

# **PENGGUNAAN APLIKASI VISUAL C++ UNTUK PEMROGRAMAN KOMPUTER**



**PROGRAM STUDI PENDIDIKAN FISIKA  
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN  
UNIVERSITAS KRISTEN INDONESIA  
JAKARTA, Indonesia  
2019**

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, atas limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan buku penggunaan aplikasi Visual C++ untuk dasar pemrograman komputer.

Modul ini diperuntukkan bagi mahasiswa Universitas Kristen Indonesia khususnya prodi Pendidikan Fisika Fakultas Keguruan dan Ilmu Pendidikan yang mengikuti matakuliah ICT II. Modul ini difungsikan sebagai petunjuk dalam menyelesaikan persoalan-persoalan matematika dengan menggunakan metode numerik. Modul ini dihususkan untuk dasar dalam pemrograman computer menggunakan Visual C++.

Penulis menyadari sepenuhnya bahwa modul ini tentu memiliki banyak kekurangan. Untuk itu penulis menerima masukan dan kritikan konstruktif dari berbagai pihak demi kesempurnaan dimasa yang akan datang. Semoga modul ini dapat bermanfaat bagi pembaca.

Jakarta, Januari 2019

Penulis,

Faradiba

# DAFTAR ISI

<b>Chapter 1</b>	Pengantar Bahasa Visual C++, Tipe Data, Dan Variabel	1
<b>Chapter 2</b>	Konstanta, Operator, Dan Debug	18
<b>Chapter 3</b>	Instruksi Pemilihan	36
<b>Chapter 4</b>	Instruksi Perulangan	45
<b>Chapter 5</b>	Array Dan Tipe Data Bentukan	54



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



### 1. 1 PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL

#### Kemampuan yang diharapkan

1. Mahasiswa memahami dan mampu memecahkan masalah algoritma dan mengimplementasikan program di microsoft visual C++.
2. Mahasiswa mampu menggunakan macam-macam operator.
3. Mahasiswa memahami dan mampu menggunakan metode untuk menampilkan berbagai format ke layar.
4. Mahasiswa mampu memanfaatkan instruksi untuk memasukkan data dalam berbagai tipe.

#### Uraian Materi

##### A. KONSEP DASAR

###### A.1 Sejarah bahasa C/C++

C. a Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories

Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories awal tahun 1980-an berdasarkan C ANSI (*American National Standard Institute*). Pertama kali *prototype* C++ muncul sebagai C yang dipercanggih dengan fasilitas kelas. Bahasa tersebut disebut C dengan kelas (*C with class*). Selama tahun 1983-1984, bahasa C dengan kelas disempurnakan dengan menambahkan fasilitas pembebanlebih operator dan fungsi yang kemudian melahirkan apa yang disebut C++. Simbol ++ merupakan operator C untuk operasi penaikan, simbol itu muncul untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C.

###### A.2 Mengenal IDE Microsoft Visual C++

Visual C++ adalah sebuah produk *Integrated Development Environment* (IDE) untuk bahasa pemrograman C dan C++ yang dikembangkan *Microsoft*. Visual C++ merupakan salah satu bagian dari paket *Microsoft Visual Studio*. Bahasa ini merupakan bahasa pemrograman tingkat tinggi (kompleks). Visual C++ adalah bahasa pemrograman yang cukup populer. Hampir semua file DLL pada sistem operasi Windows dibuat menggunakan bahasa ini. Visual C++ dapat berjalan lebih cepat pada windows karena hanya memerlukan memori yang kecil. Dalam hal ini, Ms. Visual C++ lebih unggul jika dibandingkan dengan Ms. Visual Basic. Ini menjadi salah satu alasan Ms. Visual C++ lebih dipilih programmer



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



untuk membuat berbagai aplikasi dekstop maupun antivirus. IDE dari Visual C++ dapat digunakan untuk :

1. Menulis naskah program;
2. Mengompilasi program (*compile*);
3. Melakukan pengujian terhadap program (*Debugging*); dll.

### A.3 Alasan menggunakan Visual C++ :

#### a. C++ adalah bahasa pemrograman yang populer saat ini

Dengan banyaknya programmer bahasa C++, membawa pengaruh pada semakin banyaknya kompilasi yang dikembangkan untuk berbagai platform (berpengaruh ke portabilitas).

#### b. C++ adalah bahasa pemrograman yang memiliki portabilitas tinggi

Program C++ yang kita tulis untuk satu jenis platform, bisa kita kompilasi dan jalankan di platform lain dengan tanpa ataupun hanya sedikit perubahan. Ini bisa diwujudkan dengan adanya standarisasi ANSI untuk C++.

#### c. C++ adalah bahasa pemrograman dengan kata kunci (keyword) sedikit

Kata kunci disini adalah merupakan fungsi ataupun kata dasar yang disediakan oleh kompilasi suatu bahasa pemrograman. Hal ini membawa pengaruh semakin mudahnya kita menulis program dengan C++. Pengaruh lain dari sedikitnya kata kunci ini adalah proses eksekusi program C++ yang sangat cepat.

#### d. C++ adalah bahasa pemrograman yang fleksibel

Dengan menguasai bahasa C++, kita bisa menulis dan mengembangkan berbagai jenis program mulai dari *operating system*, *word processor*, *graphic processor*, *spreadsheets*, ataupun kompilasi untuk suatu bahasa pemrograman.

#### e. C++ adalah bahasa pemrograman yang bersifat moduler

Program C++ ditulis dalam routine yang biasa dipanggil dengan fungsi. Fungsi-fungsi yang telah kita buat, bisa kita gunakan kembali (*reuse*) dalam program ataupun aplikasi lain.

Alasan paling penting memilih bahasa C++ adalah bahwa dengan bahasa ini, programmer dapat membuat sebuah aplikasi yang mempunyai kemampuan yang tinggi. Dengan C++,



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



kemampuan programmer tidak dibatasi. Programmer dapat membuat sebuah aplikasi berskala besar tanpa ada batasan.

### A.4 Menggunakan Microsoft Visual C++

Untuk memulai membuat project baru pada Visual C++ 2010, pilih **File -> New -> Project** atau **Ctrl + Shift + N**. Untuk melakukan kompilasi kode program, pilih menu **Build -> Compile** atau **Ctrl + F7**. Setelah itu untuk menjalankan program cukup tekan tombol **F5** pada keyboard.

Menu	Item	Shortcut	Penjelasan
Build	Build	Ctrl + F7	Meng-compile program
	Run+ Debugging	F5	Menjalankan program sekaligus melakukan Debugging
	Run	Ctrl + F5	Menjalankan program tanpa melakukan debungging
	Rebuild All		Membangun kembali dengan meng-compile program
	Clean		Proses pembersihan terhadap kemungkinan adanya bug
	Program reset	Enter	Menutup program aktif, melepas memori, dan kembali ke IDE Visual C++
Project dan File	New source file	Ctrl + N	Membuat source file baru
	New project	Ctrl + Shift + N	Membuat project baru
	Open file	Ctrl + O	Membuka file
	Open project	Ctrl + Shift + O	Membuka project
	Save	Ctrl + S	Menyimpan file/project dengan nama yang sama
	Save as		Menyimpan file/project dengan nama yang berbeda
	Save all	Ctrl + Shift + S	Menyimpan semua file/ project yang terbuka pada jendela IDE VS 2010
	Close		Menutup file/project yang sedang terbuka di jendela IDE VS 2010
	Exit	Alt + F4	Keluar dari program VS 2010

### A.5 Struktur Program Pada Visual C++ :

Program C maupun C++ secara umum tersusun dari beberapa bagian utama, yaitu :

#### a. Komentar

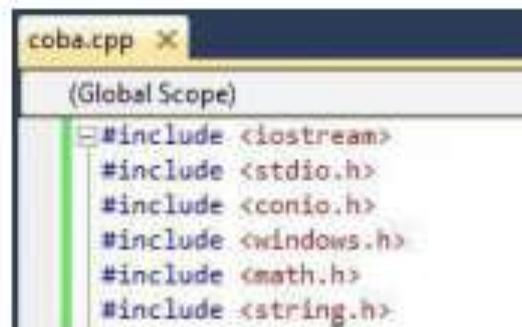
Komentar digunakan untuk memberi informasi/dokumentasi tentang program atau code yang ada. Dalam C atau C++ setiap tulisan yang diapit oleh simbol `/* ... */` untuk multi

baris atau setiap baris yang dimulai dengan simbol // dianggap komentar dan tidak akan dikompilasi oleh compiler. Contoh komentar :

<pre>/* ini adalah contoh komentar multibaris */</pre>	← Komentar multibaris
<pre>// ini adalah contoh // komentar 1 baris</pre>	← Komentar 1 baris

### b. Preprocessor Directive

*Preprocessor directive* disebut juga pengarah compiler karena fungsinya untuk mengatur proses kompilasi.



```
coba.cpp x
(Global Scope)
#include <iostream>
#include <stdio.h>
#include <conio.h>
#include <windows.h>
#include <math.h>
#include <string.h>
```

#include merupakan satu jenis pengarah preprocessor yang digunakan untuk membaca file yang dinamakan file judul (*header file*). `iostream`, `conio.h`, `stdio.h`, `windows.h`, `math.h`, dan `string.h` merupakan *file header* yang merupakan standard library dari C++. Setiap file header berhubungan dengan perintah masukan (*input*), perintah keluaran (*output*), dan fungsi-fungsi yang digunakan dalam suatu program.

File Header	Input	Output	Fungsi
iostream	cin	cout	iostream
conio.h	getchar getch getche		clrscr()
stdio.h	scanf gets	printf puts putchar	
windows.h			system()
math.h			sqrt() pow() log10() sin()
string.h			strcpy() strlen() strcat()



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



### c. Fungsi utama dan fungsi tambahan

Fungsi utama (main) harus ada dalam setiap program karena fungsi utama merupakan fungsi yang akan dieksekusi pertama kali. Lebih lanjut tentang fungsi akan dipelajari pada bab fungsi dipertemuan selanjutnya.

### d. Bagian definisi fungsi

Diawali dengan tanda “{” (kurawal buka) sebagai tanda awal fungsi dan tanda “}” (kurawal tutup) sebagai tandaberakhirnya suatu fungsi, baik fungsi utama maupun fungsi tambahan. Definisi fungsi berisi sekumpulan code yang nanti akan dieksekusi bila fungsi tersebut dipanggil. Aturan umum penulisan bahasa C++ adalah sebagai berikut.

- ✓ *case-sensitive*, yaitu bahasa C++ membedakan penulisan huruf besar dan huruf kecil, contoh  $A \neq a$ ;
- ✓ setiap *statement* diakhiri dengan tanda “;” (titik koma);
- ✓ tidak boleh ada variabel ganda, maupun konflik dengan reserved *keywords* (kata-kata bawaan dari IDE), contoh : `and`, `and_eq`, `asm`, `auto`, `bitand`, `bitor`, `bool`, `break`, `case`, `catch`, `char`, `class`, `compl`, `const`, `const_cast`, `continue`, `default`, `delete`, `do`, `double`, `dynamic_cast`, `else`, `enum`, `explicit`, `export`, `extern`, `false`, `float`, `for`, `friend`, `goto`, `if`, `inline`, `int`, `long`, `mutable`, `namespace`, `new`, `not`, `not_eq`, `operator`, `or`, `or_eq`, `private`, `protected`, `public`, `register`, `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`, `while`, `xor`, `xor_eq`;
- ✓ pada *subroutine* atau fungsi harus diapit oleh kurung kurawal (`{.....}`); dan
- ✓ setiap variabel yang digunakan wajib dideklarasikan terlebih dahulu.

