

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
SEMESTER GANJIL TAHUN AKADEMIK 2021/2022**



Disusun oleh :

**Nama : MOHAMMAD HARIFIN
NIM : 2118131
Prodi : TEKNIK INFORMATIKA S1
Kelompok : 34**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2021**

**LEMBAR PERSETUJUAN
ALGORITMA DAN PEMROGRAMAN
SEMESTER GANJIL TAHUN AKADEMIK 2021/2022**



Disusun Oleh

NAMA : MOHAMMAD HARIFIN
NIM : 2118131
PRODI : TEKNIK INFORMATIKA S-1

Mengetahui,
Kepala Laboratorium
Pemrograman Komputer

Menyetujui,
Dosen Pembimbing

A handwritten signature in black ink, appearing to read "Fahrudi Setiawan".

(Ahmad Faisol, ST, MT)
NIP.P.1031000431

(Ahmad Fahrudi Setiawan, S.Kom, M.T.)
NIP.P.1031500497

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2021**

KATA PENGANTAR

Dengan memanjatkan puji syukur ke hadirat Tuhan Yang Maha Esa, karena atas berkah rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Mata Kuliah, guna persyaratan dalam menempuh mata kuliah.

Laporan ini disusun berdasarkan percobaan dan teori dasar yang ada dalam buku panduan praktikum, teori yang diperoleh praktikan dari perkuliahan, dan tidak lupa yaitu Internet sehingga praktikan dapat menambah tidak hanya menguasai teori saja namun juga memahami serta mengaplikasikannya.

Terwujudnya laporan ini, tentunya tidak lepas dari bantuan-bantuan yang telah kami terima. Pada kesempatan ini, kami menyampaikan terima kasih yang sebesar-besarnya kepada yang terhormat:

1. Bapak Ahmad Fahrudi Setiawan, S.kom, M.T. selaku dosen pembimbing Praktikum Algoritma dan Pemrograman.
2. Bapak Joseph Dedy Irawan, ST, MT. dan Bapak Renaldi Primaswana Prasetya, S.Kom, M.Kom. selaku dosen mata kuliah Algoritma dan Pemrograman.
3. Bapak Ahmad Faisol, ST, MT. selaku Ketua Pelaksana Praktikum Algoritma dan Pemrograman Program Studi Teknik Informatika ITN Malang.
4. Instruktur Lab.Pemrograman Komputer yang telah memberi petunjuk kepada penulis selama pelaksanaan praktikum.
5. Rekan-rekan yang telah membantu dalam pelaksanaan dan penyelesaian laporan ini.

Dalam menyusun laporan ini kami menyadari bahwa laporan ini masih memiliki kekurangan, karena itu segala kritik dan saran yang membangun akan kami nanti demi perbaikan penyusunan laporan selanjutnya.

Harapan kami laporan praktikum ini bermanfaat bagi penulis sendiri maupun pembaca sekalian.

Malang, Januari 2022

Mohammad Harifin

DAFTAR ISI

KATA PENGANTARii
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
BAB I Pendahuluan	I-1
I.1 Latar Belakang	I-1
I.2 Rumusan Masalah	I-1
I.3 Tujuan.....	I-2
I.4 Manfaat.....	I-2
BAB II Pengenalan C++	II-1
II.1 Landasan Teori.....	II-1
II.2 Langkah – Langkah Praktikum	II-7
II.3 Latihan Praktikum ke-1:.....	II-7
II.4 Tugas praktikum ke-1:	II-8
II.5 Tugas praktikum ke-2:	II-9
II.6 Tugas Praktikum ke-3:	II-10
II.7 Kesimpulan	II-11
BAB III Operator dan Statement I/O	III-1
III.1 Landasan Teori	III-1
III.2 Langkah – Langkah Praktikum.....	III-9
III.3 Latihan Praktikum ke-1:	III-9
III.4 Latihan Praktikum ke-2:	III-10
III.5 Latihan Praktikum ke-3:	III-11
III.6 Tugas Praktikum ke-1:.....	III-12
III.7 Tugas Praktikum ke-2:.....	III-13
III.8 Tugas Praktikum ke-3:.....	III-14
III.9 Kesimpulan	III-16
BAB IV Percabangan.....	IV-1
IV.1 Landasan Teori	IV-1
IV.2 Langkah – Langkah Praktikum	IV-4

IV.3 Latihan Praktikum ke-1: (IF-ELSE).....	IV-4
IV.4 Latihan praktikum ke-2: (IF ELSE)	IV-5
IV.5 Latihan Praktikum ke-3: (SWITCH CASE).....	IV-6
IV.6 Tugas Praktikum ke-1 : (IF-ELSE)	IV-9
IV.7 Tugas Praktikum ke-2: (IF-ELSE)	IV-10
IV.8 Tugas praktikum ke-3: (SWITCH CASE)	IV-12
IV.9 Kesimpulan.....	IV-14
BAB V Perulangan	V-1
V.1 Landasan Teori	V-1
V.2 Langkah – Langkah Praktikum.....	V-4
V.3 Latihan Praktikum ke-1: (FOR).....	V-4
V.4 Latihan praktikum ke-2: (WHILE-DO WHILE)	V-5
V.5 Latihan praktikum ke-3: (NESTED LOOP)	V-6
V.6 Tugas Praktikum ke-1: (FOR)	V-9
V.7 Tugas Praktikum ke-2: (WHILE, DO-WHILE)	V-10
V.8 Tugas Praktikum ke-3: (NESTED FOR)	V-11
V.9 Kesimpulan	V-12
BAB VI Fungsi Void dan Non-Void	VI-1
VI.1 Landasan Teori	VI-1
VI.2 Langkah – Langkah Praktikum	VI-4
VI.3 Latihan Praktikum ke-1: (FUNGSI NON-VOID).....	VI-4
VI.4 Latihan Praktikum ke-2: (FUNGSI NON-VOID).....	VI-6
VI.5 Latihan Praktikum ke-3: (FUNGSI NON-VOID)	VI-7
VI.6 Tugas Rumah ke-1: (FUNGSI VOID)	VI-8
VI.7 Tugas Rumah ke-2: (FUNGSI VOID)	VI-9
VI.8 Tugas Rumah ke-3 : (FUNGSI VOID)	VI-10
VI.9 Kesimpulan.....	VI-12
BAB VII Array	VII-1
VII.1 Landasan Teori.....	VII-1
VII.2 Langkah – Langkah Praktikum	VII-3
VII.3 Latihan Praktikum ke-1:.....	VII-3
VII.4 Latihan praktikum ke-2:	VII-5

VII.5	Latihan Praktikum ke-3:.....	VII-7
VII.6	Tugas Praktikum ke-1:	VII-7
VII.7	Tugas Praktikum ke-2:	VII-9
VII.8	Tugas Praktikum ke-3:	VII-12
VII.9	Kesimpulan	VII-13
BAB VIII	Kesimpulan	VIII-1
DAFTAR PUSTAKA	VIII-4

DAFTAR GAMBAR

Gambar 2.1 Latihan Praktikum ke-1 (Pengenalan C++)	II-8
Gambar 2.2 Tugas praktikum ke-1 (Pengenalan C++).....	II-9
Gambar 2.3 Tugas Praktikum ke-2 (Pengenalan C++).....	II-10
Gambar 2.4 Tugas praktikum ke-3 (Pengenalan C++).....	II-11
Gambar 3.1 Latihan praktikum ke-1 (Operator dan <i>statement I/O</i>)	III-9
Gambar 3.2 Latihan praktikum ke-2 (Operator dan <i>statement I/O</i>)	III-10
Gambar 3.3 Latihan praktikum ke-3 (Operator dan <i>statement I/O</i>)	III-11
Gambar 3.4 Tugas praktikum ke-1 (Operator dan <i>statement I/O</i>).....	III-12
Gambar 3.5 Tugas praktikum ke-2 (Operator dan <i>statement I/O</i>).....	III-13
Gambar 3.6 Tugas praktikum ke-3 (Operator dan <i>statement I/O</i>).....	III-15
Gambar 4.1 Latihan praktikum ke-1: <i>if else</i> (percabangan)	IV-5
Gambar 4.2 Latihan praktikum ke-2 : <i>if else</i> (Percabangan)	IV-6
Gambar 4.3 Latihan praktikum ke-3: <i>switch case</i> (Percabangan)	IV-8
Gambar 4.4 Tugas praktikum ke-1: <i>if else</i> (Percabangan).....	IV-10
Gambar 4.5 Tugas praktikum ke-2: <i>if else</i> (Percabangan).....	IV-11
Gambar 4.6 Tugas praktikum ke-3: <i>switch case</i> (Percabangan).....	IV-13
Gambar 5.1 Latihan praktikum ke-1 (Perulangan for)	V-5
Gambar 5.2 Latihan Praktikum ke-2 (<i>while, do-while</i>)	V-6
Gambar 5.3 Latihan Praktikum ke-3 (<i>Nested loop</i>).....	V-7
Gambar 5.4 Tugas Praktikum ke-1 (Perulangan for).....	V-10
Gambar 5.5 Tugas praktikum ke-2 (<i>while, do-while</i>).....	V-11
Gambar 5.6 Tugas praktikum ke-3 (<i>Nested loop</i>).....	V-12
Gambar 6.1 Latihan Praktikum ke-1 (fungsi non- <i>void</i>).....	VI-5
Gambar 6.2 Latihan Praktikum ke-2 (fungsi non- <i>void</i>).....	VI-6
Gambar 6.3 Latihan Praktikum ke-3 (fungsi non- <i>void</i>).....	VI-7
Gambar 6.4 Tugas Praktikum ke-1 (fungsi void)	VI-9
Gambar 6.5 Tugas Praktikum ke-2 (fungsi void)	VI-10
Gambar 6.6 Tugas Praktikum ke-3 (fungsi void)	VI-11
Gambar 7.1 Latihan Praktikum ke-1 (<i>Array</i>).....	VII-4
Gambar 7.2 Latihan Praktikum ke-2 (<i>Array</i>).....	VII-6
Gambar 7.3 Latihan Praktikum ke-3 (<i>Array</i>).....	VII-7

Gambar 7.4 Tugas Praktikum ke-1 (<i>Array</i>)	VII-9
Gambar 7.5 Tugas Praktikum ke-2 (<i>Array</i>)	VII-11
Gambar 7.6 Tugas Praktikum ke-3 (<i>Array</i>)	VII-12

DAFTAR TABEL

Tabel 2.1 Penamaan benar dan salah.....	II-5
Tabel 2.1 Tipe data dasar.....	II-6
Tabel 2.2 Newline.....	II-7
Tabel 3.1 operator aritmatika.....	III-2
Tabel 3.2 tabel prioritas operator	III-3
Tabel 3.3 kondisi benar atau salah.....	III-3
Tabel 3.4 Operator Majemuk.....	III-4
Tabel 3.5 AND (&)	III-6
Tabel 3.6 OR ()	III-6
Tabel 3.7 XOR (^)	III-7
Tabel 3.8 NOT (~)	III-7
Tabel 7.1 Ilustrasi tabel <i>array</i>	VII-3

BAB I

Pendahuluan

I.1 Latar Belakang

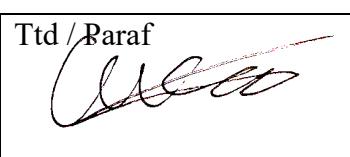
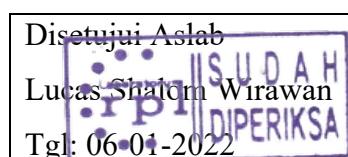
C++ adalah bahasa pemrograman yang dibuat oleh Bjarne Stroustrup yang merupakan perkembangan dari bahasa C pada awal tahun 1970-an. Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B. Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem UNIX. Pada perkembangannya, versi ANSI (American National Standard Institute). Bahasa pemrograman C menjadi versi dominan, meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded. Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (*Object Oriented Programming*).

Dev-C++ merupakan sebuah IDE (Integrated Development Environment) C / C++ yang sudah dilengkapi dengan TDM-GCC Compiler (bagian dari GNU Compiler Collection / GCC). Dev-C++ merupakan IDE gratis dan *full fixture* yang didistribusikan di bawah lisensi GNU General Public License untuk pemrograman C dan C++. Seperti yang sudah disebutkan di atas IDE merupakan lembar kerja terpadu untuk pengembangan program.

Program C++ menghasilkan program yang sederhana dan mudah digunakan, sehingga bisa digunakan untuk memecahkan suatu masalah pada kehidupan sehari-hari.

I.2 Rumusan Masalah

1. Bagaimana cara melakukan pemrograman dengan menggunakan Bahasa C++?
2. Bagaimana cara membuat program dengan menggunakan aplikasi Dev C++?



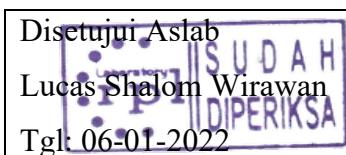
3. Bagaimana mengimplementasikan pemrograman dengan bahasa C++ dalam kehidupan sehari-hari?

I.3 Tujuan

1. Praktikan mampu memahami pemrograman dengan menggunakan bahasa C++
2. Praktikan mampu membuat program dengan menggunakan aplikasi Dev C++
3. Praktikan mampu mengimplementasikan pemrograman dengan Bahasa C++ dalam kehidupan sehari-hari.

I.4 Manfaat

1. Praktikan dapat mengerti dan memahami bahasa pemrograman C++.
2. Praktikan dapat membuat program dengan bahasa pemrograman C++.
3. Praktikan dapat membuat program dan mengaplikasikannya dalam kehidupan sehari-hari.



Ttd / Paraf

BAB II

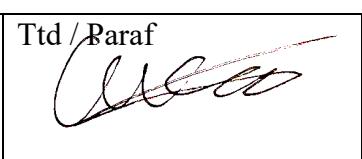
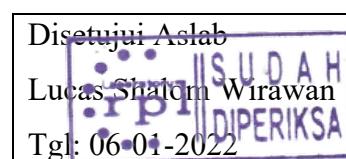
Pengenalan C++

Jumlah Pertemuan	:	100 menit
Tujuan Praktikum	:	<ul style="list-style-type: none"> 1. Mampu memahami pengertian Algoritma dan Pemrograman dengan menggunakan C++ 2. Mampu memahami konsep ‘Include’ pada C++ 3. Mampu memahami bentuk umum dari pemrograman C++ 4. Mampu memahami variabel dan tipe data pada C++ 5. Mampu membuat program sederhana dari pemahaman sebelumnya
Alat / bahan	:	<ul style="list-style-type: none"> 1. Seperangkat <i>computer</i>. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2020.

II.1 Landasan Teori

Bahasa Pemrograman C++ adalah bahasa Pemrograman Komputer Tingkat Tinggi (*High Level Language*), tapi C++ juga dimungkinkan untuk menulis Bahasa Pemrograman Tingkat Rendah (*Low Level Language*) di dalam pengkodingan

C++ merupakan penyempurnaan dari bahasa pemrograman sebelumnya yaitu bahasa C, oleh Bjarne Stroustrup pada tahun 1980. Awal C++ mempunyai nama yaitu “C with Classes” dan berganti nama menjadi C++ pada tahun 1983. Bjarne Stroustrup membuat bahasa pemrograman C++ dengan tambahan fasilitas, yang sangat berguna pada tahun itu sampai sekarang, yaitu bahasa pemrograman yang mendukung OOP (*Object Oriented Programming*).



II.1.1 Header dan Fungsi Main

Header adalah tempat kita memasukkan *library* yang ada di c++ ke dalam program, *library* tersebut berisi deklarasi berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar *Input/Output* maupun sebagai *syntax*.

Apabila kita lihat pada program baris pertama terdapat `#include <iostream>`. Tanda '#' adalah **Preprocessor directive**. *Preprocessor directive* sendiri adalah instruksi untuk *compiler* memerintahkan untuk memproses suatu arahan atau informasi sebelum memulai melakukan kompilasi pada program yang telah kita tulis, misal untuk memasukkan file *library*, dll.

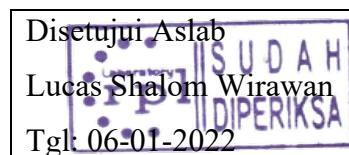
Sebagai contoh, jika dilihat dari program di atas kita akan memasukkan (*include*) *library iostream* ke dalam program kita, *library iostream* sendiri adalah *header* yang dibutuhkan untuk melakukan *input* dan *output*, dll.

Jika di perhatikan terdapat kode `using namespace std;`, *Using* adalah sebuah kata kunci yang digunakan untuk memperkenalkan suatu *Namespace* kepada suatu ruang lingkup, dengan ini kita tidak diwajibkan lagi untuk menyertakan nama dari *Namespace* dan *Scope Operator ::* setiap kali mencoba untuk mengakses entitas dari *Namespace* tersebut.

Selanjutnya fungsi `main()`, `main()` adalah blok program utama kita yang ditulis di antara { hingga }. Fungsi inilah yang akan dipanggil pertama kali pada saat eksekusi program. Apabila ada fungsi lain yang dibuat, maka fungsi tersebut akan dijalankan ketika dipanggil di fungsi utama. Hal ini karena bahasa C/C++ merupakan bahasa prosedural yang menerapkan konsep runtutan (program dieksekusi baris-per baris dari atas ke bawah secara berurutan).

II.1.2 Return 0;

Instruksi *return* menyebabkan fungsi `main()` berakhir dan mengembalikan kode yang mengikuti instruksi tersebut, dalam kasus



ini 0. Ini merupakan cara yang paling sering digunakan untuk mengakhiri program.

II.1.3 Komentar Program

Komentar adalah bagian dari kode program yang tidak ikut dibaca pada proses kompilasi. Dengan kata lain, komentar tidak akan mempengaruhi jalannya program. Dalam C++ terdapat 2 jenis komentar, yaitu :

1. Menggunakan tanda //

Tanda ini digunakan untuk menuliskan komentar yang banyaknya hanya satu baris.

```
// ini adalah komentar untuk satu baris
```

2. Menggunakan tanda /**/

Berbeda dengan sebelumnya, tanda ini dapat digunakan untuk menuliskan komentar yang banyaknya satu baris atau lebih. Komentar ini dimulai dari tanda /* sampai ditemukan tanda */.

```
/*ini adalah komentar panjang
yang banyaknya
lebih dari satu baris*/
```

II.1.4 Tanda Semicolon

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

II.1.5 Input dan Output

Statemen Input adalah *Statemen / fungsi* yang digunakan untuk menerima data dari *input keyboard*, contoh: *cin (character in)*.

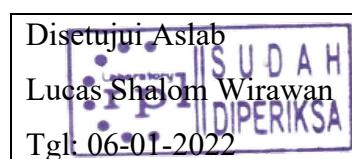
```
cin>>a;
```

Statemen Output adalah *Statemen* yang digunakan untuk menampilkan data ke layar monitor., contoh: *cout (character out)*.

```
cout<<"RPL";
```

II.1.6 Jenis Identifier

Identifier atau pengenal adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan variabel, konstanta, tipe data, dan fungsi. *Identifier* yang berperan sebagai variabel dan konstanta



berfungsi untuk menampung sebuah nilai yang digunakan dalam program.

II.1.7 Konstanta

Konstanta adalah jenis *identifier* yang bersifat konstan atau tetap, artinya nilai dari konstanta di dalam program tidak berubah. Terdapat dua cara untuk membuat sebuah konstanta, yaitu dengan menggunakan *preprocessor directive* `#define` dan menggunakan kata kunci `const`.

a. Menggunakan kata kunci `const`

Mendefinisikan sebuah konstanta dapat dilakukan dengan menggunakan kata kunci `const`. Berikut ini adalah bentuk umumnya.

```
const tipe_data nama_konstanta =
    nilai;
```

b. Menggunakan Preprocessor Directive `#define`

Keuntungan menggunakan `#define` apabila dibandingkan dengan `const` adalah kecepatan kompilasi, karena sebelum kompilasi dilaksanakan, *kompiler* pertama kali mencari *symbol* `#define` (oleh sebab itu mengapa `#` dikatakan *preprocessor directive*). `#define` adalah *preprocessor directive*, oleh sebab itu tidak diakhiri dengan tanda semicolon (`;`).

