

Modul Praktikum

Algoritma dan Pemrograman



Modul Praktikum Semester Ganjil 2019/2020
UNIVERSITAS AMIKOM YOGYAKARTA

MODUL PRAKTIKUM #1

A. Install Notepad++ terbaru

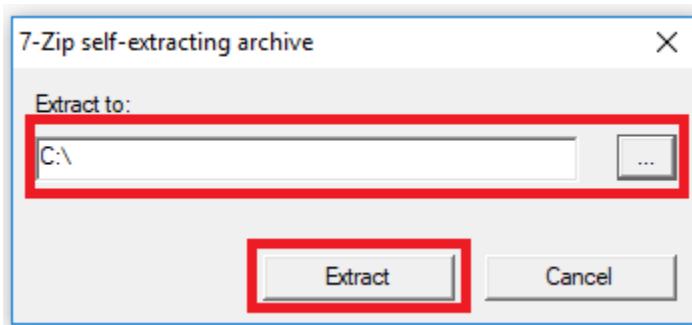
1. Link download <https://notepad-plus-plus.org/downloads/>

B. Setting MinGW Pada CommandPrompt (CMD)

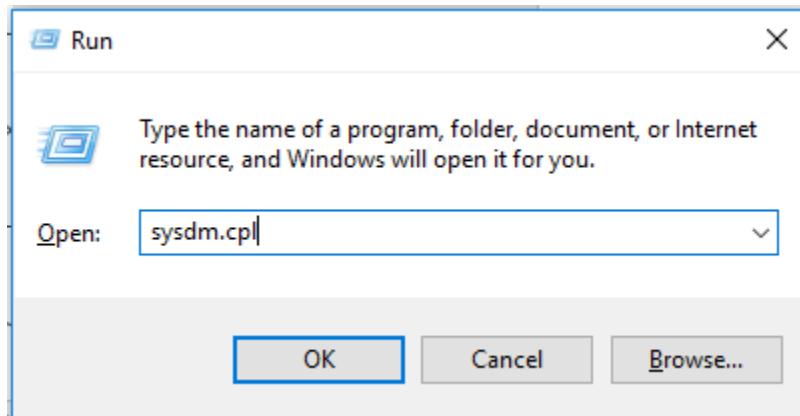
1. Download MinGW (engine compiler C/C++) terlebih dahulu melalui link:

<https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/mingw-offline-install/mingw-6.3.exe>

2. Ekstrak MinGW ke drive C:\ sehingga compiler MinGW ada di Folder C:\MinGW

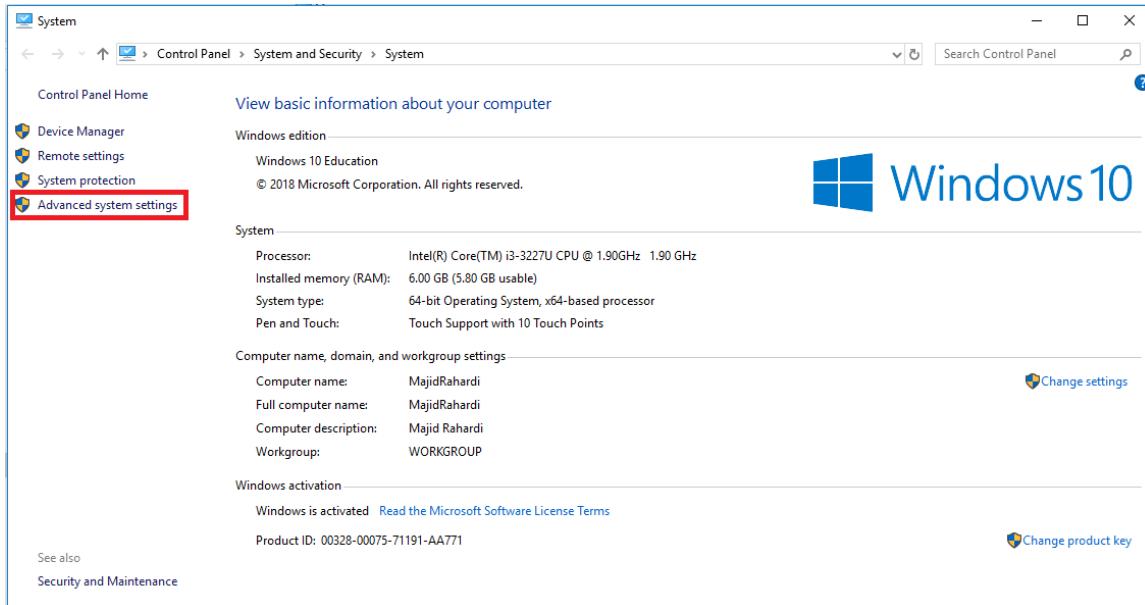


3. Buka Run (Windows + R) lalu ketik "sysdm.cpl" tanpa tanda petik kemudian pilih tab Advanced --> Environment Variables



Atau Masuk ke Control Panel\System and Security\System

Atau This PC dengan cara klik kanan – properties.

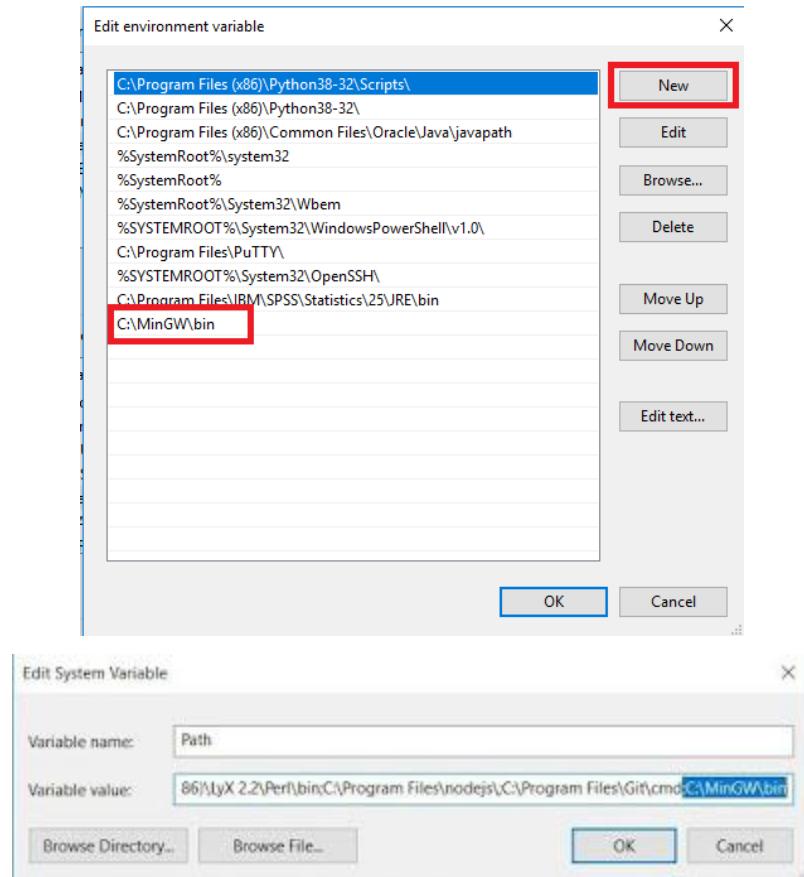


4. Klik Advanced System Settings – maka akan muncul form System Properties – klik tab Advanced – klik tab Environment Variables

User variables for Majid Rahardi	
Variable	Value
MOZ_PLUGIN_PATH	
OneDrive	C:\Users\Majid Rahardi\OneDrive
Path	C:\Users\Majid Rahardi\AppData\Local\Microsoft\WindowsApps; C...
TEMP	C:\Users\Majid Rahardi\AppData\Local\Temp
TMP	C:\Users\Majid Rahardi\AppData\Local\Temp

System variables	
Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files (x86)\Python38-32\Scripts\;C:\Program Files (x86)\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
PROCESSOR_ARCHITECTURE	AMD64

5. Klik path di system variables -> Edit -> New -> ketik C:\MinGW\bin -> Ok ->Ok ->Ok.



6. Buka Run (Windows + R) lalu ketik "cmd" tanpa tanda petik, ketikkan command g++ --version :

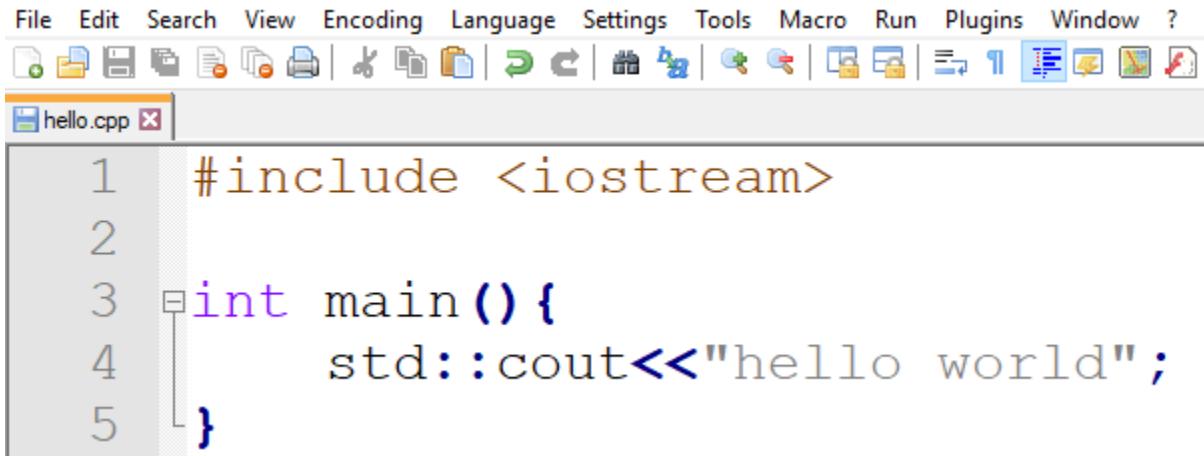
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1726]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Majid Rahardi>g++ --version
g++ (GCC) 4.5.0
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

7. MinGW selesai dipasang dan siap digunakan. Selamat membuat program !!!

C. Program Pertama Anda dengan C++

1. Buka notepad++ anda, ketikkan baris perintah berikut :



```
#include <iostream>
int main() {
    std::cout<<"hello world";
}
```

Keterangan Code :

Baris 1 adalah header library yang ditambahkan pada program untuk operasi input output. Program yang anda buat ini adalah program yang akan menampilkan tulisan “hello world”, karena membutuhkan output maka membutuhkan header library tsb.