### e. Bagian Deklarasi

Bagian yang akan mendeklarasikan variabel, konstanta, fungsi, dan lain-lain. Lebih lanjut tentang materi ini akan dibahas pada bab ini dan bab-bab selanjutnya.

### A.6 Tipe Data

Tipe data adalah suatu jenis nilai yang dapat dinyatakan dalam bentuk konstanta atau variabel dan operator yang dapat digunakan untuk mendefinisikan objek data yang akan dimanipulasi dalam sebuah program. Jenis-jenis tipe data antara lain:

Jenis Tipe Data	Penjelasan	Contoh
Tipe data dasar (primitive data type)	Tipe data bawaan dari bahasa pemrograman	<code>int</code> , <code>char</code> , <code>float</code> , <code>boolean</code>
Tipe data bentukan (defined data type)	Tipe data bentukan user	<code>struct</code> , <code>enum</code>

Beberapa jenis tipe data dasar adalah :

#### a. Tipe Data Bilangan/Angka

Tipe data bilangan/angka dapat berupa `int` (untuk bilangan bulat/integer) atau `float` dan `double` (untuk bilangan desimal/real). Tiap tipe data tersebut memiliki jangkauan nilai yang berbeda-beda. Tipe data `int`, `float`, dan `double` memiliki tambahan yang dapat mengubah batas atas maupun batas bawah dari rentang konstanta yang dapat ditampung oleh tipe tersebut. Tambahan yang dapat dipergunakan adalah:

- ✓ *signed* dan *unsigned*
- ✓ *short* dan *long*

Tipe Data	Ukuran (bit)	Jangkauan Nilai	Keterangan
<code>int</code>	16 bit	-32768 s/d 32768	Bilangan bulat
<code>short int</code> atau <code>short</code>	16 bit	-32768 s/d 32768	Bilangan bulat
<code>long int</code> atau <code>long</code>	32 bit	-2147483648 s/d 2147483648	Bilangan bulat dengan rentang yang lebih luas
<code>unsigned int</code>	16 bit	0 s/d 65535	Bilangan bulat positif
<code>unsigned short</code>	16 bit	0 s/d 65535	Bilangan bulat positif
<code>unsigned long</code>	32 bit	0 s/d 4294967295	Bilangan bulat positif dgn rentang yang lebih luas
<code>float</code>	32 bit	$3.4 \times 10^{-38}$ s/d $3.4 \times 10^{38}$	Bilangan real
<code>double</code>	64 bit	$1.7 \times 10^{-308}$ s/d $1.7 \times 10^{308}$	Bilangan real dengan rentang yang lebih luas
<code>long double</code>	80 bits	$3.4 \times 10^{-4932}$ s/d $3.4 \times 10^{4932}$	Bilangan real dengan rentang terluas

**CATATAN!** Rentang nilai ini berbeda antara satu bahasa pemrograman dengan bahasa pemrograman yang lainnya dan rentang nilai ini juga dipengaruhi oleh hardware yang digunakan.

### b. Tipe Data Teks

Pada tipe data teks menggunakan keyword “char”.Teks dapat berupa satu karakter (char) ataupun kumpulan karakter (string). Karakter bisa berupa huruf alfabet, angka, tanda baca, operator aritmatik (+, -, /, \*) dan karakter khusus (@, &, \$, #, dll).

Dalam penulisannya, untuk teks yang berupa 1 karakter caranya yaitu karakter ditulis di antara dua petik tunggal („..“). Sedangkan penulisan teks dengan karakter lebih dari satu ditulis di antara dua petik ganda (“..”). Pendeklarasiannya pun diikutsertakan panjang dari nilai string tersebut.

**CATATAN!** Perlu diingat, pada tipe data char, data tidak dapat diolah secara matematis.

### c. Tipe Data Boolean

Tipe data boolean adalah tipe data hanya mempunyai 2 kondisi benar (“true”) atau salah (“false”). Untuk kondisi benar bernilai 1 dan kondisi salah bernilai 0.

## A.7 Variabel

Variabel menyatakan objek data yang nilainya disimpan dan dapat berubah-ubah nilainya selama eksekusi berlangsung. Syarat-syarat penulisan nama variabel adalah :

- nama variabel **tidak boleh sama** dengan *reserved keywords*, *function* (fungsi), dan harus unik;
- maksimum 32 karakter**, jika lebih maka karakter selebihnya tidak akan diperhatikan oleh komputer;
- nama variabel bersifat *case-sensitive*;
- nama variabel tersusun dari **huruf, angka**, dan “\_” (*underscore*);
- nama variabel harus diawali **huruf alfabet** atau “\_” (*underscore*), tidak boleh diawali dengan angka atau karakter khusus lainnya; dan
- nama variabel tidak boleh mengandung spasi, jika ada pemisahan karakter
- maka spasi pemisah dapat diganti dengan karakter “\_” (*underscore*)

### A.8 Pendeklarasian Variabel

Bentuk umum pendeklarasian variabel adalah :

```
tipe_data<spasi>nama_variabel;
```

Contoh pendeklarasian variabel:

```
int umur;
```

tipe data ← int → nama variabel

Beberapa contoh pendeklarasian variabel berdasarkan tipe datanya :

#### a. Tipe Data Bilangan/Angka

Contoh :

```
int umur = 28;
```

```
float ipk = 3.47;
```

#### b. Tipe Data Teks

Contoh teks 1 karakter (char):

```
char nilai = 'A';
```

```
char Golongandarah = 'B';
```

Contoh teks yang lebih dari 1 karakter (string):

```
char nama [10] = "shinta";
```

nama sebagai variabel yang berisi **string** dengan panjang maksimal 10 karakter

#### c. Tipe Data Boolean

Contoh :

```
bool kondisi = true;
```

```
bool keadaan = false;
```

Berikut merupakan beberapa contoh pendeklarasian variabel yang benar dan yang salah:

#### a. Pendeklarasian yang BENAR

- ✓ Pendeklarasian variabel umur langsung dengan pemberian nilai (inisialisasi), misal :

```
int umur = 17;
```

- ✓ Pendeklarasian variabel berat badan dengan menggunakan tanda “\_” (underscore) sebagai pemisah antara kata “berat” dan “badan”.

```
int berat_badan;
```

- ✓ Pendeklarasian variabel golongan darah dengan menggunakan huruf kapital.

```
char Golongandarah = 'O';
```

- ✓ Pendeklarasian variabel nilai tugas 1 dengan mengkombinasikan huruf kapital, huruf kecil, “\_” (underscore), dan angka

```
int nilai_TUGAS_1 = 100;
```

- ✓ Pendeklarasian variabel 1 kondisi dengan menggunakan “\_” (underscore) untuk mengawali penamaan variabel karena nama variabel tersebut diawali dengan angka 1.

```
bool _1kondisi = true;
```

### b. Pendeklarasian variabel yang SALAH

```
int umur = 0.17;
```

- ✓ Pendeklarasian variabel umur tersebut salah karena nilai yang dimasukkan ke dalam variabel tidak sesuai dengan tipe data yang digunakan. Seharusnya jika ingin membuat variabel yang dapat menampung nilai berupa angka desimal menggunakan tipe data float atau double. Jadi, pemberian nilai pada suatu variabel harus sesuai dengan tipe datanya.

```
int berat badan;
```

- ✓ Pendeklarasian variabel berat badan tersebut salah karena menggunakan spasi antara kata “berat” dan “badan”

### A.9 Format Data

Format data adalah suatu kombinasi tanda-tanda khusus berupa karakter yang dipakai untuk mengatur bentuk data tampilan. Format data bisa lebih dari satu sesuai dengan banyak data yang ingin ditampilkan. Format data yang dipakai disesuaikan dengan jenis data tampilan.

### Daftar Spesifikasi Format Data

Format Data	Bentuk Tampilan Data
%c	Karakter tunggal
%d	Bilangan bulat
%e	Bilangan bernotasi ilmiah
%f	Bilangan berjenis pecahan bertitik (floating point number)
%g	Pilihan tampilan terpendek antara bentuk %e atau %f
%i	Bilangan bulat desimal (sama dengan %d)
%n	Digunakan untuk perintah menyimpan banyak karakter yang telah ditampilkan oleh perintah print() kedalam suatu petunjuk jenis integer
%o	Notasi bilangan basis delapan (oktal)
%p	Tampilan petunjuk (pointer)
%s	Tampilan deret karakter (string)
%u	Bilangan bulat decimal tak bertanda (unsign)
%x	Notasi bilangan basis 16 (hexadesimal)
%%	Tampilan tanda persen (%)

### HALAMAN UTAMA MICROSOFT VISUAL C++

#### a. Tampilan Halaman Utama Visual C++ 2010

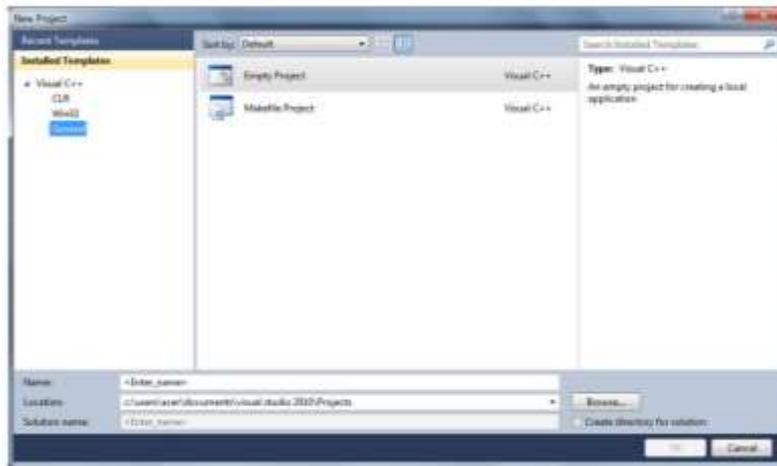


**Gambar .** Tampilan halaman utama visual C++

Gambar diatas adalah tampilan dari Microsoft Visual C++ 2010 Express. Untuk membuat sebuah program terlebih dahulu kita harus membuat sebuah project seperti yang dijelaskan pada halaman sebelumnya.

#### Langkah MembuatProject :

1. *File* → *New* → *Project*



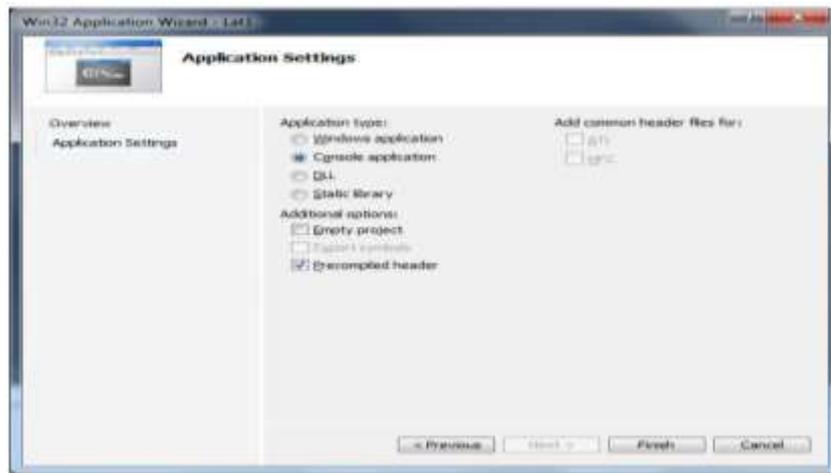
**Gambar.** Tampilan halaman membuat project baru di visual C++

1. Pilih *Win32 Console Application*.
2. Pada field **Name** : isikan sebuah nama, untuk kali ini kita beri nama Lat1.cpp.
3. Pada field **Location**: adalah tempat atau lokasi dimana kita akan menyimpan Project tersebut. Arah kan lokasi penyimpanan ke Folder tertentu di Drive **D:\**, dengan cara klik tombol *Browse...* .lalu klik tombol *OK*.

