pernyataan cin untuk membaca data dari keyboard dan meletakkannya ke sebuah variabel dituliskan sebagai berikut :

```
cin >> var;
```

#### Output (**cout**)

Penggunaan **cout** stream dihubungkan dengan operator overloaded `<<` (Sepasang tanda "*less than*").

Contoh :

```
cout << "Output "; // cetak Output kelayar
cout << 120;      // cetak angka 120 k layar
cout << x;       //cetak variable x k layar
```

Operator `<<` dikenal sebagai *insertion operator*, dimana berfungsi untuk menginput data yang mengikutinya. Jika berupa string, maka harus diapit dengan kutip ganda ("`"`), sehingga membedakannya dari variable.

Contoh :

```
cout << "Hello"; // cetak Hello kelayar
cout << Hello;   //cetak content dari variable Hello kelayar
```

#### Input (**cin**)

Menangani standard input pada C++ dengan menambahkan overloaded operator *extraction* (`>>`) pada **cin** stream. Harus diikuti dengan variable yang akan menyimpan data.

Contoh :

```
int umur;
cin >> umur;
```

Contoh diatas mendeklarasikan variabel age dengan tipe int dan menunggu input dari cin (keyborad) untuk disimpan di variabel age.

**cin** akan memproses input dari keyboard sekali saja dan tombol ENTER harus ditekan.

**cin** juga dapat digunakan untuk lebih dari satu input :

```
cin >> a >> b;  
Sama dengan :  
cin >> a;  
cin >> b;
```

Dalam hal ini data yang di input harus 2, satu untuk variabel **a** dan lainnya untuk variabel **b** yang penulisannya dipisahkan dengan : spasi, tabular atau *newline*.

### SEKUENSIAL

Sekuensi adalah struktur kontrol algoritmik yang paling sederhana. Sekuensi adalah sederetan instruksi primitif dan/atau aksi yang akan dilaksanakan (dieksekusi) oleh komputer berdasarkan urutan penulisannya.

Sekuensi adalah Proses menjelaskan alur logika berurutan atau proses penyelesaian masalah secara berurutan. Mulai pendefenisian variable, penetapan spesifikasi masalah, proses, dan pencetakan output program.

Jadi Sekuensi Adalah pemrograman sederhana yang hanya dapat memecahkan masalah-masalah yang sederhana.

Untuk penulisan instruksi atau aksi pada sekeuksi adalah sesuai dengan penulisan per baris. Sekuensi dapat juga ditulis dalam satu baris program dengan cara memisahkan penulisan setiap instruksi atau aksi dengan tanda "titik koma". Penulisan sekuensi dengan menggunakan "titik koma" sebaiknya hanya dilakukan untuk sekuensi yang jika urutan diubah tidak berpengaruh pada program.

Contoh Penulisan sekuensi perbaris :

Algoritma Sekuensi\_Perbaris

Deklarasi :

```
A, B : integer;  
X : float;
```

Deskripsi :

```
input(A, B);  
X ← A + B  
output(X)
```

Contoh Penulisan Sekuensi dengan titik koma

Algoritma Sekuensi\_Dengan\_TitikKoma

Deklarasi :

```
A, B : integer;  
X, Y : float;
```

Deskripsi :

```
input(A, B); X ← A + B; Y ← A - B;  
output(X); output(Y);
```

## C. PRAKTIK

1. Buatlah program seperti dibawah, amati hasilnya

```
#include<iostream.h>  
int main ()  
{  
    int i;  
    cout << "Masukan Nilai : ";  
    cin >> i;  
    cout << "Nilai Yang dimasukan adalah " <<i<<endl;  
    cout << "Nilai jika digandakan " << i*2 <<endl;  
    getchar();  
    return 0;  
}
```

2. Buatlah program seperti dibawah, amati hasilnya

```
#include<iostream.h>
main ()
{
    int iPertama, iKedua;
    float fHasil;
    cout<<"Masukan Bilangan Pertama :";
    cin>>iPertama;
    cout<<"Masukan Bilangan kedua : ";
    cin>>iKedua;
    fHasil = iPertama / iKedua;
    cout<<"Hasilnya : "<<fHasil;
    getch();
}
```

Hilangkan fungsi getch(), jalankan kembali dan analisa hasilnya

3. Buatlah program seperti dibawah, amati hasilnya

```
#include <iostream.h>
main () {
    string sNama;
    int iUmur;
    char jKelamin;

    cout<<"Masukan Nama : ";
    cin>>sNama;
    cout<<"Masukan Umur : ";
    cin>>iUmur;
    cout<<"Masukan Jenis Kelamin (L/P) : ";
    cin>>jKelamin;

    cout<<"Nama : "<<sNama<<endl;
    cout<<"Umur : "<<iUmur<<endl;
    cout<<"Jenis Kelamin : "<<jKelamin<<endl;
    getch();
}
```

4. Buatlah program seperti dibawah

```
#include <iostream.h>
void main()
{
    int pil;
    cout<<"***** \n";
    cout<<"Silahkan pilih menu \n";
    cout<<"***** \n";
    cout<<"1. Paket lele \n";
    cout<<"2. Paket ayam \n";
    cout<<"3. Paket nila \n";
    cout<<"Masukkan pilihan (1,2,3) = ";
    cin>>pil;
    cout<<"Pesanan anda paket "<<pil;
    getch();
}
```

5. Buatlah algoritma, flowchart dan program untuk menghitung jumlah 2 buah bilangan bulat positif.

Langkahnya :

1. Deklarasikan 3 buah variabel misal A, B dan Jumlah

2. Inputkan nilai A dan B
3. Hitung Jumlah = A + B
4. Tampilkan hasil Jumlah

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int A,B, jumlah;
    clrscr();
    cout<<"Bilangan 1 : ";
    cin>>A;
    cout<<"Bilangan 2 : ";
    cin>>B;
    jumlah = A + B;
    cout<<"Jumlah = " << jumlah<<endl;
    getch();
}
```

6. Buatlah algoritma dan program untuk menghitung konversi dari Meter Ke CM dan Ke INCI.

Rumus :

$$1 \text{ Meter} = 100 \text{ CM}$$

$$1 \text{ Inchi} = 2.54 \text{ CM}$$

Langkahnya :

1. Deklarasikan variabel Meter, CM dan Inchi
2. Hitung konversi Meter Ke CM, dengan rumus  $\text{CM} = \text{Meter} * 100$
3. Hitung konversi Meter ke INCI, dengan rumus  $\text{Inchi} = \text{Meter} * 100/2.54$
4. Tampilkan hasil CM
5. Tampilkan hasil INCI

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float m, cm, inci;
    clrscr();
    cout<<"Masukan ukuran dalam Meter : ";
    cin>>m;
    cm = m * 100;
    inci = m * 100 / 2.54;
    cout<<"Ukuran dalam CM = " << cm <<endl;
    cout<<"Ukuran dalam Inchi = " << inci <<endl;
    getch();
}
```

7. Buatlah algoritma dan program untuk menghitung keliling dan luas lingkaran

Rumus :

$$\text{Keliling} = 2\pi r$$

$$\text{Luas} = \pi r^2$$

Langkahnya :

1. Deklarasikan konstanta  $\pi$  dan variabel r (jar-jari), Keliling dan Luas
2. Hitung keliling lingkaran, dengan rumus,  $\text{Keliling} = 2 * \pi * r$
3. Hitung luas lingkaran, dengan rumus  $\text{Luas} = \pi * r * r$
4. Tampilkan keliling lingkaran
5. Tampilkan luas lingkaran

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    const float phi = 3.14;
    int r;
    float keliling, luas;
    cout<<"Jari-jari Lingkaran : ";
    cin>>r;
    keliling = 2 * phi * r;
    luas = phi * r * r;
    cout<<"Keliling lingkaran : "<<keliling<<endl;
    cout<<"Luas Lingkaran : "<<luas<<endl;
    getch();
}
```

**D. LATIHAN**

Diberikan oleh dosen pengampu pada saat praktikum.

**E. TUGAS**

1. Buatlah program persamaan linier yang menerima inputan x dan menghasilkan nilai y dengan persamaan  $y = 3x + 4$
2. Diberikan oleh dosen pengampu pada saat praktikum

## PERTEMUAN KE – 5 SEKUENSIAL

### A. TUJUAN

Mahasiswa dapat membuat algoritma dan program untuk menyelesaikan masalah sederhana

### B. TEORI SINGKAT

Sekuensi adalah struktur kontrol algoritmik yang paling sederhana. Sekuensi adalah sederetan instruksi primitif dan/atau aksi yang akan dilaksanakan (dieksekusi) oleh komputer berdasarkan urutan penulisannya.

Sekuensi adalah Proses menjelaskan alur logika berurutan atau proses penyelesaian masalah secara berurutan. Mulai pendefinisian variable, penetapan spesifikasi masalah, proses, dan pencetakan output program.

Jadi Sekuensi Adalah pemrograman sederhana yang hanya dapat memecahkan masalah-masalah yang sederhana.

Untuk penulisan instruksi atau aksi pada sekuensi adalah sesuai dengan penulisan per baris. Sekuensi dapat juga ditulis dalam satu baris program dengan cara memisahkan penulisan setiap instruksi atau aksi dengan tanda “titik koma”. Penulisan sekuensi dengan menggunakan “titik koma” sebaiknya hanya dilakukan untuk sekuensi yang jika urutan diubah tidak berpengaruh pada program.

### C. PRAKTIK

1. Buatlah algoritma dan program konversi suhu Celcius ke Reamur ke fahrenheit ke kelvin

Rumus :

$$\text{Reamur} = 4/5 * \text{Celcius}$$

$$\text{Fahrenheit} = 9/5 * \text{celcius} + 32$$

$$\text{Kelvin} = \text{Celcius} + 273$$

**Langkahnya :**

1. Deklarasikan variabel suhu Celcius, Reamur, Fahrenheit dan Kelvin
2. Hitung Rumus Reamur =  $4/5 * \text{Celcius}$
3. Hitung Rumus Fahrenheit =  $9/5 * \text{Celcius} + 32$
4. Hitung Rumus Kelvin =  $273 + \text{Celcius}$
5. Tampilkan hasil konversi Celcius ke Reamur
6. Tampilkan hasil konversi Celcius ke Fahrenheit
7. Tampilkan hasil konversi Celcius ke Kelvin

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    float celcius;
    float reamur, fahrenheit, kelvin;
    cout<<"Masukkan derajat Celcius : ";
    cin>>celcius;
    reamur = 4.0/5.0 * celcius;
    fahrenheit = 9.0/5.0 * celcius + 32;
    kelvin = celcius + 273;
    cout<<"Reamur : "<<reamur<<endl;
    cout<<"Fahrenheit : "<<fahrenheit<<endl;
    cout<<"Kelvin : "<<kelvin<<endl;
    getch();
}
```