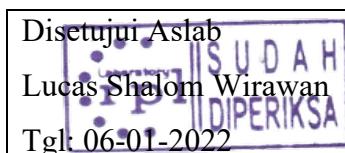
```
#define nama_konstanta nilai_tetapan
```

II.1.8 Variabel

Variabel adalah sebuah *identifier* yang mempunyai nilai dinamis. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan.

1. Deklarasi variabel

Untuk menggunakan variabel pada C++, kita harus mendeklarasikan tipe data yang akan digunakan. *Syntax* penulisan deklarasi variabel adalah dengan menuliskan tipe data yang akan digunakan diikuti dengan *identifier* yang benar. Pengertian deklarasi di sini berarti mengenalkan sebuah pengenal ke program



dan menentukan jenis data yang di simpan di dalamnya. Bentuk pendeklarasian variabel :

```
tipe_data nama_variabel;
```

Jika akan menggunakan tipe data yang sama untuk beberapa *identifier* maka dapat dituliskan dengan menggunakan tanda koma.

```
int a,b,c;
float d,e,f;
```

Nama dari suatu variabel dapat ditentukan sendiri oleh program dengan aturan sebagai berikut :

- Tidak boleh dimulai dengan karakter non huruf
- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C++ bersifat *case-sensitive*, yang artinya huruf besar dan kecil dianggap berbeda.

Jadi antara **nim**, **NIM**, dan **Nim** dianggap berbeda.

- Tidak boleh mengandung spasi.
- Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk simbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, *, (,), -, +, =, dsb.
- Panjang bebas, tetapi hanya 32 karakter pertama yang terpakai.

Contoh penamaan variabel yang benar dan salah :

Tabel 2.1 Penamaan benar dan salah

Penamaan yang Benar		Penamaan yang Salah
namasiswa		nama siswa (salah karena menggunakan spasi)
XY12		12X (salah karena dimulai dengan angka)
harga_total		harga.total (salah karena menggunakan karakter .)
JenisMotor		Jenis Motor (salah karena menggunakan spasi)
alamatrumah		for (salah karena menggunakan kata kunci bahasa pemrograman)

2. Inisialisasi variabel

Inisialisasi dapat didefinisikan sebagai proses pengisian nilai awal (nilai *default*) ke dalam suatu variabel. Dalam C++, pengisian nilai dilakukan dengan menggunakan operator sama dengan (=).

Bentuk pernyataan yang digunakan untuk memberikan nilai ke variabel yang telah dideklarasikan atau didefinisikan :

```
nama_variabel = nilai_awal;
```

Adakalanya dalam penulisan program, variabel langsung diberi nilai setelah didefinisikan. Sebagai contoh :

```
int A;  
A = 10;
```

Dua pernyataan seperti di atas sebenarnya dapat disingkat melalui pendefinisian yang disertai penugasan nilai, sebagai berikut :

```
int A = 10;
```

II.1.9 Tipe Data

1. Tipe Data Dasar

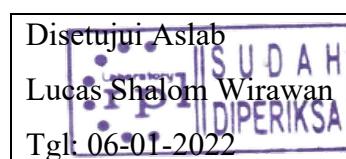
Adalah tipe data yang dapat langsung dipakai. Daftarnya dapat dilihat pada tabel di bawah ini :

Tabel 2.1 Tipe data dasar

Tipe Dasar	Ukuran Memory (byte)	Jangkuan Nilai	Jumlah digit Presisi
Char	1	-128 hingga +127	-
Int	2	-32768 hingga +32767	-
Long	4	-2.147.438.648 hingga 2.147.438.647	-
Float	4	3,4E-38 hingga 3,4E38	6-7
Double	8	1,7E-308 hingga 1,7E308	15-16
Long Double	10	3,4E-4932 hingga 1,1E4932	19

2. Karakter dan String Literal

String adalah tipe data text yang memungkinkan kita menyimpan nilai dengan bentuk text, kumpulan dari karakter. Berbeda dengan *char* gabungan dari karakter. Perbedaannya dengan tipe data *char* adalah mengenai penyimpanannya. *Char*



Ttd / Paraf

menyimpan hanya beberapa karakter saja, sedangkan *String* dapat menyimpan banyak karakter, kata, dan kalimat sekaligus.

3. Escape Sequence Characters

Escape sequence characters adalah urutan sebuah tabel, penulisan kode yang mewakili sebuah karakter yang tidak dapat diwakili secara langsung seperti *newline* yang diwakili dengan tanda \n dan petik dua dengan menggunakan tanda \".

Tabel 2.2 Newline

Karakter	Keterangan
\0	Karakter ber-ASCII nol (karakter null)
\a	Karakter bell
\b	Karakter Backspace
\f	Karakter ganti halaman (fromfeed)
\n	Karakter baris baru (newline)
\r	Karakter carriage return (ke awal baris)
\t	Karakter tab horizontal
\v	Karakter tab vertikal
\\\	Karakter \
\'	Karakter '
\"	Karakter "
\?	Karakter ?

II.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum Algoritma dan Pemrogramman.
3. Menjalankan *script* dari modul.

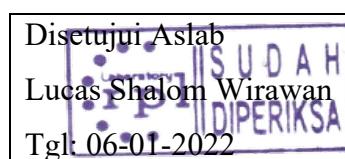
II.3 Latihan Praktikum ke-1:

Dengan perintah *cout*, buatlah tampilan seperti pada gambar berikut, kemudian berikan komentar keterangan pada setiap baris pada *source code* kalian!

1. *Source code* :

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

int main(){
    cout<<endl;
    cout<<"\t\t INSTITUT TEKNOLOGI NASIONAL"<<endl;
    cout<<"\t\t-----"< "<<endl;
    cout<<endl;
```



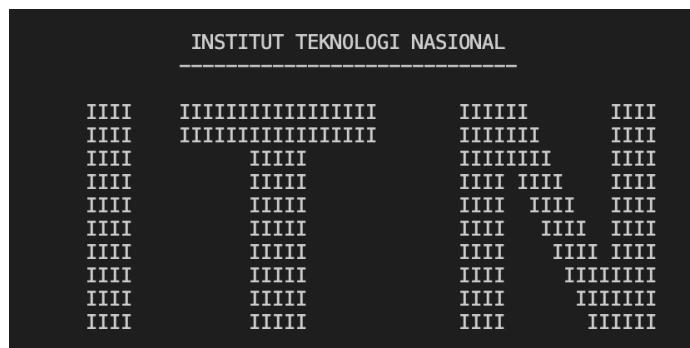
Ttd / Paraf

```

        cout<<"\tAAAAA\t"<<"BBBBBBBBBBBBBBB\t"=<<"AAAAAA
AAAA" <<endl;
        cout<<"\tAAAAA\t"<<"BBBBBBBBBBBBBBB\t"=<<"AAAAAA
AAAA" <<endl;
        cout<<"\tAAAAA\t"=<<"      AAAAA      \t"=<<"AAAAAA
AAAA"
        cout<<endl;
    }
}

```

2. Tampilan Program :



Gambar 2.1 Latihan Praktikum ke-1 (Pengenalan C++)

3. Analisa Program

Pada program ini hanya menampilkan *output* aja dengan menggunakan *cout*, karena di program ini saya hanya menampilkan teks itu maka library yang di gunakan adalah *iostream* untuk menjalankan program ini.

II.4 Tugas praktikum ke-1:

Salinlah *source code* di bawah, kemudian identifikasi kesalahan kode program. Analisa kesalahan kode beserta kode program yang sudah benar dan tampilannya !(pada bagian *source code*, berikan komentar keterangan pada baris atau bagian yang kalian perbaiki).

1. *Source code :*

```
#include <iostream>
using namespace std;
int a;
int main(){
    cout <<"Masukkan Usia Anda : ";
    cin >>a;
    cout <<"Usia Anda adalah "<<a<<" tahun";
    return 0;
}
```

2. Tampilan Program :

```
Masukkan Usia Anda : 18
Usia Anda adalah 18 tahun
arifin@MacBook-Air-Arifin T %
```

Gambar 2.2 Tugas praktikum ke-1 (Pengenalan C++)

3. Analisa Program

Yang di perbaiki yaitu dari garis ketiga *cout* *int a=10;* di ubah menjadi *int a;* . garis kelima yang ada pada Code sebelumnya yaitu *cout >>* di ubah menjadi *cout<<..* dan garis ketujuh di Code sebelumnya yaitu *cout >>* dan *<<a>>* di ubah menjadi *cout<<* dan *<<a<<*.

II.5 Tugas praktikum ke-2:

Dengan perintah *cin* dan *cout*, buatlah program C++ untuk menginputkan biodata kalian, lalu setelah di *input* kan maka biodata akan ditampilkan seperti pada tampilan program di bawah!

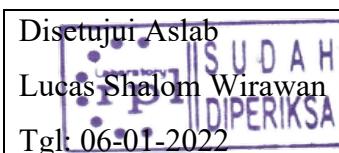
1. *Source code :*

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

int main (){

    string nama, hobi, asal, usia;

    cout << "\tMASUKKAN BIODATA ANDA\n";
    cout << "Masukkan nama anda :";
    getline (cin,nama);
    cout << "Masukkan usia anda :";
    getline (cin,usia);
    cout << "Masukkan asal anda :";
    getline (cin,asal);
    cout << "Masukkan hobi anda :";
    cin >> hobi;
```



Ttd / Paraf

```

cout << "\tHASIL BIODATA ANDA \n";
cout << "\t=====\n";
cout << "nama\t:" << nama << endl;
cout << "usia\t:" << usia << endl;
cout << "asal\t:" << asal << endl;
cout << "hobi\t:" << hobi << endl;

return 0;
}

```

2. Tampilan Program :

```

/var/guruversi/mampirzsscvcsjyupok9_4vv4vuuvgm7
MASUKKAN BIODATA ANDA
Masukkan nama anda :MOHAMMAD HARIFIN
Masukkan usia anda :18 TAHUN
Masukkan asal anda :SUMENEP, KECAMATAN ARJASA
Masukkan hobi anda :FUTSAL
    HASIL BIODATA ANDA
    =====
nama :MOHAMMAD HARIFIN
usia :18 TAHUN
asal :SUMENEP, KECAMATAN ARJASA
hobi :FUTSAL
arifin@MacBook-Air-Arifin T %

```

Gambar 2.3 Tugas Praktikum ke-2 (Pengenalan C++)

3. Analisa Program

Untuk program ini memasukkan biodata dari garis 8 sampai 14 yaitu menampilkan *output* dengan perintah *cout* di perintah *cout* terdapat teks yang di minta dan menyimpannya menggunakan perintah *getline* (*cin,nama*) dan jika udah di *input* maka akan di tampilkan lagi menggunakan perintah *cout << "nama\t:" << nama << endl;*. Dan di akhir menggunakan perintah *return* sebagai tanda akhirnya atau selesanya program.

II.6 Tugas Praktikum ke-3:

Membuat sebuah program tentang materi yang dipelajari pada bab 1 beserta hasil tampilan dan analisa program !

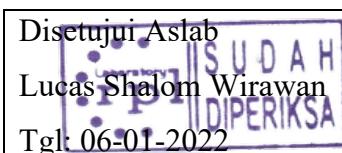
1. *Source code :*

```

//membuat program di pelajari di bab 1 c++
#include <iostream>

using namespace std;
int main(){
    int nim;
    cout << "masukkan nim anda : ";
    cin>>nim;
}

```



Ttd / Paraf

```

cout<<"\t=====\n";
<<endl;
    cout<<"\t= selamat anda telah menjadi mahasiswa ITN\n";
="<<endl;

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

}

```

2. Tampilan Program :

```

/var/folders/bm/Temporary/pZSSCVCSjyH0K9_4VV4v0000gn/17_tempCode
masukkan nim anda : 2118131
=====
= selamat anda telah menjadi mahasiswa ITN =
=====
arifin@MacBook-Air-Arifin T %

```

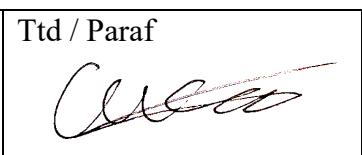
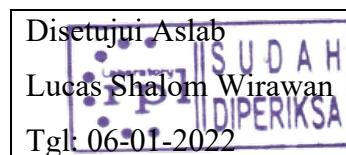
Gambar 2.4 Tugas praktikum ke-3 (Pengenalan C++)

3. Analisa Program

Dari program ini menggunakan perintah *int* untuk menjalankan *cin* dan meng *input* sebuah karakter angka yang diminta dan ditampilkan hasil *input* menggunakan *cout* yang sudah di program sebagai hasil tampilan yang isinya “selamat anda telah menjadi mahasiswa ITN”.

II.7 Kesimpulan

1. *#include <iostream>* adalah memberitahukan *compiler* agar menggunakan file *header iostream* yang ada pada direktori file-file *header*.
2. Int *main()* adalah fungsi utama dari sebuah kode bahasa C. Fungsi ini memberikan nilai balik menurut *type* datanya, dan karena memiliki nilai balik maka diberikan perintah *return* nilai.
3. *cout* adalah fungsi standar pada C++ untuk menampilkan *output* ke layar. Setelah simbol *<<* kita bisa menuliskan teks yang akan ditampilkan ke layar.
4. *cin* adalah perintah dasar C++ untuk proses *input* atau menerima data masukan dari *user* dan menggunakan simbol *>>*.



BAB III

Operator dan Statement I/O

Jumlah Pertemuan	: 100 menit
Tujuan Praktikum	: <ol style="list-style-type: none"> 1. Mampu memahami konsep Operator dan Statement I/O 2. Mampu Memahami jenis-jenis Operator 3. Mampu memahami Statement I/O 4. Mampu menerapkan Operator dan Statement I/O ke dalam program
Alat / bahan	: <ol style="list-style-type: none"> 1. Seperangkat <i>computer</i>. 2. Perangkat lunak: Dev C++ 3. Modul Praktikum Algoritma dan Pemrograman 2020.

III.1 Landasan Teori

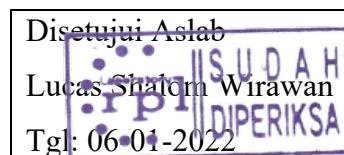
Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi. Dengan operator, kita dapat melakukan operasi perhitungan, perbandingan, manipulasi bit, dan lain-lain.

Sebagai pengenalan, terdapat istilah yang harus diketahui dalam bekerja dengan operator. Sebagai contoh :

C = 3 + 7;

Keterangan :

- C disebut *variable*
- = disebut operator *assignment*
- 3 dan 7 disebut *operand*
- 3 + 7 disebut ekspresi
- + disebut operator *aritmatika* (penambahan)
- C = 3 + 7 disebut *statement aritmatika*



Ttd / Paraf

Pada dasarnya, ada tiga jenis operator dalam pemrograman C++

- *Unary*

Sifat *unary* pada operator adalah melibatkan sebuah *operand* pada suatu operasi *aritmatika*

Contoh: -5

- *Binary*

Sifat *binary* pada operator adalah melibatkan 2 buah *operand* pada suatu operasi *artimatika*.

Contoh: $4 + 8$

- *Ternary*

Sifat *ternary* pada operator adalah melibatkan 3 buah *operand* pada suatu operasi *aritmatika*

Contoh: $10 \% 3 + 2;$

$a > b ? 1 : 0;$

III.1.1 Operator Assingment

Operator *assignment* (penugasan) adalah operator yang digunakan untuk memasukkan atau mengisikan nilai ke dalam suatu variabel. Dalam bahasa C++ digunakan operator berupa tanda sama dengan (“=”).

Contoh :

```
int A, B;
A = 55;
B = 66;
```

Pada contoh di atas, kita memasukkan nilai 55 ke dalam variabel A dan 66 ke dalam variabel B.

III.1.2 Operator Aritmatika

Operator ini digunakan untuk melakukan sebuah *opearasi* penghitungan atau *aritmatika*. Tugasnya untuk memecahkan masalah matematika.

Tabel 3.1 operator aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	X + Y
-	Pengurangan	X - Y

*	Perkalian	X * Y
/	Pembagian	X / Y
%	Sisa pembagian Integer (<i>Modulus</i>)	X % Y

Pada operator penjumlahan, pengurangan, perkalian, dan pembagian dilakukan operasi seperti biasa. Sedangkan, yang di maksud dengan sisa pembagian adalah **sisa dari hasil pembagian** bukan hasil dari pembagian.

Contohnya :

```
10 % 2
Hasil pembagian = 5
Sisa pembagian   = 0
```

Pada sebuah ekspresi bisa saja dijumpai lebih dari 1 operator. Contohnya: $x = 2 + 6 / 3$; Dalam C++ *variable* x menghasilkan nilai 4, karena operator “ / ” memiliki kedudukan yang lebih tinggi dari operator “ + ”.

Tabel 3.2 tabel prioritas operator

Kedudukan	Operator	Deskripsi
Tertinggi	++ --	Increment, dercrement
	* / %	Perkalian, pembagian, modulus
Terendah	+ -	Penjumlahan, pengurangan

III.1.3 Operator Relasional

Operator *relasional* atau hubungan digunakan untuk membandingkan hubungan antara dua buah *operand*. Pada operator ini menghasilkan kondisi **benar** atau **salah**.

Tabel 3.3 kondisi benar atau salah

Operator	Deskripsi	Contoh	
==	Sama dengan (bukan assigment)	5==5	Benar, karena 5 sama dengan 5
!=	Tidak sama dengan	5!=2	Benar, karena 5 tidak sama dengan 2
>	Lebih besar	5 > 2	Benar, karena 5 lebih besar dari 2

<	Lebih Kecil	$5 < 2$	Tidak, karena 5 lebih besar dari 2
\geq	Lebih besar atau sama dengan	$5 \geq 5$	Benar, karena 5 sama dengan 5
\leq	Lebih kecil atau sama dengan	$5 \leq 2$	Tidak, karena 5 lebih besar dari 2

III.1.4 Operator Naik dan Turun (Increment dan Decrement)

Operator *increment* merupakan operator yang dapat menambahkan (menaikkan) suatu nilai. Operator *increment* ini ditandai dengan tanda “++”. Sedangkan, operator *decrement* merupakan operator yang mengurangkan (menurunkan) suatu nilai. Operator *decrement* ini ditandai dengan tanda “--”.

Terdapat dua jenis *increment/decrement*, yaitu *pre-increment* adalah melakukan penambahan nilai sebelum suatu variabel itu diproses, dan *post-increment* yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai.

III.1.5 Operator Majemuk (Compound Assignment)

Operator majemuk digunakan untuk memendekkan penulisan operasi penugasan semacam :

```
x = x + 2 ;
y = y * 4 ;
```

Menjadi :

```
x += 2;
y *= 4;
```

Tabel 3.4 Operator Majemuk

Operator	Keterangan	Contoh Penggunaanya
$+=$	Penjumlahan	$z += x$ sama dengan, $z = z + x$
$-=$	Pengurangan	$z -= x$ sama dengan, $z = z - x$
$*=$	Perkalian	$z *= x$ sama dengan, $z = z * x$
$/=$	Pembagian	$z /= x$ sama dengan, $z = z / x$

III.1.6 Operator Bitwise (Logika)

Operator logika adalah operasi matematika yang mengoperasikan suatu nilai dalam bilangan biner. Operator *bitwise* akan bekerja dengan mengubah bilangan desimal menjadi bilangan biner, setelah itu akan melakukan operasi berdasarkan operator *bitwise* yang digunakan dan pada akhirnya nilai tersebut akan diubah menjadi bilangan desimal kembali. Operator ini hanya bisa digunakan pada *operand bertipe* data int atau char.

a. Shift Left (<<)

Operator Bitwise Shift Left (<<) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kiri.