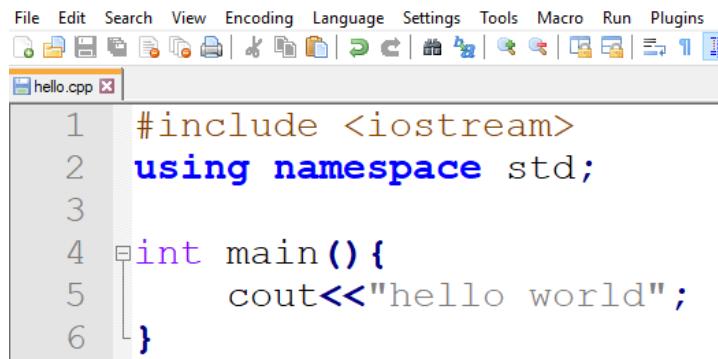
Baris 3 adalah metode main yang berisi statement perintah baris 4

Metode main atau sering dipanggil method main, adalah method yang pertama kali dijalankan ketika program anda decompile

Baris 4 adalah perintah untuk menampilkan pesan hello world menggunakan perintah cout

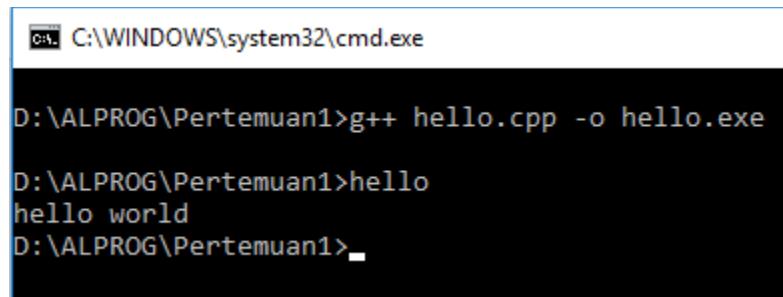
Baris 5 adalah perintah untuk mengakhiri method main, sederhananya perintah tsb digunakan untuk menghentikan program

Contoh tanpa menggunakan std:: maka dibutuhkan memanggil fungsi std dengan cara menambahkan “using namespace std;”



```
#include <iostream>
using namespace std;
int main() {
    cout<<"hello world";
}
```

2. Simpan file program dengan nama hello berekstensi cpp, misal hello.cpp
3. Buka commandPrompt anda, masuk ke direktori penyimpanan file program C++ anda
4. Setelah anda berada di lokasi file program C++ anda, ketikkan perintah g++ hello.cpp -o hello.exe, artinya anda ingin mengkompilasi file hello.cpp dan memberinya nama output hasil hello.exe
5. Setelah berhasil dan tidak muncul error, ketikkan perintah hello atau hello.exe, berikut hasil running program :



```
C:\WINDOWS\system32\cmd.exe
D:\ALPROG\Pertemuan1>g++ hello.cpp -o hello.exe
D:\ALPROG\Pertemuan1>hello
hello world
D:\ALPROG\Pertemuan1>
```

Catatan pada cmd :

1. Pindah direktori, ketikan direktorinya lalu beri titik dua, misal D:

```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1726]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Majid Rahardi>D:

D:\>
```

2. Masuk ke sebuah folder, ketikan cd spasi nama folder yang dituju, misal cd ALPROG

```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1726]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Majid Rahardi>D:

D:\>cd ALPROG

D:\ALPROG>
```

3. Pindah ke folder yang lebih atas, ketikan cd..

```
C:\> C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1726]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Majid Rahardi>D:

D:\>cd ALPROG

D:\ALPROG>cd..

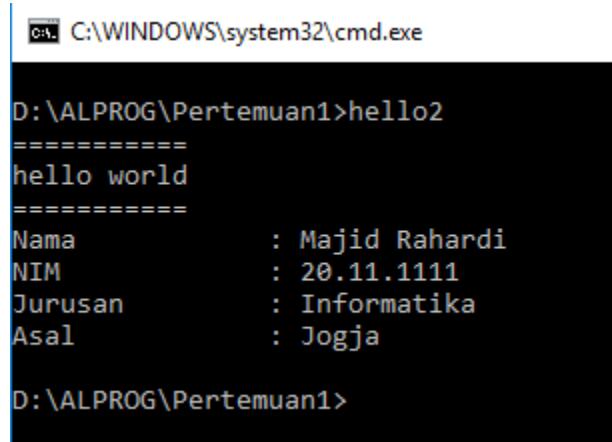
D:\>
```

4. Membersihkan layar cmd, ketikan cls

5. Melihat isi folder, ketikan dir

LATIHAN !

1. Buatlah program yang menampilkan :
 - a. Nama anda
 - b. Nim anda
 - c. Jurusan anda
 - d. Asal



```
C:\WINDOWS\system32\cmd.exe
D:\ALPROG\Pertemuan1>hello2
=====
hello world
=====
Nama : Majid Rahardi
NIM : 20.11.1111
Jurusan : Informatika
Asal : Jogja

D:\ALPROG\Pertemuan1>
```

MODUL PRAKTIKUM #2

A. Type Data

Pada dasarnya tipe data dibagi menjadi 3 jenis:

- **Tipe Data Angka** - untuk angka dan berhubungan dengan aritmetika.
- **Tipe Data Karakter** - untuk karakter dan angka bukan untuk operasi aritmetika.
- **Tipe Data Logika** - untuk logika benar (*true*) atau salah (*false*).

No	Tipe Data	Keterangan
1.	Char	Karakter / string
2.	int	Integer/bilangan bulat
3.	float	Float/bilangan pecahan
4.	double	Pecahan presisi ganda
5.	void	Tidak bertipe
6.	String	String

Cobalah coding di bawah ini !

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // deklarasi tipe data variabel
6     string nama;
7     int umur;
8     char jenis_kelamin;
9
10    // --- proses input ---
11    cout << "Siapakah namamu?" << endl;
12    cout << "jawab: ";
13    // menyimpan data ke variabel
14    getline(cin,nama);
15
16    cout << "Berapa umurmu?" << endl;
17    cout << "jawab: ";
18    // menyimpan data ke variabel
19    cin >> umur;
20
21    cout << "Jenis kelamin [L/P]: ";
22    // menyimpan data ke variabel
23    cin >> jenis_kelamin;
24
25    // --- proses output ---
26    cout << "Salam kenal, " << nama << " Sekarang anda berusia ";
27    cout << umur << " dan anda berjenis kelamin "<< jenis_kelamin;
28 }
```

B. Jenis-jenis Type Data

a. Tipe Data Integer (int)

Integer adalah salah satu tipe data numerik yang memungkinkan kita untuk menyimpan data dalam bentuk bilangan bulat.

Source Code

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

int main(){
    int x,y,z;
    x=3; y=4;

    z=x*y;
    cout << "Hasil perkalian: " << z;
}
```

Dengan menggunakan tipe data integer hal ini memungkinkan kita untuk melakukan sejumlah operasi aritmetika seperti perkalian dan lain sebagainya.

b. Tipe Data Character (char)

Character adalah salah satu tipe data yang memungkinkan kita untuk memesan memori berformat text (huruf, angka, dan simbol) dengan karakter tunggal.

Source Code

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

int main(){
    char nilai;

    cout << "Masukan nilai (A/B/C/D): "; cin>>nilai;
    cout << "Nilai anda:" << nilai;
}
```

Perlu diingat bahwa tipe data char hanya dapat menyimpan data berbentuk karakter dan hanya satu karakter, oleh karena itu apabila anda memasukan lebih dari 1 karakter maka nilai yang akan tersimpan hanya karakter pertama.

c. Tipe Data String (string)

String merupakan tipe data *text* (huruf, angka, dan simbol) yang memungkinkan kita menyimpan nilai dengan bentuk text, kumpulan dari character.

Source Code

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

int main(){
    string nohp;

    cout << "Masukan nomor HP: "; cin >> nohp;
    cout << "Nomor HP anda: " << nohp;
}
```

Sama seperti halnya tipe data char, dalam tipe data string kita bisa menggunakan karakter dan angka dengan ketentuan tidak dapat dilakukan operasi aritmetika.

Namun perbedaannya, jika dalam tipe data char kita hanya mampu menyimpan nilai satu karakter untuk tiap variabel, hal ini tidak berlaku pada tipe data string.

Hal ini di karenakan **String** merupakan tipe data composite.

d. Tipe Data Boolean (bool)

Boolean adalah salah satu tipe data yang hanya memiliki dua pilihan yaitu True (1) atau False (0). Tipe data ini biasanya digunakan untuk memberikan kondisi pada program. Atau bisa juga memastikan kebenaran dari sebuah operasi.