2. Buatlah algoritma, flowchart dan program untuk menghitung luas segitiga dengan rumus  $\text{Luas} = 0.5 * \text{Alas} * \text{Tinggi}$

**Langkahnya :**

1. Deklarasikan variabel Alas, Tinggi dan Luas
2. Hitung Rumus  $\text{Luas} = 0.5 * \text{Alas} * \text{Tinggi}$
3. Tampilkan hasil Luas Segitiga

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int Alas, Tinggi;
    float Luas;
    cout<<"Masukan Alas    = ";cin>>Alas;
    cout<<"Masukan Tinggi = ";cin>>Tinggi;
    Luas = 0.5 * Alas * Tinggi;
    cout<<"Luas Segitiga  = "<<Luas<<endl;
    getch();
}
```

3. Buatlah algoritma, flowchart dan program untuk menghitung perkalian, pembagian dan pengurangan dari 2 buah bilangan.

**Langkahnya :**

1. Deklarasikan variabel A, B, Kali, Bagi dan Kurang
2. Hitung Rumus  $\text{Kali} = A * B$ ;
3. Hitung Rumus  $\text{Bagi} = A / B$ ;
4. Hitung Rumus  $\text{Kurang} = A - B$
5. Tampilkan hasil Kali, Bagi dan Kurang

**Programnya :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    float A, B, Kali, Bagi, Kurang;
    cout<<"Masukan nilai 1 = ";cin>>A;
    cout<<"Masukan nilai 2 = ";cin>>B;
    Kali = A * B;
    Bagi = A / B;
    Kurang = A - B;
    cout<<"Hasil Kali    = "<<Kali<<endl;
    cout<<"Hasil Bagi    = "<<Bagi<<endl;
    cout<<"Hasil Kurang = "<<Kurang<<endl;
    getch();
}
```

4. Buatlah algoritma, flowchart dan program untuk luas segipanjang dengan rumus  $\text{Luas} = \text{panjang} * \text{lebar}$

**Langkahnya :**

1. Deklarasikan variabel panjang, lebarLuas
2. Hitung Rumus  $\text{Luas} = \text{panjang} * \text{lebar}$
3. Tampilkan hasil Luas Segipanjang

**Program :**

```
#include <iostream.h>
#include <conio.h>
void main()
```

```

{
int panjang, lebar;
float Luas;
cout<<"Masukan panjang = ";
cin>>panjang;
cout<<"Masukan lebar    = ";
cin>>lebar;
Luas = panjang * lebar;
cout<<"Luas Segipanjnag = "<<Luas<<endl;
getchar();
}

```

5. Buatlah algoritma, flowchart dan program untuk mencari hasil bagi dari suatu bilangan yang diinputkan, gunakan operator modulo

#### **Algoritma Modulo**

Deklarasi

Nilai1, nilai2 : integer {input}  
 modulo : integer {output}

Deklarasi

cout(nilai1, nilai2)  
 Modulo  $\leftarrow$  nilai1 % nilai2  
 cout(modulo)

#### **Program :**

```

#include<iostream.h>
#include<conio.h>
void main()
{
  int nilai1, nilai2, modulo;
  cout<<"Masukan nilai 1 = ";
  cin>>nilai1;
  cout<<"Masukan nilai 2 = ";
  cin>>nilai2;
  modulo = nilai1 % nilai2;
  cout<<"Modulo = "<<modulo<<endl;
  getch();
}

```

#### **D. LATIHAN**

Diberikan oleh dosen pengampu pada saat praktikum.

#### **E. TUGAS**

Diberikan oleh dosen pengampu pada saat praktikum

## PERTEMUAN KE – 6 PEMILIHAN (SELECTION) IF

### A. TUJUAN

1. Memahami struktur pemilihan dalam program
2. Mengetahui struktur IF yang dapat digunakan dalam pemilihan
3. Memahami konsep kondisi dan aksi dalam struktur pemilihan
4. Menerapkan pemilihan dalam menyelesaikan berbagai kasus

### B. TEORI SINGKAT

Seleksi digunakan untuk mengarahkan suatu proses itu berjalan. Seleksi adalah Suatu program untuk mengambil keputusan berdasarkan suatu kondisi. Sebelum kita membahas seleksi lebih jauh ada baiknya kita mengenal terlebih dahulu operator-operator yang akan digunakan pada saat pembuatan program dengan menggunakan seleksi.

#### Pernyataan if

Pernyataan if dapat digunakan untuk mengambil suatu keputusan berdasarkan suatu kondisi, bentuk dari pernyataan ini ada tiga macam, yaitu :

1. if tunggal (if saja)
2. else (if dengan ingkaran)
3. if di dalam if (if ganda)

#### 1. if Tunggal

Pernyataan if tunggal ini hanya digunakan jika kondisinya hanya 1.

Bentuknya :

```
if (kondisi)
    pernyataan;
```

Keterangan :

- Kondisi digunakan untuk menentukan pengambilan keputusan
- Pernyataan, berisi perintah-perintah dan akan dijalankan jika kondisi bernilai benar

#### 2. Else if

Pernyataan else-if tidak berdiri sendiri tanpa adanya if, karena bentuk pernyataan else adalah merupakan suatu ingkaran dari pernyataan if.

Bentuknya :

```
if (kondisi)
    pernyataan1;
else
    pernyataan2;
```

pada bentuk di atas, pernyataan1 dijalankan jika kondisi bernilai benar dan pernyataan2 hanya dieksekusi jika kondisi bernilai salah. Pernyataan if maupun else dapat berupa pernyataan tunggal ataupun majemuk, berikut ini ada beberapa kemungkinan :

Kemungkinan 1	Kemungkinan 2	Kemungkinan 3
<pre>if (kondisi)     pernyataan1; else {     pernyataan21;     pernyataan22;     ..... }</pre>	<pre>if (kondisi) {     Pernyataan11;     Pernyataan12;     ..... } else     pernyataan2;</pre>	<pre>if (kondisi) {     Pernyataan11;     Pernyataan12;     ..... } else {     pernyataan21;     pernyataan22;     ..... }</pre>

#### 3. if else if

Bentuknya :

```
if (kondisi1)
    pernyataan1;
```

```

else if (kondisi2)
    pernyataan2;
else if (kondisi3)
    pernyataan3;
else if (kondisi_n)
    pernyataan_n;
else
    pernyataan x;

```

Bentuk pernyataan ini bermanfaat untuk menyeleksi segala kemungkinan tindakan, penyeleksikan dilakukan secara bertingkat, begitu ada kondisi yang bernilai benar, maka pernyataan yang sesuai dengan kondisi tersebut akan dijalankan dan jika tidak ada kondisi yang memenuhi, maka pernyataanX yang akan dijalankan.

### C. PRAKTIK

1. Buatlah algoritma dan program untuk menyatakan jika bil = 0 maka bilangan nol.

#### Langkah Algoritma :

1. Deklarasikan variabel bil
2. Masukkan bil
3. Jika (bil == 0)
4. Jika "Ya" Tampilkan keterangan bilangan nol

#### Program :

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int bil;
    cout<<"Masukkan bilangan : ";
    cin>>bil;
    if (bil==0)
        cout<<"Bilangan Nol"<<endl;
    getch();
}

```

2. Buatlah algoritma dan program untuk menanyakan usia seseorang dan jika usia anda sudah di atas 10 anda boleh mencoba wahana bermain.

#### Langkah Algoritma :

1. Deklarasikan variabel usia
2. Masukkan usia
3. Jika (usia > 10)
4. Jika "Ya" Tampilkan keterangan "Boleh mencoba wahana bermain"

#### Program :

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int usia;
    cout<<"Masukkan usia anda : ";
    cin>>usia;
    if (usia > 10)
        cout<<"Boleh mencoba wahana bermain"<<endl;
    getch();
}

```

3. Berdasarkan praktek di atas diubah dengan menggunakan pernyataan else sebagai berikut jika bil 0, maka bilangan nol, Jika bil bukan 0, maka bil bukan nol

#### Langkah Algoritma :

1. Deklarasikan var bil

2. Masukkan bil
3. Jika (bil == 0)
4. Jika "Ya" Tampilkan keterangan "bilangan nol"
5. Jika "Tidak" Tampilkan keterangan "bilangan bukan nol"

**Program :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int bil;
    cout<<"Masukkan bilangan : ";
    cin>>bil;
    if (bil==0)
        cout<<"Bilangan Nol"<<endl;
    else
        cout<<"Bilangan bukan nol"<<endl;
    getch();
}
```

4. Modifikasi program nomor 2 diubah dengan menggunakan pernyataan else sebagai berikut jika usia > 10, maka "Boleh mencoba wahana bermain", Jika tidak, maka "Tidak boleh mencoba wahana bermain"

**Langkah Algoritma :**

1. Deklarasikan var usia
2. Masukkan usia
3. Jika (usia > 10)
4. Jika "Ya" Tampilkan keterangan "boleh mencoba wahana bermain"
5. Jika "Tidak" Tampilkan keterangan "Tidak boleh mencoba wahana bermain"

**Program :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int usia;
    cout<<"Masukkan usia anda : ";
    cin>>usia;
    if (usia > 10)
        cout<<"Boleh mencoba wahana bermain"<<endl;
    else
        cout<<"Tidak boleh mencoba wahana bermain"<<endl;
    getch();
}
```

5. Buatlah algoritma dan program untuk menentukan apakah sebuah bilangan itu :
  - bilangan genap
  - bilangan ganjil
  - dan jika bukan genap atau ganjil, maka 'salah memasukan bilangan'

**Langkah Algoritma :**

1. Deklarasikan variabel bil
2. Masukkan bil
3. Jika (bil % 2 == 0)
4. Jika "Ya" Tampilkan keterangan "bilangan genap"
5. Jika "Tidak" apakah (bil % 2 == 1)
6. Jika "Ya" Tampilkan keterangan "bilangan ganjil"
7. Jika "Tidak" tampilkan komentar "salah memasukan bilangan"

**Program :**

```
#include <iostream.h>
#include <conio.h>
```

```

void main()
{
    clrscr();
    int bil;
    cout<<"Masukkan sebuah bilangan : ";
    cin>>bil;
    if (bil %2 == 0)
        cout<<"Bilangan Genap "<<endl;
    else if (bil % 2 ==1)
        cout<<"Bilangan Ganjil "<<endl;
    else
        cout<<"Salah masukan bilangan";
    getch();
}

```

6. Buatlah algoritma dan program untuk mengkonversikan nilai angka menjadi nilai huruf, dengan ketentuan sebagai berikut :

Nilai Angka	Nilai Huruf
0 – 20	E
21 – 40	D
41 – 60	C
61 – 80	B
81 - 100	A

Dan jika nilai yang dimasukan lebih dari 100 atau kurang dari 1 (negatif) ada komentar "Salah Memasukan Nilai" Gunakan operator logika && yang dapat menyederhanakan penggunaan if.