Bentuk Penulisan :

```
Nilai << jumlah;
```

Contoh : 12 << 1 ?

```
0000 0000 0000 1100 = 12
//menggeser 1 bit ke kiri, Dibagian kanan
disisipkan 0, sebanyak bit yang digeser
0000 0001 0001 1000 = 24
```

b. Shift Right (>>)

Operator Bitwise Shift Right (>>) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kanan.

Bentuk Penulisan :

```
Nilai >> jumlah;
```

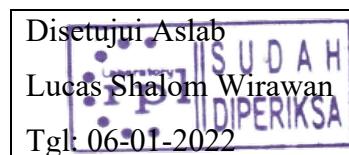
contoh :

12 >> 1 ?

```
0000 0000 0000 1100 = 12
//digeser 1 bit ke kanan, Dibagian kiri disisipkan
0, sebanyak bit yang digeser dan pada bagian
paling kanan akan hilang sebanyak bit yang di
geser
0000 0000 0000 0110 = 6
```

c. AND (&)

Membandingkan berdasarkan sifat dari AND yaitu untuk menghasilkan nilai 1 (*true*) kedua *operand* harus bernilai 1(*true*) jika tidak akan menghasilkan nilai 0 (*false*).



Ttd / Paraf

Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh :12 & 10 ?

Tabel 3.5 AND (&)

Operand 1 (12)	Operand 2 (10)	Hasil
1	1	1
1	0	0
0	1	0
0	0	0

Hasil :

$$1000 = 8$$

d. OR (|)

Membandingkan berdasarkan sifat dari OR yaitu untuk mendapatkan nilai 1 (*true*) maka salah satu atau semua *operand* harus bernilai 1 (*true*), jika semua *operand* bernilai 0 (*false*) maka akan mendapatkan nilai 0 (*false*).

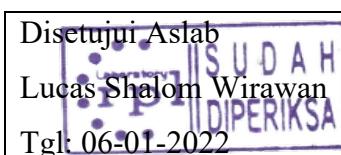
Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh :12 | 10 ?

Tabel 3.6 OR (|)

Operand 1 (12)	Operand 2 (10)	Hasil
1	1	1
1	0	1
0	1	1
0	0	0

Hasil : 1110 = 14



Ttd / Paraf

e. XOR (\wedge)

Membandingkan berdasarkan sifat dari XOR yaitu untuk mendapatkan nilai 1 (*true*) maka kedua *operand* harus memiliki nilai yang berbeda, jika kedua operand memiliki nilai yang sama maka akan mendapatkan nilai 0 (*false*).

Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh : $12 \wedge 10 ?$

Tabel 3.7 XOR (\wedge)

Operand 1 (12)	Operand 2 (10)	Hasil
1	1	0
1	0	1
0	1	1
0	0	0

Hasil : 0110 = 6

f. NOT (\sim)

Operator *Bitwise* NOT dilambangkan dengan menggunakan simbol " \sim ", Operator ini digunakan untuk membalik nilai bit dari suatu *operand*. Berikut ilustrasi penggunaan Operator *Bitwise* NOT.

Contoh : $\sim 10 ?$

Tabel 3.8 NOT (\sim)

Operand	Hasil
1	0
0	1

Hasil : $(10 + 1) * -1 = -11$

g. Logika AND (**&&**)

Operator AND akan menghasilkan nilai 1 (*true*) jika semua *operand*-nya bernilai true. Namun, jika salah satu *operand*-nya bernilai 0 (*false*) maka akan menghasilkan nilai 0 (*false*).

h. Logika OR (**||**)

Operator OR akan menghasilkan nilai 1 (*true*), jika salah satu *operand*-nya bernilai true, dan hanya akan menghasilkan nilai 0 (*false*) jika seluruh *operand*-nya bernilai 0 (*false*).

i. Logika NOT (**!**)

Operator NOT akan menghasilkan nilai kebalikan dari *operand*-nya, misal, jika *operand*-nya bernilai 1 (*true*) maka akan menghasilkan nilai 0 (*false*).

III.1.7 Operator Ternary

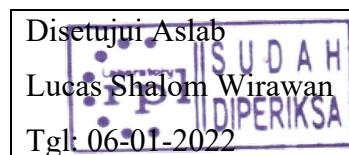
Operator *ternary* adalah operator yang digunakan dalam operasi yang melibatkan tiga buah *operand*. Adapun operator yang digunakan untuk menyatakannya adalah operator ?. berikut bentuk umum penggunaan operator *ternary*.

Kondisi ? Ekspresi2 : Ekspresi3;

III.1.8 Operator I/O

Pada C++ terdapat 2 jenis I/O dasar, yaitu :

2. Statemen Input adalah Statemen / fungsi yang digunakan untuk menerima data dari *keyboard*. Operator yang digunakan “>>”, contoh: `cin>>` (character in).
3. Statemen Output adalah Statemen yang digunakan untuk menampilkan data ke layar monitor. Operator yang digunakan “<<”, contoh: `cout<<` (character out).



III.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum Algoritma dan Pemrogramman.
3. Menjalankan script dari modul.

III.3 Latihan Praktikum ke-1:

Mencetak *output* pada program operator assingment.

1. *Source code* :

```
#include <iostream>
using namespace std;
int main()
{
    int x=6, y=4;
    cout<<"x= "<<x<<endl;
    cout<<"y= "<<y<<endl;
    x+=y; // "x+=y" sama dengan "x=x+y"
    cout<<"Setelah statemen x += y\n";
    cout<<"x= "<<x;
    return 0;
}
```

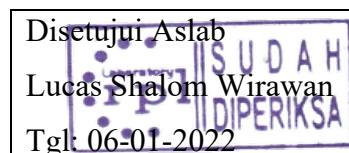
2. Tampilan program:

```
arifin@192 ~ % cd "/Users/arifin/Desktop/latihan/" && g++ 1.cpp -o 1 && "/Users/arifin/Desktop/latihan/1"
x= 6
y= 4
Setelah statemen x += y
x= 10%
arifin@192 latihan %
```

Gambar 3.1 Latihan praktikum ke-1 (Operator dan *statement* I/O)

3. Analisa program:

Pada program terdapat *konstanta* $x=6$ dan $y=4$ yang bertipe data integer. Kemudian dicetak oleh perintah $cout \ll "x= " \ll x \ll endl$ dan $cout \ll "y= " \ll y \ll endl$. Kemudian dilakukan penjumlahan antara nilai variabel x dan y dengan perintah " $x+=y$ ", dan kemudian program mencetak hasil dari penjumlahan dengan $cout \ll "x= " \ll x$. Dan program pun berakhir.



Ttd / Paraf


III.4 Latihan Praktikum ke-2:

Mencetak *output* pada program operator majemuk.

1. Source code :

```
#include <iostream>
using namespace std;
int main()
{
    int x = 2;
    cout<<"nilai x = "<<x<<endl;
    x += 3;
    cout<<"Setelah x += 3, x = "<<x<<endl<<endl;
    cout<<"nilai x setelah di tambah 3 adalah
"<<x<<endl;
    x *= 2;
    cout<<"Setelah x *= 2, x = "<<x<<endl<<endl;
    cout<<"nilai x setelah di kalikan 2 adalah
"<<x<<endl;
    x++;
    cout<<"Setelah x++ , x = "<<x<<endl;
    return 0;
}
```

2. Tampilan program :

```
N/latihan/" && g++ 3.cpp -o 3 && "/Users/ar
PEMOGRAMAN/latihan/"3
nilai x = 2
Setelah x += 3, x = 5

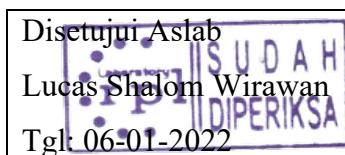
nilai x setelah di tambah 3 adalah 5
Setelah x *= 2, x = 10

nilai x setelah di kalikan 2 adalah 10
Setelah x++ , x = 11
arifin@192 latihan %
```

Gambar 3.2 Latihan praktikum ke-2 (Operator dan *statement* I/O)

3. Analisa Program :

Pada program diatas terdapat variabel $x=2$ yang bertipe data integer, kemudian ditambah dengan 3 dengan operator ($x+=3$). Nilai x menjadi 5, dan program mencetak *cout* “nilai x setelah ditambah 3 adalah ”. Jadi sekarang $x=5$, kemudian x dikali dengan 2 yang menggunakan operator ($x*=2$). Nilai x menjadi 10, setelah itu nilai x di *increment* dengan ($x++$). Dari seimua itu hasilnya akan di tampilkan menggunakan *cout*. Kemudian program berakhir.



III.5 Latihan Praktikum ke-3:

Mencetak *output* pada program operator bitwise.

1. *Source code* :

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

int main(){
    int x , y;

    x=9, y=2;
    cout<<"a. 9 >> 2 = "<<(x>>y)<<endl;
    x=7, y=3;
    cout<<"b. 7 << 3 = "<<(x<<y)<<endl;
    x=16, y=3;
    cout<<"c. 16 ^ 3 = "<<(x^y)<<endl;
    x=5, y=3;
    cout<<"d. 5 & 3 = "<<(x&y)<<endl;
    x=567;
    cout<<"e. ~ 567 ="<<(~x)<<endl;

    return 0;
}
```

2. Tampilan program:

```
N/latihan/`` && g++ z.cpp -o z
PEMOGRAMAN/latihan/"2
a. 9 >> 2 = 2
b. 7 << 3 = 56
c. 16 ^ 3 = 19
d. 5 & 3 = 1
e. ~ 567 =-568
arifin@192 latihan %
```

Gambar 3.3 Latihan praktikum ke-3 (Operator dan *statement* I/O)

3. Analisa program:

Pada program diatas terdapat dua *variabel* x dan y yang menggunakan tipe data integer dan kemudian di berikan nilai *Konstanta* x=9 dan y=2 kemudian di lakukan proses *shift right*, menggeser nilai bilangan biner 9 ke kanan sebanyak 2 *bit* hasilnya akan di tampilkan menggunakan cout. Kemudian untuk nilai *konstanta* x=7 dan y=3 dilakukan proses *shift left*, menggeser nilai dalam bilangan biner 7 ke kiri sebanyak 3 *bit* dan hasilnya akan di tampilkan menggunakan cout. Kemudian untuk *konstanta* x=16 dan y=3 dilakukan proses XOR an hasilkanya akan di tampilkan

menggunakan cout. Kemudian berikutnya untuk *konstanta* $x=5$ dan $y=3$ dilakukan untuk menentukan logika AND sehingga hasilnya 1 dan di cetak menggunakan cout. Dan yang terakhir untuk nilai *konstanta* $x=567$ proses NOT, membalik nilai bit dari suatu *operand* sehingga hasilnya menjadi -567. Dan akan di tampilkan menggunakan cout.

III.6 Tugas Praktikum ke-1:

Buatlah sebuah program untuk menghitung volume bangun ruang (pilih salah satu, seperti pada contoh di bawah ini, beserta hasil tampilan dan analisa program!)

1. Source code :

```
#include <iostream>

using namespace std;

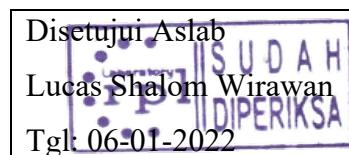
int main(){
    float alas, tinggi, v;
    cout<<"\t\t=====\n";
    cout<<"\t\t Program Perhitungan Volume Limas \n";
    cout<<"\t\t=====\n";
    cout<<endl;
    cout<<endl;
    cout<<"\nMasukkan Luas Alas (cm) : ";
    cin>>alas;
    cout<<"\nMasukkan Tinggi (cm) : ";
    cin>> tinggi;
    v=(alas*tinggi)*1/3;
    cout<<"\nJadi Volume Limas Tersebut Adalah "<<v<<"\n";
    cout<<endl;
}
```

2. Tampilan Program :

```
arifin@MacBook-Air-Arifin: ~ % cd "/Users/arifin/Desktop"
arifin@MacBook-Air-Arifin: ~ % g++ tugas-algo.cpp -o tugas-algo
arifin@MacBook-Air-Arifin: ~ %
=====
Program Perhitungan Volume Limas
=====

Masukkan Luas Alas (cm) : 24
Masukkan Tinggi (cm) : 12.8
Jadi Volume Limas Tersebut Adalah 102.4 cm3
arifin@MacBook-Air-Arifin: ~ %
```

Gambar 3.4 Tugas praktikum ke-1 (Operator dan statement I/O)



Ttd / Paraf

3. Analisa Program

Dari Analisah program ini saya menggunakan *float* untuk bilangan *decimal*. Jadi *user input* bilangan yaitu menggunakan *cin* sebagai *input* dari apa yang di minta program. Jika udah di *input* oleh *user* dari apa yang di minta program maka menampilkannya menggunakan *cout* sebagai hasil dari *inputan user* yang di inginkan.

III.7 Tugas Praktikum ke-2:

Buatlah program untuk mengkonversikan suhu sesuai dengan ketentuan sesuai pada gambar berikut, di mana suhu *celcius* adalah *inputan user*!

1. Source code :

```
#include <iostream>

using namespace std;
int main(){
    float celc, fahr;
    float ream, kelv;
    cout<<"\t\t=====\\n";
    cout<<"\t\t Program Konversi Suhu \\n";
    cout<<"\t\t=====\\n";
    cout << "\\nMasukan Suhu Celcius: ";
    cin >> celc;

    fahr = celc*1.8 + 32;
    ream = (fahr-32)*0.444;
    kelv = (1.25*ream) + 273;

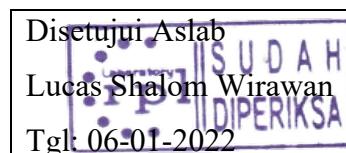
    cout << "\\njadi, " << celc << " derajat celcius\\t= " <<
    fahr << " derajat Fahrenheit";
    cout << "\\n         " << fahr << " derajat Fahrenheit\\t= " <<
    ream << " derajat reamur";
    cout << "\\n         " << ream << " derajat reamur\\t= " <<
    kelv << " derajat kelvin";
}
```

2. Tampilan Program :

```
arifin@MacBook-Air-Arifin:~/Desktop$ g++ tugas-algo.cpp -o tugas-algo && ./tugas-algo
=====
Program Konversi Suhu
=====

Masukan Suhu Celcius: 80
jadi, 80 derajat celcius      = 176 derajat Fahrenheit
176 derajat Fahrenheit      = 63.936 derajat reamur
63.936 derajat reamur       = 352.92 derajat kelvin%
arifin@MacBook-Air-Arifin:~/Desktop %
```

Gambar 3.5 Tugas praktikum ke-2 (Operator dan *statement I/O*)



Ttd / Paraf

3. Analisa Program

Untuk program ini adalah menghitung konversi suhu. Dari program ini *user* akan diminta memasukkan nilai suhu (dalam skala celcius) yang nilainya akan di tumpung pada variabel celc. Dan setelah di masukkan oleh maka hasil akan di tampilkan menggunakan *cout* dari masing-masing konversinya.

III.8 Tugas Praktikum ke-3:

Membuat sebuah program tentang materi yang dipelajari pada bab 2 beserta hasil tampilan dan analisa program !

1. Source code :

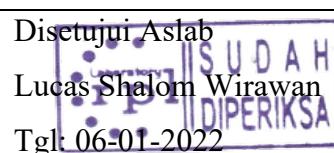
```
#include<iostream>

using namespace std;

main()
{
    int pilihan;
    cout << "selamat datang di program calculator" ;
    cout<<"\n[1] Penjumlahan "<<endl;
    cout<<"[2] perkalian "<<endl;
    cout<<"[3] pengurangan "<<endl;
    cout<<"[4] pembagian "<<endl;
    cout<<"\nmasukkan pilihan anda : ";
    cin>>pilihan;

    if (pilihan==1)

    {
        int nilai1,nilai2, hasil;
        cout<<"\nmasukkan nilai pertama : " ;
        cin>>nilai1;
        cout<<"masukan nilai kedua : " ;
        cin>>nilai2;
        hasil = nilai1+nilai2;
        cout<<"hasil penjumlahan :"<<hasil;
    }
    else if (pilihan==2)
    {
        int nilai1,nilai2,hasil;
        cout<<"\nmasukkan nilai pertama : " ;
        cin>>nilai1;
        cout<<"masukan nilai kedua : " ;
        cin>>nilai2;
        hasil = nilai1*nilai2;
        cout<<"hasil perkalian :"<<hasil ;
    }
    else if (pilihan==3)
    {
        int nilai1,nilai2,hasil;
        cout<<"\nmasukkan nilai pertama : " ;
```



Ttd / Paraf

```

        cin>>nilail;
        cout<<"masukan nilai kedua : " ;
        cin>>nilai2;
        hasil = nilai1-nilai2;
        cout<<"hasil pengurangan :"<<hasil;
    }
    else if (pilihan==4)
    {
        int nilai1,nilai2,hasil;
        cout<<"\nmasukkan nilai pertama : " ;
        cin>>nilai1;
        cout<<"masukan nilai kedua : " ;
        cin>>nilai2;
        hasil = nilai1/nilai2;
        cout<<"hasil pembagian:"<<hasil;
    }

}

```

2. Tampilan Program :

```

arifin@MacBook-Air-Arifin Desktop % g++ tugas_drgs.cpp -o tugas_drgs & ./tugas_drgs
selamat datang di program calculator
=====
[1] Penjumlahan
[2] perkalian
[3] pengurangan
[4] pembagian

masukkan pilihan anda : 4

masukkan nilai pertama : 50
masukan nilai kedua : 5
hasil pembagian:10%
arifin@MacBook-Air-Arifin Desktop %

```

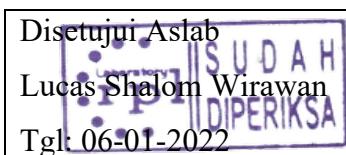
Gambar 3.6 Tugas praktikum ke-3 (Operator dan *statement I/O*)

3. Analisa Program

Dari Analisah program ini pada baris 15 sebagai *inputan* dari *user* jika *user* telah meng *input* sebuah pilihan yang telah di tampilkan oleh cout maka nilai variabel dari sebut pilihan tersebut akan masuk ke dalam kondisi. jika pilihan nilai 1 maka if (pilihan==1) yang akan di jalankan, begitu pula juga Seterusnya. Dan jika user memasukkan pilihan yang tidak sesuai maka program tersebut tidak akan berjalan. Setelah pilihan di pilih maka *user* akan di minta meng *input* nilai variabel yang akan dihitung oleh *user* dan setelah di masukkan akan menampilkan *output* menggunakan cout sebagai hasil yang di minta *user*.

III.9 Kesimpulan

1. Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi.
2. Operator *assignment* (penugasan) adalah operator yang digunakan untuk memasukkan atau mengisikan nilai ke dalam suatu variabel. Dalam bahasa C++ digunakan operator berupa tanda sama dengan (“=”).
3. Operator aritmatika digunakan untuk melakukan sebuah *opearasi* penghitungan atau *aritmatika*. Tugasnya untuk memecahkan masalah matematika.
4. Operator *relasional* atau hubungan digunakan untuk membandingkan hubungan antara dua buah *operand*.
5. Operator *increment* merupakan operator yang dapat menambahkan (menaikkan) suatu nilai.



Ttd / Paraf


BAB IV

Percabangan

Jumlah Pertemuan	:	100 menit
Tujuan Praktikum	:	<ol style="list-style-type: none"> 1. Mampu memahami konsep tentang Kondisi 2. Mampu memahami konsep IF dan switch case 3. Mampu membuat program dengan menambahkan konsep Kondisi.
Alat / bahan	:	<ol style="list-style-type: none"> 1. Seperangkat <i>computer</i>. 2. Perangkat lunak: Dev C++ 3. Modul Praktikum Algoritma dan Pemrograman 2020.

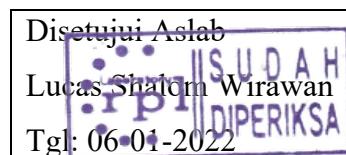
IV.1 Landasan Teori

Salah satu permasalahan yang pasti akan dijumpai dalam pembuatan program adalah suatu percabangan. Percabangan yang dimaksud di sini adalah suatu pemilihan *statement* yang akan dieksekusi di mana pemilihan tersebut didasarkan atas kondisi tertentu.

Dalam kehidupan sehari-hari terkadang kita dihadapkan pada kondisi di mana kita harus memilih suatu tindakan. Sebagai contoh ketika anda ingin menonton film di bioskop. Konsepnya seperti ini: Jika anda memiliki tiket, maka anda boleh masuk untuk menonton film.

Nah, di sana ada sebuah kondisi yang harus terpenuhi yaitu memiliki tiket. Artinya jika anda tidak memiliki tiket (kondisi = FALSE) maka anda tidak diperbolehkan untuk menonton film di bioskop. Kurang lebih itulah contoh nyata konsep percabangan.

Intinya selama kondisi terpenuhi maka program akan dijalankan. Begitu juga dalam membuat program, kita akan sedikit banyak berurusan dengan pilihan. Di sini kita akan mengenal dua fungsi percabangan yaitu *if else* dan *switch case*.



Ttd / Paraf

IV.1.1 Struktur if Satu Kondisi

Struktur ini merupakan struktur yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa. Bentuk umum dari struktur percabangan yang memiliki satu kondisi adalah sebagai berikut :

```
if (kondisi) {
    pernyataan1;
    pernyataan2;
    // kode yang di jalankan jika pernyataan
benar
    .....
}
```

Pernyataan IF di atas mempunyai pengertian, “ Jika kondisi bernilai benar, maka perintah/pernyataan akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”.

Jika ’pernyataan’ yang dijalankan hanya sebaris, maka tanda {} boleh ditiadakan. Seperti di bawah ini :

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

IV.1.2 Struktur if Dua Kondisi

Pada struktur ini terdapat sebuah *statement* khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).

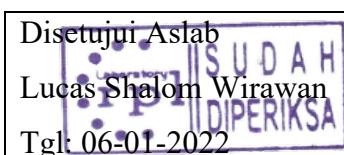
Bentuk umum dari struktur if adalah sebagai berikut :

```
if (kondisi) {
    pernyataan_jika_kondisi_terpenuhi;
    .....
}
else{
    pernyataan_jika_kondisi_tidak_terpenuhi;
    .....
}
```

IV.1.3 Struktur if Tiga Kondisi atau Lebih

Percabangan ini merupakan perluasan dari struktur yang memiliki dua kondisi, yaitu dengan menyisipkan satu atau lebih kondisi ke dalamnya. Untuk struktur yang memiliki tiga kondisi atau lebih, bentuk umum dari struktur ini adalah adalah seperti yang tampak di bawah ini.