Source Code

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

int main()
{
    int angka;
    bool hasil;
    cout << "Masukan angka = "; cin >> angka;
    hasil = angka > 10;
    cout << hasil;
}
```

Pada contoh program diatas, kita menggunakan 2 buah variabel yaitu variable "angka" dengan tipe data integer, dan variable "hasil" dengan tipe data boolean.

Nah, kita akan mengambil nilai/value untuk variabel hasil dengan membandingkan nilai pada variabel angka terhadap bilangan 10.

Apabila nilai pada variabel angka lebih dari 10 maka hasil bernilai 1 (true) dan jika angka lebih kecil dari 10 maka hasilnya bernilai 0 (false).

e. Tipe Data Floating Point (float)

Floating Point adalah tipe data numerik yang memungkinkan untuk menyimpan nilai dalam memori bersifat bilangan pecahan atau real, maupun eksponensial.

Source Code

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

int main(){
    float jari, hasil ;
    const float pi=3.14;

    cout << "Masukan Jumlah jari-jari = "; cin >> jari;
    hasil = (jari * pi) * 2;

    cout << "Keliling dari Lingkaran adalah " << hasil;
}
```

f. Tipe Data Double Floating Point (double)

Double Floating Point sama seperti float yaitu salah satu tipe data yang bersifat menyatakan bilangan pecahan atau real, maupun eksponensial. Bedanya adalah penyimpanan angka masimal lebih besar daripada float dan otomatis double juga akan membutuhkan memori yang lebih besar.

Source Code

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

int main(){
    double jari, hasil ;
    const double pi=3.1428;

    cout << "Masukan Jumlah jari-jari = "; cin >> jari;
    hasil = jari*(jari * pi);
    cout << "Luas lingkaran: " << hasil;
}
```

MODUL PRAKTIKUM #3 - Operator

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengenal operator dalam bahasa pemrograman C++ secara baik.
- 2) Menjelaskan jenis-jenis operator dalam bahasa C++ secara benar.
- 3) Dapat menggunakan operator dalam bahasa C++ untuk memanipulasi data secara benar
- 4) Menghitung bilangan berdasarkan penaikan dan penurunan secara tepat

B. Materi Operator C++

Operator adalah sebuah simbol. Simbol yang digunakan untuk melakukan operasi tertentu.

Misalnya: Kita ingin menjumlahkan nilai dari variabel x dan y, maka kita bisa menggunakan operator penjumlahan (+). Sehingga menjadi $x + y$.

Ditinjau berdasarkan sifatnya, operator dapat dibedakan menjadi tiga macam, yaitu :

- 1) unary - operator yang melibatkan sebuah operand
- 2) binary - operator yang melibatkan dua operand
- 3) ternary - operator yang melibatkan tiga operand

```
-a //unary
a + b //binary
a ? b : c //ternary
```

1. Operator Unary

Operator unary adalah operator yang hanya melibatkan sebuah operand. Adapun yang termasuk dalam kategori operator unary ini adalah sebagai berikut:

Operator	Jenis Operasi	Contoh
+	Membuat nilai positif	+7
-	Membuat nilai negatif	-7
++	Increment	C++
--	Decrement	C--

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     // Mendeklarasikan Variabel dan Memberi Nilai
6     int A = 10;
7     int B = 10;
8
9     // Melakukan Increment
10    A++; //sama dengan A+1
11
12    // Melakukan Decrement
13    B--; //sama dengan B-1
14
15    //Menampilkan nilai A akhir
16    cout<<"Nilai A akhir : "<<A<<endl;
17
18    //Menampilkan nilai B akhir
19    cout<<"Nilai B akhir : "<<B<<endl;
20 }
```

Hasil :

Command Prompt

```
D:\ALPROG\Pertemuan3>InDe.exe
Nilai A akhir : 11
Nilai B akhir : 9
```

```
D:\ALPROG\Pertemuan3>
```

2. Operator Binary

Operator binary adalah operator yang digunakan dalam operasi yang melibatkan dua buah operand. Jenis-jenis binary adalah :

a) Operator Aritmatika

Operator aritmatika merupakan operator yang digunakan untuk melakukan operasi aritmatika.

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa Bagi	%

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 7, b = 5;
6
7     cout << "Hasil a + b = " << a + b << endl;
8     cout << "Hasil a - b = " << a - b << endl;
9     cout << "Hasil a * b = " << a * b << endl;
10    cout << "Hasil a / b = " << a / b << endl;
11    cout << "Hasil a % b = " << a % b << endl;
12 }
```

```

D:\ALPROG\Pertemuan3>aritmatika.exe
Hasil a + b = 12
Hasil a - b = 2
Hasil a * b = 35
Hasil a / b = 1
Hasil a % b = 2

D:\ALPROG\Pertemuan3>
```

b) Operator Penugasan

Operator penugasan (Assignment Operator) merupakan operator untuk memberikan tugas pada variabel. Biasanya untuk mengisi nilai. Operator Penugasan terdiri dari:

Nama Operator	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=
Pengisian dan shift left	<<=
Pengisian dan shift right	>>=
Pengisian dan bitwise AND	&=
Pengisian dan bitwise OR	=
Pengisian dan bitwise XOR	^=

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a=5, b=10;
6
7     b += a; //pengisian dan penambahan, ini sama seperti b = b + a
8     cout << "Hasil b += a adalah " << b << endl;
9
10    a=5, b=10;
11    b -= a; //pengisian dan pengurangan, ini sama seperti b = b - a
12    cout << "Hasil b -= a adalah " << b << endl;
13
14    a=5, b=10;
15    b *= a; //pengisian dan perkalian ini, sama seperti b = b * a
16    cout << "Hasil b *= a adalah " << b << endl;
17
18    a=5, b=10;
19    b /= a; //pengisian dan pembagian, ini sama seperti b = b / a
20    cout << "Hasil b /= a adalah " << b << endl;
21
22    a=5, b=10;
23    b %= a; //pengisian dan penambahan, ini sama seperti b = b % a
24    cout << "Hasil b %= a adalah " << b << endl;
25 }
```

```
Command Prompt  
D:\ALPROG\Pertemuan3>penugasan.exe  
Hasil b += a adalah 15  
Hasil b -= a adalah 5  
Hasil b *= a adalah 50  
Hasil b /= a adalah 2  
Hasil b %= a adalah 0  
D:\ALPROG\Pertemuan3>
```

c) Operator Pembanding

Operator pembanding adalah operator untuk membandingkan dua buah nilai. Operator ini juga dikenal dengan operator relasi. Nilai yang dihasilkan dari operasi pembanding akan berupa true dan false. Pada bahasa C++, nilai true akan sama dengan 1 dan false akan sama dengan 0. Operator pembanding terdiri dari:

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama dengan	!=
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 4, b = 3;
6     bool hasil;
7
8     hasil = a > b;
9     cout << "a > b = " << hasil << endl;
10
11    hasil = a < b;
12    cout << "a < b = " << hasil << endl;
13
14    hasil = a >= b;
15    cout << "a >= b = " << hasil << endl;
16
17    hasil = a <= b;
18    cout << "a <= b = " << hasil << endl;
19
20    hasil = a == b;
21    cout << "a == b = " << hasil << endl;
22
23    hasil = a != b;
24    cout << "a != b = " << hasil << endl;
25 }
```

```
Command Prompt
D:\ALPROG\Pertemuan3>pembanding.exe
a > b = 1
a < b = 0
a >= b = 1
a <= b = 0
a == b = 0
a != b = 1
D:\ALPROG\Pertemuan3>
```

d) Operator Logika

Operator Logika digunakan untuk membuat operasi logika. Operator logika terdiri dari:

Nama Operator	Simbol di Java
Logika AND	&&
Logika OR	
Negasi/kebalikan	!

Pernyataan 1	Pernyataan 2	Logika AND
true	true	true
true	false	false
false	true	false
false	false	false

Pernyataan 1	Pernyataan 2	Logika OR
true	true	true
true	false	true
false	true	true
false	false	false

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a = 1; // true
6     int b = 0; // false
7     bool hasil;
8
9     cout << "a = " << a << endl;
10    cout << "b = " << b << endl;
11
12    // logika AND
13    hasil = a && b;
14    cout << "a && b = " << hasil << endl;
15
16    // logika OR
17    hasil = a || b;
18    cout << "a || b = " << hasil << endl;
19
20    // logika NOT
21    cout << "!a = " << !a << endl;
22 }
```

Command Prompt

```
D:\ALPROG\Pertemuan3>logika.exe
a = 1
b = 0
a && b = 0
a || b = 1
!a = 0

D:\ALPROG\Pertemuan3>
```

e) Operator Bitwise

Operator bitwise merupakan operator yang digunakan untuk operasi berdasarkan bit (biner) dari sebuah nilai. Operator bitwise terdiri dari:

Nama Operator	Simbol di Java
AND	&
OR	
XOR	^
NOT/komplemen	~
Left Shift	<<
Right Shift	>>

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a = 6, b = 3, hasil;
6
7     // menggunakan operator bitwise and
8     hasil = a & b;
9     cout << "a & b = " << hasil << endl;
10
11    // menggunakan operator bitwise or
12    hasil = a | b;
13    cout << "a | b = " << hasil << endl;
14
15    // menggunakan operator bitwise xor
16    hasil = a ^ b;
17    cout << "a ^ b = " << hasil << endl;
18
19    // menggunakan operator bitwise not
20    hasil = ~a;
21    cout << "~a = " << hasil << endl;
22
23    // menggunakan operator bitwise left shift
24    hasil = a << 1;
25    cout << "a << 1 = " << hasil << endl;
26
27    // menggunakan operator bitwise left shift
28    hasil = a >> 1;
29    cout << "a >> 1 = " << hasil << endl;
30 }
```

```

Command Prompt

D:\ALPROG\Pertemuan3>bitwise.exe
a & b = 2
a & b = 7
a ^ b = 5
~a = -7
a << 1 = 12
a >> 1 = 3