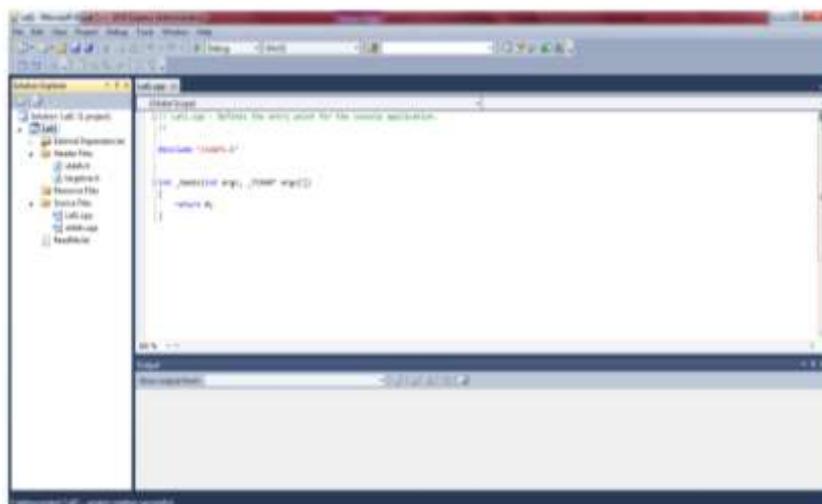


**Gambar .** Tampil halaman win 32 wizard

4. Setelah muncul jendela seperti gambar diatas. Lalu kita klik *Next*. Setelah itu klik *Finish*.



**Gambar .** Tampil halaman application settings



**Gambar .** Tampilan halaman project baru di C++

5. Apabila muncul seperti gambar di atas, artinya anda telah berhasil membuat sebuah Project pada Microsoft Visual C++.

### Langkah Praktikum Awal

a. Program menampilkan kalimat

1. Tuliskan kode program seperti padagambar berikut

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
```

```
    cout<<" Hello Word, Saya ingin menjadi programmer C++ ";  
    return 0;  
}
```

2. Jalankan program dengan cara menekan tombol *Start Debugging* (F5) yang terletak pada area toolbar *Microsoft Visual C++ 2010*.
3. Perhatikan hasilnya .....
4. Setelah itu tambah kan perintah berikut pada kode program yang telah anda buat sebelumnya.

```
#include "stdafx.h"  
#include <iostream>  
using namespace std;  
int _tmain (int argc, _TCHAR* argv[])  
{  
    cout<<" Hello Word, Saya ingin menjadi programmer C++ ";  
    system ("PAUSE");  
    return 0;  
}
```

5. Jalankan kembali program dengan menekan tombol *Start Debugging* atau dengan cara, menekan tombol F5 pada *keyboard* anda.

### b. Program menampilkan kalimat dengan menambahkan `Getch ()`

1. Buatlah sebuah project baru dengan nama `Lat2.cpp`.
2. Ketikkan perintah berikut

```
#include "stdafx.h"  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
int _tmain (int argc, _TCHAR* argv[])  
{  
    cout <<" Hello Word, Saya ingin menjadi programmer C++  
    "<<endl;  
    cout <<" Ini adalah latihan ke-2 tentang penggunaan  
    getch() "<<endl;  
    cout <<" untuk menggantikan system PAUSE  
    "<<endl<<endl;  
    cout <<" Tekan Tombol ENTER untuk mengahiri program ";  
    getch ();  
  
    return 0;  
}
```



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



3. Perhatikan hasilnya kemudian bandingkan dengan Lat1cpp yang telah anda buat sebelumnya.
4. Output yang dihasilkan:

```
Hello Word, Saya ingin menjadi programmer C++
Ini adalah latihan ke-2 tentang penggunaan getch()
untuk menggantikan system PAUSE
Tekan Tombol ENTER untuk mengahiri program
```

### B. Penugasan Kelas

1. Buat program seperti di bawah ini! Simpan dengan nama LatihanPratikum1.cpp!

Pseudocode :

**Program** LatihanPratikum1

*{Tanpa masukan, hanya menampilkan angka, kata, dan kalimat yang dimasukkan ke dalam variabel}*

**Kamus Lokal :**

Nama : string  
Umur : integer  
Tinggi\_badan : double  
Ipk : float  
Gol\_darah : char  
Jurusan : string

**Algoritma :**

```
nama ← "Desi Ratnasari"
umur ← 20
tinggi_badan ← 153.5
ipk ← 3.82
gol_darah ← 'O'
jurusan ← "Pendidikan Informatika"
output (" Nama : ", nama)
output (" Umur : ", umur)
```

**output** (“ Tinggi : ”, tinggi\_badan)

**output** (“ Goldar : ”, gol\_darah)

**output** (“ Jur : ”, jurusan)

**output** (“ IPK : ”, ipk)

**Source Code Program :**

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

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    //pendeklarasian variabel dan inisialisasi
    char nama[30] = "Desi Ratnasari";
    int umur = 20;
    double tinggi_badan = 153.5;
    float ipk = 3.82;
    char gol_darah = '0';
    char jurusan[30] = "Pendidikan Informatika";

    //menampilkan ke layar
    printf("Nama \t : "); puts(nama);
    printf("Umur \t : %d tahun\n", umur);
    printf("Tinggi \t : %g cm\n", tinggi_badan);
    cout<<"Goldar \t : "; putchar(gol_darah);
    cout<<endl;
    cout<<"Jur \t : "<<jurusan<<endl;
    cout<<"IPK \t : "<<ipk<<endl;

    system("pause");
    return 0;
}
```

Output yang dihasilkan :

```
Nama      : Desi Ratnasari
Umur      : 20 tahun
Tinggi    : 153.5 cm
Goldar    : 0
Jur       : Pendidikan Informatika
IPK       : 3.82
Press any key to continue . . .
```

2. Buat program seperti di bawah ini! Simpan dengan nama **LatihanPratikum2.cpp!**

Pseudocode :

**Program** LatihanPratikum2

*{Menampilkan hasil atau nilai dari suatu variabel bertipe data BOOLEAN}*

### Kamus Lokal :

kondisi : **boolean**

### Algoritma :

kondisi ← true **output** (kondisi)

kondisi ← false **output** (kondisi)

### Source Code Program :

```
#include "stdafx.h"
#include <iostream>
#include <Windows.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    //mendeklarasikan variabel menampilkan jika
    kondisibernilai BENAR
    bool kondisi;
    kondisi = true;
    cout<<"Kondisi BENAR bernilai : ";
    cout<<kondisi <<endl;

    //menampilkan jika kondisi bernilai SALAH
    kondisi = false;
    cout<<"Kondisi SALAH bernilai : ";
    cout<<kondisi <<endl;

    system("pause");
    return 0;
}
```

Output yang dihasilkan :

```
Kondisi BENAR bernilai : 1
Kondisi SALAH bernilai : 0
Press any key to continue . . .
```

### C. Rangkuman

Visual C++ dapat berjalan lebih cepat pada windows karena hanya memerlukan memori yang kecil. Dalam hal ini, Ms. Visual C++ lebih unggul jika dibandingkan dengan Ms.



# CHAPTER 1

## PENGANTAR BAHASA VISUAL C++, TIPE DATA, DAN VARIABEL



Visual Basic. Ini menjadi salah satu alasan Ms. Visual C++ lebih dipilih programmer untuk membuat berbagai aplikasi dekstop maupun antivirus. IDE dari Visual C++ dapat digunakan untuk :

1. Menulis naskah program;
2. Mengompilasi program (compile);
3. Melakukan pengujian terhadap program (Debugging); dll.

### D. Evaluasi Formatif

1. Buatlah algoritma dan program dengan visual C++ untuk menampilkan biodata diri anda secara lengkap, dengan variabel yang ditampilkan adalah nim, nama, program studi, alamat, telepon, kota, asal sekolah, hobby, agama, tempat tgl/lahir, status keluarga.

Tampilan Outputnya :

```
BIODATA PRIBADI
=====
Nim          :
Nama         :
Program Studi :
Alamat      :
Telepon     :
Kota        :
Asal Sekolah :
Hobby       :
Agama      :
Tempat Tgl/Lahir :
Status Keluarga :
=====
```

**Catatan :** Tugas dibuat dalam bentuk hardcopy (disusun rapi), buat algoritmanya, cantumkan source code programnya, printscrint outputnya.

## 2.1 KONSTANTA, OPERATOR, DAN DEBUG

### Kemampuan yang diharapkan

1. Mahasiswa mengetahui konstanta dan operator
2. Mahasiswa mampu mendeklarasikan konstanta dan operator
3. Mahasiswa mengerti cara melakukan Debugging

### Uraian Materi

#### A. KONSEP DASAR

##### A.1 Konstanta

Konstanta adalah nilai numeris ataupun karakter yang tetap/tidak berubah selama program berlangsung. Penggunaan konstanta pada bahasa C++ ada 2 cara yaitu :

```
#define phi 3.14
#define nama "helmi"
```

Atau

```
const float pi= 3.14;
const char nama = "helmi";
```

Khusus untuk konstanta karakter selalu ditulis dengan diawali dan diakhiri dengan tanda petik tunggal (,). Beberapa konstanta karakter dapat ditulis dengan diawali tanda \ (penempatan tanda \ setelah tanda petik tunggal). Karakter yang berawalan tanda \ disebut rangkaian karakter escape (escape sequence).

Sequence	Karakter	Keterangan
\a	Bel	Bunyi bel
\b	BS	Mundur 1 karakter (backspace)
\f	FF	Ganti halaman (form feed)
\n	LF	Ganti baris (line feed)
\r	CR	Kembali ke awal kolom (carriage return)
\t	HT	Tabulasi horizontal (horizontal tab)
\v	VT	Tabulasi vertical (vertikal tab)
\	\	Backslash
\'	'	Petik tunggal
\"	"	Petik ganda
\?	?	Tanda tanya
\DDD	Apapun	DDD adalah digit nilai octal
\xHHH	Apapun	HHH adalah digit nilai hexadecimal

\0	null	Karakter ASCII=0
----	------	------------------

### A.2 Operator

Operator merupakan simbol yang digunakan dalam suatu program untuk melakukan suatu operasi atau manipulasi, misalnya untuk :

- ✓ menjumlahkan dua nilai;
- ✓ memberikan nilai ke suatu variabel (assignment); dan
- ✓ membandingkan kesamaan dua nilai.

#### a. Operator Pengerjaan

Operator pengerjaan berupa tanda sama dengan (=) yang digunakan untuk mengisi nilai yang berada di sebelah kanannya variabel yang ditunjuk disebelah kirinya.

**Contoh:**

```

a = 7
a = b = c = 7
```

maka nilai a dan b akan sama dengan nilai c yaitu 7.

#### b. Operator Aritmatika

Operator ini digunakan untuk perhitungan dasar aritmatika. Operator aritmatika ini digolongkan menjadi dua yaitu operator unary dan operator binary. Operator unary yaitu operator yang dikenakan terhadap satu buah nilai (operand).

**Contoh :**

```

c = -b
c = -a
```

**Tabel operator aritmatika unary**

Operator	Keterangan
+	Tanda plus
-	Tanda minus
++	Prefix :preincrement; postfix: postincrement
--	Prefix :predecrement; postfix: postdecrement

#### c. Operator peningkatan dan penurunan

Bahasa C++ mempunyai operator unik, yaitu operator peningkatan/increment (++) dan operator penurunan/ decrement (--). Operator peningkatan (++) digunakan untuk menambahkan nilai 1 pada nilai sebelumnya dan operator penurunan (--) digunakan untuk mengurangi nilai 1 pada nilai sebelumnya. Operator binary yaitu operator yang dikenakan terhadap dua buah nilai (operand).