**Langkah Algoritma :**

1. Deklarasikan variabel nilai
2. Deklarasikan variabel nilai\_huruf
3. Masukan nilai
4. Jika (nilai >0) && (nilai <= 20)
5. Tampilkan nilai huruf 'E'
6. Jika (nilai >20) && (nilai <= 40)
7. Tampilkan nilai huruf 'D'
8. Jika (nilai >40) && (nilai <= 60)
9. Tampilkan nilai huruf 'C'
10. Jika (nilai >60) && (nilai <= 80)
11. Tampilkan nilai huruf 'B'
12. Jika (nilai >80) && (nilai <= 100)
13. Tampilkan nilai huruf 'A'
14. Jika tidak
15. Tampilkan komentar 'Salah Memasukan Nilai'

**Program :**

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int nilai;
    char nilai_huruf;
    cout<<"Masukkan nilai angka : ";
    cin>>nilai;
    if ((nilai>=0) && (nilai<=20))
        nilai_huruf='E';
    else if ((nilai>20) && (nilai<=40))
        nilai_huruf='D';
    else if ((nilai>40) && (nilai<=60))
        nilai_huruf='C';

```

```

else if ((nilai>60) && (nilai<=80))
    nilai_huruf='B';
else if ((nilai>80) && (nilai<=100))
    nilai_huruf='A';
else
    cout<<"Salah masukkan nilai"<<endl;
cout<<"Nilai huruf : "<<nilai_huruf<<endl;
getch();
}

```

7. Buatlah Flowchart, algoritma dan program untuk menentukan diskon bagi pembeli berdasarkan :

Total_Pembelian	Discount
> 1000000	20 %
500000 - 1000000	15 %
100000 – 500000	10 %
< 100000	0 %

**Langkah Algoritma :**

1. Deklarasikan variabel Tot\_Beli, Discount dan Bayar
2. Masukkan Total Beli
3. Jika (Tot\_Beli > 1000000)
4. Hitung Discount = Tot\_Beli \* 0.2
5. Hitung Bayar = Tot\_Beli - Discount
6. Jika "Tidak" apakah (Tot\_Beli > 500000)
7. Jika "Ya" Hitung Discount = Tot\_Beli \* 0.15
8. Hitung Bayar = Tot\_Beli - Discount
9. Jika "Tidak" apakah (Tot\_Beli > 100000)
10. Jika "Ya" Hitung Discount = Tot\_Beli \* 0.1
11. Hitung Bayar = Tot\_Beli - Discount
12. Jika "Tidak" Tampilkan pesan "Tidak Ada Discount"
13. Hitung Discount = Tot\_Beli \* 0
14. Hitung Bayar = Tot\_beli - Discount
15. Tampilkan Discount dan Bayar

**Program :**

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    float Tot_Beli, Discount, Bayar;
    cout<<"Total Pembelian : ";
    cin>>Tot_Beli;
    if (Tot_Beli > 1000000)
    {
        Discount = Tot_Beli * 0.2;
        Bayar = Tot_Beli - Discount;
    }
    else if (Tot_Beli > 500000)
    {
        Discount = Tot_Beli * 0.15;
        Bayar = Tot_Beli - Discount;
    }
    else if (Tot_Beli > 100000)
    {
        Discount = Tot_Beli * 0.1;
        Bayar = Tot_Beli - Discount;
    }
    else
    {
        cout<<"Tidak ada discount"<<endl;
    }
}

```

```
Discount = Tot_Beli * 0;
Bayar = Tot_Beli -Discount;
}
cout<<"Discoun beli = "<<Discount<<endl;
cout<<"Jumlah bayar = "<<Bayar<<endl;
getch();
}
```

#### D. LATIHAN

Diberikan oleh dosen pengampu pada saat praktikum.

#### E. TUGAS

Buatlah algoritma dan program untuk menjumlahkan dan menghitung nilai mahasiswa dari tiga nilai yang didapat dari nilai\_tugas 20%, nilai\_uts 30% dan nilai\_uas 40%, kemudian konversikan hasil angka yang didapat kedalam bentuk huruf, dengan ketentuan sebagai berikut :

Nilai Angka	Nilai Huruf
0 – 20	E
21 – 40	D
41 – 60	C
61 – 80	B
81 - 100	A

Dan jika nilai yang dimasukan lebih dari 100 atau kurang dari 1 (negatif) ada komentar "Salah Memasukan Nilai"

## PERTEMUAN KE – 7 PEMILIHAN (SELECTION) SWITCH CASE

### A. TUJUAN

1. Memahami struktur pemilihan dalam program
2. Mengetahui struktur switch case yang dapat digunakan dalam pemilihan
3. Menerapkan switch case dalam menyelesaikan berbagai kasus

### B. TEORI SINGKAT

#### Pernyataan Switch

Adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksi.

Bentuknya :

```
switch (ungkapan)
{
    case ungkapan1:
        pernyataan1;
        break;
    case ungkapan2:
        pernyataan2;
        break;
    .....
    default:
        pernyataan_x;
}
```

Keterangan :

- ungkapan1, ungkapan2 dan seterusnya dilakukan secara berurutan dimulai dari yang pertama, sekiranya cocok pernyataan yang mengikuti **case** dijalankan.
- **break** ditemukan dari eksekusi pernyataan **switch** berakhir
- **default** hanya akan dijalankan jika ungkapan pada bagian case tidak ada yang cocok.

Catatan penting :

Semua program yang dibuat dengan pernyataan switch dapat diubah menggunakan pernyataan IF, tetapi tidak semua program yang telah dibuat dengan if dapat diubah dengan menggunakan pernyataan switch, karena ada batasan-batasan pada pernyataan switch.

Seperti program yang telah dibuat di atas dengan menggunakan pernyataan switch, maka program tersebut dapat kita ubah dengan menggunakan pernyataan if.

### C. PRAKTIK

1. Buatlah algoritma dan program, Bila ingin mengetahui kode bulan dan hari sebagai berikut :

Bulan : 1, 3, 5, 7, 8, 10, 12      jumlah hari 31 hari  
Bulan : 4, 6, 9, 11              jumlah hari 30 hari  
Bulan : 2                            jumlah hari 28 atau 29

Dan jika selain angka-angka tersebut dimasukkan muncul komentar "salah memasukkan kode bulan"

**Langkah algoritmanya :**

1. Deklarasikan variabel kode\_bulan
2. Masukkan kode\_bulan
3. Jika kode\_bulan = 1, 3, 5, 7, 8, 10, 12
4. tampilan jumlah hari = 31
5. Jika kode\_bulan = 4, 6, 9, 11
6. tampilan jumlah hari = 30
7. Jika kode\_bulan = 2
8. tampilan jumlah hari = 28 atau 29
9. Jika kode\_bulan yang dimasukkan salah
10. tampilkan komentar "Salah memasukan kode bulan"

**Program :**

```
#include <iostream.h>
#include <conio.h>
void main()
```

```

{
clrscr();
int kode_bulan;
cout<<"Masukkan kode bulan [1..12] : ";
cin>>kode_bulan;
switch (kode_bulan) {
case 1:case 3:case 5: case 7: case 8:case 10:case 12:
    cout<<"Jumlah hari = 31 hari"<<endl;
    break;
case 4:case 6: case 9: case 11:
    cout<<"Jumlah hari = 30 hari"<<endl;
    break;
case 2:
    cout<<"Jumlah hari = 28 atau 29 hari"<<endl;
    break;
default:
    cout<<"Salah masukan kode bulan"<<endl;
    break;
}
getch();
}

```

2. Kasus jurusan yang ada di STMIK AKAKOM dengan ketentuan :

Pilih 1 jurusan MI/D3

Pilih 2 jurusan TK/D3

Pilih 3 jurusan KA/D3

Pilih 4 jurusan TI/S1

Pilih 5 jurusan SI/S1

Jika yang dipilih selain (1, 2, 3, 4, 5) beri komentar "Salah masukan pilihan"

#### Langkah algoritma :

1. Definisikan/masukan variabel pil
2. Masukan pil
3. Jika (pil == 1)
4. tampilkan jurusan MI/D3
5. Jika "Tidak" apakah (pil == 2)
6. Jika "Ya" tampilkan jurusan TK/D3
7. Jika "Tidak" apakah (pil == 3)
8. Jika "Ya" tampilkan jurusan KA/D3
9. Jika "Tidak" apakah (pil == 4)
10. Jika "Ya" tampilkan jurusan TI/S1
11. Jika "Tidak" apakah (pil == 5)
12. Jika "Ya" tampilkan jurusan SI/S1
13. Jika "Tidak" tampilkan komentar "Salah masukan pilihan"

#### Program :

```

#include <iostream.h>
#include <conio.h>
void main()
{
clrscr();
int pil;
cout<<"Masukkan Pilihan : ";
cin>>pil;
switch (pil) {
case 1:cout<<"Jurusan MI/D3"<<endl;break;
case 2:cout<<"Jurusan TK/D3"<<endl;break;
case 3:cout<<"Jurusan KA/D3"<<endl;break;
case 4:cout<<"Jurusan TI/S1"<<endl;break;
case 5:cout<<"Jurusan SI/S1"<<endl;break;

```

```

default:cout<<"Salah masukkan pilihan"<<endl;break;
}
getch();
}

```

3. Buatlah algoritma dan program pemilihan kendaraan untuk menempuh perjalanan dari Yogyakarta menuju Surabaya, dengan ketentuan sebagai berikut :
- pilihan 1 → naik pesawat terbang
  - pilihan 2 → naik kereta api
  - pilihan 3 → naik bus umum
  - pilihan 4 → naik kapal laut
  - Jika pilihan bukan 1, 2, atau 3 → naik mobil pribadi