```
if (kondisi1) {
```



Ttd / Paraf

```

        statemen_jika_kondisi_terpenuhi;
} else if (kondisi2){
    statemen_jika_kondisi2_terpenuhi;
}
else if (kondisi3){
    statemen_jika_kondisi3_terpenuhi;
}
.....
else {

    statemen_jika_semua_kondisi_diatas_tidak_terpenuhi;
}

```

IV.1.4 Struktur Switch

Selain menggunakan *statemen if*, C++ juga menawarkan untuk dapat melakukan percabangan (pemilihan) dengan menggunakan *statemen switch*. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian. Pernyataan *if...else if* jamak dapat dibangun dengan pernyataan *switch*.

Bentuk umum dari *Switch* :

```

switch (ekspresi)
{
    case konstanta1:
        pernyataan1;
        break;
    case konstanta2:
        pernyataan2;
        break;
    case konstanta3:
        pernyataan3;
        break;
    default:
        pernyataan_lain;
}

```

Hal-hal yang perlu diperhatikan :

1. Di belakang *keyword case* harus diikuti oleh sebuah konstanta, tidak boleh diikuti oleh ekspresi ataupun variabel.
2. Konstanta yang digunakan ber tipe int atau char.
3. Jika bentuknya seperti di atas maka apabila ekspresi sesuai dengan konstanta2 maka pernyataan2, pernyataan3 sampai dengan pernyataan lain dieksekusi. Untuk mencegah hal tersebut, gunakan *keyword break*. Jika *keyword break* digunakan maka setelah pernyataan2 dieksekusi program langsung keluar dari

pernyataan *switch*. Selain digunakan dalam *switch*, keyword *break* banyak digunakan untuk keluar dari pernyataan yang berulang (*looping*).

4. Pernyataan lain (*default*) dieksekusi jika konstanta1 sampai konstanta N tidak ada yang memenuhi *ekspresi*.

IV.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum Algoritma dan Pemrograman.
3. Menjalankan script dari modul.

IV.3 Latihan Praktikum ke-1: (IF-ELSE)

Program untuk mendeteksi bilangan prima dan bilangan non prima.

1. *Source code* :

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

int main(){
    cout<<"-----";
    cout<<"\n\tMENDETEKSI BILANGAN PRIMA DAN NON PRIMA";
    cout<<"\n-----";
    int n,i,cek;
    cout<<"\nMASUKKAN BILANGAN : ";
    cin>>n;
    // 0 dan 1 bukan angka prima
    if (n == 0 || n == 1) {
        cek++;
    }
    else
        for(i=2; i<=n; i++){ //dimulai dari 2 hingga n
            if(n % i == 0){
                cek++;
            }
        }
    if(cek==1){
        cout<< n <<" bilangan prima"<<endl;
    }else{
        cout<< n <<" bukan bilangan prima"<<endl;
    }
}
```



Disetujui Aslab

Lucas Shalom Wirawan

Tgl: 06-01-2022

Ttd / Paraf

2. Tampilan Program :

```
IAH/PRAKTIKUM PEMOGRAMAN/TUGAS 3/"mendeteksi_bilangan_prima_
----- MENDETEKSI BILANGAN PRIMA DAN NON PRIMA -----
MASUKKAN BILANGAN : 45
45 bukan bilangan prima
arifin@MacBook-Air-Arifin TUGAS 3 %
```

Gambar 4.1 Latihan praktikum ke-1: *if else* (percabangan)

3. Analisa Program

Untuk menampilkan sebuah bilangan prima kita menggunakan perulangan *for* untuk mencetak sebuah bilangan dari 2 hingga n, dan juga menggunakan kondisi *if* untuk memulai pengecekan apakah n habis dibagi dengan i melalui perintah $n \% i == 0$. Jika kondisi ini terpenuhi, *update* isi variabel cek menjadi *false* dan langsung hentikan perulangan.

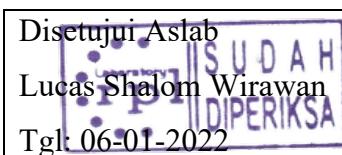
IV.4 Latihan praktikum ke-2: (IF ELSE)

Program untuk menentukan pengelompokan warna-warna pelangi

1. Source code :

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

int main(){
    cout<<"\tWARNA-WARNA PELANGI";
    cout<<"\n1. Merah";
    cout<<"\n2. Jingga";
    cout<<"\n3. Kuning";
    cout<<"\n4. Hijau";
    cout<<"\n5. Biru";
    cout<<"\n6. Nila";
    cout<<"\n7. Ungu";
    cout<<"\n=====";
    cout<<"\nMasukkan warna-warna Pelangi (1/2/3)";
    int warna;
    cout<<endl;
    cin>>warna;
    if (warna==1)
        cout<<"Merah";
    if (warna==2)
        cout<<"Jingga";
    if (warna==3)
        cout<<"Kuning";
    if (warna==4)
        cout<<"Hijau";
    if (warna==5)
        cout<<"Biru";
    if (warna==6)
        cout<<"Nila";
    if (warna==7)
        cout<<"Ungu";
}
```



Ttd / Paraf

2. Tampilan Program :

```
g: ~ /Users/arifin/Documents/DATA KULIAH/
warna_pelangi
WARNA-WARNA PELANGI
1. Merah
2. Jingga
3. Kuning
4. Hijau
5. Biru
6. Nila
7. Ungu
=====
Masukkan warna-warna Pelangi (1/2/3)
1
Merah
arifin@MacBook-Air-Arifin TUGAS 3 %
```

Gambar 4.2 Latihan praktikum ke-2 : *if else* (Percabangan)

3. Analisa Program

Untuk nilai *inputan* yang digunakan mulai dari 1 hingga 7 untuk *inputan* ini di tampung ke dalam variabel warna yang sekarang di sebagai tipe data integer. Dan untuk *outputnya* menggunakan if yaitu satu blok jadi jika pilihan benar maka akan di *eksikusikan*, jika pilihan itu bernilai *false* maka blok perintah tidak di *eksikusikan*.

IV.5 Latihan Praktikum ke-3: (SWITCH CASE)

Buatlah suatu program untuk Bioskop Manarela yang baru buka dikota malang.

Pilihan menu program :

1. Film Antares
2. Film Single
3. Film Mariposa

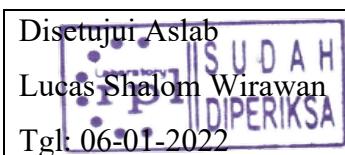
Ketentuan Harga Tiket Bioskop :

- a. Film Antares : Rp. 30000
- b. Film Single : Rp.35000
- c. Film Mariposa : Rp 25000

Setiap pembelian Tiket bioskop pelanggan mendapatkan diskon 10%

1. *Source code* :

```
#include <iostream>
using namespace std;
int main()
{
    int pilihan, harga, jumlah, tiket, total, kembalian,
    pembayaran,b;
```

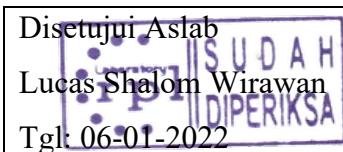


Ttd / Paraf

```

cout<<"\n\tPILIHAN TIKET";
cout<<"\n1. Film Antares";
cout<<"\n2. Film Single";
cout<<"\n3. Film Mariposa";
cout<<"\nSetiap pembelian Tiket bioskop pelanggan mendapatkan diskon 10%";
cout<<"\nmasukkan pilihan :";
cin>>pilihan;
switch (pilihan)
{
    case 1:
        cout<<"\nTIKET ANTRES";
        cout<<"\n=====";
        cout<<"\nMasukkan jumlah tiket :";
        cin>>tiket;
        total = tiket*30000;
        jumlah =total*0.1;
        pembayaran=total-jumlah;
        cout<<"Total :"<<pembayaran;
        cout<<"\nMasukkan pembayaran Rp.";
        cin>>b;
        kembalian =b-pembayaran;
        cout<<"\n-----";
    ;
        cout<<"\n Kembalian anda adalah
Rp."<<kembalian;
        cout<<"\n-----";
    ;
        break;
    case 2:
        cout<<"\nTIKET SINGLE";
        cout<<"\n=====";
        cout<<"\nMasukkan jumlah tiket :";
        cin>>tiket;
        total = tiket*35000;
        jumlah =total*0.1;
        pembayaran=total-jumlah;
        cout<<"Total :"<<pembayaran;
        cout<<"\nMasukkan pembayaran Rp.";
        cin>>b;
        kembalian =b-pembayaran;
        cout<<"\n-----";
    ;
        cout<<"\n Kembalian anda adalah
Rp."<<kembalian;
        cout<<"\n-----";
    ;
        break;
    case 3:
        cout<<"\nTIKET MARIPOSA";
        cout<<"\n=====";
        cout<<"\nMasukkan jumlah tiket :";
        cin>>tiket;
        total = tiket*25000;
        jumlah =total*0.1;
        pembayaran=total-jumlah;
        cout<<"Total :"<<pembayaran;
        cout<<"\nMasukkan pembayaran Rp.";

```



Ttd / Paraf


```

        cin>>b;
        kembalian =b-pembayaran;
        cout<<"\n-----"
";
        cout<<"\n Kembalian anda adalah
Rp."<<kembalian;
        cout<<"\n-----"
";

        break;
default :
        cout<<"tidak ada pilihan";
        cout<<"\n=====";
}

return 0;
}

```

2. Tampilan Program :

```

PILIHAN TIKET
1. Film Antares
2. Film Single
3. Film Mariposa
Setiap pembelian Tiket bioskop pelanggan mendapatkan diskon 10%
masukkan pilihan :1

TIKET ANTRES
_____
Masukkan jumlah tiket :3
Total :81000
Masukkan pembayaran Rp.100000

_____
Kembalian anda adalah Rp.19000
arifin@MacBook-Air-Arifin praktikum 4 %

```

Gambar 4.3 Latihan praktikum ke-3: *switch case* (Percabangan)

3. Analisa Program

Analisa program ini *user* akan meng *input* pilihan tiket filmnya dan disimpan pada variabel pilihan, nantinya nilai dari variabel ini akan di bandingkan pada setiap *case* pada blok *switch*. Apa bila salah satu *case* bernilai *true* maka pernyataan pada *case* tersebut yang di jalankan. *User* juga akan memasukkan jumlah tiket yang ingin di beli dan akan disimpan nilainya pada variabel tiket yang ingin dibeli dan disimpan nilainya pada variabel tiket dan setiap pembelian mendapatkan *discon* 10%. Jika udah meng *input* jumlah tiket maka akan menampilkan total pembayaran. Jika nilai pilihan tersebut *false* maka akan menjalankan *default*. *input* angka yang di minta maka akan di jalankan sampai proses di temukan. Jika nilai pilihan tersebut *false* maka akan menjalankan *default*.

IV.6 Tugas Praktikum ke-1 : (IF-ELSE)

Buatlah program dimana dalam suatu proyek atasan ingin memberikan komisi kepada karyawan dengan ketentuan seperti berikut :

- Jika karyawan lembur dibawah 3 jam maka akan mendapatkan komisi sebesar Rp. 10.000
- Jika karyawan dapat lembur diatas 3 jam maka akan mendapatkan komisi sebesar Rp. 20.000
- Dan jika karyawan lembur diatas 5 jam maka akan mendapatkan komisi sebesar Rp. 50.000

1. Source code :

```
#include <iostream>
using namespace std;
int main(){
    int l=0;
/*Jika karyawan lembur dibawah 3 jam maka akan mendapatkan komisi sebesar Rp. 10.000
Jika karyawan dapat lembur diatas 3 jam maka akan mendapatkan komisi sebesar Rp. 20.000
Dan jika karyawan lembur diatas 5 jam maka akan mendapatkan komisi sebesar Rp. 50.000*/
    cout<<endl;
    cout << "Masukkan jumlah jam lembur : ";
    cin>>l;
    cout<<endl;

    if (l<3){
        int l;
        l=10000;
        cout<<"anda mendapatkan komesi sebesar Rp"<<l;
        cout<<"\n-----";
    }else
        if (l>=3 && l<=4) {
            int l;
            l=20000;
            cout<<"anda mendapatkan komesi sebesar Rp"<<l;
            cout<<"\n-----";
        }else
            if (l>=5) {
                int l;
                l=50000;
                cout<<"anda mendapatkan komesi sebesar Rp"<<l;
                cout<<"\n-----";
            }else
                return 0;
}
```



Ttd / Paraf

2. Tampilan Program :

```
Masukkan jumlah jam lembur : 2
anda mendapatkan komesi sebesar Rp10000
arifin@MacBook-Air-Arifin TUGAS 3 %
```

Gambar 4.4 Tugas praktikum ke-1: *if else* (Percabangan)

3. Analisa Program

Untuk baris yang ke-4 yaitu menggunakan *integer* untuk menjalankan sebuah tipe data angka. Jika kondisi di baris 13 tidak terpenuhi (yang artinya variabel 1 kurang dari 3), maka program akan lanjut ke kondisi *else if* berikutnya. Di sini saya menggabungkan dua buah kondisi pemeriksaan dengan operator logika **&&** (operator AND). Kondisi *if(1>=3 && 1<=4)* hanya akan terpenuhi jika isi variabel berada dalam rentang 3 sampai 4.

IV.7 Tugas Praktikum ke-2: (IF-ELSE)

Buatlah progam di bawah ini menggunakan metode *If-Else* seperti yang sudah kalian pelajari.

1. *Source code* :

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

int main(){
    cout<<"\n-----";
    cout<<"\nMENU MAKANAN WARUNG BISMILLAH";
    cout<<"\n-----";
    cout<<"\n1. Ayam bakar : 10000";
    cout<<"\n2. Ayam goreng : 10000";
    cout<<"\n3. Air mineral : 5000";
    cout<<"\n-----";
    int m;
    cout<<endl;
    cout<<"\nMasukkan menu yang anda inginkan : ";
    cin>>m;
    if(m==1) {
        int j,h;
        cout<<"Berapa Ayam bakar yang anda inginkan : ";
        cin>>j;
        h=j*10000;
        cout<<"\ntotal ayam bakar anda adalah "<<h;
    }else if(m==2) {
        int a,h2;
```

```

cout<<"Berapa Ayam bakar yang anda inginkan :
";
cin>>a;
h2=a*10000;
cout<<"\ntotal ayam goreng anda adalah "<<h2;
}else if(m==3){
    int b,h3;
    cout<<"Berapa Ayam bakar yang anda inginkan :
";
    cin>>b;
    h3=b*5000;
    cout<<"\ntotal ayam bakar anda adalah "<<h3;
}else{
    cout<<"MAAF MENU YANG ANDA TUJU TIDAK VALID";
}

return 0;
}

```

2. Tampilan Program :

```

-----  

MENU MAKANAN WARUNG BISMILLAH  

-----  

1. Ayam bakar : 10000  

2. Ayam goreng : 10000  

3. Air mineral : 5000  

-----  

Masukkan menu yang anda inginkan : 1  

Berapa Ayam bakar yang anda inginkan : 3  

total ayam bakar anda adalah 30000%  

arifin@MacBook-Air-Arifin TUGAS 3 %

```

Gambar 4.5 Tugas praktikum ke-2: *if else* (Percabangan)

3. Analisa Program

Sebuah variabel m sebagai integer. Variabel m ini kemudian di pakai untuk menampung dari perintah *cin>>m* di baris 15. Mulai dari 16 hingga 34 terdapat 3 kali pemeriksaan kondisi. Yakni satu untuk setiap blok *if else*. Dalam setiap kondisi, isi variabel m akan di periksa apakah berisi angka. Jika salah satu terpenuhi, maka *block* kode program yang sesuai akan di eksikusi.

IV.8 Tugas praktikum ke-3: (SWITCH CASE)

Identifikasi kesalahan dan lengkapi program dibawah ini dan tampilan program beserta analisanya. dengan ketentuan berikut :

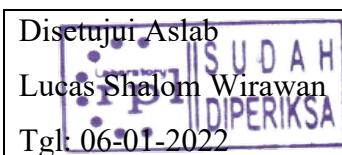
- a. Input indikator dari 1-2 maka akan tercetak “Hari senin dan Selasa memakai seragam putih abu-abu”
- b. Input indikator 3-4 maka akan tercetak “Hari rabu dan kamis memakai seragam batik”
- c. Input indikator 5 maka akan tercetak “Hari Jumat memakai baju pramuka”

1. *Source code :*

```
//Program dari tugas
/*#include <iostream>
using namespace std;

int main(){
    cout<<"\tProgram menentukan hari dan seragam SMA
Garuda"<<endl<<endl;
    cout<<"Masukkan Angka : ";cin>>hari;
    case (hari)
    {
        switch 1 ... 2 :
            cout<<"Hari seni dan selasa menggunakan
seragam putih abu-abu";
            cout<<endl;
            switch 3 ... 4 :
                cout<<"Hari rabu dan kamis menggunakan seragam
batik";
                cout<<endl;
                switch 5:
                    cout<<"Hari jum'at menggunakan seragam
pramuka";
                    cout<<endl;
                    default:
                        cout<<"pilihan tidak tersedia";
                }
            return 0;
}*/
#include <iostream>
using namespace std;

int main(){
    int hari;
    cout<<"\tProgram menentukan hari dan seragam SMA
Garuda"<<endl<<endl;
    cout<<"Masukkan Angka : ";cin>>hari;
    switch (hari)
    {
        case 1 ... 2 :
            cout<<"Hari seni dan selasa menggunakan
seragam putih abu-abu";
```



Ttd / Paraf


```

        cout<<endl;
        break;
    case 3 ... 4 :
        cout<<"Hari rabu dan kamis menggunakan seragam
batik";
        cout<<endl;
        break;
    case 5:
        cout<<"Hari jum'at menggunakan seragam
pramuka";
        cout<<endl;
        break;
    default:
        cout<<"pilihan tidak tersedia";
        break;
    }
    return 0;
}

```

2. Tampilan Program :

```

47 menentukan_seragam
Program menentukan hari dan seragam SMA Garuda

Masukkan Angka : 1
Hari seni dan selasa menggunakan seragam putih abu-abu

```

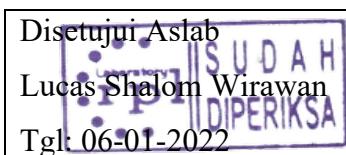
Gambar 4.6 Tugas praktikum ke-3: *switch case* (Percabangan)

3. Analisa Program

Dari Analisah program ini yaitu memperbaiki kesalahan jalannya program pada saat di *running*. Kesalahan terhadap program tersebut yaitu pada case dan switch, karena pada program tersebut terbalik seharusnya *switch* terlebih dahulu baru *case*. Dan pada program tersebut juga tidak ada *break*, jadi saat di *running* akan berlanjut terhadap *output case* selanjutnya.