**Contoh :**

```

c = a + b
c = a * b
```

**Tabel operator aritmatika binary**

Operator	Keterangan
*	Perkalian
/	Pembagian
%	Modulus atau sisa pembagian
+	Penjumlahan
-	Pengurangan

#### d. Operator Hubungan

Operator hubungan digunakan untuk membandingkan dua elemen nilai dan akan dihasilkan nilai perbandingan, yaitu betul (bernilai 1) atau salah (bernilai 0). Operator ini banyak diaplikasikan pada penyelesaian suatu kondisi di statementt IF.

Operator	Keterangan
>	Lebih besar dari
>=	Lebih besar dari sama dengan
<	Lebih kecil dari
<=	Lebih kecil dari sama dengan
==	Sama dengan
!=	Tidak sama dengan

**Contoh :**

```
#include "stdafx.h"
#include <iostream>
#include <Windows.h>

using namespace std;

int main ()
{
    int nilai =90;
    kondisi = true;
    cout<<(nilai>80)<<endl;
    cout<<(nilai<80)<<endl;

    system("pause");
    return 0;
}
```

#### e. Operator Logika

Operator logika bisa digunakan untuk menghubungkan dua buah ungkapan kondisi menjadi sebuah ungkapan kondisi.

**Tabel Operator Logika**

Operator	Keterangan
&&	AND (dan)
	OR (atau)
!	NOT (bukan)

Bentuk umum pemakaian operator logika && dan || :

```
<ungkapan1>&&<ungkapan2>;
<ungkapan1> || <ungkapan2>;
```

Bentuk umum pemakaian operator logika ! :

```
!<ungkapan>;
```

#### Contoh :

```
#include "stdafx.h"
#include <iostream>
#include <Windows.h>

using namespace std;

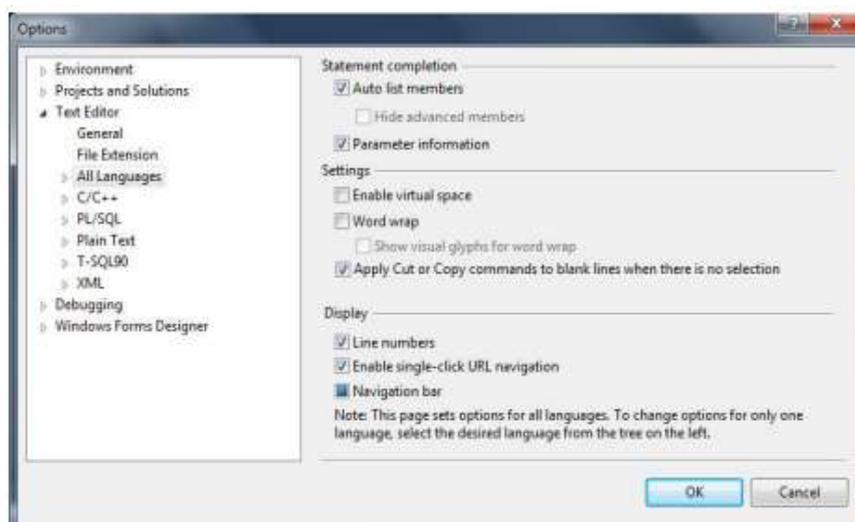
int main ()
{
    int nilai =90;
    cout<<((nilai>80)&& (nilai<100))<<endl;
    cout<<((nilai<80)&& (nilai<100))<<endl;

    system("pause");
    return 0;
}
```

#### A.3 Debug

Debugging adalah proses menemukan atau melacak kesalahan pada kode sebuah program. Versi debug ++ digunakan untuk memudahkan programmer dalam melakukan proses Debugging atau melacak kesalahan pada kode program. Pada versi debug ini akan memberikan informasi-informasi yang berguna saat akan dilakukan Debugging. Versi debug juga bisa disebut sebagai debugger pada Microsoft Visual C++.

Sebelum memulai proses Debugging program menggunakan Visual C++ ada baiknya apabila fitur line numbers diaktifkan terlebih dahulu. Hal ini untuk mempermudah programmer dalam mencari baris kode yang bermasalah apabila program gagal dieksekusi.

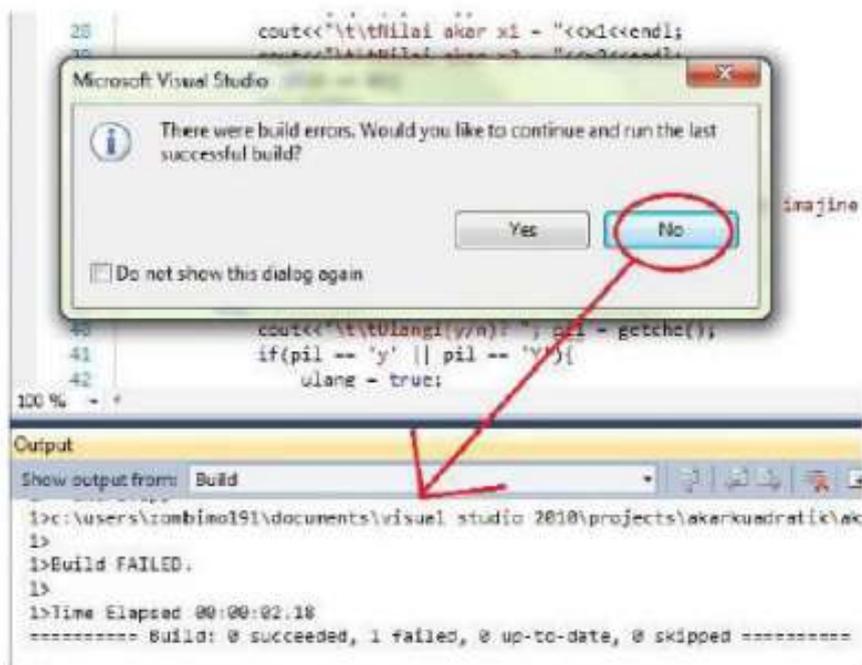


Pada kotak sebelah kiri expand bagian Text Editor kemudian pilih All Languages | General dan pada bagian Display disebelah kanan centang fitur Line numbers. Untuk memulai proses Debugging sama dengan cara menjalankan program yaitu cukup dengan menekan tombol F5 pada keyboard. Berikut adalah contoh proses Debugging dengan metode try and error dan hasil yang tidak sesuai.

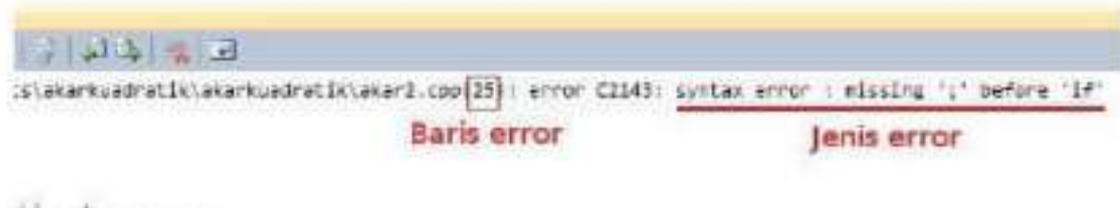
#### a. try and error



Apabila muncul pesan seperti di atas, maka dapat dipastikan terdapat kesalahan pada kode program yang ditulis. Untuk mengatasinya klik No kemudian pindah kebagian Output yang berada di bawah code editor seperti berikut:



Cari teks yang memberikan informasi tentang error yang dimaksud seperti tampak pada contoh di bawah ini :



Angka (25) menunjukkan baris yang error dan **sintaks error : missing „;“ before „if“** menunjukkan jenis error-nya. Selanjutnya klik pada bagian teks informasi error dan klik icon  untuk berpindah ke bagian yang error pada *code editor*.



te, 0 skipped =====

## B. Penugasan Kelas

### Langkah Pratikum

1. Buatlah program untuk menampilkan hasil penjumlahan operator aritmatika menggunakan increment dan decrement, simpan dengan nama aritmatika.cpp!, serta lakukanlah pengujian debugging menggunakan metode hasil yang tidak sesuai.

**Berikut bentuk dari pseudocode-nya :**

#### **Program** aritmatika

*{Program untuk menampilkan hasil penjumlahan operator aritmatika dengan menggunakan increment dan decrement}*

#### **Kamus Lokal :**

a : integer

#### **Algoritma :**

a ← 5

**output** (“ Nilai awal a : ”, a)

**output** (“ Nilai a ketika melakukan a++ : ”, a++)

**output** (“Nilai a ketika melakukan ++a : ”, ++a)

**output** (“Nilai a ketika melakukan a-- : ”, a--)

**output** (“Nilai a ketika melakukan --a : ”, --a)

**output** (“ IPK : ”, ipk)

#### Source Code Program :

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

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    int a = 5;
    cout<<"Nilai awal a: "<<a<<endl;
    cout<<"Nilai awal a ketika melakukan a++: ";
    cout<<a++<<endl;cout<<"Nilai awal a ketika melakukan
a++: ";
    cout<<++a<<endl;cout<<"Nilai awal a ketika melakukan
++a: ";
    cout<<a--<<endl;cout<<"Nilai awal a ketika melakukan a--
: ";
    cout<<--a<<endl;cout<<"Nilai awal a ketika melakukan --
a: ";

    system("pause");
    return 0;
}
```

#### Output yang dihasilkan :

```
Nilai awal a : 5
Nilai a ketika melakukan a++ : 5
Nilai a ketika melakukan ++a : 7
Nilai a ketika melakukan a-- : 7
Nilai a ketika melakukan --a : 5
Press any key to continue . . .
```

#### 2. Buatlah sebuah project baru dengan nama **Penjumlahan.cpp**.

- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    string Kall;
    int Bill;
    int Bil2;

    Kall="Tulis Nama Anda";
    Bill=2;
    Bil2=4;

    cout <<" Selamat Datang " <<Kall <<endl<<endl;
    cout <<" Bilangan Pertama adalah " <<Bill <<endl;
    cout <<" Bilangan Kedua adalah " <<Bil2<<endl<<endl;
    cout <<" Penjumlahan kedua bilangan tersebut adalah
    " <<Bill+Bil2<<endl;

    system("pause");
    return 0;
}
```

#### Output yang dihasilkan :

```
Selamat Datang  Tulis Nama Anda

Bilangan Pertama adalah 2
Bilangan Kedua adalah 4

Penjumlahan kedua bilangan tersebut adalah 6
Press any key to continue . . .
```

### 3. Buatlah project dengan nama **LatihanString.cpp**

Ketikkan Kode Program seperti berikut :

```
#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    string sNama;
    int Bill, Bil2;

    cout << " Nama Anda : ";
    cin >> sNama;
    cout << endl << " Selamat Datang " << sNama << endl << endl;
    cout << " Silahkan menggunakan aplikasi ini
    " << endl << endl;

    system("pause");
    return 0;
}
```

#### Output yang dihasilkan :

```
Nama  Anda : Desi Ratnasari

Selamat Datang Desi

Silahkan menggunakan Aplikasi ini

Press any key to continue . . .
```

#### Penjelasan Kode Program :

- `#include <string>`: menambahkan library string, agar kita dapat menggunakan tipe data string pada badan program.
- `string sNama;` :mendeklarasikan variable sNama dengan tipe data string. sNama yang nantinya akan digunakan untuk menampung data yang berupa string.
- `int Bill, Bil2` :mendeklarasikan variable Bill dan Bil2 dengan tipe data Integer. Bill dan Bil2 nanti adalah variable yang nantinya akan digunakan untuk menampung data yang berupa bilangan bulat.