**Langkah Algoritma :**

1. Deklarasikan variabel pil
2. Masukkan pil
3. Jika (pil == 1)
4. tampilkan "naik pesawat terbang"
5. Jika "Tidak" apakah (pil == 2)
6. Jika "Ya" tampilkan "naik kereta api"
7. Jika "Tidak" apakah (pil == 3)
8. Jika "Ya" tampilkan "naik bus umum"
9. Jika "Tidak" apakah (pil == 4)
10. Jika "Ya" tampilkan "naik kapal laut"
11. Jika "Tidak" tampilkan "naik mobil pribadi"

**Program :**

```

#include <iostream.h>
#include <conio.h>
void main()
{
clrscr();
int pil;
cout<<"Masukkan Pilihan Kendaraan : ";
cin>>pil;
switch (pil) {
case 1:cout<<"Naik pesawat terbang"<<endl;break;
case 2:cout<<"Naik kereta api"<<endl;break;
case 3:cout<<"Naik bus umum"<<endl;break;
case 4:cout<<"Naik Kapal Laut"<<endl;break;
default:cout<<"Naik mobil pribadi"<<endl;break;
}
getch();
}

```

4. Buatlah algoritma dan program untuk menghasilkan suatu pilihan, yang berguna untuk mencari :
- Luas Segi Tiga (dianggap pilihan 1)
  - Luas Persegi Panjang (dianggap pilihan 2)
  - Luas Lingkaran (dianggap pilihan 3)
  - Dan jika pilihan diluar dari 1, 2 dan 3, beri komentar "Anda Salah Memilih"

**Langkah Algoritma :**

1. Deklarasikan konstanta phi = 3.14
2. Deklarasikan variabel pil, alas, tinggi, panjang, lebar, jari, luassg, luaspg dan luaslg
3. Masukkan pil
4. Jika (pil == 1)
5. Masukkan alas dan tinggi
6. Hitung luassg = 0.5 \* alas \* tinggi
7. Tampilkan Luas Segi Tiga
8. Jika "Tidak" apakah (pil == 2)

9. Jika "Ya" masukan panjang dan lebar
10. Hitung  $luaspj = panjang * lebar$
11. Tampilkan Luas Persegi Panjang
12. Jika "Tidak" apakah ( $pil == 3$ )
13. Jika "Ya" masukan jari
14. Hitung  $luaslg = phi * jari * jari$
15. Tampilkan Luas Lingkaran
16. Jika "Tidak" tampilkan komentar "Anda Salah Memilih"

**Program :**

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int pil;
    const phi = 3.14;
    double alas, tinggi, panjang, lebar, jari;
    double luassg, luaspj, luaslg;
    cout<<"Masukkan Pilihan : ";
    cin>>pil;
    switch(pil)
    case 1:
    {
        cout<<"Menghitung Luas Segi Tiga"<<endl;
        cout<<"====="<<endl;
        cout<<"Masukan alas = ";cin>>alas;
        cout<<"Masukan tinggi = ";cin>>tinggi;
        luassg = 0.5 * alas * tinggi;
        cout<<"Luas SegiTiga = "<<luassg<<endl; break;
    case 2:
        cout<<"Menghitung Luas Persegi Panjang"<<endl;
        cout<<"====="<<endl;
        cout<<"Masukan panjang = ";cin>>panjang;
        cout<<"Masukan lebar = ";cin>>lebar;
        luaspj = panjang * lebar;
        cout<<"Luas PersegiPanjang = "<<luaspj<<endl; break;
    case 3:
        cout<<"Menghitung Luas Lingkaran"<<endl;
        cout<<"====="<<endl;
        cout<<"Masukan jari-jari = ";cin>>jari;
        luaslg = phi * jari * jari;
        cout<<"Luas Lingkaran = "<<luaslg<<endl; break;
    default:
        cout<<"Anda Salah Memilih !!"<<endl;
    }
    getch();
}
```

5. Buatlah flowchart, algoritma dan program untuk menentukan hari senin, selasa, rabu, kamis, jumat, sabtu atau minggu berdasarkan nomor hari.

**Langkah Algoritma :**

1. Deklarasikan variabel no hari
2. Masukan no hari
3. Jika ( $nohari == 1$ )
4. tampilkan "Hari ke 1 adalah Minggu"
5. Jika "Tidak" apakah ( $nohari == 2$ )
6. tampilkan "Hari ke 2 adalah Senin"
7. Jika "Tidak" apakah ( $nohari == 3$ )
8. tampilkan "Hari ke 2 adalah Selasa"
9. Jika "Tidak" apakah ( $nohari == 4$ )

10. tampilkan "Hari ke 2 adalah Rabu"
11. Jika "Tidak" apakah (nohari == 5)
12. tampilkan "Hari ke 2 adalah Kamis"
13. Jika "Tidak" apakah (nohari == 6)
14. tampilkan "Hari ke 2 adalah Jumat"
15. Jika "Tidak" apakah (nohari == 7)
16. tampilkan "Hari ke 2 adalah Sabtu"
17. Jika tidak tampilkan "Tidak terdapat nama hari"

**Program :**

```
#include <iostream.h>
int main(){
int nohari;

    cout<<"Masukkan nomor hari (1...7) : ";
    cin>>nohari;
    switch(nohari)
    {
        case 1:
            cout<<"Hari ke-"<<nohari<<" : adalah Minggu";
            break;
        case 2:
            cout<<"Hari ke-"<<nohari<<" : adalah Senin";
            break;
        case 3:
            cout<<"Hari ke-"<<nohari<<" : adalah Selasa";
            break;
        case 4:
            cout<<"Hari ke-"<<nohari<<" : adalah Rabu";
            break;
        case 5:
            cout<<"Hari ke-"<<nohari<<" : adalah Kamis";
            break;
        case 6:
            cout<<"Hari ke-"<<nohari<<" : adalah Jumat";
            break;
        case 7:
            cout<<"Hari ke-"<<nohari<<" : adalah Sabtu";
            break;
        default:
            cout<<"Tidak terdapat nama hari ke-"<<nohari;
    }
    return 0;
}
```

**D. LATIHAN**

Diberikan oleh dosen pengampu pada saat praktikum.

**E. TUGAS**

Ubahlah praktikum diatas menjadi bentuk if

## PERTEMUAN KE – 8 PEMILIHAN (SELECTION) BERTINGKAT

### A. TUJUAN

Mampu menyelesaikan kasus dengan menggunakan konsep seleksi bertingkat

### B. TEORI SINGKAT

Suatu bentuk pemilihan (selektion) bertingkat sangat diperlukan dalam pemrograman, bentuk seperti inilah yang dinamakan sebagai seleksi bersarang. Seleksi bertingkat merupakan penggunaan seleksi di dalam seleksi. Seleksi yang digunakan pada pertemuan ini adalah seleksi bertingkat dua.

### C. PRAKTIK

1. Buatlah program seperti di bawah

```
#include <iostream.h>
int main ()
{
    int a = 100;
    int b = 300;

    if( a == 100 )
    {
        if( b == 200 )
        {
            cout << "Nilai a=100 dan b=200" << endl;
        }
    }
    cout << "Nilai a : " << a << endl;
    cout << "Nilai b : " << b << endl;

    return 0;
}
```

2. Modifikasi praktik di atas agar nilai a dan b bisa dimasukkan melalui keyboard
3. Buatlah program seperti di bawah, cobalah beberapa alternatif masukkan, amati hasilnya

```
#include <iostream.h>
void main ()
{
    int kw,a,b;
    char pil;
    cout<<" produk a = dancow \n";
    cout<<" produk b = bendera \n";
    cout<<" maskan pilihan : "; cin>>pil;
    cout<<" Masukkan Jenis KW 1,2 : "; cin>>kw;

    if (pil=='a'){
        if (kw==1){
            cout<<" Kw 1 Dancow";
            cout<<" Harga = 25.000";}
        if (kw==2){
            cout<<" Kw 2 Dancow";
            cout<<" Harga = 20.000";}}

    if (pil=='b'){
        if (kw==1){
            cout<<" Kw 1 Bendera";
            cout<<" Harga = 25.000";}
        if (kw==2){
            cout<<" Kw 2 Bendera";
            cout<<" Harga = 20.000";}}
}
```

4. Buatlah program seperti di bawah, cobalah beberapa alternatif masukkan, amati hasilnya

```
#include<iostream.h>
int main() {
    int num;
    cout<<" Masukkan nomor dari 1 sampai 999 = ";
    cin>>num;
    if(num>0)
    {
        cout<<num<<" Merupakan bilangan positif "<<endl;
        if(num<10)
        {
            cout<<num<<" satu digit ";
        }
        else if(num<100)
        {
            cout<<num<<" dua digit ";
        }
        else if(num<1000)
        {
            cout<<num<<" tiga digit ";
        }
    }
    return 0;
}
```

5. Buatlah program seperti di bawah, cobalah beberapa alternatif masukkan, amati hasilnya

```
#include<iostream.h>
void main()
{
    float pendapatan, jasa, bonus, total;
    cout<<"Pendapatan Hari ini Rp. ";
    cin>>pendapatan;

    if(pendapatan>=0 && pendapatan <=300000)
    {
        jasa=10000;
        bonus=0.1*pendapatan;
    }
    else
    {
        if(pendapatan<=500000)
        {
            jasa=20000;
            bonus=0.15*pendapatan;
        }
        else if(pendapatan<=1000000)
        {
            jasa=50000;
            bonus=0.2*pendapatan;
        }
        else
        {
            jasa=100000;
            bonus=0.3*pendapatan;
        }
    }
    total=bonus+jasa;
    cout<<"Jasa Penjualan Rp."<<jasa<<endl;
    cout<<"Bonus Rp."<<bonus<<endl;
    cout<<"====="<<endl;
    cout<<"Hasil Total Rp."<<total<<endl;
}
```

```
}
```

6. Buatlah program seperti di bawah, cobalah beberapa alternatif masukkan, amati hasilnya