IV.9 Kesimpulan

1. Struktur *if* satu kondisi ini merupakan struktur yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa.
2. struktur *if* dua kondisi ini terdapat sebuah *statemen* khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).
3. Struktur *if* 3 kondisi/lebih percabangan ini merupakan perluasan dari struktur yang memiliki dua kondisi, yaitu dengan menyisipkan satu atau lebih kondisi ke dalamnya.
4. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian.



Ttd / Paraf

BAB V

Perulangan

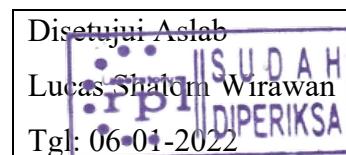
Jumlah Pertemuan	:	100 menit
Tujuan Praktikum	:	<ul style="list-style-type: none"> 1. Mampu memahami konsep perulangan 2. Mampu melakukan perulangan dengan bantuan “For” 3. Mampu melakukan perulangan dengan bantuan “While” 4. Mampu melakukan perulangan dengan bantuan “Do-While” 5. Mampu membuat program berdasarkan perulangan
Alat / bahan	:	<ul style="list-style-type: none"> 3. Seperangkat <i>computer</i>. 4. Perangkat lunak: Dev C++ 5. Modul Praktikum Algoritma dan Pemrograman 2020.

V.1 Landasan Teori

Perulangan data atau yang biasa disebut dengan “looping” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan. Dengan *loop*, suatu proses yang berulang dapat diimplementasikan menggunakan *statement-statement* yang pendek, tanpa harus menuliskan *statement* berulang-ulang.

Dalam bahasa C++ terdapat tiga buah jenis struktur pengulangan, yaitu :

1. Struktur *For*
2. Struktur *While*
3. Struktur *Do-While*
4. Struktur *Nested For*



Ttd / Paraf


V.1.1 Struktur For

Merupakan salah satu bentuk fungsi *looping*, digunakan untuk melaksanakan pernyataan berulang kali terhadap sejumlah nilai yang telah ditetapkan. *Looping* yang dilakukan oleh *for* telah diketahui batas awal, batas kondisi dan perubahannya. Tipe data dari variabel yang akan digunakan sebagai indeks haruslah tipe data yang mempunyai urutan yang teratur, misalnya tipe data int (0,1,2,...) atau *char*('a','b','c',...).

Bentuk umum:

```
for (inisialisasi ; kondisi ; perubahan) {
    Statement;
}
```

VI.1.2 Struktur For dengan Banyak Variabel

Struktur *for* di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu di antaranya akan digunakan sebagai indeks pengulangan.

VI.1.3 Struktur While

Struktur perulangan *WHILE* adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur. Pengulangan hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar). Hal ini berarti jika kondisi yang didefinisikan tidak terpenuhi (bernilai salah) maka *statement-statement* yang terdapat dalam blok pengulangan pun tidak akan pernah dieksekusi oleh program.

Cek Kondisi?
while (true)

Blok kode
yang diulang

Bentuk umumnya :

```
while (kondisi) {
    statemen_statemen_yang_akan_diulang;
    perubahan;
}
```

Algoritma Program :

- Sebuah *while loop* mengecek kondisi
- Selama kondisi yang dicek bernilai *true*, kode di dalam *while loop* di jalankan
- Kondisi di cek lagi
- Proses ini berlanjut hingga saat kondisi bernilai *false*
- Saat kondisi yang di cek bernilai *false*, perulangan berakhir

VI.1.4 Struktur Do-While

Berbeda dengan struktur *while* yang melakukan pemeriksaan kondisi di awal blok perulangan, pada struktur *do-while* kondisi justru ditempatkan di bagian akhir. Hal ini tentu menyebabkan struktur pengulangan ini minimal akan **melakukan satu kali** proses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah).

Blok kode
yang diulang

Cek Kondisi?
while (true)

Bentuk umum :

```
do {
    statemen_yang_akan_diulang;
    perubahan;
} while (kondisi);
```

Algoritma Program :

- Kode di dalam perulangan di jalankan
- kemudian kondisi di cek

- Selama kondisi yang dicek bernilai *true*, kode di dalam *do* di jalankan
- Kondisi di cek lagi
- Proses ini berlanjut hingga saat kondisi bernilai *false*
- Saat kondisi yang di cek bernilai *false*, perulangan berakhir

VI.1.5 Struktur Nested For

Pernyataan *nested for* adalah suatu perulangan *for* di dalam perulangan *for* yang lain. Di dalam penggunaan *nested for*, perulangan yang di dalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang di luar diselesaikan.

Biasanya, *nested loops* digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom. *Loop* luar, biasanya digunakan untuk mendefinisikan baris. Sedangkan *loop* dalam, digunakan untuk mendefinisikan kolom.

Bentuk umum :

```
for (inisialisasi1 ; kondisi1 ; perubahan1)
{
    for (inisialisasi2 ; kondisi2 ; perubahan2)
    {
        Statement;
    }
}
```

V.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum *Algoritma dan Pemrogramman*.
3. Menjalankan *script* dari modul.

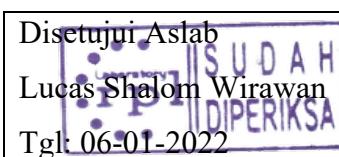
V.3 Latihan Praktikum ke-1: (FOR)

Buatlah program sesuai dengan tampilan program di bawah ini:

1. *Source code* :

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

int main(){
    int n,m,a,b;
    cout<< "a. ";
    for (n=5;n<=15;n++) {
        cout<<"    "<<n;
    }
}
```



Ttd / Paraf

```

        cout<<endl;
        cout<< "b. ";
        for (m=20;m>=11;m--) {
            cout <<"    ";cout<<m;
        }
        return 0;
    }
}

```

2. Tampilan Program :

```

arifin@MacBook-Air-Arifin tugas5 % cd g++-10.1.0-rc1pp-0-10.1-0-0.36.1-0.36.1-1.1.1
PEMOGRAMAN/tugas5/"for1
a.   5   6   7   8   9   10  11  12  13  14  15
b.  20  19  18  17  16  15  14  13  12  11
arifin@MacBook-Air-Arifin tugas5 %

```

Gambar 5.1 Latihan praktikum ke-1 (Perulangan for)

3. Analisa Program

Untuk Analisah dari program ini yaitu menampilkan *output* 5 sampai dengan 15 dan 20 sampai dengan 11. Untuk tampilan *output* 5-15 yaitu menggunakan *for*, di mana *for* tersebut terdapat kondisi $n \leq 15$ dan nilai variabel awal di beri nilai 5. Pada bagian *modifier* dilakukan penaikan (*increment*) maka perulangan akan terus berjalan sehingga kondisi bernilai benar. Sedangkan untuk tampilan *output* 20-11 yaitu menggunakan *for* di mana *for* tersebut terdapat kondisi $n \leq 11$ dan nilai variabel awal di beri nilai 20. Pada bagian *modifier* dilakukan penurunan (*decrement*) maka perulangan akan terus berjalan sehingga kondisi bernilai benar.

V.4 Latihan praktikum ke-2: (WHILE-DO WHILE)

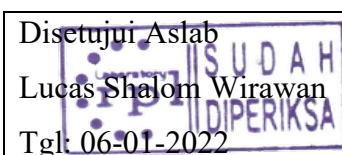
Buatlah program sesuai dengan tampilan program dibawah ini:

1. *Source code* :

```

#include <iostream>
using namespace std;
int main()
{
    int P;
    P = 1;
    cout<<"\tPERULANGAN WHILE";
    while (P<14)
    {
        cout<<"\nX. : "<<P<<endl;
        cout<<"Belajar perulangan while C++"<<endl;
        P+=3;
    }
    return 0;
}

```



Ttd / Paraf

2. Tampilan Program :

```
ifin/Documents/DATA KULIAH/PRAKTIKUM PEMOG
PERULANGAN WHILE
X. : 1
Belajar perulangan while C++

X. : 4
Belajar perulangan while C++

X. : 7
Belajar perulangan while C++

X. : 10
Belajar perulangan while C++

X. : 13
Belajar perulangan while C++
arifin@MacBook-Air-Arifin latihan6 %
```

Gambar 5.2 Latihan Praktikum ke-2 (*while, do-while*)

3. Analisa Program

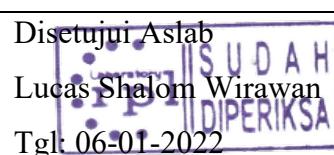
Dari program ini pada baris enam terdapat *inisialisasi* $p=1$, dan pada baris delapan terdapat kondisi yang akan melakukan perulangan sampai kondisi tersebut lebih besar dari empat belas. Selama kondisi belum lebih besar dari 14 maka akan melakukan perulangan seterusnya sampai kondisi di atas empat belas dan perulangan akan berhenti. variabel a dengan *inisialisasi* 5. Dan pada baris ke 20 terdapat juga perubahan $a*=5$. Dan diberikan kondisi $a<=3125$ pada baris 23. Untuk baris yang ke 24 itu terdapat variabel p dengan *inisialisasi* ‘A’. Dan pada baris ke 29 terdapat juga perubahan $p+=2$. Dan diberikan kondisi $p<='K'$ pada baris 26. Untuk baris yang ke 32 itu terdapat variabel b dengan *inisialisasi* ‘Z’. Dan pada baris ke 37 terdapat juga perubahan $b-=3$. Dan diberikan kondisi $b>='N'$ pada baris 34. Jika kondisi bernilai *true* maka kode di dalam *do* akan di jalankan dan kondisi akan di cek lagi sampai kondisi bernilai *false*. Saat kondisi bernilai *false* maka perulangan akan berhenti.

V.5 Latihan praktikum ke-3: (NESTED LOOP)

Buatlah program dengan hasil sebagai berikut :

1. *Source code* :

```
#include <iostream>
using namespace std;
int main() {
    int a,b,z=1;
    int y=10;
    cout<<"A. "<<endl;
    for (a=1;a<=3;a++) {
        for(b=1;b<=3;b++) {
```



Ttd / Paraf

```

        cout<<z<<"\t";
        z=z+2;
    }
    cout<< endl;
}
cout<<endl;
cout<<"B. "<<endl;
for (a=1;a<=5;a++) {
    for(b=5;b>=a;b--) {
        cout<<b<<" ";
    }
    cout<< endl;
}
cout<<endl;
cout<<"C. "<<endl;
for (a=1;a<=5;a++) {
    for(b=10;b>=y;b-=2) {
        cout<<y<<" ";
    }
    y-=2;
    cout<< endl;
}
cout<<endl;
cout<<"D. "<<endl;
for (a=5;a<=9;a+=2) {
    for(b=9;b>=a;b--) {
        cout<<a<<" ";
    }
    cout<< endl;
}

return 0;
}

```

2. Tampilan Program :

```

/Documents/DATA KULIAH/PRAKTIKUM PEMOGRAMAN/tugas
A.
1      3      5
7      9      11
13     15     17

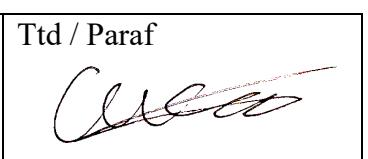
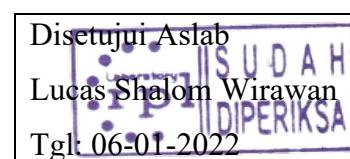
B.
5 4 3 2 1
5 4 3 2
5 4 3
5 4
5

C.
10
8 8
6 6 6
4 4 4 4
2 2 2 2 2

D.
5 5 5 5 5
7 7 7
9
arifin@MacBook-Air-Arifin tugas 7 %

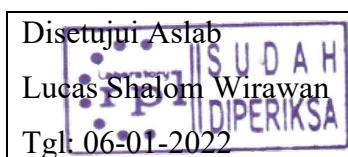
```

Gambar 5.3 Latihan Praktikum ke-3 (*Nested loop*)



3. Analisa Program

Dari program di atas terdapat *variable* a, b, z=1, dan y=10 yang ber *type* integer. Pada program di baris 6 sampai 13, terdapat dua *for* yaitu *for* pertama yang terdapat program *for* (*a*=1;*a*<=3;*a*++) dan *for* kedua *for* (*b*=1 ;*b*<=3 ;*b*++) dari *for* pertama yang digunakan untuk *output vertical*, dan sedangkan untuk *for* kedua mengerjakan *output horizontal* yang akan meng *outputkan* dari variabel z dari variabel z tersebut yang terdapat operator *z=z+2*. Pada saat di jalankan maka pertama yang dijalankan adalah *for* yang pertama di mana saat *for* pertama bernilai *true* maka akan di lanjutkan dengan *for* yang kedua dari setiap kolom tersebut *outpunya* akan di tambah 2 yang sudah di operator kan. Untuk baris yang 15 sampai 21 juga terdapat 2 *for* di mana *for* yang pertama yaitu *for* (*a*=1;*a*<=5;*a*++) sedangkan yang *for* kedua yaitu *for*(*b*=10;*b*>=y;*b*-=2). Dari program pada *for* pertama sebagai *vertical* sedangkan *for* kedua sebagai *horizontal*. Dari perulangan *for* kedua yang sebagai kolom di mana setiap kolom akan berubah-ubah dengan berkurangnya nilai, jadi setiap kolom akan berkurang seperti halnya pada gambar di atas di baris pertama yaitu *output* nya 5 sampai 1 sedangkan pada baris kedua 5 sampai 2 dst. Untuk baris yang 22 sampai baris 27 juga terdapat 2 *for* di mana dalam *for* pertama yaitu *for* (*a*=1;*a*<=5;*a*++) pada perulangan *for* pertama terdapat operator *y-=2* sedangkan pada baris yang kedua yaitu *for*(*b*=10;*b*>=y;*b*-=2). Saat program dijalankan maka yang pertama di *eksikusikan* adalah *for* pertama dan selanjutnya *for* yang kedua. Pada program di atas setiap kolom akan berubah ubah dan setiap perulangan akan terjadi pengurangan, jadi saat dijalankan *output* yang pertama di tampilan adalah nilai 10 di mana pada *for* kedua terdapat *inisialisasi* *b*=10 dan akan terjadi perubahan kolom di mana pada *for* kedua terdapat perubahan *b-=2* dan saat terjadi perulangan akan berkurang 2 yang di mana di *for* pertama terdapat perubahan *y-=2* di setiap terjadi perulangan. Untuk baris yang ke 32 sampai 38 juga terdapat 2 *for* yaitu *for* pertama yaitu *for* (*a*=5;*a*<=9;*a*+=2) sedangkan *for* yang kedua *for* (*b*=9 ;*b*>=a ;*b*--). Saat program dijalankan maka yang pertama di *eksikusikan* adalah *for*



pertama dan selanjutnya *for* yang kedua. Dari *for* pertama sebagai *vertical* dan *for* yang kedua horizontal. Di *for* pertama terdapat *inisialisasi* $a=5$ dan kondisi $a \leq 9$ dan perubahannya $a+=2$ jadi saat di jalankan program tersebut makan di setiap baris akan bertambah 2 dan setiap kolom akan berubah dari 5 ke 9.

V.6 Tugas Praktikum ke-1: (FOR)

Buatlah program untuk menentukan bilangan ganjil mulai dari 60 hingga data ke n. Data masukan adalah data ke n. Lengkapi kode dengan benar dan tampilannya beserta analisanya.

1. *Source code :*

```
#include <iostream>
using namespace std;
int main () {
    /*INI ADALAH PROGRAM YANG BELUM DI PERBAIKI
    int bil, a;
    cout<<"menampilkan banyak data 60 s/d ";cin>>bil;
    cout<<endl;
    cout <<"bilangan ganjil antara 60 s/d "<<<<" adalah
    : "<<endl;
        for (int i=a;a<=100;i++) {
            if (i>=60){
                if(i%2!=0){
                    cout <<a<<;
                }
            }
        }*/
    //INI ADALAH PROGRAM YANG SUADAH DI PERBAIKI
    int bil, a;
    cout<<"menampilkan banyak data 60 s/d ";cin>>bil;
    cout<<endl;
    cout <<"bilangan ganjil antara 60 s/d "<<bil<<""
    adalah : "<<endl;
        for (int i=60;i<=bil;i++) {
            if (i%2==1){
                cout <<i<<" ";
            }
        }
    cout <<endl;
    return 0;
}
```



Ttd / Paraf


2. Tampilan Program :

```
ROLIAN/PRAKTIKUM PEMrograman/tugas5/ - dd g++ tglv.cpp & tglv dd /o
ers/arifin/Documents/DATA KULIAH/PRAKTIKUM PEMrograman/tugas5/"for4
menampilkan banyak data 60 s/d 100

bilangan ganjil antara 60 s/d 100 adalah :
61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
arifin@MacBook-Air-Arifin tugas5 %
```

Gambar 5.4 Tugas Praktikum ke-1 (Perulangan for)

3. Analisa Program

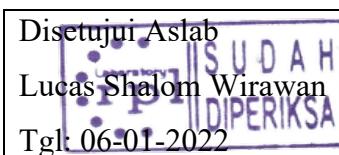
Di dalam program ini di lakukan perbaikan kesalahan dalam melakukan program. Untuk baris yang ke delapan mendapat perbaikan dari “<<<<” menjadi “<<bil<<” untuk menampilkan hasil *inputan*. Dan untuk yang ke sembilan itu terdapat perubahan variabel dari *for* (int i=a;a<=100;i++) menjadi *for* (int i=60;i<=100;i++) jadi untuk baris kesepuluh itu di hapus. Dan untuk baris kesebelas juga terdapat perubahan dari *if* (i%2!=0) menjadi *if* (i%2==1) dan ini adalah untuk menentukan bilangan ganjil. Dan untuk *cout* dalam *if* itu terdapat perubahan dari *cout* <<a<<; menjadi *cout* <<i<<” “;

V.7 Tugas Praktikum ke-2: (WHILE, DO-WHILE)

Lengkapi kode dengan benar dan tampilannya beserta analisanya.

1. Source code :

```
#include <iostream>
using namespace std;
int main()
{
    //program yang salah
    /*char A;
    int B=2,C=3;
    while (A<='L', B<=10, C<=11)
    {
        cout <<"Nilai A = " <<A<<endl;
        A++;
        cout <<"Nilai B = " <<B<<endl;
        B+=2;
        cout <<"Nilai C = " <<C<<endl;
        C+=2;
        cout<<endl;
    }*/
    //program yang benar
    char A='H';
    int B=2,C=3;
    while (A<='L')
    {
        cout <<"Nilai A = " <<A<<endl;
        A++;
        cout <<"Nilai B = " <<B<<endl;
        B+=2;
        cout <<"Nilai C = " <<C<<endl;
    }
}
```



Ttd / Paraf