D:\ALPROG\Pertemuan3>_

```

- Bitwise AND

Bitwise AND

$$\begin{array}{r}
 6 \text{ & } 3 \\
 \textcircled{0110} \\
 \textcircled{0011} \\
 \hline
 \textcircled{0010} = 2
 \end{array}$$

Pernyataan 1	Pernyataan 2	Logika AND
true	true	true
true	false	false
false	true	false
false	false	false

- Bitwise OR

Bitwise OR

$$\begin{array}{r}
 6 \text{ | } 3 \\
 \textcircled{0110} \\
 \textcircled{0011} \\
 \hline
 \textcircled{0111} = 7
 \end{array}$$

Pernyataan 1	Pernyataan 2	Logika OR
true	true	true
true	false	true
false	true	true
false	false	false

- Bitwise XOR

Bitwise XOR

$$\begin{array}{r}
 6 \wedge 3 \\
 \textcircled{0110} \\
 \textcircled{0011} \\
 \hline
 0101 = 5
 \end{array}$$

Pernyataan 1	Pernyataan 2	Logika XOR
true	true	false
true	false	true
false	true	true
false	false	false

- Bitwise NOT

Bitwise NOT

$$\begin{array}{l}
 6 \rightarrow 0110 \\
 \sim 6 \rightarrow 1001 = -7
 \end{array}$$

bit untuk
simbol negatif
dalam komplemen dua

Komplemen Dua

1000	= -8
1001	= -7
1010	= -6
1100	= -4
1101	= -3
1110	= -2
1111	= -1
0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7

- Bitwise Left Shift

Left Shift

$6 = 0\ 1\ 1\ 0$

$6 \ll 1 = \underbrace{1\ 1}_{\text{geser 1 bit ke kiri}} \underbrace{0\ 0} = 12$

- Bitwise Right Shift

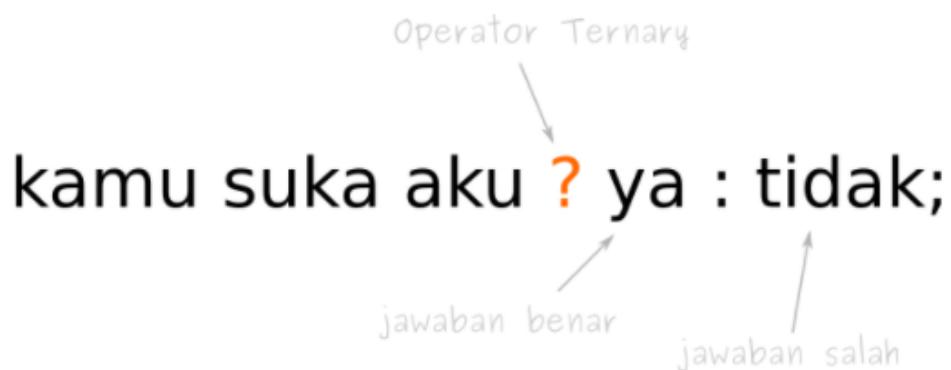
Right Shift

$6 = 0\ 1\ 1\ 0$

$6 \gg 1 = \underbrace{0\ 0}_{\text{geser 1 bit ke kanan}} \underbrace{1\ 1} = 3$

3. Operator Ternary

Operator Ternary - operator yang dikenakan untuk tiga buah nilai (operand)



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 4;
6
7     //menampilkan nilai a
8     cout << "nilai a adalah " << a << endl;
9
10    // menggunakan operator ternary
11    string hasil = a > 1 ? "benar": "salah";
12
13    cout << "a > 1 adalah " << hasil << endl;
14 }
```

Command Prompt

```
D:\ALPROG\Pertemuan3>ternary.exe
nilai a adalah 4
a > 1 adalah benar

D:\ALPROG\Pertemuan3>
```

Kerjakanlah Tugas Berikut !

- 1.Buatlah aplikasi menghitung luas dan keliling Persegi Panjang !**
- 2.Buatlah aplikasi menghitung luas dan keliling Lingkaran !**

MODUL PRAKTIKUM #4 - Percabangan

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengenal bentuk-bentuk percabagan dalam pemrograman dengan baik
- 2) Menjelaskan jenis-jenis bentuk percabangan secara benar
- 3) Menggunakan bentuk-bentuk percabangan dalam pembuatan program secara tepat
- 4) Mengembangkan bentuk-bentuk percabangan yang dikombinasikan dengan bentuk pernyataan lain dalam suatu program yang kompleks secara benar

B. Materi Percabangan C++

- Percabangan yaitu suatu pemilihan statement yang akan dieksekusi di mana pemilihan tersebut didasarkan atas kondisi tertentu.
- Statement kendali yang digunakan untuk mengambil suatu keputusan atau memilih bagian program yang akan dikerjakan sesuai dengan kondisi atau syarat yang diberikan.
- Statement kendali memerlukan suatu kondisi atau syarat sebagai dasar pengambilan keputusan. Salah satu kondisi yang umum digunakan adalah berupa keadaan benar atau salah (true or false).
- Selain percabangan, struktur ini juga disebut: *control flow*, *decision*, struktur kondisi.

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Penyeleksian kondisi dapat diibaratkan sebagai katup atau kran yang mengatur jalannya air. Bila katup terbuka maka air akan mengalir dan sebaliknya bila katup tertutup air tidak akan mengalir atau akan mengalir melalui tempat lain. Fungsi penyeleksian kondisi penting artinya dalam penyusunan bahasa C++ terutama untuk program yang kompleks.

Bahasa C++ ada 2 jenis percabangan,yaitu :

- 1) IF
 - Statement **if**
 - Statement **if-else**
 - Statement **if-else-if**
- 2) SWITCH Case

Dalam menuliskan kondisi/syarat selalu digunakan Operator Relasi/Pembanding atau Operator Logika sebagai sarana untuk melakukan proses pengecekan :

- Operator Relasi/Pembanding

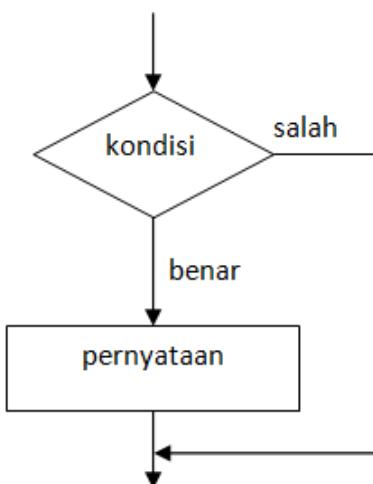
Operator	Arti
>	Lebih besar
<	Lebih kecil
==	Sama dengan
>=	Lebih atau sama
<=	Kurang atau sama
!=	Tidak sama dengan

- Operator Logika

Operator	Keterangan
&&	And (Semua kondisi harus bernilai benar)
	Or (Salah satu harus bernilai benar)
!	Not (Kebalikan dari kondisi sebenarnya)

➤ IF

Percabangan *if* merupakan percabangan yang hanya memiliki satu blok pilihan saat kondisi bernilai benar atau hanya mengakomodir jika syarat bernilai true/benar saja.



Contoh IF 1 :

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     string password;
6
7     cout << "===== Login Wifi =====" << endl;
8     cout << "Masukan password: ";
9     cin >> password;
10
11    // percabangan if
12    if (password == "amikom"){
13        cout << "Selamat login wifi berhasil !" << endl;
14    }
15 }
```

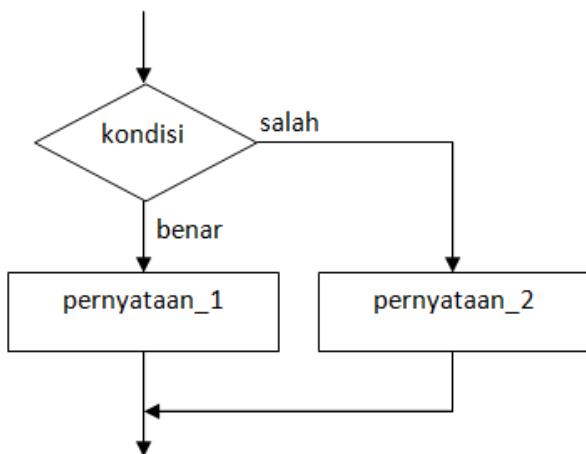
Contoh IF 2 :

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int nilai;
6
7     cout<<"Masukkan Nilai = ";
8     cin>>nilai;
9
10    // percabangan if
11    if (nilai>0) {
12        cout<<"Angkanya adalah bilangan positif";
13    }
14 }
```

➤ IF ELSE

Percabangan *if else* merupakan percabangan yang memiliki dua blok pilihan. Blok pilihan pertama untuk kondisi benar, dan pilihan kedua untuk kondisi salah (*else*).

```
if (kondisi)
{   blok_ini_dijalankan_jika_kondisi_benar;  }
else
{   blok_ini_dijalankan_jika_kondisi_salah;  }
```



Contoh IF Else 1 :

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     string password;
6
7     cout << "===== Login Wifi =====" << endl;
8     cout << "Masukan password: ";
9     cin >> password;
10
11    // percabangan if else
12    if (password == "amikom"){
13        cout << "Selamat login wifi berhasil !" << endl;
14    } else {
15        cout << "Password salah, coba lagi!" << endl;
16    }
17 }
```

Contoh IF Else 2 :

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     string nim, password;
6
7     cout << "===== Login Dashboard Mahasiswa =====" << endl;
8     cout << "Masukan nim: ";
9     cin >> nim;
10    cout << "Masukan password: ";
11    cin >> password;
12
13    // percabangan if else
14    if ((nim == "20.11.1111") && (password == "amikom")){
15        cout << "Selamat Login Dashboard Mahasiswa Berhasil !" << endl;
16    } else {
17        cout << "Kombinasi nim dan password salah, coba lagi!" << endl;
18    }
19 }
```