```
#include <iostream.h>
void main(){
int a,b,c;

    cout<<"\npilihan 1 : Laki-laki";
    cout<<"\npilihan 2 : Perempuan";
    cout<<"\nmasukkan pilihan : ";
    cin>>a;

    if (a==1)
    {
        cout<<"masukkan usia : ";
        cin>>b;
        if (b>35)
        {
            cout<<"maaf, anda di tolak karena terlalu tua";
        }
        else if ((b>23)&&(b<35))
        {
            cout<<"selamat, anda di terima";
        }
        else
        {
            cout<<"maaf, anda di tolak anda masih muda";
        }
    }
    else
    {
        cout<<"Silahkan pilih pendidikan anda?";
        cout<<"\n1. D3";
        cout<<"\n2. S1";
        cout<<"\n";
        cin>>c;
        if (c==1)
        {
            cout<<"Anda kuliah jenjang D3";
        }
        else
        {
            cout<<"Anda kuliah jenjang S1";
        }
    }
    getch();
}
```

#### D. LATIHAN

Diberikan oleh dosen pengampu pada saat praktikum.

#### E. TUGAS

Buatlah algoritma dan program untuk menentukan suatu tahun termasuk kabisat atau tidak.

Keterangan :

1. satu tahun disebut tahun kabisat jika habis dibagi 4 dengan pengecualian adalah tahun yang habis dibagi 100 dan tidak habis di bagi 400
2. Misal tahun 2000 termasuk sebagai tahun kabisat. Tahun1998 termasuk sebagai tahun bukan kabisat.

**Contoh output :**

Jika input tahun 1998 maka output :

menentukan tahun kabisat  
tahun: 1998  
bukan tahun\_kabisat

Jika input tahun 2000 maka output :

menentukan tahun kabisat  
tahun: 2000  
tahun\_kabisat

## PERTEMUAN KE – 9 PERULANGAN /ITERASI WHILE

### A. TUJUAN

Mahasiswa diharapkan dapat membuat algoritma serta program untuk iterasi bertingkat dalam kasus sederhana

### B. TEORI SINGKAT

Salah satu kelebihan komputer dibandingkan dengan manusia adalah kemampuan untuk melaksanakan suatu perintah berulang kali tanpa mengenal lelah dan bosan. Di dalam algoritma, pengulangan atau kalang (*repetition* atau *loop*) dapat dilakukan sejumlah kali, atau sampai kondisi berhenti pengulangan tercapai.

#### Struktur Pengulangan

Struktur pengulangan secara umum terdiri atas dua bagian:

1. Kondisi pengulangan, yaitu ekspresi boolean yang harus dipenuhi untuk melakukan pengulangan. Kondisi ini ada yang menyatakan secara eksplisit oleh pemrogram atau dikelola sendiri oleh komputer (implisit)
2. Badan (*body*) pengulangan, yaitu bagian algoritma yang diulang

Disamping itu, struktur pengulangan biasanya disertai dengan bagian

1. Inisialisasi, yaitu aksi yang dilakukan sebelum pengulangan dilakukan pertama kali
2. Terminasi, yaitu aksi yang dilakukan setelah pengulangan selesai dilaksanakan

Inisialisasi dan terminasi tidak selalu harus ada, namun pada berbagai kasus inisialisasi umumnya diperlukan.

Struktur pengulangan secara umum :

```
<inisialisasi>  
Awal pengulangan  
  Badan pengulangan  
  Akhir pengulangan  
<terminasi>
```

Yang dalam hal ini awal dan akhir pengulangan dinyatakan sebagai kata kunci yang tergantung pada konstruksi pengulangan yang digunakan. Selain itu, *<inisialisasi>* dan *<terminasi>* adalah bagian yang opsional (tidak selalu harus dituliskan).

Di dalam algoritma terdapat beberapa macam konstruksi pengulangan yang berbeda. Beberapa konstruksi dapat dipakai untuk masalah yang sama, namun ada konstruksi pengulangan yang hanya cocok dipakai untuk masalah tertentu. Pemilihan konstruksi pengulangan untuk masalah tertentu dapat mempengaruhi kebenaran algoritma. Pemilihan konstruksi pengulangan yang tepat bergantung pada masalah yang akan diprogram.

#### Pernyataan while

Pernyataan **while** merupakan salah satu pernyataan yang berguna untuk memproses suatu pernyataan atau beberapa pernyataan beberapa kali. Bentuk pernyataan ini :

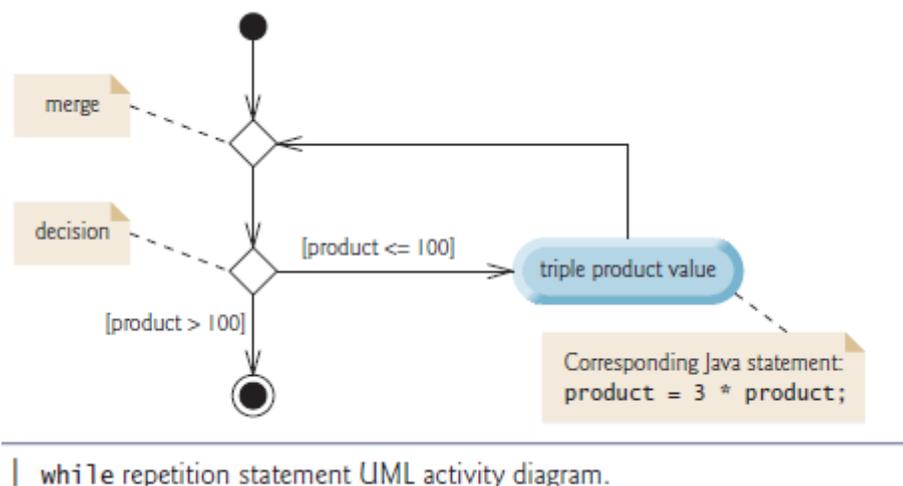
```
while (ungkapan)  
  pernyataan;
```

Bagian *pernyataan* tertentu saja bisa berupa pernyataan manajemen sehingga bentuknya menjadi seperti berikut:

```
while (ungkapan){  
  pernyataan1;  
  pernyataan2;  
  ...  
  pernyataanN;  
}
```

Bagian pernyataan yang mengikuti **while** akan dieksekusi selama *ungkapan* pada **while** bernilai benar (tidak sama dengan nol). Perlu diketahui, pengujian terhadap ungkapan pada **while** dilakukan sebelum bagian *pernyataan*. Oleh karena itu ada kemungkinan bagian pernyataan pada **while** tidak dijalankan sama sekali, yaitu kalau kondisi yang pertama kali bernilai salah.

Contoh flowchart untuk perulangan while.



Jika menggunakan pernyataan **while**, pastikan bahwa suatu saat ungkapan pada **while** bernilai salah.

### C. PRAKTIK

Praktik 1. Memperoleh faktor persekutuan terbesar.

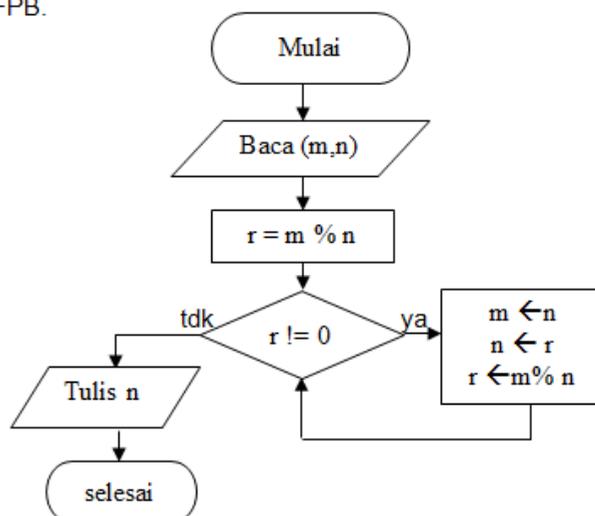
Algoritma untuk memperoleh faktor persekutuan terbesar (FPB). Algoritma sebagai berikut :

1. Masukan (m,n)
2.  $r \leftarrow \text{sisapembagian}(m,n)$
3. Ulang Selama  $r \neq 0$ 
  - $m \leftarrow n$
  - $n \leftarrow r$
  - $r \leftarrow \text{sisapembagian}(m,n)$

Akhir-Ulang

4. Tampilkan n

Flowchat FPB.



Kode program dengan C++

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

```

int main()
{
    int m, n, r;
    cout << "Masukkan m: ";
    cin >> m;
    cout << "Masukkan n: ";
    cin >> n;
    r = m % n;
    while (r != 0)
    {
        m = n;
        n = r;
        r = m % n;
    }
    cout << "FPB: " << n << endl;
    return 0;
}

```

Perintah :

```
r = m % n;
```

digunakan untuk memperoleh sisa pembagian  $m$  dengan  $n$  dan hasilnya diletakkan pada variabel  $r$ .

Hasil eksekusi program :

```

D:\00 materi\bahan\algoritma\Source Code\Bab 6\fpb.exe
Masukkan m: 15
Masukkan n: 3
FPB: 3
-----
Process exited after 5.546 seconds with return value 0
Press any key to continue . . .

```

Praktik 2. Menampilkan suatu teks 6 kali

Algoritma untuk menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar".

1. Bil  $\leftarrow$  1
  2. ULANG SELAMA bil  $\leq$  6
    - Tampilkan ("selamat belajar")
    - bil  $\leftarrow$  bil + 1
- AKHIR-ULANG

Kode program