- d. `cout >> " Nama Anda : "` : digunakan untuk menampilkan tulisan " Nama Anda : "(tanpa tanda petik) pada layar hasil compile.
- e. `cin << sNama;` : digunakan untuk menyimpan input dari Keyboard user ke dalam variable `sNama`. Nantinya variable `sNama` akan berisi data berupa string yang telah dimasukkan oleh user melalui keyboard.
- f. `cout << endl << "Selamat Datang " << sNama << endl << endl;` : digunakan untuk menampilkan bacaan "Selamat Datang " (tanpa tanda petik) yang diikuti dengan isi dari variable `sNama` pada layar hasil kompilasi.
- g. Untuk perintah yang lainnya cukup jelas, sesuai dengan pratikum sebelumnya.

**- Tambahkan Kode Program Seperti di bawah ini.**

```
#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    string sNama;
    int Bil1,Bil2;

    cout <<" Nama Anda  : ";
    cin >> sNama;
    cout <<endl<<" Selamat Datang "<<sNama<<endl<<endl;
    cout <<" Silahkan menggunakan aplikasi ini
    "<<endl<<endl;

    cout <<"Masukkan bilangan bulat 1 : ";
    cin >> Bil1;
    cout <<"Masukkan bilangan bulat 2 : ";
    cin >> Bil2;

    cout <<endl<<endl;

    cout <<Bil1 <<" + " <<Bil2 <<" = " <<Bil1+Bil2
    <<endl<<endl;

    system("pause");
    return 0;
}
```

#### Output yang dihasilkan:

```
Nama Anda : Desi Ratnasari
Selamat Datang Desi
Silahkan menggunakan Aplikasi ini
Masukkan bilangan bulat 1 : 7
Masukkan bilangan bulat 2 : 6

7 + 6 = 13
Press any key to continue . . .
```

#### Penjelasan Kode Program :

- `cin >> Bil1;` : digunakan untuk menyimpan input dari Keyboard user ke dalam variable `Bil1`. Nantinya variable `Bil1` akan berisi data berupa bilangan bulat yang telah dimasukkan oleh user melalui keyboard.
- `cin >> Bil2;` : digunakan untuk menyimpan input dari Keyboard user ke dalam variable `Bil2`. Nantinya variable `Bil2` akan berisi data berupa bilangan bulat yang telah dimasukkan oleh user melalui keyboard.
- `cout << Bil1 << "+" << Bil2 <<"= " << Bil1+Bil2 << endl << endl;` : digunakan untuk menampilkan isi dari variable `Bil1`, karakter "+", variable `Bil2`, karakter "=" dan Hasil penjumlahan dari `Bil1` dan `Bil2`.

#### - Tambahkan kode program berikut pada project anda

```
#include "stdafx.h"
#include <iostream>
#include <string>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    string sNama;
    int Bil1, Bil2;

    cout << " Nama Anda : ";
    cin >> sNama;
    cout << endl << " Selamat Datang " << sNama << endl << endl;
    cout << " Silahkan menggunakan aplikasi ini
    " << endl << endl;
```

```
cout <<"Masukkan bilangan bulat 1 : ";
cin >> Bil1;
cout <<"Masukkan bilangan bulat 2 : ";
cin >> Bil2;

cout <<endl<<endl;

cout <<Bil1 <<" + " <<Bil2 <<" = " <<Bil1+Bil2
<<endl<<endl;

cout <<Bil1 <<" - " <<Bil2 <<" = " <<Bil1-Bil2
<<endl<<endl;
cout <<Bil1 <<" Div " <<Bil2 <<" = " <<Bil1/Bil2
<<endl<<endl;
cout <<Bil1 <<" Mod " <<Bil2 <<" = " <<Bil1%Bil2
<<endl<<endl;

system("pause");
return 0;
}
```

#### Output yang dihasilkan :

```
Nama Anda : Desi Ratnasari

Selamat Datang Desi
Silahkan menggunakan Aplikasi ini
Masukkan bilangan bulat 1 : 9
Masukkan bilangan bulat 2 : 7

9 + 7 = 16
9 - 7 = 2
9 Div 7 = 1
9 Mod 7 = 2
Press any key to continue . . .
```

#### Penjelasan Kode Program :

- a. /: merupakan Div apabila digunakan pada variable dengan tipe data integer, dan pembagian biasa apabila digunakan pada variable dengan tipe data float atau double.
- b. %: merupakan Mod, yaitu sisa hasil pembagian yang telah dijelaskan pada pertemuan sebelumnya.

#### 4. Menggunakan Perintah Konstanta

1. Menghitung Luas lingkaran yang diketahui jari-jari lingkaran 5

- ✓ Buatlah sebuah project baru dengan nama **LatihanKonstanta.cpp**.
- ✓ Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    const float phi = 3.14;
    float r = 5;
    float luas;

    cout<< endl <<" Program Menghitung Luas Lingkaran
" <<endl<<endl;
    cout<<"Luas Lingkaran"<<"= "<<phi*r*r<<endl<<endl;

    system("pause");
    return 0;
}
```

**Output yang dihasilkan :**

```
Program Menghitung Luas Lingkaran
Luas Lingkaran = 78.5
Press any key continue . . .
```

**Penjelasan :**

Konstanta adalah suatu nilai yang tetap (tidak berubah) dalam sebuah program. Contoh sebuah konstanta adalah saat kita menentukan sebuah tetapan dengan sebuah nilai, misalnya kita tetapkan pi adalah bernilai 3.14, dalam Bahasa C ++ penetapan konstanta tersebut dapat dilakukan dengan menggunakan preprocessor directive #define.

```
#include "stdafx.h"
#include <iostream>
#define phi 3.14

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
```

```
float r = 5;
float luas;

cout<< endl <<" Program Menghitung Luas Lingkaran
"<<endl<<endl;
cout<<"Luas Lingkaran"<<"= "<<phi*r*r<<endl<<endl;

system("pause");
return 0;
}
```

#### 5. Menggunakan Perintah Manipulator

- Buatlah project dengan nama **Manipulator.cpp**
- Ketikkan Kode Program seperti berikut :

```
#include "stdafx.h"
#include <iostream>
#include <iomanip>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    int bil = 23;
    char st[] = "Halo";

    cout << bil << endl;
    cout << setw (4) << bil << endl;
    cout << setw (5) << bil << endl;

    cout << bil << endl;
    cout << setw (6) << setfill (' ') << bil <<endl;
    cout << setw (7) << setfill ('*') << bil <<endl;

    cout << st << endl;
    cout << setw (6) << setfill (' ') << st <<endl;
    cout << setw (7) << setfill ('*') << st <<endl;

    system("pause");
    return 0;
}
```

#### Output yang dihasilkan :

```

23
  23
   23
    23
   ***23
  Halo
   Halo
  ***Halo
Press any key continue . . .

```

#### Penjelasan :

- ✓ `setw (n)` = untuk menentukan lebar tampilan sebanyak `n` karakter
- ✓ `setfill (karakter)` = untuk menambahkan karakter pada bagian ruang yang ditentukan melalui `setw` yang normalnya tidak berisi karakter apa pun.
- ✓ `setprecision (n)` = untuk menentukan jumlah digit pecahan.

#### 6. Menggunakan Perintah Matematika

- Buatlah project dengan nama **Peluru.cpp**
- Ketikkan Kode Program seperti berikut :

```

#include "stdafx.h"
#include <iostream>
#include <cmath>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    //Program menghitung jarak kecepatan peluru

    double kecepatan, sudut, jarak;
    kecepatan = 32;
    sudut = 40;

    jarak = 2*kecepatan*kecepatan*sin(sudut*3.14/
180)*cos(sudut*3.14/180)/9.8;

    cout <<"jarak ="<< jarak;
    cout <<"\n"<<endl;

    system("pause");
    return 0;
}

```

#### Output yang dihasilkan :