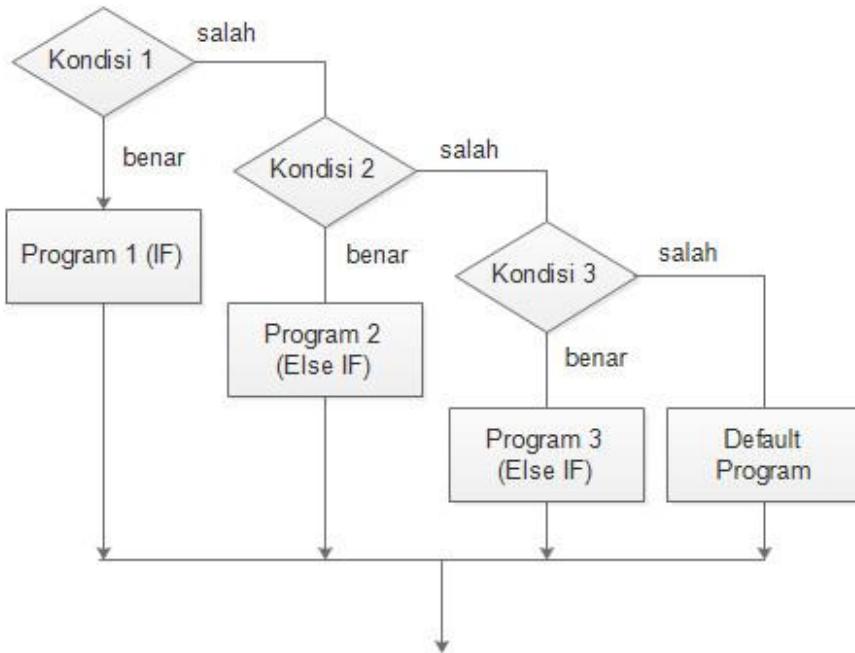
Contoh IF Else 3 :

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int nilai;
6     cout<<"Masukkan Nilai = ";
7     cin>>nilai;
8     if (nilai % 2 == 0)
9         cout<<nilai<<" adalah bilangan genap";
10    else
11        cout<<nilai<<" adalah bilangan ganjil";
12 }
```

➤ IF ELSE IF

Percabangan *if/else/if* merupakan percabangan yang memiliki lebih dari dua blok pilihan.

```
if (kondisi_1)
    {blok_disini_dijalankan_jika_kondisi_1_benar;}
else if (kondisi_2)
    {blok_disini_dijalankan_jika_kondisi_2_benar;}
else if (kondisi_3)
    {blok_disini_dijalankan_jika_kondisi_3_benar;}
else
    {blok_ini_dijalankan_jika_semua_kondisi_diatas_tdk_benar;}
```



Contoh IF Else IF 1 :

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int nilai;
6     string grade;
7
8     cout << "==== Program Grade Nilai ===" << endl;
9     cout << "Inputkan nilai akhir: ";
10    cin >> nilai;
11
12    // menggunakan percabangan if/esle/if
13    if (nilai > 80) {
14        grade = "A";
15    } else if (nilai > 60) {
16        grade = "B";
17    } else if (nilai > 40) {
18        grade = "C";
19    } else if (nilai > 20) {
20        grade = "D";
21    } else {
22        grade = "E";
23    }
24
25    cout << "Grade anda: " << grade << endl;
26 }
```

Contoh IF Else Bersarang (Nested) :

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     string username, password;
6
7     cout << "==== Welcome ===" << endl;
8     cout << "Username: ";
9     cin >> username;
10
11    if (username == "admin"){
12        cout << "Password: ";
13        cin >> password;
14        if (password == "amikom"){
15            cout << "Selamat datang, login berhasil" << endl;
16        } else {
17            cout << "Password salah, coba lagi" << endl;
18        }
19    } else {
20        cout << "Username tidak terdaftar" << endl;
21    }
22 }
```

➤ Switch Case

```
switch(variabel){  
    case <value>:  
        // blok kode  
        break;  
    case <value>:  
        // blok kode  
        break;  
    default:  
        // blok kode  
}
```

Contoh Switch Case :

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main(){  
5     char grade;  
6  
7     cout << "Inputkan grade: ";  
8     cin >> grade;  
9  
10    switch (grade){  
11        case 'A':  
12            cout << "Luar biasa!" << endl;  
13            break;  
14        case 'B':  
15            cout << "Bagus!" << endl;  
16            break;  
17        case 'C':  
18            cout << "Cukup!" << endl;  
19            break;  
20        case 'D':  
21        case 'E':  
22            cout << "Anda remidi" << endl;  
23            break;  
24        default:  
25            cout << "Grade Salah!" << endl;  
26    }  
27 }
```

➤ Percabangan Operator Ternary

- Percabangan menggunakan operator ternary merupakan bentuk lain dari percabangan if/else.
- Bisa dibilang: Bentuk singkatnya dari if/else.

```
(kondisi) ? true : false
```

Contoh Percabangan Ternary :

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int jawaban;
6
7     cout << "Berapakah hasil 3+4?" << endl;
8     cout << "jawab> ";
9     cin >> jawaban;
10
11    string hasil = (jawaban == 7) ? "Benar" : "Salah";
12    cout << "Jawaban anda: " << hasil << endl;
13 }
```

➤ Tugas : Kerjakanlah coding di bawah ini !!!

```
1 #include <iostream>
2 #include <stdlib.h>
3 using namespace std;
4
5 int main(){
6     float pilihan, panjang, lebar, jari;
7     const float phi = 3.14;
8     menu:
9     cout<<"======"<<endl;
10    cout<<"1.Menghitung Luas dan Keliling Persegi Panjang"<<endl;
11    cout<<"2.Menghitung Luas dan Keliling Lingkaran"<<endl;
12    cout<<"3.Exit/Keluar"<<endl;
13    cout<<"======"<<endl;
14    cout<<"Masukan Angka Pilihan Anda = ";
15    cin>>pilihan;
16    system("cls");
17
18    if (pilihan == 1) {
19        cout<<"Masukan Nilai Panjang \t\t= ";
20        cin>>panjang;
21        cout<<"Masukan Nilai Lebar \t\t= ";
22        cin>>lebar;
23        cout<<"Luas Persegi Panjang \t\t= "<< panjang*lebar<<endl;
24        cout<<"Keliling Persegi Panjang \t\t= "<< 2*(panjang+lebar)<<endl;
25        goto menu;
26
27    } else if (pilihan == 2) {
28        cout<<"Masukan Nilai Jari-jari \t\t= ";
29        cin>>jari;
30        cout<<"Luas Lingkaran \t\t\t= "<< phi*jari*jari<<endl;
31        cout<<"Keliling Lingkaran \t\t\t= "<< 2*phi*jari<<endl;
32        goto menu;
33
34    } else if (pilihan == 3) {
35        cin.ignore();
36
37    } else {
38        cin.ignore();
39        cout<<"Pilihan salah, tekan enter untuk melanjutkan";
40        cin.get();
41        system("cls");
42        goto menu;
43    }
44 }
```

MODUL PRAKTIKUM #5 - Perulangan

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengenal bentuk perulangan for, while, dan do while dengan benar.
- 2) Menjelaskan prinsip kerja bentuk-bentuk perulangan dengan tepat.
- 3) Dapat menggunakan bentuk-bentuk perulangan dalam aplikasinya pada pembuatan program secara tepat.
- 4) Dapat mengembangkan bentuk-bentuk perulangan dalam pemrograman secara benar.

B. Materi Percabangan C++

Perulangan atau yang sering disebut dengan “looping”, merupakan proses yang dilakukan secara berulang-ulang. Dengan menggunakan proses perulangan, penulisan kode program dapat menjadi lebih efektif dan efisien. Dalam bahasa C++, terdapat beberapa macam struktur perulangan, antara lain: *for*, *while* dan *do while*.

➤ Latar Belakang Perulangan

Jika kamu diminta menampilkan tulisan “Selamat datang” sebanyak 5 kali, apa yang kamu kerjakan ? Melakukan cout<<“Selamat datang”<<endl; sebanyak 5 kali ?

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout<<"Selamat datang"<<endl;
6     cout<<"Selamat datang"<<endl;
7     cout<<"Selamat datang"<<endl;
8     cout<<"Selamat datang"<<endl;
9     cout<<"Selamat datang"<<endl;
10 }
```

Apakah hal itu efektif dan efisien ?

➤ Kategori Perulangan

- counted loop (sudah diketahui jumlah perulangannya)
- uncounted loop (belum diketahui jumlah perulangannya)

➤ Jenis-jenis Perulangan

- For (counted loop)
- While (uncounted loop)
- Do while (uncounted loop)

➤ Perulangan FOR

Bentuk umum perulangan for adalah sebagai berikut :

```
for(inisialisasi;syarat;penambahan/pengurangan){
    pernyataan;
}
```

Keterangan :

- Inisialisasi :pernyataan untuk menyatakan keadaan awal dari variabel kontrol.
- Syarat :ekspresi relasi yang menyatakan kondisi untuk keluar dari perulangan.
- Penambahan/pengurangan :pengatur perubahan nilai variabel kontrol.