```

        C+=2;
        cout<<endl;
    }

}

```

2. Tampilan Program :

```

arifin@MacBook-Air-Arifin tugas6 %

```

Gambar 5.5 Tugas praktikum ke-2 (*while, do-while*)

3. Analisa Program

Di dalam program ini yaitu melakukan perbaikan di mana tipe data *char* terdapat variabel “A” dan di variabel A tersebut di beri tambahan menjadi “A='H’ ” pada tipe data *char* dan sebagai *inisialisasi* atau nilai awal. Untuk baris yang delapan di dalam kondisi *while* terdapat 3 kondisi yaitu (A<='L’, B<=10, C<=11) dan pada kondisi tersebut di sisakan hanya satu kondisi yaitu (A<='L’). jadi perubahannya menjadi *while* (A<='L’).

V.8 Tugas Praktikum ke-3: (NESTED FOR)

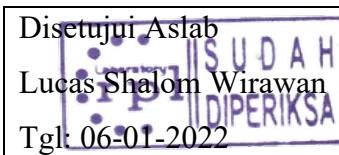
Buatlah program seperti dibawah ini kemudian analisalah progam tersebut.

1. *Source code* :

```

#include <iostream>
using namespace std;
int main () {
    int a,b,c,d;
    for (a=0;a<=9;a++) {
        for (b=0;b<=6;b++) {
            c=a+b;
            cout<<"("<<a<<","<<c<<") ";
        }
        cout<< endl;
    }
}

```



Ttd / Paraf

2. Tampilan Program :

```
uments/DATA KULIAH/PRAKTIKUM PEMOGRAMAN/tugas 7/"t
(0,0)(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)
(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)(1,7)
(2,2)(2,3)(2,4)(2,5)(2,6)(2,7)(2,8)
(3,3)(3,4)(3,5)(3,6)(3,7)(3,8)(3,9)
(4,4)(4,5)(4,6)(4,7)(4,8)(4,9)(4,10)
(5,5)(5,6)(5,7)(5,8)(5,9)(5,10)(5,11)
(6,6)(6,7)(6,8)(6,9)(6,10)(6,11)(6,12)
(7,7)(7,8)(7,9)(7,10)(7,11)(7,12)(7,13)
(8,8)(8,9)(8,10)(8,11)(8,12)(8,13)(8,14)
(9,9)(9,10)(9,11)(9,12)(9,13)(9,14)(9,15)
arifin@MacBook-Air-Arifin tugas 7 %
```

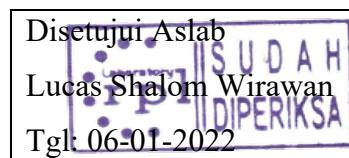
Gambar 5.6 Tugas praktikum ke-3 (*Nested loop*)

3. Analisa Program

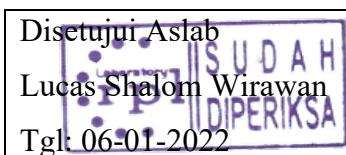
Dari program di atas terdapat *variable* a, b, dan c yang bertipe integer. Dan pada program tersebut menggunakan 2 *for*, di mana *for* per pertama terdapat *for* (a=0;a<=9;a++) sedangkan *for* kedua *for*(b=0; b<=6;b++) dan terdapat operator di dalam *for* kedua c=a+b di mana saat terjadi perulangan juga akan dijumlahkan dan di *outputkan* dalam cout<<"(<<a<< ,<< c<<)". Pada saat dijalankan maka yang dieksikusikan pertama adalah *for* pertama yang sebagai *vertical* dan selanjutnya meng eksikusikan *for* kedua yang sebagai horizontal di mana kedua *for* tersebut akan berlanjut jika bernilai *true*.

V.9 Kesimpulan

1. Perulangan data atau yang biasa disebut dengan “looping” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan.
2. Dalam bahasa C++ terdapat tiga buah jenis struktur pengulangan, yaitu : Struktur For, Struktur While, Struktur Do-While, Struktur Nested For.
3. Struktur for di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu diantara-Nya akan digunakan sebagai indeks pengulangan.
4. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan.
5. Struktur perulangan WHILE adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur.
6. Pengulangan hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar).



7. Berbeda dengan struktur *while* yang melakukan pemeriksaan kondisi di awal blok perulangan, pada struktur *do-while* kondisi justru ditempatkan di bagian akhir.
8. Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan.
9. Pernyataan *nested for* adalah suatu perulangan *for* di dalam perulangan *for* yang lain.
10. *Nested loops* digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom.



Ttd / Paraf

BAB VI

Fungsi Void dan Non-Void

Jumlah Pertemuan	:	240 menit
Tujuan Praktikum	:	<ol style="list-style-type: none"> 1. Mampu memahami konsep Fungsi Void dan Non-Void 2. Mampu membuat program dengan bantuan Fungsi Void dan Non-Void
Alat / bahan	:	<ol style="list-style-type: none"> 1. Seperangkat <i>computer</i>. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrograman 2020.

VI.1 Landasan Teori

VI.1.1 Non Void (Fungsi dengan Nilai Kembalian)

Fungsi merupakan *statement* yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu. Intinya kita ingin membagi-bagi program dari program utama (main program) sehingga program dapat lebih rapih dan juga lebih terstruktur dan dapat dijalankan berulang kali.

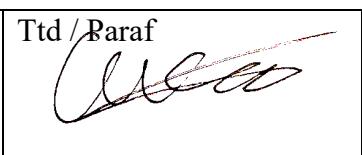
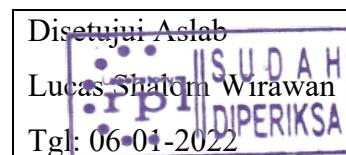
Dalam sebuah program C++ minimal kita memiliki 1 *method*, karena pada dasarnya main() juga merupakan sebuah *method*. *Method* main adalah program utama kita.

Jenis-jenis fungsi ada dua yaitu:

1. *Non void* adalah fungsi dengan nilai balik atau fungsi dengan *return value* ketika fungsi dipanggil.
2. *Void* adalah fungsi yang tidak mengembalikan nilai atau fungsi yang tidak menghasilkan nilai balik ketika fungsi dipanggil.

Keuntungan memakai fungsi :

- Menguraikan tugas pemrograman rumit menjadi langkah-langkah yang lebih sederhana atau kecil.
- Mengurangi duplikasi kode (kode yang sama ditulis berulang-ulang) dalam program.



Fungsi non-void disebut juga *function*. Disebut *non-void* karena mengembalikan nilai kembalian yang berasal dari keluaran hasil proses *function* tersebut .

Ciri-ciri dari jenis fungsi *non-void* adalah sebagai berikut:

- Ada *keyword return*
- Ada tipe data yang mengawali fungsi
- Tidak ada *keyword void*
- Memiliki nilai kembalian

Bentuk umumnya:

```
tipe_data_kembalian nama_fungsi (parameter1,
parameter2,...)
{
    /* Badan Fungsi */
    statemen_yang_akan_dilakukan;
    return nilai_yang_akan_dikembalikan;
}
```

Dari bentuk umum tersebut, yang perlu dijelaskan, yaitu:

a. **Tipe_nilai_balik**

Tipe nilai balik adalah tipe data dari nilai yang akan dikembalikan oleh fungsi. Khusus untuk Fungsi yang tidak ditulis tipenya maka akan dianggap bertipe int atau *char*.

b. **Nama_fungsi**

Nama fungsi adalah nama yang akan digunakan dalam pemanggilan. Pemanggilan fungsi harus disesuaikan dengan daftar parameter yang disertakan pada saat pendefinisian fungsi.

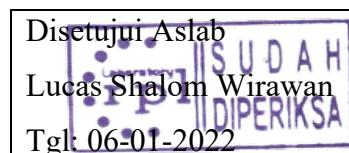
c. **Parameter**

Fungsi bisa saja memiliki satu atau lebih parameter. Parameter itu sendiri merupakan variabel berisi nilai yang akan disertakan ke dalam proses yang dilakukan oleh fungsi. Jika tak ada parameter, judul fungsi berupa:

```
tipe_nilai_balik nama_fungsi()
```

d. **Badan fungsi**

Kode atau *statemen-statemen* yang akan dilaksanakan di dalam fungsi guna menyelesaikan tugas spesifik tertentu. Untuk



bagian ini harus diawali dengan tanda { dan diakhiri dengan tanda }.

1. Fungsi dengan parameter

Fungsi dengan parameter yang mana parameter ini akan digunakan sebagai nilai masukkan (*input*) dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai kembalian (*return value*).

Secara teori, parameter yang terdapat pada pendefinisian sebuah fungsi disebut dengan parameter formal, sedangkan parameter yang terdapat pada saat pemanggilan disebut dengan parameter aktual.

2. Fungsi tanpa parameter

Fungsi tanpa parameter bisa didefinisikan sebagai fungsi yang setiap kali pemanggilan fungsi tersebut, hasil yang didapatkan selalu bernilai tetap.

VI.1.2 Void (Fungsi tanpa Nilai Balik)

C++ tidak mengenal istilah prosedur (*procedure*) seperti pada saat kita melakukan pemrograman dengan menggunakan bahasa Pascal. Dalam bahasa Pascal, prosedur didefinisikan sebagai suatu proses yang tidak mengembalikan nilai. Untuk melakukan hal ini di dalam C++, maka harus membuat suatu fungsi dengan tipe *void*, yang berarti tidak memiliki nilai balik (*return value*).

Ciri-ciri dari jenis fungsi *void* adalah sebagai berikut:

- Secara umum tidak ada *keyword return*.
- Tidak adanya tipe data di dalam deklarasi fungsi.
- Menggunakan *keyword void*.

Bentuk umum dari pembuatan fungsi tanpa nilai balik ini adalah seperti yang terlihat di bawah ini.

```
void nama_fungsi (parameter1, parameter2, ...)
{
    Statement_yang_akan_dilakukan;
    ....
}
```

Dari fungsi *void*, kita tidak bisa mengembalikan nilai apa pun, tapi kita bisa mengembalikan sesuatu selain nilai. Beberapa diantaranya seperti di bawah ini.

a. *Return* di fungsi *void*

Fungsi *void* tidak dapat mengembalikan nilai apa pun. Tapi kita bisa menggunakan pernyataan *return*. *return* digunakan untuk menunjukkan bahwa fungsi tersebut dihentikan.

b. Fungsi *void* dapat mengembalikan fungsi *void* lainnya

Fungsi *void* dapat me *return* kan fungsi lain, dengan syarat fungsi tersebut harus fungsi *void* juga.

VI.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum Algoritma dan Pemrogramman.
3. Menjalankan script dari modul.

VI.3 Latihan Praktikum ke-1: (FUNGSI NON-VOID)

Membuat program memilih perhitungan luas dan keliling lingkaran dengan menerapkan fungsi dengan parameter dan fungsi tanpa parameter.

1. *Source code* :

```
#include <iostream>
using namespace std;
//dengan parameter
double luas (int r){//int r: parameter formal yang diambil dari inputan
int hasil;
const float pi = 3.14;
hasil = pi * r * r;
return hasil;
}
//tanpa parameter
double keliling(){
int d=8;
const float pi = 3.14;
return (pi*d);
}

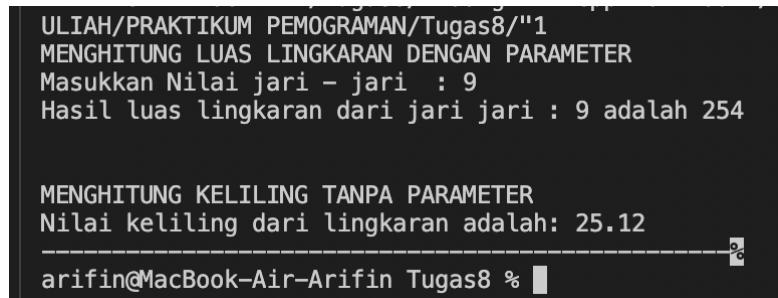
int main(){
int r;
cout<<"MENGHITUNG LUAS LINGKARAN DENGAN PARAMETER"<<endl;
cout<<"Masukkan Nilai jari - jari : ";
cin>>r;
cout<<"Hasil luas lingkaran dari jari jari : "<<r<<" adalah "<<luas(r)<<endl;
cout<<"\n"<<endl;
```

```

cout<<"MENGHITUNG KELILING TANPA PARAMETER"<<endl;
cout<<"Nilai keliling dari lingkaran adalah:
"<<keliling()<<endl;
cout<<"-----";
";
return 0;
}

```

2. Tampilan Program ::



```

ULIAH/PRAKTIKUM PEMOGRAMAN/Tugas8/"1
MENGHITUNG LUAS LINGKARAN DENGAN PARAMETER
Masukkan Nilai jari - jari : 9
Hasil luas lingkaran dari jari jari : 9 adalah 254

MENGHITUNG KELILING TANPA PARAMETER
Nilai keliling dari lingkaran adalah: 25.12
-----@arifin@MacBook-Air-Arifin Tugas8 %

```

Gambar 6.1 Latihan Praktikum ke-1 (fungsi non-void)

3. Analisa Program

Di dalam program tersebut menggunakan 2 fungsi non *void*, yang satu menggunakan parameter dan satunya tidak menggunakan parameter. Pada program luas menggunakan parameter sedangkan keliling tidak menggunakan parameter. Untuk program yang menggunakan parameter harus ada *inputan* yang masuk ke dalam program. Pada saat program di jalankan *user* akan di minta untuk meng *input* nilai variabel dari nilai jari-jari. Dan saat udah di *input* oleh *user* program tersebut akan menuju ke fungsi yang di mana terdapat tipe data *double* yaitu luas(int r). Dan selanjutnya yaitu akan dilakukan proses perhitungan luas dan nilai akan dikembalikan menggunakan *return*. Dan akan di tampilkan luas dari jari-jari yang telah *user input* menggunakan luas (r).

VI.4 Latihan Praktikum ke-2: (FUNGSI NON-VOID)

Menampilkan NAMA, NIM menerapkan fungsi dengan parameter,
NAMA, NIM kalian menggunakan fungsi tanpa parameter!

1. *Source code :*

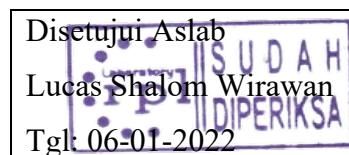
```
#include <iostream>
using namespace std;
string bio(string nama) {
    string a;
    a=nama;
    return a;
}
int induk(int nim) {
    int a;
    a= nim;
    return a;
}
int data() {
    cout<<"\nNama : Mohammad harifin";
    cout<<"\nNim : 2118131";
}

int main(){
string nama;
int nim;
cout<<"Masukkan Nama : ";
getline(cin,nama);
cout<<"Masukkan Nim : ";
cin >>nim;
cout<<"\n======"<<endl;
cout<<"Hasil dengan Menggunakan Parameter :";
cout<<bio(nama)<<" "<<induk(nim);
cout<<endl;
cout<<"======"<<endl;
cout<<"Nama dan Nim tanpa parameter anda adalah :";
cout<< data();
cout<<"\n======"<<endl;
return 0;
}
```

2. Tampilan Program ..

```
Masukkan Nama : MOHAMMAD HARIFIN
Masukkan Nim : 2118131
=====
Hasil dengan Menggunakan Parameter :MOHAMMAD HARIFIN 2118131
=====
Nama dan Nim tanpa parameter anda adalah :
Nama : Mohammad harifin
Nim : 21181311
=====
arifin@MacBook-Air-Arifin Tugas8 %
```

Gambar 6.2 Latihan Praktikum ke-2 (fungsi non-void)



Disetujui Aslab

Lucas Shalom Wirawan

Tgl: 06-01-2022

Ttd / Paraf

3. Analisa Program

Pada Analisah dari program ini saat di jalankan *user* akan meng *input* nama dan nim, yang diberikan variabel nama dan nim pada parameter dan akan di *output* kan pemanggilan dengan menggunakan bio(nama) dan induk (nim) yang sudah di dapatkan. Dan pada program yang tidak menggunakan parameter akan dilakukan pemanggilan fungsi yang telah ditetapkan.

VI.5 Latihan Praktikum ke-3: (FUNGSI NON-VOID)

Membuat sub-rutin menentukan nilai dari fungsi $f(x) = x^4 - 20x + 19$ nilai x adalah inputan user!

1. Source code :

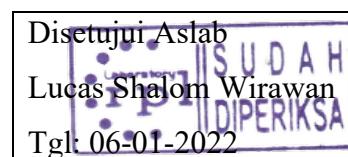
```
#include <iostream>
using namespace std;
int pangkat(int x)
{
    int pangkat, hasil;
    pangkat = x * x * x * x;
    hasil=pangkat-(20*x)+19 ;
    return hasil;
}
int main()
{
    int x;
    cout<<"Masukkan nilai X: ";
    cin>>x;
    cout<<"\nHasil dari f(x)=x^4-20 x+19 =
"<<pangkat(x)<<endl;
    cout<<"-----";
    return 0;
}
```

2. Tampilan Program :.

```
GRAMAN/Tugas8/" && g++ 3.cpp -o 3 && "/
RAMAN/Tugas8/"3
Masukkan nilai X: 7

Hasil dari f(x)=x^4-20 x+19 = 2280
-----
arifin@MacBook-Air-Arifin Tugas8 %
```

Gambar 6.3 Latihan Praktikum ke-3 (fungsi non-void)



Disetujui Aslab
Lucas Shalom Wirawan
Tgl: 06-01-2022

Ttd / Paraf


3. Analisa Program

Pada Analisa dari program di atas , di mana *user* meng *input* nilai variabel x. selanjutnya saat *user* sudah meng *input* nilai x maka akan di tampung menggunakan fungsi dari parameter yaitu int x . di dalam int x tersebut terdapat pangkat 4 dari nilai x yang di *input* kan *user*. Saat sudah di proses di temukannya hasil maka akan di kembalikan nilainya dengan *return*. Dan di tampilkan pada main() menggunakan panggilan pangkat (x).

VI.6 Tugas Rumah ke-1: (FUNGSI VOID)

Buatlah program inputan dinamis dengan bentuk perulangan seperti tampilan dibawah ini serta beri Analisa programnya!

1. *Source code* :

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

void Perulangan(int tgl,string bln,int thn ,int perul){
    int tanggal;
    for(int i=1,tanggal=tgl; i<=perul;i++,tanggal++) {
        cout<<tanggal<<"-"<<bln<<"-"<<thn<<endl;
    }
}
int main(){
    int tgl,thn,perul;
    string bln;
    cout<<" Masukkan Tanggal :";
    cin>>tgl;
    cout<<" Masukkan bulan :";
    cin>>bln;
    cout<<" Masukkan Tahun :";
    cin>>thn;
    cout<<"-----"<<endl;
    cout<<"Masukkan Banyak Perulangan : ";
    cin>>perul;
    Perulangan(tgl,bln,thn,perul);
}
```



Ttd / Paraf

2. Tampilan Program :

```
EMOGRAMAN/tugas9/" && g++ tugas1.cpp -o tuga
ULIAH/PRAKTIKUM PEMOGRAMAN/tugas9/"tugas1
Masukkan Tanggal :10
Masukkan bulan :januari
Masukkan Tahun :2021
-----
Masukkan Banyak Perulangan : 10
10-januari-2021
11-januari-2021
12-januari-2021
13-januari-2021
14-januari-2021
15-januari-2021
16-januari-2021
17-januari-2021
18-januari-2021
19-januari-2021
arifin@MacBook-Air-Arifin tugas9 %
```

Gambar 6.4 Tugas Praktikum ke-1 (fungsi void)

3. Analisa Program

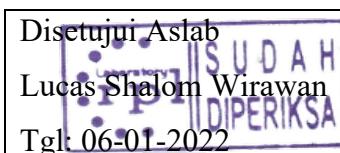
Pada Analisa program di atas adalah menggunakan *for* untuk melakukan seperti pada gambar di atas. Saat program di jalankan dimana *user* akan meng *input* tanggal, tahun, banyak perulangan ,yang menggunakan tipe data integer sedangkan bulan menggunakan tipe data *string*. Saat sudah meng *input* maka akan di panggil fungsi *void* dengan nama *perulangan()* yang menggunakan parameter .Pada *void* perulangan tersebut terdapat *for* yang akan melakukan perulangan dengan *inisialisasi* *i=1* dan *tanggal=tgl* sedangkan kondisi *i<=perul* dan akan melakukan *increment* pada nilai variabel *n i* dan *tanggal*. Pada *for* tersebut akan melakukan perulangan sampai kondisi tersebut terpenuhi. Dan untuk *output* yang di lakukan perulangan yaitu *cout<<tanggal<<"-“ <<bln<<"-“ <<thn<<endl;*.