```

#include <iostream>
using namespace std;

int main()
{
    int bil;

    bil = 1;
    while (bil <= 6)
    {
        cout << "Selamat Belajar" << endl;
        bil = bil + 1;
    }

    return 0;
}

```

### Praktik 3. Algoritma nilai rata-rata

Membuat algoritma untuk menghitung nilai rata-rata yang meminta  $n$  buah bilangan bulat dimasukkan dari keyboard, kemudian menampilkan nilai rata-rata dari keseluruhan bilangan tersebut.

Algoritma dimulai dengan mula-mula meminta jumlah data yang akan dimasukkan dari keyboard (berupa  $n$ ). Selanjutnya,  $n$  buah data dibaca dari keyboard dan setiap kali data selesai dibaca, nilainya ditambahkan ke variabel  $jum$ . Dengan demikian, setelah data terakhir dimasukkan,  $jum$  berisi penjumlahan  $n$  buah nilai yang dimasukkan dari keyboard.

Algoritma nilai rata-rata:

1. Masukkan ( $n$ )
2.  $Jum \leftarrow 0$
3.  $Pencacah \leftarrow 0$
4. ULANG SELAMA  $pencacah < n$ 
  - Masukkan( $bil$ )
  - $jum \leftarrow jum + bil$  //jumlahkan
  - $pencacah \leftarrow pencacah + 1$
- AKHIR-ULANG
5.  $rataRata \leftarrow jum/n$
6. tampilkan( $rataRata$ )

Tuliskan kode program yang mengimplementasi algoritma nilai rata-rata

Praktik 4. Buatlah flowchart dan program untuk menampilkan deret angka dengan batas tertentu.

Contoh :

**Masukkan angka : 6**

**Output :**

**6**

**5**

**4**

**3**

**2**

**1**

### Praktik 5. Menampilkan deret bilangan

Buatlah **algoritma** dan **program** yang meminta sebuah bilangan bulat dimasukkan dari keyboard. kemudian menampilkan deret seperti berikut :

**Masukkan sebuah bilangan : 50**

**Keluaran :**

1 2 4 7 11 16 22 29 37 46

Bila bilangan yang dimasukkan adalah 50. Bantuan : Angka terakhir yang ditampilkan dalam deret harus tidak lebih dari bilangan yang dimasukkan dari keyboard.

Praktik 6. Buatlah flowchat dan program untuk menampilkan deret kelipatan suatu bilangan yang diinput dan dengan batas nilai tertentu yang diinput dari keyboard.

Contoh :

**Masukan angka kelipatan : 4**

**Masukan Batas nilai : 32**

**4 8 12 16 20 24 28 32**

## D. LATIHAN

Diberikan oleh masing-masing dosen pengampu praktikum

## E. TUGAS

1. Buatlah algoritma dan program untuk menerima 10 data suhu air, kemudian mencetak pesan dilayar monitor tentang bentuk air sesuai dengan data suhu yang dimasukkan.
2. Buatlah algoritma dan program untuk menerima 50 bilangan, kemudian mencetak pesan ke layar apakah bilangan ganjil atau genap.

## PERTEMUAN KE – 10 PERULANGAN /ITERASI DO - WHILE

### A. TUJUAN

Mahasiswa dapat membuat program untuk menyelesaikan kasus dengan menggunakan iterasi do-while

### B. TEORI SINGKAT

Pernyataan **do-while** juga berguna untuk mengulang proses.

```
do{
    pernyataan1;
    pernyataan2;
    ...
    pernyataanN;
}while (ungkapan)
```

Bagian *pernyataan1* hingga *pernyataanN* dijalankan secara berulang sampai *ungkapan* bernilai salah (sama dengan nol). Namun berbeda dengan **while**, pengujian *ungkapan* dilakukan di belakang (setelah bagian *pernyataan*)

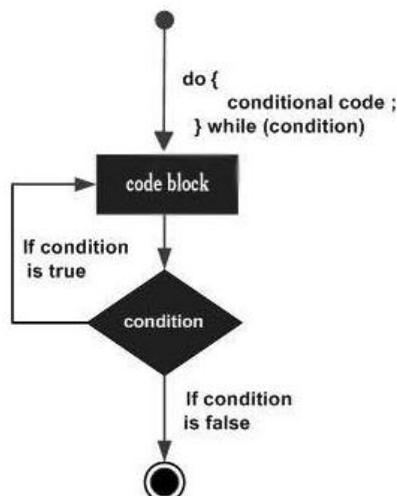
Atau dapat dituliskan dengan bentuk

```
ULANG
    Pernyataan1
    ...
    pernyataanN
SAMPAI kondisi
```

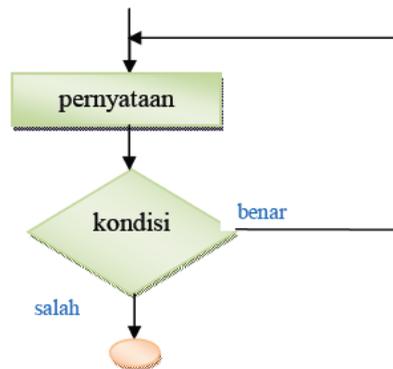
Diterjemahkan dalam bahasa pemrograman

```
do
{
    Pernyataan1
    ...
    pernyataanN
} while (kondisi==true)
```

Diagram activity untuk perulangan dengan do... while



Flowchart untuk perulangan dengan do... while



Contoh program :

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

int main ()
{
    // Local variable declaration:
    int a = 10;

    // do loop execution
    do
    {
        cout << "value of a: " << a << endl;
        a = a + 1;
    }while( a < 20 );

    return 0;
}
```

Program diatas akan memberikan hasil

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

Pada dasarnya struktur perulangan do....while sama saja dengan struktur while, hanya saja pada proses perulangan dengan while, seleksi berada di while yang letaknya di atas sementara pada perulangan do....while, seleksi while berada di bawah batas perulangan. Jadi dengan menggunakan struktur do...while sekurang-kurangnya akan terjadi satu kali perulangan.

Pada struktur do-while kondisi pengecekan ditempatkan di bagian akhir. Hal ini menyebabkan struktur pengulangan ini minimal akan melakukan satu kali proses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah).

### C. PRAKTIK

Praktik 1. Kode program menampilkan i dengan pengulangan dengan do... while

```
// DoWhile001
```

```
#include <iostream>

using namespace std;
```

```

int main()
{
    int i = 4;
    do
    {
        cout << i << " ";
        i++;
    }
    while (i < 10);
    cout << endl;
    return 0;
}

```

#### // DoWhile002

```

#include <iostream>
using namespace std;
int main()
{
    int i = 14;
    do
    {
        cout << i << " ";
        i++;
    }
    while (i < 10);
    cout << endl;
    return 0;
}

```

#### // DoWhile003

```

#include <iostream>
using namespace std;
int main()
{
    int i = 0;
    do
    cout << i++ << " ";
    while (i < 10);
    cout << endl;

    return 0;
}

```

Praktik 2. Menampilkan suatu teks 6 kali

Algoritma untuk menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar".

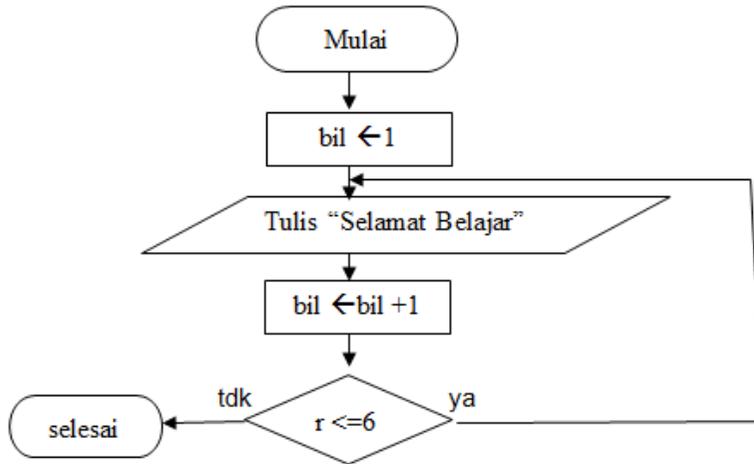
1. Mulai
2. Bil  $\leftarrow$  1
3. ULANG
  - Tampilkan ("selamat belajar")
  - bil  $\leftarrow$  bil + 1
- SAMPAI bil  $\leq$  6
4. Selesai

Atau Algoritma untuk menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar".

1. Mulai

2. Bil  $\leftarrow$  1
3. Do
  - Tampilkan ("selamat belajar")
  - bil  $\leftarrow$  bil + 1
- While bil  $\leq$  6
4. Selesai

Flowchart menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar"



Kode Program menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar"

```

#include <iostream>
using namespace std;
int main()
{
    int bil = 1;
    do
    {
        cout << "Selamat Belajar"<<endl;
        bil = bil + 1;
    }
    while (bil <=6);
    cout << endl;
    return 0;
}
  
```

### Praktik 3. Membuat menu

#### Kode program

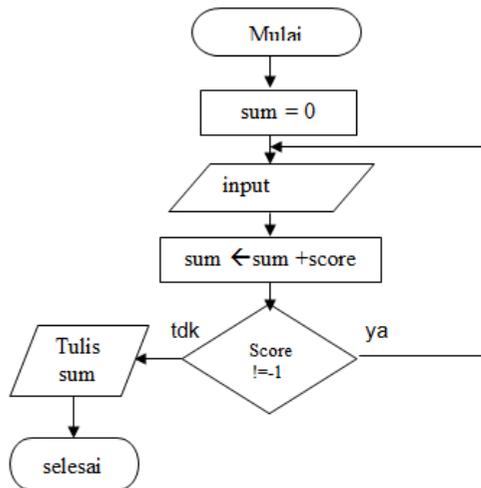
```

#include <iostream>
int main()
{
    using namespace std;
    int nSelection;
    do
    {
        cout << "Please make a selection: " << endl;
        cout << "1) Addition" << endl;
        cout << "2) Subtraction" << endl;
        cout << "3) Multiplication" << endl;
        cout << "4) Division" << endl;
        cout << "?";
        cin >> nSelection;
    } while (nSelection != 1 && nSelection != 2 &&
            nSelection != 3 && nSelection != 4);
    return 0;
}
  
```

```
}
```

#### Praktik 4. Flowchart dan program untuk menghitung jumlah score

##### Flowchart menghitung jumlah score



##### Program menghitung jumlah score

```
#include <iostream>
using namespace std;
int main()
{
    int score, sum = 0;
    do
    {
        cout << "Masukan Nilai (-1 untuk Keluar): ";
        cin >> score;
        if (score != -1)
            sum += score;
    } while (score != -1);

    cout << "Hasil Penjumlahan = " << sum << endl;
    return 0;
}
```

Praktik 5. Buat flowchart dan program dengan memodifikasi praktik 4 sehingga menghasilkan keluaran nilai rata-rata dari score yang telag diinputkan.

Praktik 6. Buatlah flowchart dan program untuk menampilkan deret angka dengan batas tertentu menggunakan perulangan do...while.

Contoh :

**Masukkan angka : 6**

**Output :**

**6  
5  
4  
3  
2  
1**

#### **D. LATIHAN**

Latihan diberikan oleh dosen pengampu pada saat praktikum.

#### **E. TUGAS**

1. Buatlah algoritma dan program untuk menerima beberapa data suhu air, kemudian mencetak pesan dilayar monitor tentang bentuk air sesuai dengan data suhu yang dimasukkan. Batas jumlah bilangan yang diinputkan ditentukan oleh user.
2. Buatlah algoritma dan program untuk menerima beberapa bilangan, kemudian mencetak pesan ke layar apakah bilangan ganjil atau genap. Batas jumlah bilangan yang diinputkan ditentukan oleh user.