✓ Contoh For 1

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     for(int i=1; i<=5; i++)
6         cout<<"Selamat datang\n";
7 }
```

Selamat datang
Selamat datang
Selamat datang
Selamat datang
Selamat datang

✓ Contoh For 2

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     for(int i=1; i<=10; i++) {
6         cout<<"No ke = " <<i<<endl;
7     }
8 }
```

No ke = 1
No ke = 2
No ke = 3
No ke = 4
No ke = 5
No ke = 6
No ke = 7
No ke = 8
No ke = 9
No ke = 10

✓ Contoh For 3

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int i, n;
6     float nilai, rata, jml=0, kali=1;
7     cout<<"Input banyak nilai = ";
8     cin>>n;
9
10    for(i=1; i<=n; i++) {
11        cout<<"Input Nilai "<<i<<" = ";
12        cin>>nilai;
13        jml+=nilai;
14        //jml=jml+nilai;
15        kali*=nilai;
16        rata=jml/n;
17    }
18    cout<<"\nTotal nilai = "<<jml;
19    cout<<"\nTotal perkalian = "<<kali;
20    cout<<"\nRata nilai = "<<rata;
21 }
```

Input banyak nilai = 3

Input Nilai 1 = 2

Input Nilai 2 = 3

Input Nilai 3 = 4

Total nilai = 9

Total perkalian = 24

Rata nilai = 3

➤ Perulangan WHILE

Perulangan while akan terus berlanjut selama kondisinya bernilai benar (*true*) dan akan berhenti bila kondisinya bernilai salah.

- Bentuk umum perulangan while adalah :

while (syarat)

{ *instruksi*;

...

}

✓ Contoh While 1

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char ulangi = 'y';
6     int i = 0;
7
8     // perulangan while
9     while(ulangi == 'y'){
10         cout<<"Apakah kamu mau mengulang?\n";
11         cout<<"Jawab (y/t) : ";
12         cin >> ulangi;
13
14         // increment i
15         i++;
16     }
17
18     cout<<"Perulangan Selesai!\n";
19     cout<<"Kamu mengulang sebanyak "<<i<<" kali \n";
20 }
```

```

C:\ Command Prompt
D:\ALPROG\Pertemuan5>while1a.exe
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): t
Perulangan Selesai!
Kamu mengulang sebanyak 3 kali

D:\ALPROG\Pertemuan5>
```

➤ **Perulangan DO WHILE**

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

While vs Do While

While	Do while
<ul style="list-style-type: none">• Bisa jadi tidak akan pernah dikerjakan jika syarat tidak dipenuhi.• Dikarenakan sebelum instruksi dikerjakan, syarat dicek terlebih dahulu.	<ul style="list-style-type: none">• Minimal dikerjakan satu kali walaupun syarat tidak dipenuhi.• Dikarenakan instruksi dikerjakan dahulu, baru syarat dicek untuk melanjutkan perulangan.

✓ Contoh Do While 1

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char ulangi = 'y';
6     int i = 0;
7
8     do {
9         cout<<"Apakah kamu mau mengulang?\n";
10        cout<<"Jawab (y/t): ";
11        cin >> ulangi;
12
13        // increment i
14        i++;
15    } while(ulangi == 'y');
16
17    cout<<"Perulangan Selesai!\n";
18    cout<<"Kamu mengulang sebanyak "<<i<<" kali \n";
19 }
```

Command Prompt

```
D:\ALPROG\Pertemuan5>dowhile1a.exe
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): y
Apakah kamu mau mengulang?
Jawab (y/t): t
Perulangan Selesai!
Kamu mengulang sebanyak 3 kali

D:\ALPROG\Pertemuan5>
```

✓ Contoh Do While 2

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int i = 1;
6     do {
7         cout<<"Selamat datang\n";
8         i++;
9     } while(i<0);
10 }
```

Output :

Selamat datang

Catatan : Meskipun syaratnya bernilai salah/false, perulangan do while minimal akan tetap memproses *body perulangan* sebanyak 1 kali.

Good Luck !

MODUL PRAKTIKUM #6 – Perulangan Bersarang

A. Tujuan

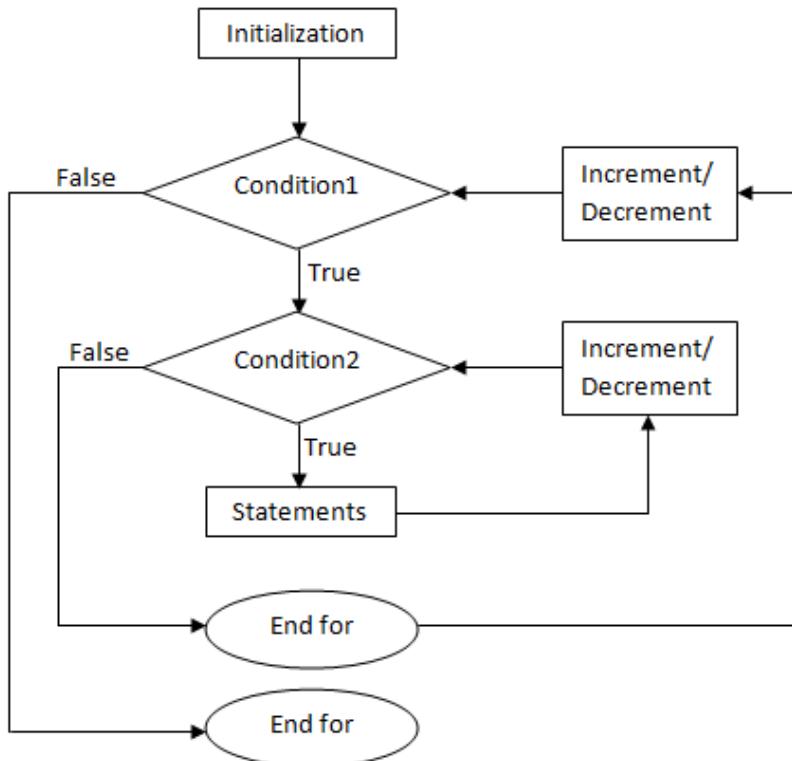
Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengenal bentuk perulangan for, while, dan do while dengan benar.
- 2) Menjelaskan prinsip kerja bentuk-bentuk perulangan dengan tepat.
- 3) Dapat menggunakan bentuk-bentuk perulangan dalam aplikasinya pada pembuatan program secara tepat.
- 4) Dapat mengembangkan bentuk-bentuk perulangan dalam pemrograman secara benar.

B. Materi Percabangan C++

Perulangan atau yang sering disebut dengan “looping”, merupakan proses yang dilakukan secara berulang-ulang. Dengan menggunakan proses perulangan, penulisan kode program dapat menjadi lebih efektif dan efisien. Dalam bahasa C++, terdapat beberapa macam struktur perulangan, antara lain: *for*, *while* dan *do while*.

➤ Flowchart Perulangan Bersarang FOR



➤ Perulangan Bersarang FOR

Bentuk umum perulangan for adalah sebagai berikut :

```

1  for (initialization; condition; increment/decrement) {
2    for (initialization; condition; increment/decrement) {
3      statement;
4    }
5    statement;
6  }

```

Keterangan :

- Inisialisasi : pernyataan untuk menyatakan keadaan awal dari variabel kontrol.
- Syarat : ekspresi relasi yang menyatakan kondisi untuk keluar dari perulangan.
- Penambahan/pengurangan : pengatur perubahan nilai variabel kontrol.

✓ Contoh Nested Looping For 1

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5    int tinggi,baris,kolom;
6    cout<<"Masukan jumlah tinggi = ";
7    cin>>tinggi;
8
9    for(baris=1; baris<=tinggi; baris++) {
10      for(kolom=1; kolom<=baris; kolom++) {
11        ...
12        cout<<"*";
13      }
14      cout<<endl;
15    }

```

D:\ALPROG\Pertemuan6>segitiga1.exe
Masukan jumlah tinggi = 5
*
**

✓ Contoh Nested Looping For 2

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=tinggi; kolom>=baris; kolom--) {
11            cout<<"*";
12        }
13        cout<<endl;
14    }
15 }
```

D:\ALPROG\Pertemuan6>segitiga2.exe
Masukan jumlah tinggi = 5

**
*

✓ Contoh Nested Looping For 3

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=tinggi; kolom>baris; kolom--) {
11            cout<<" ";
12        }
13
14        for(kolom=1; kolom<=baris; kolom++) {
15            cout<<"*";
16        }
17        cout<<endl;
18    }
19 }
```

D:\ALPROG\Pertemuan6>segitiga3.exe
Masukan jumlah tinggi = 5
*
**

✓ Contoh Nested Looping For 4

```

1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=1; kolom<baris; kolom++) {
11            cout<<" ";
12        }
13
14        for(kolom=tinggi; kolom>=baris; kolom--) {
15            cout<<"*";
16        }
17        cout<<endl;
18    }
19 }
```

D:\ALPROG\Pertemuan6>segitiga4.exe
Masukan jumlah tinggi = 5

**
*

✓ Contoh Nested Looping For 5

```

1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=tinggi; kolom>baris; kolom--) {
11            cout<<" ";
12        }
13
14        for(kolom=1; kolom<=baris*2-1; kolom++) {
15            cout<<"*";
16        }
17        cout<<endl;
18    }
19 }
```

D:\ALPROG\Pertemuan6>segitiga5.exe
Masukan jumlah tinggi = 5
*

✓ Contoh Nested Looping For 6

```

1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=1; kolom<=baris; kolom++) {
11            cout<<" ";
12        }
13        for(kolom=tinggi; kolom>=baris*2-tinggi; kolom--) {
14            cout<<"*";
15        }
16        cout<<endl;
17    }
18 }
```

D:\ALPROG\Pertemuan6>segitiga6.exe
 Masukan jumlah tinggi = 5

 *

✓ Contoh Nested Looping For 7

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int tinggi,baris,kolom;
6     cout<<"Masukan jumlah tinggi = ";
7     cin>>tinggi;
8
9     for(baris=1; baris<=tinggi; baris++) {
10        for(kolom=1; kolom<=baris; kolom++) {
11            cout<<"*";
12        }
13        cout<<endl;
14    }
15
16    for(baris=2; baris<=tinggi; baris++) {
17        for(kolom=tinggi; kolom>=baris; kolom--) {
18            cout<<"*";
19        }
20        cout<<endl;
21    }
22 }
```

D:\ALPROG\Pertemuan6>segitiga7.exe
 Masukan jumlah tinggi = 5
 *
 **

 **
 *

✓ Contoh Nested Looping For 8

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     int tinggi,baris,kolom;
5     cout<<"Masukan jumlah tinggi = ";
6     cin>>tinggi;
7     for(baris=1; baris<=tinggi; baris++) {
8         for(kolom=tinggi; kolom>baris; kolom--) {
9             cout<<" ";
10        }
11        for(kolom=1; kolom<=baris; kolom++) {
12            cout<<"*";
13        }
14        cout<<endl;
15    }
16    for(baris=2; baris<=tinggi; baris++) {
17        for(kolom=1; kolom<baris; kolom++) {
18            cout<<" ";
19        }
20        for(kolom=tinggi; kolom>=baris; kolom--) {
21            cout<<"*";
22        }
23        cout<<endl;
24    }
25 }
```