VI.7 Tugas Rumah ke-2: (FUNGSI VOID)

Buat program membuat persegi Panjang bilangan, jumlah Panjang dan lebar merupakan inputan user dengan menerapkan void beri Analisa programnya,!

1. *Source code :*

```
#include <iostream>
using namespace std;
void persegi(int panjang, int lebar) {
    for(int x=1;x<=panjang;x++) {
        for(int y=1;y<=lebar;y++) {
            cout<<x<<" ";
```



Ttd / Paraf

```

        }
        cout<<endl;
    }

int main(){
    int panjang,lebar;
    cout<<"=====Membuat Persegi Panjang
Bilangan=====<<endl;
    cout<<"Masukkan Panjang : ";
    cin>>panjang;
    cout<<"Masukkan Lebar :";
    cin>>lebar;
    cout<<"-----"=<<endl;
    persegi(panjang,lebar);
}

```

2. Tampilan Program :

```

$ cd PEMOGRAMAN/tugas9/ && g++ tugas9.cpp -o tugas
$ ./tugas
=====Membuat Persegi Panjang Bilangan=====
Masukkan Panjang : 8
Masukkan Lebar :5
-----
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
8 8 8 8
arifin@MacBook-Air-Arifin tugas9 %

```

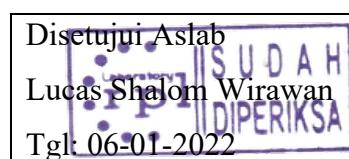
Gambar 6.5 Tugas Praktikum ke-2 (fungsi void)

3. Analisa Program

Pada Analisah program di atas yaitu menggunakan 2 *for* pada fungsi *void* dimana *for* pertama (*for(int x=1;x<=panjang;x++)*) yang sebagai *vertical* sedangkan untuk *for* kedua (*for(int y=1;y<=lebar;y++)*) yang sebagai horizontal. Pada saat program dijalankan *user* akan meng *input* nilai *varibel* Panjang dan lebar yang menggunakan tipe data integer. Saat sudah meng *input* di mana fungsi *void* *persegi()* di panggil seperti pada baris 20. Dari *inputan user* tersebut akan di eksekusikan oleh dua *for* yang di mana *output* yang di tampilkan pada dua tersebut saat nilai dari perulangan bernilai *true* adalah *cout<<x<<" "* ;.

VI.8 Tugas Rumah ke-3 : (FUNGSI VOID)

Membuat sebuah program tentang materi yang dipelajari pada bab 9 beserta hasil tampilan dan analisa program !



Ttd / Paraf

1. *Source code :*

```
#include <iostream>
using namespace std;
void segitiga (int a){
    int b,c,d;
    for(b=1; b<=a; b+=2) {
        for(c=1; c<=a-b; c++) {
            cout << " ";
        }
        for(d=1; d<=b; d++) {
            cout << " *";
        }
        cout<<endl;
        cout <<endl;
    }
}
void tentukan (){
    return segitiga (7);
}
int main (){
    tentukan();
}
```

2. Tampilan Program :

```
EMOGRAMAN/tugas9/" && g++ Tugas3.cpp -o Tugas3
ULIAH/PRAKTIKUM PEMOGRAMAN/tugas9/"Tugas3
*
* * *
* * * * *
* * * * * *
arifin@MacBook-Air-Arifin tugas9 %
```

Gambar 6.6 Tugas Praktikum ke-3 (fungsi void)

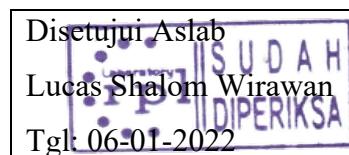
3. Analisa Program

Pada program di atas di mana pada program tersebut menggunakan 2 *void* yang pertama *void* *tentukan()* dan yang kedua *void* *segitiga*. Pada saat program di jalankan maka yang di panggil adalah *void* yang kedua yaitu *tentukan ()* sedang pada *void* *tentukan ()* tersebut memanggil *segitiga(7)* yang akan masuk ke parameter *void* *segitiga* yang *pertipe* data integer. Dan pada program tersebut menggunakan tiga *for* yaitu *for* pertama terdapat program *for(b=1; b<=a; b+=2)* dan *for* kedua *for(c=1; c<=a-b; c++)* yang terdapat *output cout<<" "*; sedangkan *for* ketiga

for(d=1; d<=b; d++) yang terdapat *output cout<< " *";* . Pada saat program ini di jalankan maka yang pertama eksekusikan yaitu *for* pertama selanjutnya *for* kedua dan selanjutnya *for* ketiga di mana nilai variabel tersebut bernilai *true* dan akan terjadi perulangan sampai bernilai *false*.

VI.9 Kesimpulan

1. Fungsi merupakan *statement* yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu. Intinya kita ingin membagi-bagi program dari program utama (main program) sehingga program dapat lebih rapi dan juga lebih terstruktur dan dapat dijalankan berulang kali.
3. *Non void* adalah fungsi dengan nilai balik atau fungsi dengan *return value* ketika fungsi dipanggil..
4. *Void* adalah fungsi yang tidak mengembalikan nilai atau fungsi yang tidak menghasilkan nilai balik ketika fungsi dipanggil.
5. Fungsi dengan parameter yang mana parameter ini akan digunakan sebagai nilai masukkan (*input*) dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai kembalian (*return value*).
6. Fungsi tanpa parameter bisa didefinisikan sebagai fungsi yang setiap kali pemanggilan fungsi tersebut, hasil yang didapatkan selalu bernilai tetap.
7. Dalam bahasa Pascal, prosedur didefinisikan sebagai suatu proses yang tidak mengembalikan nilai. Untuk melakukan hal ini di dalam C++, maka harus membuat suatu fungsi dengan tipe *void*, yang berarti tidak memiliki nilai balik (*return value*).
8. Dalam sebuah program C++ minimal kita memiliki 1 *method*, karena pada dasarnya *main()* juga merupakan sebuah *method*.



BAB VII

Array

Jumlah Pertemuan	:	100 menit
Tujuan Praktikum	:	<ol style="list-style-type: none"> 1. Mampu memahami konsep array 2. Mampu membuat program penerapan array
Alat / bahan	:	<ol style="list-style-type: none"> 1. Seperangkat <i>computer</i>. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrograman 2020.

VII.1 Landasan Teori

Selama ini kita menggunakan satu variabel untuk menyimpan 1 buah nilai dengan tipe data tertentu. Misalnya :

```
int a1, a2, a3, a4, a5;
```

Deklarasi variabel di atas digunakan untuk menyimpan 5 data integer di mana masing-masing variabel diberi nama a1, a2, a3, a4, dan a5. Jika kita memiliki 10 data, 100 data integer bahkan mungkin data yang ingin kita proses tidak kita ketahui atau bersifat dinamis? Kita tidak mungkin menggunakan variabel seperti di atas.

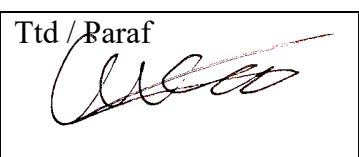
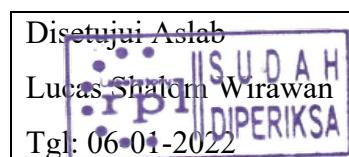
Solusi?? → Array

Array adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat *memory* yang berbeda-beda dan selanjutnya disebut dengan elemen *array*. Elemen *array* tersebut kemudian dapat kita akses melalui indeks yang terdapat di dalamnya namun penting sekali untuk diperhatikan bahwa dalam C++, Indeks *array* selalu di mulai dari 0, bukan 1.

Untuk mendeklarasian sebuah *array* kita harus menggunakan tanda [...] (*bracket*).

Bentuk umum dari variabel *array* dapat ditulis seperti di bawah ini :

```
Tipe_data nama_variabel[indeks]; //ketentuan
```



```
int data[4]; //contoh
```

Pada contoh di atas, data yang dapat ditampung yaitu 4 data. Indeks *array* selalu dimulai dari angka 0. Sehingga pada kasus di atas, data yang tersimpan dimulai dari indeks ke 0 hingga 3.

Namun anda juga dapat mendeklarasikan *array* sekaligus memberikan *inisialisasi* tanpa harus memberikan tambahan deklarasi ukuran *array*, sebagai berikut:

```
int daftarNilai [] = {10,9,8,10,9};
```

VII.1.1 Array 1 Dimensi

Contoh sebelumnya yang kita praktekkan yaitu merupakan *array* 1 dimensi. Cara di atas merupakan cara untuk melakukan inisialisasi. Untuk mengisikan data ke dalam *element-element array* kita dapat melakukannya langsung untuk setiap *element*, contohnya :

```
DATA[0]=1
DATA[1]=2
DATA[2]=3
DATA[3]=4
.....
Dst
```

Namun cara di atas tidak efisien karena jika memerlukan data yang banyak harus menuliskan *script* yang banyak juga. Cara yang lebih umum dan banyak digunakan untuk mengisikan nilai ke dalam *element array* adalah dengan menggunakan perulangan (*looping*). Contoh jika ingin mengisikan 4 elemen *array*, maka kita dapat menulis *sintaks* seperti berikut :

```
for(int index=0; index<4; index++) {
    cout<<"DATA["<<index<<"] =
"; cin>>data[index];
}
```

VII.1.2 Array 2 Dimensi

Hampir sama dengan *array* 1 dimensi, yang membuat *array* 2 dimensi berbeda adalah daya tampung datanya. Jika di *array* 1 dimensi, kita perlu *inisialisasi* 1 angka untuk menunjukkan berapa data yang bisa dimasukkan. Di *array* 2 dimensi, kita bisa menginisialisasikan 2 angka. *Array* seperti ini sering di gunakan

untuk pemrosesan *matrik*, Bentuk umum dari *array* 2 dimensi adalah sebagai berikut :

```
tipedata nama_array[jmlh baris][jmlah kolom];
float bil[2][5];
```

Tabel 7.1 Ilustrasi tabel *array*

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]

Contoh di atas menunjukkan bahwa kita dapat mengisi data dari *array* 00, 01, 02, 03, 04, 10, 11 dan seterusnya. Sehingga, data yang diisi totalnya menjadi $2 \times 5 = 10$ data.

VII.2 Langkah – Langkah Praktikum

1. Buka Aplikasi Dev C++.
2. Buka Modul Praktikum *Algoritma dan Pemrogramman*.
3. Menjalankan *script* dari modul.

VII.3 Latihan Praktikum ke-1:

Buatlah program *inputan* tinggi badan mahasiswa dalam 1 kelas. Kemudian tentukan tinggi badan tertinggi dan terendah, seperti gambar di bawah!

1. *Source code* :

```
#include <iostream>
using namespace std;
int main(){
    int banyak;
    int max, min;
    cout<<endl;
    cout<<"Masukkan banyak data : ";
    cin>>banyak;
    int data[banyak];
    for (int i = 1; i <= banyak; i++)
    {
        cout<<"Tinggi badan mahasiswa ke-<<i<<" : ";
        cin>>data[i];
    }
    cout<<"\nData Tinggi badan mahasiswa semester 1";
    cout<<endl;
    for (int a = 1; a <= banyak; a++)
    {
        cout<<"\nTinggi badan mahasiswa ke-<<a<<" :
    " <<data[a];
    }
    cout<<endl;
    max=data[0];
```

```

for (int b = 0; b < banyak; b++)
{
    if (data[b]>max) {
        max=data[b];
    }else {
        min=data[b];
    }
}
cout<<"\nTinggi badan tertinggi : "<<max;
cout<<"\nTinggi badan terendah : "<<min;

}

```

2. Tampilan Program :

```

/Users/arifin/Documents/DATA KULIAH/PRAKTIKUM PEMO
atihan1

Masukkan banyak data : 4
Tinggi badan mahasiswa ke-1 : 170
Tinggi badan mahasiswa ke-2 : 169
Tinggi badan mahasiswa ke-3 : 149
Tinggi badan mahasiswa ke-4 : 158

Data Tinggi badan mahasiswa semester 1

Tinggi badan mahasiswa ke-1 : 170
Tinggi badan mahasiswa ke-2 : 169
Tinggi badan mahasiswa ke-3 : 149
Tinggi badan mahasiswa ke-4 : 158

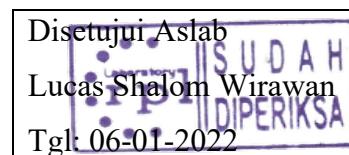
Tinggi badan tertinggi : 170
Tinggi badan terendah : 149%
arifin@MacBook-Air-Arifin latihan 10 %

```

Gambar 7.1 Latihan Praktikum ke-1 (*Array*)

3. Analisa Program

Pada Analisah program di atas yaitu menggunakan tipe data integer untuk melakukan peng *input* an nilai variabel tinggi badan. Pada saat program di jalankan *user* akan meng *input* data yang menggunakan *cin>>banyak;*, dan akan dimasukkan ke dalam data *array* yang bertipe data integer yaitu *int data[banyak];*. Dan selanjutnya yaitu setelah user sudah meng *input* banyak data maka akan menuju ke *for*(*for (int i = 1; i <= banyak; i++)*) untuk melakukan perulangan banyak data yang telah di tentukan oleh *user*. Dan pada perulangan *for* tersebut *user* akan diminta untuk memasukkan data dengan menggunakan *cin>>data[i];* sampai dari kondisi *for* tersebut *false*. Dan selanjutnya akan menuju ke *for* yang kedua (*for (int a = 1; a <= banyak; a++)*) yang akan menampilkan *imputan* dari *user* yang sebelumnya menggunakan



Ttd / Paraf

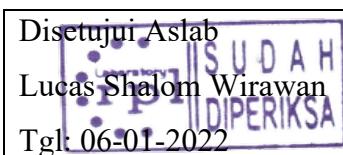
perulangan dengan menggunakan *output* (*cout*<<"\nTinggi badan mahasiswa ke-<<a<<" : "<<data[a];). Untuk variabel *max* sudah ditetapkan dengan nilai *array* *data[0]*. Selanjutnya menuju ke *for* yang ke tiga yang akan melakukan perulangan untuk mencari nilai terbesar dan terkecil pada saat awal di *input* nilai terbesar(*max*) terdapat pada nilai awal lalu akan dilakukan pengecekan *if*(*data[b]>max*) jika *true* maka nilai *max* diganti dengan nilai inputan tadi dengan menggunakan (*max*=*data[b]*);), jika nilainya *false* maka nilai *min* diganti dengan nilai *inputan* tadi dengan menggunakan (*min*=*data[b]*). Setelah proses selesai maka nilai terkecil dan terbesar akan ditampilkan dalam variabel *min* dan *max*.

VII.4 Latihan praktikum ke-2:

Buatlah program menentukan grafik poin dengan (*) berdasarkan *inputan user*

1. Source code :

```
#include <iostream>
using namespace std;
int main(){
    int banyak;
    int poin[10];
    string nama[10];
    cout<<"Masukkan bannyak data : ";
    cin>>banyak;
    for (int i = 1; i <= banyak; i++)
    {
        cout<<"\nData ke-<<i;
        cout<<"Masukkan nama = ";
        cin>>nama[i];
        cout<<"Masukkan poin = ";
        cin>>poin[i];
    }
    cout<<endl;
    cout<<"Grafik data ";
    cout<<"\n-----";
    for (int a = 1; a <= banyak; a++)
    {
        cout<<"\nData ke-<<a<<" <<nama[a]<<" : ";
        for (int b = 1; b <= poin[a]; b++)
        {
            cout<<" *";
        }
    }
    cout<<endl;
    cout<<"-----";
    ";
}
```



Ttd / Paraf




2. Tampilan Program :

```

PRAKTIKUM PEMUGRAMAN/Latihan 10/" && g++ Latihan2.cpp
/Users/arifin/Documents/DATA KULIAH/PRAKTIKUM PEMO
Latihan2
Masukkan banyak data : 3

Data ke-1
Masukkan nama = Arifin
Masukkan poin = 10

Data ke-2
Masukkan nama = ica
Masukkan poin = 8

Data ke-3
Masukkan nama = aisyah
Masukkan poin = 9

Grafik data
-----
Data ke-1 Arifin : * * * * * * * * *
Data ke-2 ica : * * * * * * *
Data ke-3 aisyah : * * * * * * *
arifin@MacBook-Air-Arifin latihan 10 %

```

Gambar 7.2 Latihan Praktikum ke-2 (*Array*)

3. Analisa Program

Dari Analisah program di atas yaitu menggunakan tipe data integer pada variabel banyak, dan pada variabel poin dengan banyak *array* poin[10], sedangkan nama menggunakan tipe data *string* dengan banyak *array* [10]. Pada saat program dijalankan akan menuju ke *for* (for (int i = 1; i <= banyak; i++)) untuk melakukan perulangan banyak data. Di dalam *for* ini terdapat variabel nama [i] dan poin[i] yang akan digunakan sebagai *inputan user* dari banyak data yang telah *user* tentukan. Dan selanjutnya akan menuju ke dua *for* yang di mana dua *for* tersebut juga dapat di panggil *nested loop* yaitu *for* dalam *for*. Dan pada *for* pertama (for (int a = 1; a <= banyak; a++)) yaitu akan menampilkan nama dengan *output* (cout<<"\nData ke-"<<a<<" " <<nama[a] <<" : ";) dan berlanjut ke *for* kedua saat nilai dari *for* pertama *true*. Dari *for* kedua (for (int b = 1; b <= poin[a]; b++)) yaitu melakukan perulangan dengan kondisi b=poin[a] dengan *output* (cout<<" *";).

VII.5 Latihan Praktikum ke-3:

Membuat program menggunakan Array.