## PERTEMUAN KE – 11 PERULANGAN /ITERASI FOR

### A. TUJUAN

Mahasiswa dapat membuat program untuk menyelesaikan kasus dengan menggunakan iterasi for

### B. TEORI SINGKAT

#### For Statement

Ini adalah statement **perulangan** yang paling sering digunakan. Statement for memiliki 3 parameter, yaitu nilai awal (initial value), tes kondisi yang menentukan akhir loop, dan penentu perubahan nilai.

Sintak Program

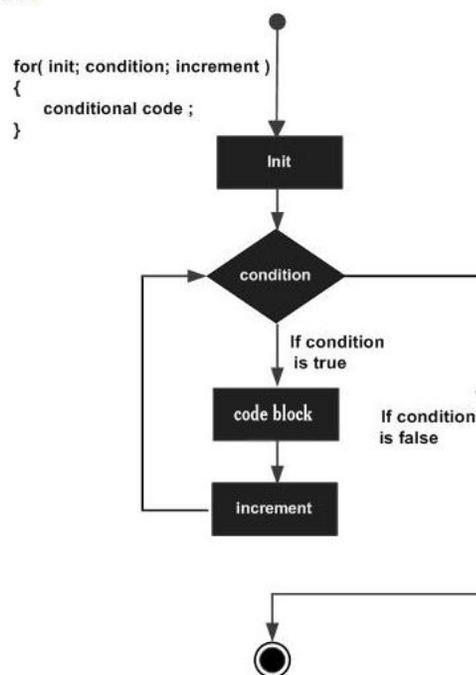
```
for ( init; condition; increment )  
{  
    statement(s);  
}
```

Keterangan :

- **init** : ekspresi yang digunakan untuk melakukan inisialisasi terhadap variable-variabel tertentu, terutama variable yang digunakan untuk melakukan iterasi. Init-exp dapat berupa ekspresi maupun pendefinisian variable.
- **condition**: ekspresi yang memegang control terhadap proses perulangan tersebut, pada bagian ini akan ditentukan apakah proses perulangan akan tetap berlangsung atau tidak.
- **Increment** : digunakan untuk mengatur perubahan nilai variable. Umumnya nilai variable tersebut bertambah / berkurang 1 (satu)

Diagram aktivitas

Flow Diagram:



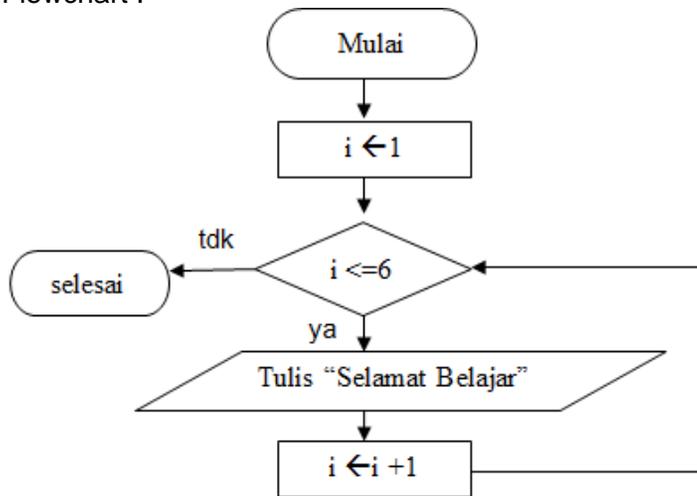
### C. PRAKTIK

Praktik 1. Buatlah algoritma untuk menampilkan 6 buah baris yang berisi tulisan "Selamat Belajar" menggunakan perulangan for.

Algoritma :

UNTUK bil  $\leftarrow$  1 s/d 6  
Tampilkan ("Selamat belajar")  
AKHIR-UNTUK

Flowchart :



Program :

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

int main()
{
    int bil;
    for (bil = 1; bil <= 6; bil++)
    {
        cout << "Selamat belajar\n";
    }

    return 0;
}
```

Praktik 2. Buatlah algoritma untuk menghitung faktorial dengan menggunakan bentuk For. Perlu diketahui suatu faktorial didefinisikan seperti berikut :

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$$

Sebagai contoh :

$$2! = 2 \times 1 = 2$$

$$3! = 3 \times 2 \times 1 = 6$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

Algoritma :

1. Masukkan(n)
2. Hasil  $\leftarrow$  0
3. UNTUK bil  $\leftarrow$  n S/D 1 LANGKAH -1  
Hasil  $\leftarrow$  hasil x bil  
AKHIR-UNTUK
4. Tampilkan(hasil)

### Kode program untuk menghitung faktorial

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

int main()
{
    int bil, n;
    long int hasil;

    cout << "n = ";
    cin >> n;

    hasil = 1;
    for (bil = n; bil >= 1; bil--)
    {
        hasil = hasil * bil;
    }

    cout << "n! = " << hasil << endl;

    return 0;
}
```

Praktik 3. Membuat algoritma dan program untuk membaca sebuah string, kemudian menampilkan karakter dalam string tersebut per satu baris

#### Algoritma

1. Mulai
2. Masukkan (SuatuString)
3.  $Jum \leftarrow \text{jumlahKarakter}(\text{SuatuString})$
4. Untuk indeks  $\leftarrow 0$  s/d  $jum - 1$   
    Karakter  $\leftarrow \text{SuatuString}(\text{indeks})$   
    Tampilkan(Karakter)
- Akhir-Untuk
5. Selesai

#### Program

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

int main(){
    char string[80];
    int indeks, jum;
    char karakter;
    cout <<"Masukkan suatu string: ";
    cin.getline(string, 80);
    jum = strlen(string);

    for (indeks = 0; indeks < jum; indeks++){
        karakter = string[indeks];
        cout <<karakter <<endl;
    }

    return 0;
}
```

#### Praktik 4. Algoritma dan Program Bunga Majemuk

Amir menabung sebesar 5 juta dan setiap tahun mendapatkan bunga majemuk 7%. Buat algoritma dan program untuk menampilkan uang amir dari akhir tahun pertama hingga akhir tahun ke delapan.

Algoritma :

1. Mulai
2. Pokok  $\leftarrow$  5000000
3. UNTUK tahun  $\leftarrow$  1 s/d 8  
    Bunga  $\leftarrow$  pokok \* 7/100  
    Tampilkan (tahun, pokok, bunga)  
    Pokok  $\leftarrow$  pokok + bunga  
    Akhir untuk
4. Selesai

Program

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

int main()
{
    long int pokok, bunga;
    int tahun;

    pokok = 5000000;
    for (tahun = 1; tahun <= 8; tahun++)
    {
        bunga = pokok * 7 / 100;
        cout << setw(2) << tahun
              << setw(11) << pokok
              << setw(11) << bunga
              << endl;

        pokok = pokok + bunga;
    }

    return 0;
}
```

#### D. LATIHAN

- Latihan diberikan oleh dosen pengampu pada saat praktikum.

#### E. TUGAS

1. Buatlah algoritma( boleh dalam bentuk flowchat, activity diagram, bahasa formal atau algoritma) untuk membuat segitiga dengan karakter bintang sebagai berikut

```
*
**
***
****
*****
```

2. Buatlah algoritma( boleh dalam bentuk flowchat, activity diagram, bahasa formal atau algoritma) untuk membuat segitiga dengan karakter bintang sebagai berikut

```
*
**
***
****
*****
```

## PERTEMUAN KE – 12 PERULANGAN /ITERASI BERSARANG

### A. TUJUAN

Mahasiswa diharapkan dapat membuat algoritma serta program untuk iterasi bertingkat dalam kasus sederhana

### B. TEORI SINGKAT

#### Sintaks :

Sintaks bersarang untuk pernyataan loop dalam C ++ adalah sebagai berikut:

```
for ( init; condition; increment )
{
    for ( init; condition; increment )
    {
        statement(s);
    }
    statement(s); // Anda dapat menempatkan statements.
}
```

Sintaks bersarang untuk pernyataan while loop dalam C ++ adalah sebagai berikut:

```
while(condition)
{
    while(condition)
    {
        statement(s);
    }
    statement(s); // Anda dapat menempatkan statements.
}
```

Sintaks bersarang untuk pernyataan do...while loop dalam C ++ adalah sebagai berikut:

```
do
{
    statement(s); // Anda dapat menempatkan statetments.
    do
    {
        statement(s);
    }while( condition );
}while( condition );
```

### C. PRAKTIK

Praktik 1. Cobalah praktikum di bawah, jaankan, amati hasilnya dan jelaskan

```
#include <iostream.h>
main()
{
    int x, y;
    for(y=1;y<=4;y++)
    {
        for(x=1;x <=3;x++)
        cout << "y=" << y << "x=" << x << "\t";
        cout <<endl;
    }
    return 0;
}
```

Praktik 2. Cobalah praktikum di bawah, jaankan, amati hasilnya dan jelaskan

```
#include<iostream.h>
#include<conio.h>
main()
{
int a, b;
clrscr();
for(a = 1; a <= 5; a++)
{
cout<<endl;
for(b = a; b <= 5; b++)
cout<<a;
}
}
```

Praktik 3. Cobalah praktikum di bawah, jaankan, amati hasilnya dan jelaskan

```
#include
#include

void main()
{
clrscr();
int i,j,n,k;
cout<<"Masukkan batas : ";
cin>>n;
for(i=1; i<=n; i++)
{
for (k=1;k<=i;k++){
cout<<" ";
}
for(j=n; j>=i; j--)
{
cout<<"*";
}
cout<
}
getch();
}
```

Praktik 4. Cobalah praktikum di bawah, jaankan, amati hasilnya dan jelaskan

```
#include <iostream.h>
#include <conio.h>
#include <string.h>

int main ()
{
int i,j,panjang;
char nama[80];

cout <<"Masukkan nama anda : ";
cin >>nama;

cout <<"\n" ;

cout << "======"<<"\n";
cout << "Nama : " << nama <<"\n";
cout << "======"<<"\n";
```

```

cout <<"\n" ;
cout <<"Hasil Proses " <<"\n" ;
cout <<"_____ " <<"\n" ;
cout <<"\n" ;

panjang = strlen(nama);
for(i=0; i<panjang; i++)
{
for (j=i; j<panjang; j++)
cout <<nama[j];

cout <<"\n";
}

getch ();
}

```

Praktik 5. Ubahlah ke bentuk while dan do-while

#### D. LATIHAN

- Latihan diberikan oleh dosen pengampu pada akhir praktikum.