```
D:\ALPROG\Pertemuan6>segitiga8.exe
Masukan jumlah tinggi = 5
*
**
***
****
*****
****
 ***
 **
 *
```

✓ Contoh Nested Looping For 9

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     int tinggi,baris,kolom;
5     cout<<"Masukan jumlah tinggi = ";
6     cin>>tinggi;
7     for(baris=1; baris<=tinggi; baris++) {
8         for (kolom=tinggi; kolom>baris; kolom--) {
9             cout<<" ";
10        }
11        for (kolom=1; kolom<=baris*2-1; kolom++) {
12            cout<<"*";
13        }
14        cout<<endl;
15    }
16    for(baris=2; baris<=tinggi; baris++) {
17        for(kolom=1; kolom<baris; kolom++) {
18            cout<<" ";
19        }
20        for(kolom=tinggi; kolom>=baris*2-tinggi; kolom--) {
21            cout<<"*";
22        }
23        cout<<endl;
24    }
25 }
```

```
D:\ALPROG\Pertemuan6>segitiga9.exe
Masukan jumlah tinggi = 5
*
 ***
 ****
 *****
 ******
 *****
 ****
 ***
 *
```

➤ Perulangan Bersarang WHILE

Perulangan while akan terus berlanjut selama kondisinya bernilai benar (*true*) dan akan berhenti bila kondisinya bernilai salah.

- Bentuk umum perulangan bersarang while adalah :

```
1 initialization;
2 while (condition1){
3     while (condition2){
4         statement2;
5         increment/decrement2;
6     }
7     statement1;
8     increment/decrement1;
9 }
```

✓ Contoh Nested Looping While

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi, baris, kolom;
6     cout << "Tinggi segi tiga : ";
7     cin >> tinggi;
8     cout << endl;
9     baris = 1;
10    while (baris <= tinggi){
11        kolom = 1;
12        while (kolom <= baris) {
13            cout << '*';
14            kolom++;
15        }
16        cout << endl;
17        baris++;
18    }
19 }
```

➤ Perulangan Bersarang DO-WHILE

- Bentuk umum perulangan bersarang do-while adalah :

```
1 initialization1;
2 do{
3     initialization2;
4     do{
5         statement2;
6     }while (condition2);
7     statement1;
8 } while (condition1);
```

✓ Contoh Nested Looping Do-While

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int tinggi, baris, kolom;
6     cout << "Tinggi segi tiga : ";
7     cin >> tinggi;
8     cout << endl;
9     baris = 1;
10    do {
11        kolom = 1;
12        do {
13            cout << '*';
14            kolom++;
15        } while (kolom <= baris);
16        cout << endl;
17        baris++;
18    } while (baris <= tinggi);
19}
```

Good Luck !

MODUL PRAKTIKUM #8 – Flowchart and Pseudocode

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Memahami cara penyajian algoritma dengan Flowchart, Pseudocode dan Bahasa Deskriptif
- 2) Mengenal bentuk/simbol dari Flowchart
- 3) Mampu membuat Flowchart dengan benar
- 4) Mampu menjelaskan proses/bagian-bagian Flowchart

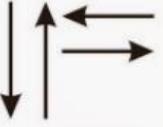
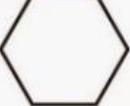
B. Materi

Bentuk penyajian untuk algoritma dibagi menjadi 3 (tiga) bentuk penyajian, yaitu :

- Algoritma dengan Flowchart
- Algoritma dengan Pseudocode
- Algoritma dengan Bahasa Deskriptif

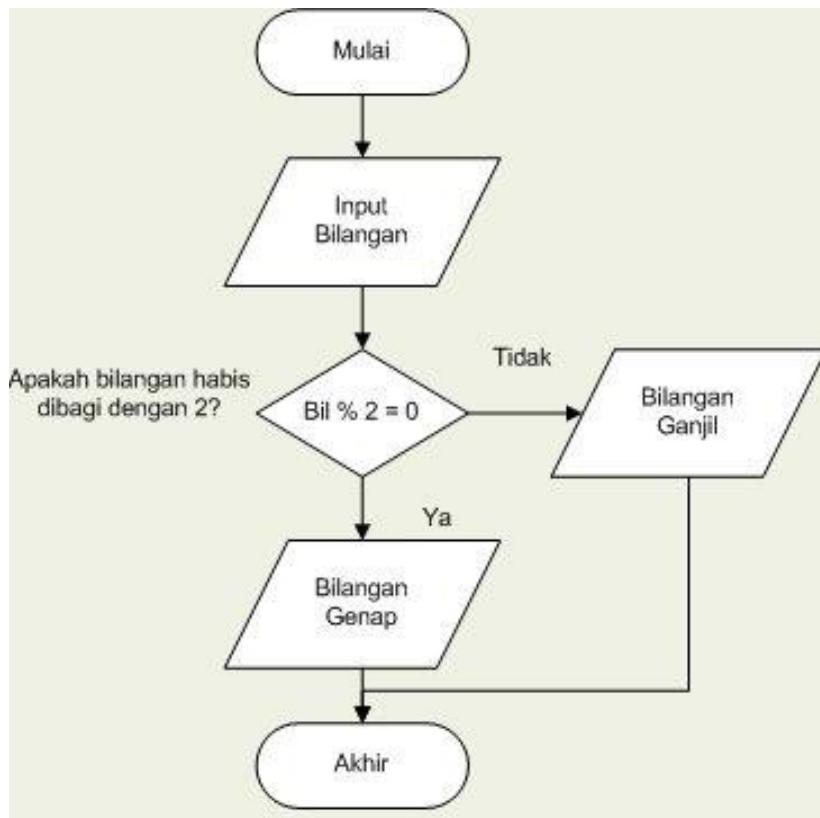
➤ Flowchart

Flowchart (Diagram Alir) adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.

	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

- Contoh Flowchart



➤ **Pseudocode**

Penyajian algoritma dengan *pseudocode* berarti menggunakan kode yang mirip dengan kode pemrograman yang sebenarnya.

○ **Contoh Pseudocode**

```
DEKLARASI
a,b: Integer

ALGORITMA
a ← 15 / (3 + 2)
b ← a / 2
IF (a + b = 8 / 2) THEN
    Write(a)
ELSE
    Write(b)
END IF
```

➤ **Bahasa Deskriptif**

Sifat Umum :

- Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
- Tidak tergantung pada suatu bahasa pemrograman.

○ **Contoh Bahasa Deskriptif**

Contoh : Menghitung rata-rata tiga buah data, Algoritma dengan struktur bahasa Indonesia :

- 1) Baca bilangan a, b, dan c
- 2) Jumlahkan ketiga bilangan tersebut
- 3) Bagi jumlah tersebut dengan 3
- 4) Tulis hasilnya

○ **Contoh Bahasa Deskriptif**

Soal : Pak Tani hendak pulang dari ladang membawa sekeranjang sayur bersama 2 peliharaan yaitu macan dan kambing. Untuk bisa sampai ke rumah dia harus menyeberang sungai menggunakan sampan dengan kapasitas 2 muatan. Faktanya adalah, jika Pak Tani membiarkan hanya kambing dan macan berdua, maka kambing akan dimakan macan. Bila kambing bersama sayur, maka sayur akan dimakan kambing. Jika ada Pak Tani maka tidak akan terjadi kejadian makan memakan. Bagaimana cara Pak Tani dapat menyeberang bersama kedua peliharaan dan sekeranjang sayur? Tuliskan langkahnya menggunakan kalimat deskriptif.

Jawaban :

Pak tani membawa kambing dari X ke Y

Pak tani menyeberang dari Y ke X

Pak tani membawa macan dari X ke Y

Pak tani membawa kambing dari Y ke X

Pak tani membawa sayur dari X ke Y

Pak tani menyeberang dari Y ke X

Pak tani membawa kambing dari X ke Y

ATAU

Pak tani membawa kambing dari X ke Y

Pak tani menyeberang dari Y ke X

Pak tani membawa sayur dari X ke Y

Pak tani membawa kambing dari Y ke X

Pak tani membawa macan dari X ke Y

Pak tani menyeberang dari Y ke X

Pak tani membawa kambing dari X ke Y

C. Tugas

Buatlah flowchart dari soal responsi praktikum yang sudah anda kerjakan sebelumnya, sesuai NIM masing-masing !!!

1) Soal NIM Ganjil



```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int i, n;
6     float nilai, nilaiterkecil;
7     cout<<"Input banyak nilai = ";
8     cin>>n;
9
10    for(i=1; i<=n; i++) {
11        cout<<"Input Nilai "<<i<<" = ";
12        cin>>nilai;
13        if (i==1) {
14            nilaiterkecil = nilai;
15        }
16
17        if (nilai<nilaiterkecil) {
18            nilaiterkecil = nilai;
19        }
20    }
21
22    cout<<"\nNilai Terkecil adalah = "<<nilaiterkecil;
23}

```

2) Soal NIM Genap



```

1 #include<iostream>
2 using namespace std;
3
4 int main(){
5     int i, n;
6     float nilai, nilaiterbesar;
7     cout<<"Input banyak nilai = ";
8     cin>>n;
9
10    for(i=1; i<=n; i++) {
11        cout<<"Input Nilai "<<i<<" = ";
12        cin>>nilai;
13        if (i==1) {
14            nilaiterbesar = nilai;
15        }
16
17        if (nilai>nilaiterbesar) {
18            nilaiterbesar = nilai;
19        }
20    }
21
22    cout<<"\nNilai Terbesar adalah = "<<nilaiterbesar;
23}

```

MODUL PRAKTIKUM #9 – Array

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Memahami konsep array dalam pemrograman C++ secara benar.
- 2) Mengenal implementasi array dengan baik.
- 3) Dapat menggunakan bentuk-bentuk array dalam aplikasinya pada pembuatan program secara tepat.
- 4) Dapat mengembangkan bentuk-bentuk array dalam pemrograman secara benar.