```
Jarak =102.889
Press any key continue . . .
```

#### C. Rangkuman

- Konstanta adalah nilai numeris ataupun karakter yang tetap/tidak berubah selama program berlangsung.
- Operator merupakan simbol yang digunakan dalam suatu program untuk melakukan suatu operasi atau manipulasi. Operator terdiri dari Operator pengerjaan, operator aritmatika, operator hubungandan operator logika.
- Debugging adalah proses menemukan atau melacak kesalahan pada kode sebuah program. Versi debug ++ digunakan untuk memudahkan programmer dalam melakukan proses Debugging atau melacak kesalahan pada kode program. Pada versi debug ini akan memberikan informasi-informasi yang berguna saat akan dilakukan Debugging. Versi debug juga bisa disebut sebagai debugger pada Microsoft Visual C++.

#### D. Evaluasi Formatif

1. Jawablah dengan singkat pertanyaan di bawah ini:
  - a. Apa perbedaan antara variabel dengan konstanta?
  - b. Bagaimana cara pendeklarasian konstanta dalam bahasa C++?
  - c. Konstanta karakter escape untuk memindahkan kursor ke baris baru adalah?
  - d. Tuliskan bentuk umum pendeklarasian variabel dalam bahasa C++ ?
2. Buatlah algoritma dan program dengan C++ untuk mencari total detik dari masukan berupa jam, menit dan detik. Masukan dari program adalah nilai jam, menit dan detik yang merupakan bilangan bulat. Keluaran yang diminta adalah total detik dari nilai-nilai yang dimasukkan.
3. Buatlah algoritma dan program dengan C++ untuk mencari volume dan luas permukaan bola apabila di beri masukan berupa nilai jari-jari bola. Masukan yang diinginkan adalah jari-jari bola dengan tipe bilangan pecahan. Keluaran yang diharapkan adalah volume dan luas bola.



## CHAPTER 2

### KONSTANTA, OPERATOR, DAN DEBUG



4. Buatlah program C++ untuk membaca dua buah bilangan bulat dari keyboard dan kemudian menampilkan jumlah kedua bilangan tersebut.

**Catatan :** Tugas dibuat dalam bentuk hardcopy (disusun rapi), buat algoritmanya, cantumkan source code programnya, printscrint outpunya.

### 3.1 INSTRUKSI PEMILIHAN

#### Kemampuan yang diharapkan

1. Mahasiswa mengetahui penggunaan pemilhan dalam pemrograman.
2. Mahasiswa mampu melakukan pemecahan masalah algoritma pemilihan.
3. Mahasiswa mengerti cara mengimplementasikan kondisi pemilihan dalam visual C++.

#### Uraian Materi

#### A. KONSEP DASAR

##### A.1 Bentuk Umum IF dan Variasinya

Bentuk IF yang juga dikenal dengan istilah IF Statement, memiliki bentuk umum sebagai berikut :

If kondisi then

Aksi-1

[else

Aksi-2]

End if

Kondisi adalah ekspresi boolean yang bernilai benar atau salah, bisa berupa:

Sebuah nilai boolean: true atau false

Sebuah variabel boolean

Sebuah perbandingan data

Dua perbandingan data atau lebih yang digabung

Aksi berupa satu statement beberapa statement, dimana tiap statement dapat berupa:

Statement pengisian nilai seperti  $a \leftarrow 5$

Statement input data

Statement output data

Statement pemilihan (If Statement atau Case Statement)

Statement pengulangan (For, Repeat atau While Statement)

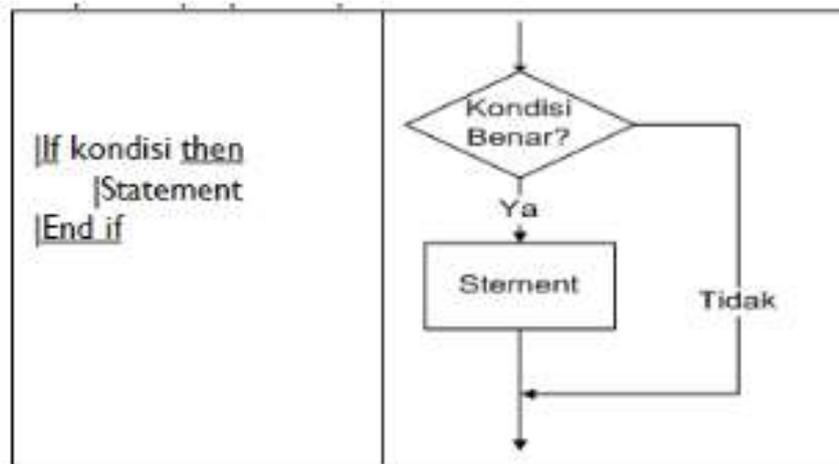
[else Aksi-2], tanda [ ] menyatakan opsional (boleh ada/tidak ada), dimana kalau tidak ada, berarti setelah Aksi-1 langsung selesai.

Dari bentuk umum yang telah dijelaskan, maka variasi bentuk IF ini banyak dan tidak berhingga. Di antaranya yang penting dapat disebutkan berikut:

- ✓ if tanpa else (satu pilihan, mengerjakan atau tidak)
- ✓ if dengan else (dua pilihan)
- ✓ if bersarang dimana dalam if ada if lagi, karena Statement dapat berupa satu perintah pemilihan. Salah satu bentuk if bersarang adalah if untuk memilih salah satu dari banyak pilihan.

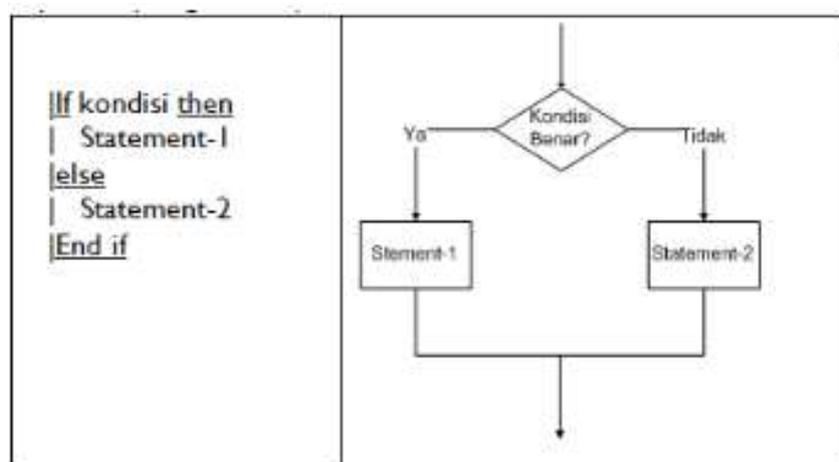
**Contoh-contoh variasi:**

**a. Satu pilihan (tanpa ELSE)**



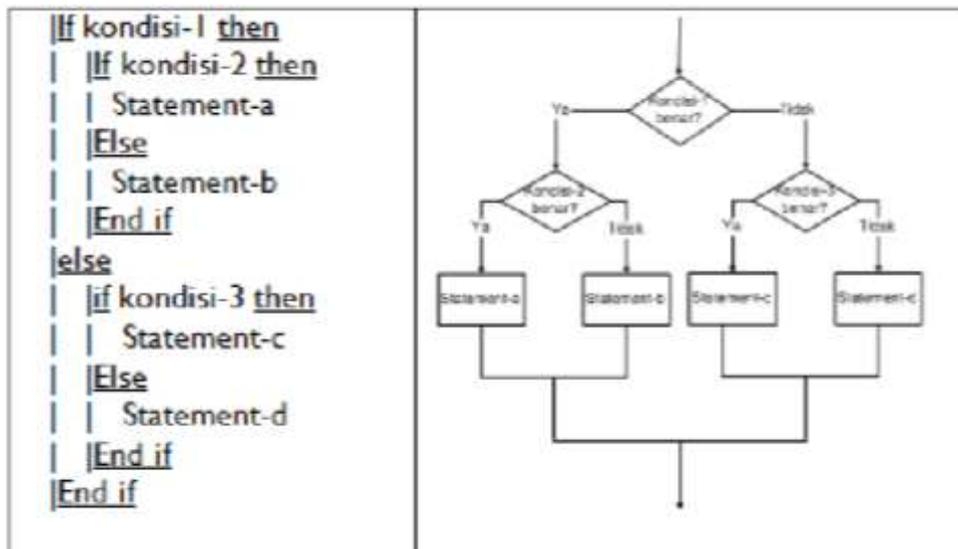
Pada variasi ini, apabila kondisi bernilai benar maka Statement dikerjakan dan apabila kondisi bernilai salah maka Statement tidak dikerjakan.

**b. Dua pilihan (dengan ELSE)**



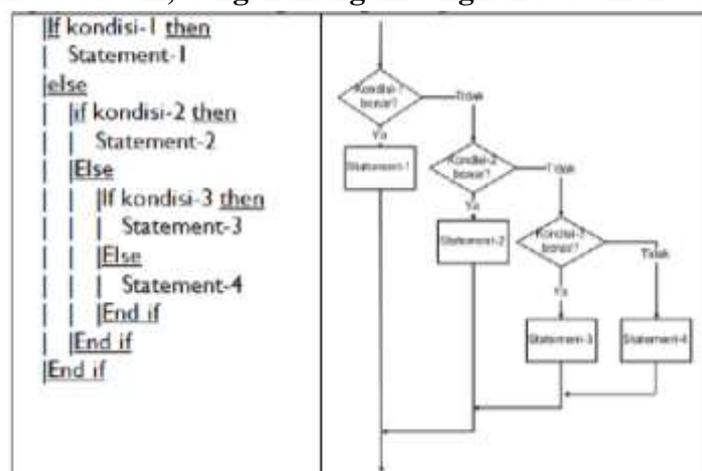
Pada variasi ini, apabila kondisi bernilai benar maka Statement-1 yang dikerjakan dan apabila kondisi bernilai salah maka Statement-2 yang dikerjakan (tidak pernah 2 statement ini dikerjakan semua).

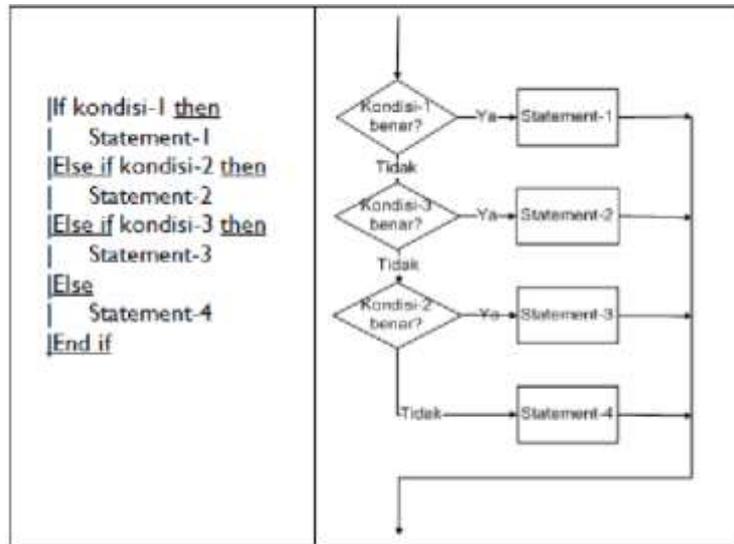
- c. Tiga pilihan atau lebih, misal Statement-1 dan Statement-2 pada contoh-2 dikembangkan menjadi bentuk if lagi sehingga jadi 4 pilihan:



Pada variasi ini, apabila kondisi-1 bernilai benar maka dilanjutkan memeriksa kondisi-2. Apabila kondisi-2 bernilai benar maka Statement-a yang dikerjakan dan apabila kondisi-2 bernilai salah maka Statement-b yang dikerjakan. Sedangkan apabila kondisi-1 salah maka dilanjutkan memeriksa kondisi-3. Apabila kondisi-3 bernilai benar maka Statement-c yang dikerjakan dan apabila kondisi-3 bernilai salah maka Statement-d yang dikerjakan. (dari 4 statement yang ada hanya salah satu yang dikerjakan).

- d. Tiga pilihan atau lebih, dengan mengembangkan Statement setelah ELSE





Pada variasi 1 dan 2 ini, apabila kondisi-1 bernilai benar maka Statement-1 yang dikerjakan dan selesai, tetapi apabila kondisi-1 bernilai salah maka dilanjutkan memeriksa kondisi-2. Apabila kondisi-2 bernilai benar maka Statement-2 yang dikerjakan dan selesai, tetapi apabila kondisi-2 bernilai salah maka dilanjutkan memeriksa kondisi-3. Apabila kondisi-3 bernilai benar maka Statement-3 yang dikerjakan, tetapi apabila kondisi-3 bernilai salah maka Statement-4 yang dikerjakan. (dari 4 statement yang ada hanya salah satu yang dikerjakan. Penggunaan variasi 2 lebih baik, karena tulisan **tidak makin**

**ke kanan** walaupun **pilihan semakin banyak**.

**e. Bentuk bebas (jumlahnya tak terhingga), salah satunya**

```

|If kondisi-1 then
| |Statement-a
| |If kondisi-2 then
| | |If kondisi-3 then
| | | |Statement-b
| | |Else
| | | |If kondisi-4 then
| | | | |Statement-c
| | | |Else
| | | | |Statement-d
| | | |End if
| | |End if
| |Else
| | |Statement-e
| |End if
| |Statement-f
|Else

```

```
| |Statement-g  
| |if kondisi-5 then  
| | |Statement-h  
| |End if  
| |if kondisi-6 then  
| | |Statement-i  
| |End if  
| |Statement-j  
| |End if
```

### B. Penugasan Kelas

#### Langkah Pratikum

1. Buatlah sebuah project baru dengan nama **Pemilihan.cpp**.

- Ketikkan perintah berikut

```
#include "stdafx.h"  
#include <iostream>  
#include <conio.h>  
  
using namespace std;  
  
int _tmain (int argc, _TCHAR* argv[])  
{  
    printf("masukkan angka: ");  
    int nilai;  
  
    scanf("%d", &nilai);  
    if(nilai >= 65)  
    {  
        printf("Selamat Anda lulus");  
    }  
    else  
    {  
        printf("Anda tidak lulus");  
    }  
  
    getch();  
    return 0;  
}
```

#### Output yang dihasilkan :

```
Masukkan angka : 67  
Selamat Anda Lulus_
```

2. Buatlah sebuah project baru dengan nama **Genap.cpp**.
  - Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    int bilangan;
    cout <<"Masukkan sebuah bilangan bulat : ";
    cin >> bilangan;

    if (bilangan % 2 == 0)
        cout <<"Bilangan Genap " <<endl;
    else
        cout <<"Bilangan Ganjil" <<endl;

    getch();
    return 0;
}
```

#### Output yang dihasilkan :

```
Masukkan sebuah bilangan bulat : 7
Bilangan Ganjil
```

Program diatas menentukan bilangan genap dan bilangan ganjil.

3. Buatlah algoritma maupun program yang meminta sebuah nilai ujian dimasukkan dari keyboard dan memvalidasi nilai tersebut. Berikan komentar “Absah” jika nilai tersebut berada antara 0 sampai dengan 100 dan berikan komentar “Tidak Absah” untuk keadaan sebaliknya.

#### Algoritmanya :

- ✓ Masukkan nilai  $\geq 0$  DAN nilai  $\leq 100$  MAKA
- ✓ Tampilkan (“Absah”)
- ✓ SEBALIKNYA
- ✓ Tampilkan (“Tidak Absah”)
- ✓ AKHIR-JIKA

- Buatlah sebuah project baru dengan nama **Valid.cpp**.

- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    double nilai;
    cout <<"Masukkan nilai Ujian : ";
    cin >> nilai;

    if (nilai >= 0 && nilai <= 100)
        cout <<"Absah " <<endl;
    else
        cout <<"Tidak Absah" <<endl;

    getch();
    return 0;
}
```

#### Output yang dihasilkan :

```
Masukkan nilai Ujian : -67
Tidak Absah
```

4. Tabel berikut memperlihatkan daftar predikat kelulusan seorang sarjana berdasarkan indeks prestasi kumulatif-nya.

IP Kumulatif	Predikat Kelulusan
2,00 <= IP <= 2,75	Lulus Memuaskan
2,75 < IP <= 3,50	Lulus Sangat Memuaskan
3,50 <= IP <= 4,00	Lulus Dengan Pujian

#### Algoritmanya :

- ✓ Masukkan (IP)
- ✓ JIKA IP >= 2 DAN IP <= 2,75 MAKA
- ✓ Tampilkan (“Lulus memuaskan”)
- ✓ SEBALIKNYA
- ✓ JIKA IP >= 2,75 DAN IP <= 3,50 MAKA
- ✓ Tampilkan (“Lulus sangat memuaskan”)
- ✓ SEBALIKNYA

- ✓ JIKA IP  $\geq 3,50$  DAN IP  $\leq 4,00$  MAKA
- ✓ Tampilkan (“Lulus dengan pujian”)
- ✓ SEBALIKNYA
- ✓ Tampilkan (“Data IP tidak valid”)
- ✓ AKHIR-JIKA
- ✓ AKHIR-JIKA
- ✓ AKHIR-JIKA
- Buatlah sebuah project baru dengan nama **Predikat.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    double ip;

    cout << "IP Kumulatif :";
    cin >> ip;

    if (ip >= 2.00 && ip <= 2.75)
        cout << "Lulus Memuaskan" << endl;
    else
        if (ip > 2.75 && ip <= 3.50)
            cout << "Lulus Sangat Memuaskan " << endl;
        else
            if (ip > 3.50 && ip <= 4.00)
                cout << "Lulus Dengan Pujian" << endl;
            else
                cout << " Data IP tidak Valid" << endl;

    getch();
    return 0;
}
```

#### Output yang dihasilkan :

```
IPK Kumulatif : 3.99
Lulus Dengan Pujian
```

#### C. Rangkuman

- Instruksi pemilihan biasa juga dikenal dengan istilah IF Statement dengan bentuk umum if (kondisi) →aksi-1→else→aksi-2→end if.
- Beberapa variasi dari instruksi pemilihan yaitu: Satu pilihan (tanpa ELSE), Dua pilihan (dengan ELSE), Tiga pilihan atau lebih misal Statement-1 dan Statement-2 pada contoh-2 dikembangkan menjadi bentuk if lagi sehingga jadi 4 pilihan, Tiga pilihan atau lebih, dengan mengembangkan Statement setelah ELSE, Bentuk bebas (jumlahnya tak terhingga), salah satunya.

#### D. Evaluasi Formatif

1. Buatlah algoritma dan program yang membaca data kode bulan dari *keyboard*, kemudian menampilkan nama bulan.
2. Buatlah algoritma dan program yang meminta data bulan [1,2,3,4,5,6,7,8,9,10,11,12] dimasukkan dari *keyboard*, kemudian menentukan jumlah hari dalam bulan tersebut.
3. [ **Menentukan bilangan terbesar di antara tiga bilangan** ] Diketahui tiga buah bilangan yang dimasukkan dari *i*. Tuliskan algoritma dan program untuk menentukan bilangan terbesar.

**Catatan :** Tugas dibuat dalam bentuk hardcopy (disusun rapi), buat algoritmanya, cantumkan source code programnya, printscreen outputnya.

### 4.1 INSTRUKSI PERULANGAN

#### Kemampuan yang diharapkan

1. Mahasiswa mengetahui penggunaan pengulangan dalam pemrograman.
2. Mahasiswa mampu melakukan pemecahan masalah algoritma pengulangan.
3. Mahasiswa mengerti cara mengimplementasikan kondisi pengulangan dalam visual C++.

#### Uraian Materi

### A. KONSEP DASAR

#### A.1 Sintaks WHILE

Pengulangan dengan menggunakan WHILE merupakan sebuah pengulangan yang dikendalikan oleh suatu kondisi tertentu, dimana kondisi tersebut yang akan menentukan apakah perulangan itu akan terus dilaksanakan atau dihentikan. Kondisi tersebut akan dicek disetiap awal iterasi, apakah sebuah kondisi terpenuhi atau tidak. Jika kondisi terpenuhi (bernilai benar), maka iterasi akan dilanjutkan. Jika kondisi tidak terpenuhi, maka iterasi dihentikan.

Perulangan dengan WHILE dapat digunakan pada struktur perulangan yang diketahui jumlah iterasinya dan juga pada struktur perulangan yang tidak diketahui jumlah iterasinya, tetapi harus selalu terdapat kondisi berhenti.

Struktur pengulangan WHILE adalah:

```
1 {inisialisasi}
2 WHILE(kondisi)
3     aksi
4     ubah pencacah (pencapaian kondisi berhenti)
5 ENDWHILE
```

Contoh pengulangan While ini:

1. Algoritma untuk menampilkan karakter ‘ \* ‘ sebanyak 5 kali

ALGORITMA Tampil\_Bintang

IS : -

FS : Jumlah bintang yg tampil = 5

KAMUS DATA

i : integer

```
1 BEGIN
2   i ← 0 {inisialisasi pencacah i}
3   WHILE (i < 5) {jumlah iterasi}
4     Output ('*') {aksi}
5     i ← i + 1 {pencapaian kondisi berhenti}
6   ENDWHILE
7 END
```

Output dari algoritma ini:

```
*
*
*
*
*
```

2. Algoritma untuk menampilkan iterasi ke 1 sampai 5 dengan pengulangan.

ALGORITMA Iterasi\_Angka

IS :-

FS : Tampil angka 1 hingga 5

KAMUS DATA

i : integer

```
1 BEGIN
2   i ← 1 {inisialisasi pencacah i}
3   WHILE (i <= 5)
4     Output ('Ini adalah iterasi ke-',i)
5     i ← i + 1
6   ENDWHILE
7 END
```

Output dari algoritma ini:

```
Ini adalah iterasi ke-1
```

Ini adalah iterasi ke-2  
Ini adalah iterasi ke-3  
Ini adalah iterasi ke-4  
Ini adalah iterasi ke-5

#### A.2 Sintaks DO...WHILE

Sintaks DO... WHILE... melakukan pengulangan serupa dengan sintaks WHILE. Penggunaan sintaks ini juga tidak harus menyebutkan jumlah pengulangan yang harus dilakukan, karena dapat digunakan untuk pengulangan dengan jumlah iterasinya yang belum diketahui, tetapi harus mempunyai kondisi berhenti.

Bedanya, jika pada sintaks WHILE kondisi dievaluasi/ diuji sebelum aksi pengulangan dilakukan, sedangkan pada sintaks DO...WHILE pengujian kondisi dilakukan setelah aksi pengulangan dilakukan.

Struktur pengulangan DO...WHILE yaitu:

```
1 {inisialisasi}
2 DO
3     aksi
4     ubah pencacah
5 WHILE (kondisi)
```

Pada struktur pengulangan dengan sintaks DO... WHILE..., aksi akan terus dilakukan hingga kondisi yang dicek di akhir pengulangan, bernilai benar. Dengan sintaks ini, pengulangan pasti dilakukan minimal satu kali, yakni pada iterasi pertama sebelum pengecekan kondisi. WHILE dengan DO WHILE seringkali memberikan hasil yang sama, tetapi ada kalanya hasilnya akan berbeda, sehingga harus berhati-hati dalam penggunaan kondisi antara WHILE dengan DO WHILE.

#### A.3 Sintaks FOR

Sintaks pengulangan FOR merupakan sintaks yang relatif paling mudah digunakan. Sintaks ini serupa dengan sintaks WHILE... DO... dalam hal pengecekan kondisi dilakukan di awal. Dalam menggunakan struktur pengulangan dengan sintaks FOR, pemrogram harus mendefinisikan nilai awal dan nilai akhir pencacah yang menunjukkan jumlah iterasi. Setiap kali iterasi berlangsung, nilai pencacah akan diubah.

Jika pencacah sudah mencapai nilai akhir yang ditentukan, maka pengulangan akan berhenti.

## B. Penugasan Kelas

### Langkah Pratikum

1. Buatlah algoritma dan program untuk menampilkan 6 buah baris yang berisi tulisan “Selamat Belajar Edi Ismanto”

Algoritma :

. bil <- 1

ULANG SELAMA bil <= 6

tampilkan ( “Selamat Belajar Mahasiswa”)

bil <- bil +1

AKHIR –ULANG

**Program :**

- Buatlah sebuah project baru dengan nama **enambaris.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    Int bil;
    bil =1;
    while(bil <= 6)
    {
        cout << " Selamat Belajar Mahasiswa "<< endl;
        bil = bil + 1;
    }

    getch();
    return 0;
}
```

#### Output yang dihasilkan :

```
Selamat Belajar Mahasiswa
```

2. Buatlah algoritma dan program untuk menampilkan deretan seperti berikut.

```
10
9
8
7
6
5
4
3
2
1
```

#### Program :

- Buatlah sebuah project baru dengan nama **deretturun.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int_tmain (int argc, _TCHAR* argv[])
{
    Int bil;
    bil =10;
    while(bil >= 1)
    {
        cout << bil << endl;
        bil = bil - 1;
    }
    getch();
    return 0;
}
```

Output yang dihasilkan :

```
10
9
8
7
6
5
4
3
2
1
```

3. Buatlah algoritma dan program yang meminta n buah bilangan bulat dimasukkan dari keyboard, kemudian menampilkan nilai rata-rata dari keseluruhan bilangan tersebut.