#### E. TUGAS

Buatlah program aplikasi yang memiliki fasilitas input data, mengurutkan data dengan metode selection sort, mengurutkan data dengan metode bubble sort dan menampilkan data baik data awal maupun data yang telah terurut. Fasilitas program dikemas dalam bentuk menu pilihan.

Contoh Tampilan menu:

Aplikasi Pengurutan

1. Masukan data
2. Pengurutan Dengan metode Selection Sort
3. Pengurutan Dengan metode Bubble Sort
4. Tampilkan data awal
5. Tampilkan data terurut
6. Keluar

Masukkan Pilihan anda (1...6)?

## PERTEMUAN KE – 13 PENGGABUNGAN SELEKSI DAN ITERASI

### A. TUJUAN

Mahasiswa diharapkan dapat membuat algoritma dan program untuk berbagai kasus yang menggunakan seleksi dalam perulangan

### B. TEORI SINGKAT

Pada dasarnya sebuah program dieksekusi secara runtut dari mulai statement yang pertama kali dibaca dilanjutkan dengan statement yang dibaca berikutnya. Tetapi alur pemrosesan itu bisa diubah dengan menggunakan **seleksi dan perulangan** sehingga memungkinkan sebuah program menjalankan tugas yang lebih kompleks. Alur pemrosesan dimulai dari bagian utama program. Seleksi dan iterasi/perulangan dapat digabungkan dengan dua kemungkinan, yang pertama seleksi dalam perulangan, gambaran sederhana dari model ini dapat dilihat sebagai berikut :

```
for (ungkapan1;ungkapan2;ungkapan3)
{
    if(kondisi)
    {
        Statement1;
    }
    else
        Statement2;
}
```

Dalam model tersebut, statement1 dalam seleksi akan dijalankan selama ungkapan2 dalam perulangan bernilai benar dan kondisi di dalam if juga bernilai benar. Statement2 akan dijalankan jika ungkapan2 dalam perulangan bernilai benar tetapi kondisi dalam if bernilai salah.

### C. PRAKTIK

#### 1. Ketikkan program di bawah dan jalankan

```
#include<iostream.h>
int main()
{
    int i;
    for(i=1;i<=10;i++)
    {
        if (i%2==0)
            cout<<i<<" adalah bilangan genap"<<endl;
        else
            cout<<i<<" adalah bilangan ganjil"<<endl;
    }
    return 0;
}
```

#### 2. Cobalah program di bawah, amati hasilnya dan jelaskan

```
#include <iostream.h>
void main()
{
    int bil, input1, input2;
    cout<<"Masukkan Sebuah Bilangan awal = ";
    cin>>input1;
    cout<<"Masukkan Sebuah Bilangan akhir = ";
    cin>>input2;
    for(bil = input1;bil<=input2;bil++)

    {
        if(bil%2==0) {
```

```
        cout<<bil;
    }
}
}
```

### 3. Cobalah program di bawah untuk menampilkan bilangan prima

```
#include <iostream.h>
#include <conio.h>

void main()
{
    int i, j;

    for(i=2; i<100; i++) {
        for(j=2; j <= (i/j); j++)
            if(!(i%j)) break;
        if(j > (i/j)) cout << i << " is prime\n";
    }
    getch();
}
```

### 4. Cobalah program di bawah, jalankan, amati hasilnya dan jelaskan

```
#include <iostream.h>
void main(void)
{
    for(int v=1; v <= 10; v++)
    {
        if(v == 5)
            continue;
        cout<<v<<endl;
    }
}
```

### 5. Ketikkan program di bawah

```
#include<iostream>
using namespace std;
int main(){
    int input1, input2, prima, pindah;
    cout<<"Masukkan nilai pertama = ";
    cin>>input1;
    cout<<"Masukkan nilai kedua = ";
    cin>>input2;
    if(input1>input2)
    {
        pindah = input1;
        input1 = input2;
        input2 = pindah;
    }

    cout<<"\n\nBilangan Prima antara "<<input1<<" sampai
"<<input2<<" = ";

    for(int a=input1;a<=input2;a++)
    {
        prima=0;
        for(int b=1;b<=a;b++)
        {
            if(a%b==0)
            {
                prima++;
            }
        }
    }
}
```

```

}

if(prima==2){
cout<<a<<" , ";
}
}
cout<<"\n\n";
}

```

6. Cobalah sintak di bawah, jalankan dan amati hasilnya.

```

#include <iostream.h>
#include <conio.h>
/**-----
Nama : .....
NIM  : .....
-----**/
void main() {
    int pilihan;
    cout << "MENU PILIHAN" << endl;
    cout << "-----" << endl;
    cout << "1. hitung luas lingkaran" << endl;
    cout << "2. hitung luas persegi" << endl;
    cout << "3. hitung luas segitiga" << endl;
    cout << "4. exit" << endl;
    cout << "-----" << endl;
    while (pilihan>=0){
    cout << "pilihan anda (1/2/3/4) : "; cin >> pilihan;
        if (pilihan== 1) {
            cout << "-----" << endl;
            cout << "Menghitung Luas Lingkaran" << endl;
            float r;
            cout << "Masukan jari-jari      : "; cin >> r;
            float L1 = 3.14*r*r;
            cout << "Luas Lingkaran      : " << L1 << endl;
            cout << "-----" << endl;
        }
        else if (pilihan== 2){
            cout << "-----" << endl;
            cout << "Menghitung Luas Persegi" << endl;
            float p;
            cout << "Masukan Panjang      : "; cin >> p;
            float l;
            cout << "Masukan Lebar        : "; cin >> l;
            float Lp = p*l;
            cout << "Luas Persegi        : " << Lp << endl;
            cout << "-----" << endl;
        }
        else if (pilihan== 3){
            cout << "-----" << endl;
            cout << "Menghitung Luas Segitiga" << endl;
            float a;
            cout << "Masukan Alas          : "; cin >> a;
            float t;
            cout << "Masukan Tinggi        : "; cin >> t;
            float Ls = 0.5*a*t;
            cout << "Luas Segitiga        : " << Ls << endl;
            cout << "-----" << endl;
        }
        else if (pilihan==4) {
            exit(4);
        }
    }
}

```

```
    else {  
        cout << "Kode Yang Anda Masukan Salah...!!!"<< endl;  
    }  
}
```

#### **D. LATIHAN**

- Latihan diberikan oleh dosen pengampu pada akhir praktikum.

#### **E. TUGAS**

- Tugas diberikan oleh dosen pengampu pada akhir praktikum

## PERTEMUAN KE – 14 PENGgabUNGAN SELEKSI DAN ITERASI

### A. TUJUAN

Mahasiswa diharapkan dapat membuat algoritma dan program untuk berbagai kasus yang menggunakan perulangan dalam seleksi

### B. TEORI SINGKAT

Seleksi dan iterasi/perulangan dapat digabungkan dengan dua kemungkinan, yang pertama seleksi dalam perulangan (sudah dibahas di pertemuan 13) dan yang kedua perulangan dalam seleksi, gambaran sederhana dari model ini kedua dapat dilihat sebagai berikut :

```
if(kondisi)
{
    for(ungkapan1;ungkapan2;ungkapan3)
    {
        Statement;
    }
}
```

Dalam model ini, perulangan baru akan dijalankan jika kondisi dalam if bernilai benar. Statement akan terus dijalankan selama ungkapan2 dalam for masih bernilai benar.

### C. PRAKTIK

#### 1. Cobalah program di bawah

```
#include<iostream.h>
int main()
{
    int nilai,i;
    cout<<"masukkan nilai = ";
    cin>>nilai;
    if(nilai>5)
    {
        for(i=nilai;i>=1;i--)
        {
            cout<<i<<" ";
        }
    }
    else
    {
        for(i=nilai;i<=10;i++)
        {
            cout<<i<<" ";
        }
    }
    return 0;
}
```

#### 2. Cobalah program di bawah, amati hasilnya dan jelaskan di laporan

```
#include <iostream.h>
#include <conio.h>
void main (){
    int nilai, i;
    cout << "Masukkan angka bulat (0 - 100) ";
    cin>>nilai;
    if (nilai>=60){
        for (i=0; i<=5; i++){
            cout << "Selamat, tingkatkan lagi...!" <<endl;
        }
    }
}
```

```

    }
}
else {
    for (i=0; i<=5; i++){
        cout << "Silahkan belajar lagi" <<endl;
    }
}
getch();
}

```

### 3. Cobalah program dibawah, jalankan dan amati hasilnya

```

#include <iostream.h>
#include <conio.h>
void main (){
    char nama[10];
    int i, acak;
    cout << "Masukkan nama anda = ";
    gets (nama);
    if (strcmp (nama,"Ali")==0)
    {
        cout << " Tiga angka keberuntunganmu adalah =" << endl;
        randomize ();
        for(i=1;i<=3; i++){
            acak = random(20);
            cout<<acak<<endl;
        }
    }
    else
        cout <<" Siapun nama Anda, semoga beruntung!" <<
endl;
    getch();
}

```

### 4. Ketikkan program di bawah, jalankan dan amati hasilnya

```

#include<iostream.h>
void main()
{
    int pil;
    cout<<"pilihan = "<<endl;
    cout<<"1. deret fibonacci"<<endl;
    cout<<"2. deret faktorial "<<endl;
    cout<<"masukkan pilihan = ";
    cin>>pil;
    cout<<"======"<<endl;
    if(pil==1)
    {
        int a,x,y=0;
        int z=1;
        int n;
        cout<<"banyaknya deret =";
        cin>>n;
        for (a=1; a<=n; a++)
        {
            x=y+z;
            z=y;
            y=x;
            cout<<x<<" ";
        }
    }
}

```

```
else if(pil==2)
{
    int hasil=1,bil;
    cout<<"masukkan faktorial =";
    cin>>bil;
    int i;
    for (i = 1; i<=bil; i++)
    {
        hasil=i*hasil;
    }
    cout<<hasil;
}
else
cout<<"salah masukkan pilihan";
}
```

#### **D. LATIHAN**

- Latihan diberikan oleh dosen pengampu pada akhir praktikum.

#### **E. TUGAS**

- Tugas diberikan oleh dosen pengampu pada akhir praktikum