1. *Source code :*

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

int main ()
{
    char grade[100]={};

    grade[90] = 'A';
    grade[80] = 'B';
    grade[70] = 'C';
    cout<<"\tGrade Nilai"<<endl;
    cout<<"90 : "<<grade[90]<<endl;
    cout<<"80 : "<<grade[80]<<endl;
    cout<<"70 : "<<grade[70]<<endl;

    return 0;
}
```

2. Tampilan Program ::

```
ts/DATA KULIAH/PRAKTIKUM PEMOGRAMAN/tugas1
Grade Nilai
90 : A
80 : B
70 : C
arifin@MacBook-Air-Arifin tugas10 %
```

Gambar 7.3 Latihan Praktikum ke-3 (*Array*)

3. Analisa Program

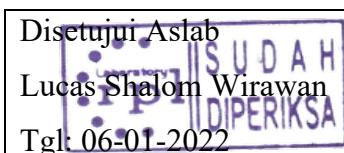
Pada Analisah program diatas terdapat deklarasi *array* yang bertipe data *char* dengan jumlah *array* sebanyak 100. Pada baris yang ke delapan udah di tentukan bahwa *grade* dengan banyak data 90 maka akan menampilkan A. dengan menggunakan *grade[90] = 'A'* dan akan di tampilkan menggunakan *cout<<"90 : "<<grade[90]<<endl*.

VII.6 Tugas Praktikum ke-1:

Perbaiki program matriks berikut dan tampilkan program beserta analisa nya

1. *Source code :*

```
/*#include <iostream>
using namespace std;
int main(){
    int n,i,j;
```



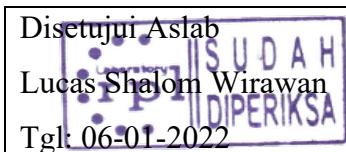
Ttd / Paraf

```

cout<<"Masukkan ukuran ordo matri pertama dan
kedua= ";cin>>n;
int matri_1[n][n],matri_2[n][n];
cout<<endl;
for (i = 0; i < n; i++)
{
    for (j=0; j<n; j++)
    {
        cout<<"Baris "<<i<<" kolom "<<j<<" : ";
        cin>>matrix_1 [n][n];
    }
}
cout<<"Tampilan matrix \n";
for (i = 1; i < n; i++)
{
    for (j=1; j<n; j++)
    {
        cout<<matrix_1 [n][n]<<" ";
    }
    cout<<endl;
}
}*/
```

```

#include <iostream>
using namespace std;
int main(){
    int n,i,j;
    cout<<"Masukkan ukuran ordo matri pertama dan kedua
= ";cin>>n;
    int matrik_1[n][n];
    cout<<endl;
    for (i = 0; i < n; i++)
    {
        for (j=0; j<2; j++)
        {
            cout<<"Baris "<<i<<" kolom "<<j<<" : ";
            cin>>matrik_1 [i][j];
        }
    }
    cout<<"Tampilan matrix \n";
    for (i = 0; i<n; i++)
    {
        for (j=0; j<2; j++)
        {
            cout<<matrik_1 [i][j]<<" ";
        }
        cout<<endl;
    }
}
```



Ttd / Paraf


2. Tampilan Program :

```
rs/arifin/Documents/DATA KULIAH/PRAKTIKUM PEMOGRAMAN/latihan1
Masukkan ukuran ordo matri pertama dan kedua = 4
Baris 0 kolom 0 : 2
Baris 0 kolom 1 : 1
Baris 1 kolom 0 : 3
Baris 1 kolom 1 : 6
Baris 2 kolom 0 : 5
Baris 2 kolom 1 : 7
Baris 3 kolom 0 : 9
Baris 3 kolom 1 : 8
Tampilan matrix
2 1
3 6
5 7
9 8
arifin@MacBook-Air-Arifin latihan 10 %
```

Gambar 7.4 Tugas Praktikum ke-1 (*Array*)

3. Analisa Program

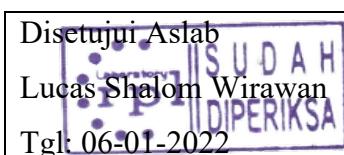
Pada Analisah program di atas yaitu melakukan perbaikan pada *matrik*. Program di atas menggunakan *array* 2 dimensi yang dapat mengeluarkan *output* variabel i yang sebagai baris dan variabel j yang sebagai kolom. Pada saat program di jalankan *user* akan meng *input* ukuran ordo *matrik* pertama dan kedua dengan menggunakan variabel n. dan selanjutnya yaitu menuju ke *for* dalam *for* yang di mana *for* pertama (*for* (*i* = 0; *i* < *n*; *i*++)) sebagai baris dan *for* kedua (*for* (*j*=0; *j*<2; *j*++)) sebagai kolom yang akan melakukan perulangan sampai nilai dari variabel *i* *false*. Dan pada *for* kedua terdapat *inputan user* yang menggunakan (*cin*>>*matrik_1* [*i*][*j*] ;). Dan selanjutnya yaitu menampilkan *matrik* menggunakan perulangan dua *for* yaitu *for* dalam *for* dengan *output* (*cout*<<*matrik_1* [*i*][*j*]<<" ");) dan akan berlanjut melakukan perulangan sampai kondisi bernilai *false*.

VII.7 Tugas Praktikum ke-2:

Buatlah program nilai mahasiswa yang terdiri di dalamnya terdapat nama mahasiswa, nim dan nilai. Hasil *inputan* akan ditampilkan beserta keterangan grade. beri Analisa programnya!

1. Source code :

```
#include <iostream>
using namespace std;
int main(){
    int b,a;
    string nama[10],status[10];
    int nilai[10],nim[10];
    cout<<"Masukkan banyak data = ";
```



Ttd / Paraf

```

    cin>>a;
    cout<<endl;
    for (b=0; b<a;b++) {
        cout<<"Data ke-"<<b;
        cout<<"\n-----";
        cout<<endl;
        cout<<"Masukkan nama : ";
        fflush(stdin); //buat clear input buffer
        getline(cin,nama[b]);
        cout<<"Masukkan nim : ";
        cin>>nim[b];
        cout<<"Masukkan nilai : ";
        cin>>nilai[b];
        if(nilai[b]>=90 && nilai[b]<=100) {
            status[b]="A";
        }else if (nilai[b]>=80 && nilai[b]<=90) {
            status[b]="B";
        }else if (nilai[b]>=70 && nilai[b]<=80) {
            status[b]="C";
        }else if (nilai[b]<70) {
            status[b]="D";
        }
        cout<<endl;
    }
    cout<<endl;
    cout<<"\n\t\tDATA MAHASISWA";
    cout<<"\n\t=====";
    cout<<endl;
    cout<<"\n-----";
    cout<<"\n NO      NAMA \t NIM \t\tNILAI \t STATUS";
    cout<<"\n=====";
    cout<<"\n=====<<endl;
    for (int i = 0; i < a; i++)
    {
        cout<<" " <<i <<" " <<nama[i] <<"\t"
" <<nim[i] <<"\t" <<nilai[i] <<"\t" <<status[i] <<endl;
        cout<<"-----";
    }
}

```



Ttd / Paraf

2. Tampilan Program :

```

rs/arifin/Documents/DATA KULIAH/PRAKTIKUM PEMROGRAMAN/latihan 10/"tu
2
Masukkan banyak data = 4

Data ke-0
-----
Masukkan nama : Arifin
Masukkan nim : 2118131
Masukkan nilai : 96

Data ke-1
-----
Masukkan nama : Aisyah
Masukkan nim : 2118114
Masukkan nilai : 88

Data ke-2
-----
Masukkan nama : Azzahra
Masukkan nim : 2118137
Masukkan nilai : 67

Data ke-3
-----
Masukkan nama : iqbal
Masukkan nim : 2118119
Masukkan nilai : 77

=====
DATA MAHASISWA
=====

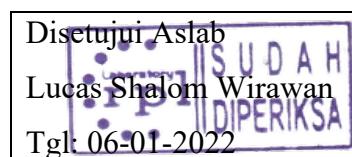
NO NAMA NIM NILAI STATUS
=====
0 Arifin 2118131 96 A
1 Aisyah 2118114 88 B
2 Azzahra 2118137 67 D
3 iqbal 2118119 77 C
=====
arifin@MacBook-Air-Arifin latihan 10 %

```

Gambar 7.5 Tugas Praktikum ke-2 (Array)

3. Analisa Program

Pada Analisa program di atas yaitu menggunakan tipe data integer pada variabel a dan b, sedangkan yang variabel nilai dan nim mempunya banyak *array* [10] dan juga menggunakan tipe data *string* pada variabel nama dengan *array* [10] dan variabel status dengan *array* [10]. Dan saat program dijalankan *user* akan meng *input* banyak data dengan menggunakan *cin>>a;*. dan selanjutnya yaitu menuju ke *for* (*for (b=0; b<a;b++)*) untuk melakukan perulangan dari banyak data yang telah *user* tentukan. Di dalam *for* terdapat *input* an data yang di mana *user* akan diminta untuk memasukkan yang udah di tentukan. Pada baris 15 terdapat (*fflush(stdin);*) untuk membersihkan *input buffer*, jika tidak menggunakan ini maka secara otomatis akan menuju ke *inputan* berikutnya. Setelah *user* meng *input* data maka akan menuju *if else* yang menentukan nilai status dengan kondisi yang telah di tetapkan. Dan pada saat *for* pertama *kodisinya* terpenuhi maka akan *muju* ke *for* selanjutnya (*for (int i = 0; i < a; i++)*) untuk melakukan perulangan dengan menampilkan hasil dari *inputan* yang telah di masukkan oleh *user*



Ttd / Paraf

dengan *output* (*cout*<<"<<i<<" "=<<nama[i]<< "\t "<<nim[i]<< "\t "<< nilai[i] <<"\t" <<status[i] <<endl;).

VII.8 Tugas Praktikum ke-3:

Membuat sebuah program tentang materi yang dipelajari pada bab 10 beserta hasil tampilan dan analisa program !

1. Source code :

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

int main()
{
    int nilaiTugas[5]={75, 80, 90, 100, 95};
    for(int i = 0; i<5; i++){
        cout<<"Nilai tugas-<<i+1<<" adalah "<<
nilaiTugas[i]<<endl;
    }

    return 0;
}
```

2. Tampilan Program :

```
[arifin:~/Documents/tugas10] arifin:~/Documents/tugas10%
/DATA KULIAH/PRAKTIKUM PEMOGRAMAN/tugas10/
Nilai tugas-1 adalah 75
Nilai tugas-2 adalah 80
Nilai tugas-3 adalah 90
Nilai tugas-4 adalah 100
Nilai tugas-5 adalah 95
arifin@MacBook-Air-Arifin tugas10 %
```

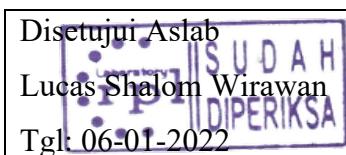
Gambar 7.6 Tugas Praktikum ke-3 (Array)

3. Analisa Program

Analisa program di atas, pada baris 6 dideklarasikan sebuah array. *Array* bernama *Taskvalue*, merupakan tipe integer dengan panjang 5 elemen dengan nilai 75, 80, 90, 100, 95. Dan selanjutnya dari nilai setiap elemen *array* tersebut akan di tampilkan menggunakan *for* (*for*(*int* *i* = 0; *i*<5; *i*++)) yang akan melakukan perulangan sampai kondisi bernilai *false*. Dan setiap terjadi perulangan akan menampilkan *output* (*cout*<<"Nilai tugas-<<*i*+1<<" adalah "<< nilaiTugas[*i*]<<endl;) sampai kondisi terpenuhi.

VII.9 Kesimpulan

1. *Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat *memory* yang berbeda-beda dan selanjutnya disebut dengan elemen *array*.
2. Jika memerlukan data yang banyak harus menuliskan *script* yang banyak juga. Cara yang lebih umum dan banyak di gunakan untuk mengisikan nilai ke dalam *element array* adalah dengan menggunakan perulangan (*looping*).
3. *Array 2 dimensi* Hampir sama dengan *array 1 dimensi*, yang membuat *array 2 dimensi* berbeda adalah daya tampung datanya. Jika di *array 1 dimensi*, kita perlu *inisialisasi* 1 angka untuk menunjukkan berapa data yang bisa dimasukkan. Di *array 2 dimensi*, kita bisa menginisialisasikan 2 angka.



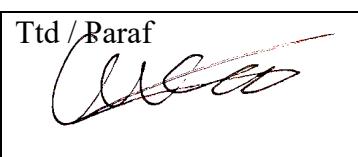
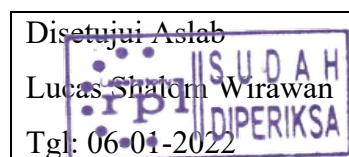
Ttd / Paraf

BAB VIII

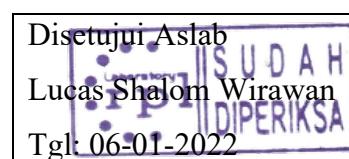
Kesimpulan

Dengan adanya kegiatan praktikum Algoritma dan Pemrogramman. Penulis dapat menyimpulkan sebagai berikut :

1. `#include <iostream>` adalah memberitahukan *compiler* agar menggunakan file *header iostream* yang ada pada direktori file-file *header*.
2. Int *main()* adalah fungsi utama dari sebuah kode bahasa C. Fungsi ini memberikan nilai balik menurut *type* datanya, dan karena memiliki nilai balik maka diberikan perintah *return* nilai.
3. *cout* adalah fungsi standar pada C++ untuk menampilkan *output* ke layar. Setelah simbol `<<` kita bisa menuliskan teks yang akan ditampilkan ke layar.
4. *cin* adalah perintah dasar C++ untuk proses *input* atau menerima data masukan dari *user* dan menggunakan simbok `>>`.
5. Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi.
6. Operator *assignment* (penugasan) adalah operator yang digunakan untuk memasukkan atau mengisikan nilai ke dalam suatu variabel. Dalam bahasa C++ digunakan operator berupa tanda sama dengan (“=”).
7. Operator aritmatika digunakan untuk melakukan sebuah *opearasi* penghitungan atau *aritmatika*. Tugasnya untuk memecahkan masalah matematika.
8. Operator *relasional* atau hubungan digunakan untuk membandingkan hubungan antara dua buah *operand*.
9. Struktur *if* satu kondisi ini merupakan struktur yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa.
10. struktur *if* dua kondisi ini terdapat sebuah *statemen* khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).
11. Struktur *if* 3 kondisi/lebih percabangan ini merupakan perluasan dari struktur yang memiliki dua kondisi, yaitu dengan menyisipkan satu atau lebih kondisi ke dalamnya.

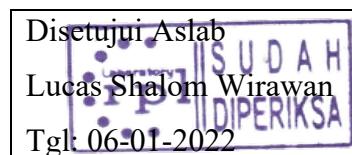


12. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian.
13. Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan.
14. Dalam bahasa C++ terdapat tiga buah jenis struktur pengulangan, yaitu : Struktur For, Struktur *While*, Struktur *Do-While*, Struktur *Nested For*.
15. Struktur for di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu diantara-Nya akan digunakan sebagai indeks pengulangan.
16. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan.
17. Struktur perulangan WHILE adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur.
18. Pengulangan hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar).
19. Berbeda dengan struktur *while* yang melakukan pemeriksaan kondisi di awal blok perulangan, pada struktur *do-while* kondisi justru ditempatkan di bagian akhir.
20. Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan.
21. Pernyataan *nested for* adalah suatu perulangan *for* di dalam perulangan *for* yang lain.
22. *Nested loops* digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom
23. Fungsi merupakan *statement* yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu. Intinya kita ingin membagi-bagi program dari program utama (main program) sehingga program dapat lebih rapi dan juga lebih terstruktur dan dapat dijalankan berulang kali.
24. *Non void* adalah fungsi dengan nilai balik atau fungsi dengan *return value* ketika fungsi dipanggil..



Ttd / Paraf

25. *Void* adalah fungsi yang tidak mengembalikan nilai atau fungsi yang tidak menghasilkan nilai balik ketika fungsi dipanggil.
26. Fungsi dengan parameter yang mana parameter ini akan digunakan sebagai nilai masukkan (*input*) dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai kembalian (*return value*).
27. Fungsi tanpa parameter bisa didefinisikan sebagai fungsi yang setiap kali pemanggilan fungsi tersebut, hasil yang didapatkan selalu bernilai tetap.
28. Dalam bahasa Pascal, prosedur didefinisikan sebagai suatu proses yang tidak mengembalikan nilai. Untuk melakukan hal ini di dalam C++, maka harus membuat suatu fungsi dengan tipe *void*, yang berarti tidak memiliki nilai balik (*return value*).
29. *Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat *memory* yang berbeda-beda dan selanjutnya disebut dengan elemen *array*.
30. Jika memerlukan data yang banyak harus menuliskan *script* yang banyak juga. Cara yang lebih umum dan banyak di gunakan untuk mengisikan nilai ke dalam *element array* adalah dengan menggunakan perulangan (*looping*).
31. *Array 2 dimensi* Hampir sama dengan *array 1 dimensi*, yang membuat *array 2 dimensi* berbeda adalah daya tampung datanya. Jika di *array 1 dimensi*, kita perlu *inisialisasi* 1 angka untuk menunjukkan berapa data yang bisa dimasukkan. Di *array 2 dimensi*, kita bisa menginisialisasikan 2 angka.

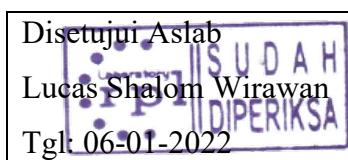


Ttd / Paraf


DAFTAR PUSTAKA

- Laboratorium Pemrograman dan RPL. 2021. *Modul Praktikum Algoritma Dan Pemrogramman*. Laboratorium Pemrograman Insitut Teknologi Nasional Malang.
- Utami, E. (2005). 10 Langkah Belajar Logika dan Algoritma. menggunakan Bahasa C dan C++ di GnuLlinux. Penerbit Andi.
- Dewi, L. J. E. (2012). Media Pembelajaran Bahasa Pemrograman C++. Jurnal Pendidikan Teknologi dan Kejuruan, 7(1).
- Document. Tanpa tahun. *Operator dan statemen I/O*. (online). (<https://dokumen.tips/documents/praktikum-c-3-operator-dan-statement-io.html/>).
- Pdfcoffee. Tanpa tahun. *Operator dan statemen I/O*. (online). (<https://pdfcoffee.com/praktikum-c-3-operator-dan-statement-i-o-pdf-free.html>).
- Hanief, S., Jepriana, I. W., & Kom, S. (2020). *Konsep Algoritme dan Aplikasinya dalam Bahasa Pemrograman C++*. Penerbit Andi.
- Nugraha, M. (2021). *Dasar Pemrograman Dengan C++ Materi Paling Dasar Untuk Menjadi Programmer Berbagai Platform*. Deepublish.
- Indahyanti, U., & Rahmawati, Y. (2020). *Buku Ajar Algoritma Dan Pemrograman Dalam Bahasa C++*. Umsida Press, 1-146.
- FIRLIANA, R., & Kasih, P. (2018). Algoritma dan Pemrograman C++.
- DuniaIlkom. 2020. *Tutorial Belajar C++ Part 32: Perulangan FOR Bahasa C++*. (online). (<https://www.duniaIlkom.com/tutorial-belajar-c-plus-plus-perulangan-for-bahasa-c-plus-plus/>).
- Kelasprogramer. 2020. *Perulangan For Pada C++ Pembahasan Lengkap*. (online). (<https://kelasprogrammer.com/perulangan-for-cpp/>).
- Couresehero. Tanpa tahun. *Fungsi non void disebut juga function disebut non*. (online). (<https://www.coursehero.com/file/p3oit0i/Fungsi-non-void-disebut-juga-function-Disebut-non-void-karena-mengembalikan/>).
- Pratama, M. A. (2020). STRUKTUR DATA ARRAY DUA DIMENSI PADA PEMROGRAMAN C++.

Arifin, R. W., & Setiyadi, D. (2020). Bubble Sort and Quick Sort Sort Method Algorithm in C++ Programming Language. INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS: Journal of Information Systems, 4(2), 178-187.



Ttd / Paraf





**LEMBAR ASISTENSI PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
SEMESTER GANJIL TAHUN AKADEMIK 2021/2022**

Nama : MOHAMMAD HARIFIN
NIM : 2118131

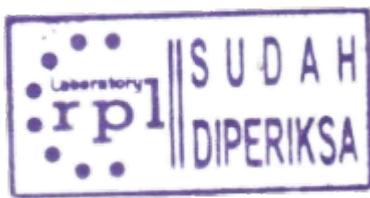


No.	Tanggal	Asistensi			Paraf	
		Konsep	Program	Hasil Akhir		
1	6 Januari 2022	Instruktur	✓ Pengenalan C++ ✓ Operator dan Statement I/O ✓ Percabangan ✓ Perulangan ✓ Fungsi Void dan Non-Void Array	FIX	FIX	
2		Dosen				

Batas Akhir: 10 Januari 2022

Asisten,

(Lucas Shalom Wirawan)
2018033



Malang, 14 Januari 2022
Dosen, Pembimbing,

(Ahmad Fahrudi Setiawan, S.Kom, M.T.)
NIP.P.1031500497