**Program :**

- Buatlah sebuah project baru dengan nama **rata-rata.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int_tmain (int argc, _TCHAR* argv[])
{
    Int n, pencacah;
    Double bil, jum, rata_rata;
    cout << " Jumlah data = ";
    cin >> n;
    jum = 0;
    pencacah = 1;
    while(pencacah <= n)
    {
        cout << " Bilangan "<< pencacah << " = ";
        cin >> bil;
        jum = jum + bil;
        pencacah = pencacah + 1;
    }
    rata_rata = jum / n;
    cout << "Rata-rata = "<< rata_rata << endl;

    getch();
    return 0;
}
```

**Output yang dihasilkan :**

```
Jumlah data = 3
Bilangan 1 = 9
Bilangan 1 = 8
Bilangan 1 = 7
Rata-rata = 8
```

4. Buatlah algoritma untuk membuat segitiga yang disusun dari simbol \* dengan tinggi setinggi ditentukan melalui keyboard.

**Program :**

- Buatlah sebuah project baru dengan nama segitiga.cpp.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include <stdio.h>
#include <conio.h>

using namespace std;

int _tmain (int argc, _TCHAR* argv[])
{
    Int i, j, tinggi;
    cout << " Masukkan Tinggi Segitiga : ";
    cin >> tinggi;
    for(i=1; i<= tinggi; i++)
    {
        for(j=1; j<=i; j++)
            cout << " * ";
        cout <<endl;
    }

    getch();
    return 0;
}
```

**Output yang dihasilkan :**

```
Masukkan Tinggi Segitiga = 8
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

#### C. Rangkuman

- Beberapa variasi dari instruksi perulangan yaitu: Sintaks WHILE, Sintaks DO... WHILE dan Sintaks FOR
- Perulangan dengan WHILE dapat digunakan pada struktur perulangan yang diketahui jumlah iterasinya dan juga pada struktur perulangan yang tidak diketahui jumlah iterasinya, tetapi harus selalu terdapat kondisi berhenti.
- Penggunaan sintaks DO...WHILE juga tidak harus menyebutkan jumlah pengulangan yang harus dilakukan, karena dapat digunakan untuk perulangan dengan jumlah iterasinya yang belum diketahui, tetapi harus mempunyai kondisi berhenti.
- Dalam menggunakan struktur pengulangan dengan sintaks FOR, pemrogram harus mendefinisikan nilai awal dan nilai akhir pencacah yang menunjukkan jumlah iterasi. Setiap kali iterasi berlangsung, nilai pencacah akan diubah. Jika pencacah sudah mencapai nilai akhir yang ditentukan, maka pengulangan akan berhenti.

#### D. Evaluasi Formatif

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

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.

2. Buatlah algoritma dan program untuk menampilkan deret berikut dengan menggunakan struktur pengulangan.

100000000

10000000

1000000

100000

10000

1000

100

10

1

3. Buatlah algoritma dan program untuk menampilkan semua bilangan ganjil yang terletak antara 1 sampai dengan 25 yang tidak habis dibagi dengan 5.
4. Buatlah algoritma dan program untuk menyusun hasil seperti berikut.

1

2 6

3 7 10

4 8 11 13

5 9 12 14 15

**Catatan :** Tugas dibuat dalam bentuk hardcopy (disusun rapi), buat algoritmanya, cantumkan source code programnya, printscrint outputnya.

### 5.1 ARRAY DAN TIPE DATA BENTUKAN

#### Kemampuan yang diharapkan

1. Mahasiswa mengetahui penggunaan array dalam pemrograman.
2. Mahasiswa mampu melakukan pemecahan masalah algoritma array.
3. Mahasiswa mengerti cara mengimplementasikan array dalam visual C++.

#### Uraian Materi

##### A. KONSEP DASAR

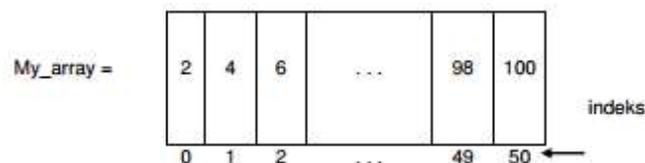
Tipe data array adalah tipe data terstruktur yang merujuk kepada sebuah atau sekumpulan elemen yang mempunyai tipe data yang sama melalui indeks. Array biasanya disebut juga sebagai tabel, vektor atau larik.

Elemen dari array dapat diakses langsung jika dan hanya jika indeks terdefinisi (telah ditentukan nilainya sesuai dengan domain yang didefinisikan untuk indeks tersebut). Struktur data array disimpan dengan urutan yang sesuai dengan definisi indeks secara kontigu (berurutan) dalam memori komputer. Karena itu indeks haruslah merupakan suatu tipe data yang memiliki keterurutan (ada suksesor dan predesesor), misal tipe integer dan karakter.

Dilihat dari dimensinya, array dapat dibagi menjadi Array Satu Dimensi, Array Dua Dimensi dan Array Multi-Dimensi

##### a. Array Satu Dimensi

Representasi array satu dimensi



untuk mendeklarasikan variabel dengan tipe data array satu dimensi pada notasi algoritma, digunakan pola sebagai berikut:

...

KAMUS DATA

```
Nama_variabel : array[x..y] of tipe_data
```

```
...
```

Keterangan:

Nilai x merupakan nilai awal indeks pada array, dan nilai y merupakan nilai akhir pada indeks array.

#### **b. Array Dua Dimensi**

Array dua dimensi merupakan array yang terdiri dari m buah baris (row) dan n buah kolom (column). Bentuk array semacam ini menggunakan 2 (dua) buah kelompok indeks yang masing-masing direpresentasikan sebagai indeks baris dan kolom. Jika ingin memasukkan atau membaca sebuah nilai pada matriks maka, harus diketahui terlebih dahulu indeks baris dan kolomnya.

Untuk mendeklarasikan variabel dengan tipe data array dua dimensi pada notasi algoritma, digunakan pola sebagai berikut:

```
...
```

```
KAMUS DATA
```

```
Nama_variabel : array[x..y,t..u] of tipe_data
```

```
...
```

Keterangan:

Nilai x merupakan nilai awal indeks baris pada array, dan nilai y merupakan nilai akhir indeks baris array. Nilai t merupakan nilai awal indeks kolom pada array,

dan nilai u merupakan nilai akhir indeks kolom array.

#### **c. Array Multi-Dimensi**

Dalam menggambarkan array multidimensi, hanya terbatas hingga dimensi ke-3, yakni dengan menggunakan bangun ruang, namun dalam kenyataannya, tipe data array ini dapat dibentuk menjadi lebih dari tiga dimensi atau menjadi ndimensi.

Untuk mendeklarasikan variabel dengan tipe data array n-dimensi pada notasi algoritma, digunakan pola sebagai berikut:

```
...  
KAMUS DATA  
Nama_variabel: array[a..b,t..u,x..y,..] of tipe_data  
...
```

### B. Penugasan Kelas

#### Langkah Pratikum

1. Buatlah algoritma dan program yang mula-mula menyimpan data huruf vocal ke dalam larik, kemudian tampilkan isi larik.

#### Algoritma :

V <- [“ A”, “E”, “I”,”O”,”U”]

UNTUK I <- 0 S/D 4

Tampilkan (V[i])

AKHIR-UNTUK

#### Program :

- Buatlah sebuah project baru dengan nama **vokal.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"  
#include <iostream>  
#include<stdio.h>  
#include <conio.h>  
  
using namespace std;  
  
int _tmain (int argc, _TCHAR* argv[])  
{  
    Char kar [5];  
    int i;  
    kar[0] = 'A';  
    kar[1] = 'E';  
    kar[2] = 'I';  
    kar[3] = 'O';  
    kar[4] = 'U';  
    for(i = 0; i < 5; i++)  
        cout << kar[i] << endl;  
  
    getch();  
    return 0;  
}
```

Output yang dihasilkan :

```
A
E
I
O
U
```

2. Menukarkan dua angka yang bersebelahan. Terdapat larik seperti berikut.  
[8,9,4,7,6,1,5,3,10,4,16,3]

Tulislah algoritma dan program yang melakukan pertukaran 2 nilai yang berdekatan.  
Hasilnya akan menjadi sebagai berikut.

[9,8,7,4,1,6,3,5,4,10,3,16]

**Program :**

- Buatlah sebuah project baru dengan nama **TukarLarikNilai.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include<stdio.h>
#include <conio.h>

using namespacestd;

int_tmain (intargc, _TCHAR* argv[])
{
    Intdata[] = {8,9,4,7,6,1,5,3,10,4,16,3};
    Inttmp;
    int i;
    for(i = 0; i < 11; i += 2)
    {
        tmp = data [i];
        data[i] = data[i+1];
        data[i+1] = tmp;
    }
    /* Tampilkan isi Larik */
    for(i = 0; i < 12; i++)
        cout << data[i] <<" ";
    cout << endl;

    getche();
    return 0;
}
```

#### Output yang dihasilkan :

```
9 8 7 4 1 6 3 5 4 10 3 16
```

3. Sepuluh siswa mengikuti ujian dengan nilai hasil berkisar antara 0 sampai dengan 100. Tulislah algoritma untuk membaca seluruh data tersebut ( yang mencakup nama dan nilai ujian ), kemudian :

- Hitunglah nilai rata-ratanya, dan
- Tampilkan siswa yang gagal ( yaitu yang nilainya kurang dari 60 )

#### Program :

- Buatlah sebuah project baru dengan nama **DaftarSiswa.cpp**.
- Ketikkan perintah berikut

```
#include "stdafx.h"
#include <iostream>
#include<stdio.h>
#include <conio.h>

using namespacestd;

int _tmain (intargc, _TCHAR* argv[])
{
    struct siswa
    {
        char nama[25];
        double nilai;
    };

    struct siswa daftar[10];
    int pencacah;
    double jum;
    int ada_gagal;

    pencacah = 0;

    while (pencacah < 10)
    {
        cout << " Nama Siswa : ";
        cin >> daftar[pencacah].nama;
```

```
while (1)
{
    cout << "Nilai : ";
    cin >> daftar[pencacah].nilai;

    if (daftar[pencacah].nilai >= 0 &&
        daftar[pencacah].nilai <= 100)
        break;
    else
        cout << " Nilai harus terletak antara
            "<< "0 sampai dengan 100\n";
    }
    pencacah++;
}

//Jumlah semua nilai
jum = 0;
pencacah = 0;
while (pencacah < 10)
{
    jum = jum + daftar[pencacah].nilai;
    pencacah++;
}

//Menampilkan siswa yang gagal
cout << "Siswa yang gagal : \n";
ada_gagal = 0;
pencacah = 0;
while (pencacah < 10)
{
    if (daftar[pencacah].nilai < 60)
    {
        ada_gagal = 1;
        cout << daftar[pencacah].nama << endl;
    }
    pencacah++;
}
if (ada_gagal == 0)
    cout << " Tak ada\n";

getche();
return 0;
}
```

#### Output yang dihasilkan :

```
Nama Siswa : Desi
Nilai : 99
Nama Siswa : Ratna
Nilai : 70
Nama Siswa : Sari
Nilai : 45
Nama Siswa : Fajar
Nilai : 55
Nama Siswa : Septi
Nilai : 80
Nama Siswa : Taat
Nilai : 65
Nama Siswa : Nia
Nilai : 78
Nama Siswa : Masta
Nilai : 40
Nama Siswa : Siti
Nilai : 95
Nama Siswa : Farah
Nilai : 83
Siswa yang gagal :
Sari
Fajar
Masta
```

#### C. Rangkuman

- Tipe data *array* adalah tipe data terstruktur yang merujuk kepada sebuah atau sekumpulan elemen yang mempunyai tipe data yang sama melalui indeks. *Array* biasanya disebut juga sebagai **tabel, vektor atau larik**.
- Elemen dari *array* dapat diakses langsung **jika dan hanya jika** indeks terdefinisi (telah ditentukan nilainya sesuai dengan domain yang didefinisikan untuk indeks tersebut). Struktur data *array* disimpan dengan urutan yang sesuai dengan definisi indeks **secara kontigu (berurutan) dalam memori** komputer. Karena itu indeks haruslah merupakan suatu tipe data yang memiliki keterurutan (ada suksesor dan predesesor), misal tipe integer dan karakter.
- Dilihat dari dimensinya, *array* dapat dibagi menjadi *Array Satu Dimensi*, *Array Dua Dimensi* dan *Array Multi-Dimensi*

#### D. Evaluasi Formatif

1. Buatlah algoritma dan program untuk memperoleh larik yang berisi semua elemen pada larik A yang tidak terdapat pada B.
2. Terdapat larik seperti berikut.  
[8, 9, 4, 7, 6, 1, 5, 3]  
Ubahlah isinya menjadi berikut.  
[3, 5, 1, 6, 7, 4, 9, 8]
3. Terdapat larik yang berisi data nomor dan nama seperti berikut.

Nomor	Nama
10	Rian
6	Sita
3	Fanny
1	Eerwin
2	Ahmad
4	Ridwan
5	Desi
9	Triana
8	Ali
7	Yohana

Tuliskan algoritma dan program yang menampilkan data nomor dan nama dengan tampilanurut menurut nomor.

**Catatan :** Tugas dibuat dalam bentuk hardcopy (disusun rapi), buat algoritmanya, cantumkan source code programnya, printscrint outputnya.