B. Materi

Pengertian Array yaitu :

- Array (larik) adalah variabel yang mampu menyimpan banyak data dalam 1 variabel dengan tipe data yang sama.
- Array memiliki elemen-elemen yang digunakan untuk menyimpan banyak data.
- Cara memanggil data array pada elemen tertentu adalah dengan mengakses index elemen data tersebut.
- Index array secara default dimulai dari 0.

➤ Ilustrasi

0	1	2	3	4	5	6	7	indeks
10	44	2	76	1	56	70	7	value

➤ Keunggulan

- Dapat menyimpan banyak data dalam satu variable array.
- Sembarang elemen di array dapat diacu secara langsung.
- Jika berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemen-elemen tetangga, baik elemen pendahulu atau elemen penerus

➤ Kelemahan

- Array harus bertipe homogen. Kita tidak dapat mempunyai array dimana satu elemen adalah karakter, elemen lain bilangan, dan elemen lain adalah tipe-tipe lain.

➤ Latihan Array (1)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat array kosong
6     int nilai[5];
7
8     // mengisi array
9     nilai[0] = 32;
10    nilai[1] = 42;
11    nilai[2] = 76;
12    nilai[3] = 31;
13    nilai[4] = 57;
14
15    // mencetak isi array
16    cout << "Nilai ke-1: " << nilai[0] << endl;
17    cout << "Nilai ke-2: " << nilai[1] << endl;
18    cout << "Nilai ke-3: " << nilai[2] << endl;
19    cout << "Nilai ke-4: " << nilai[3] << endl;
20    cout << "Nilai ke-5: " << nilai[4] << endl;
21 }
```

➤ Latihan Array (2)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat dan mengisi array
6     int nilai[5] = {32,42,76,31,57};
7
8     // mencetak isi array
9     cout << "Nilai ke-1: " << nilai[0] << endl;
10    cout << "Nilai ke-2: " << nilai[1] << endl;
11    cout << "Nilai ke-3: " << nilai[2] << endl;
12    cout << "Nilai ke-4: " << nilai[3] << endl;
13    cout << "Nilai ke-5: " << nilai[4] << endl;
14 }
```

➤ Latihan Array (3)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat dan mengisi array
6     int nilai[5] = {32,42,76,31,57};
7
8     // mencetak isi array
9     for (int i=0; i<=4; i++) {
10         cout << "Nilai ke-"<<i+1<<": " << nilai[i] << endl;
11     }
12 }
```

➤ Latihan Array (4)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat dan mengisi array
6     char huruf[5] = {'a', 'b', 'c', 'd', 'e'};
7
8     // mencetak isi array
9     for (int i=0; i<=4; i++) {
10         cout << "Nilai ke-"<<i+1<<": " << huruf[i] << endl;
11     }
12 }
```

➤ Latihan Array (5)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string buah[5] = {"Apel", "Jeruk", "Jambu", "Durian", "Mangga"};
6
7     for (int i=0; i<=4; i++) {
8         cout << "Nilai ke-<<i+1<<: " << buah[i] << endl;
9     }
10 }
```

➤ Latihan Array (6)

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int i, n;
6     cout<<"Input banyak nilai = ";
7     cin>>n;
8     int nilai[n]; //membuat array
9
10    //menyimpan nilai ke array
11    for(i=0; i<n; i++) {
12        cout<<"Input Nilai ke "<<i+1<< " = ";
13        cin>>nilai[i];
14    }
15
16    //menampilkan nilai dari array
17    for(i=0; i<n; i++) {
18        cout<<"Nilai ke "<<i+1<< " = "<<nilai[i]<<endl;
19    }
20 }
```

➤ Latihan Array (7)

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     int i, n, jumlah=0, terkecil, terbesar;
5     cout<<"Input banyak nilai = ";
6     cin>>n;
7     int nilai[n]; //membuat array
8     for(i=0; i<n; i++) {
9         cout<<"Input Nilai ke "<<i+1<<" = ";
10        cin>>nilai[i];
11        jumlah = jumlah + nilai[i];
12        if (i==0) {
13            terkecil = nilai[0];
14            terbesar = nilai[0];
15        }
16        if (nilai[i]<terkecil) {
17            terkecil = nilai[i];
18        }
19        if (nilai[i]>terbesar) {
20            terbesar = nilai[i];
21        }
22    }
23    cout<<"\nData Nilai";
24    for(i=0; i<n; i++) {
25        cout<<" "<<nilai[i];
26    }
27    cout<<"\nJumlah Nilainya Adalah = "<<jumlah;
28    cout<<"\nNilai Terkecil = "<<terkecil;
29    cout<<"\nNilai Terbesar = "<<terbesar;
30 }
```

Good Luck !

MODUL PRAKTIKUM #10 – Array Multi Dimensi

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Memahami konsep array multi dimensi dalam pemrograman C++ secara benar.
- 2) Mengenal implementasi array multi dimensi dengan baik.
- 3) Dapat menggunakan bentuk-bentuk array multi dimensi dalam aplikasinya pada pembuatan program secara tepat.
- 4) Dapat mengembangkan bentuk-bentuk array multi dimensi dalam pemrograman secara benar.

B. Materi

Pengertian Array yaitu :

- Array (larik) adalah variabel yang mampu menyimpan banyak data dalam 1 variabel dengan tipe data yang sama.
- Array memiliki elemen-elemen yang digunakan untuk menyimpan banyak data.
- Cara memanggil data array pada elemen tertentu adalah dengan mengakses index elemen data tersebut.
- Index array secara default dimulai dari 0.

➤ Ilustrasi Array 2 Dimensi

Angka	kolom[0]	kolom[1]	kolom[2]	kolom[3]	kolom[4]	kolom[5]	Kolom[6]
Baris[0]	0	1	2	3	4	5	6
Baris[1]	10	11	12	13	14	15	16
Baris[2]	26	27	28	29	30	31	32

$$\text{Angka}[1][2] = 12$$

➤ Latihan Array 2 Dimensi (1)

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat array 2 dimensi kosong
6     int a[2][3]; // 2 baris dan 3 kolom
7
8     // mengisi nilai array 2 dimensi
9     a[0][0] = 5;
10    a[0][1] = 10;
11    a[0][2] = 15;
12    a[1][0] = 20;
13    a[1][1] = 25;
14    a[1][2] = 30;
15
16    // mencetak isi array
17    cout<<"Nilai index ke-[0][0]: "<<a[0][0]<<endl;
18    cout<<"Nilai index ke-[0][1]: "<<a[0][1]<<endl;
19    cout<<"Nilai index ke-[0][2]: "<<a[0][2]<<endl;
20    cout<<"Nilai index ke-[1][0]: "<<a[1][0]<<endl;
21    cout<<"Nilai index ke-[1][1]: "<<a[1][1]<<endl;
22    cout<<"Nilai index ke-[1][2]: "<<a[1][2]<<endl;
23 }
```

➤ Latihan Array 2 Dimensi (2)

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // membuat dan mengisi nilai array 2 dimensi
6     int a[2][3] = {{5,10,15}, {20,25,30}};
7
8     // mencetak isi array
9     cout<<"Nilai index ke-[0][0]: "<<a[0][0]<<endl;
10    cout<<"Nilai index ke-[0][1]: "<<a[0][1]<<endl;
11    cout<<"Nilai index ke-[0][2]: "<<a[0][2]<<endl;
12    cout<<"Nilai index ke-[1][0]: "<<a[1][0]<<endl;
13    cout<<"Nilai index ke-[1][1]: "<<a[1][1]<<endl;
14    cout<<"Nilai index ke-[1][2]: "<<a[1][2]<<endl;
15 }
```

➤ Latihan Array 2 Dimensi (3)

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a[2][3] = {{5,10,15}, {20,25,30}};
6
7     for (int i=0; i<=1; i++) {
8         for (int j=0; j<=2; j++) {
9             cout<<"Nilai index ke-["<<i<<""]["<<j<<"] : "<<a[i][j]<<endl;
10        }
11    }
12 }
```

➤ Latihan Array 2 Dimensi (4)

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a[2][3]; //membuat wadah array 2 dimensi
6
7     //mengisi nilai array 2 dimensi
8     for (int i=0; i<=1; i++) {
9         for (int j=0; j<=2; j++) {
10            cout<<"Inputkan nilai index ke-["<<i<<""]["<<j<<"] : ";
11            cin>>a[i][j];
12        }
13    }
14
15     //menampilkan nilai array 2 dimensi
16     for (int i=0; i<=1; i++) {
17         for (int j=0; j<=2; j++) {
18             cout<<"Nilai index ke-["<<i<<""]["<<j<<"] : "<<a[i][j]<<endl;
19         }
20     }
21 }
```

➤ Latihan Array 2 Dimensi (5)

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a[2][3]; //membuat ordo matrik 2x3
6
7     //mengisi nilai ordo matrik 2x3
8     for (int i=0; i<=1; i++) {
9         for (int j=0; j<=2; j++) {
10            cout<<"Inputkan nilai index ke-["<<i<<""]["<<j<<"] : ";
11            cin>>a[i][j];
12        }
13    }
14
15     //menampilkan nilai ordo matrik 2x3
16     cout<<"\nNilai matrik 2x3";
17     for (int i=0; i<=1; i++) {
18         cout<<endl;
19         for (int j=0; j<=2; j++) {
20             cout<<a[i][j]<<" ";
21         }
22     }
23 }
```

➤ Latihan Array 2 Dimensi (6)

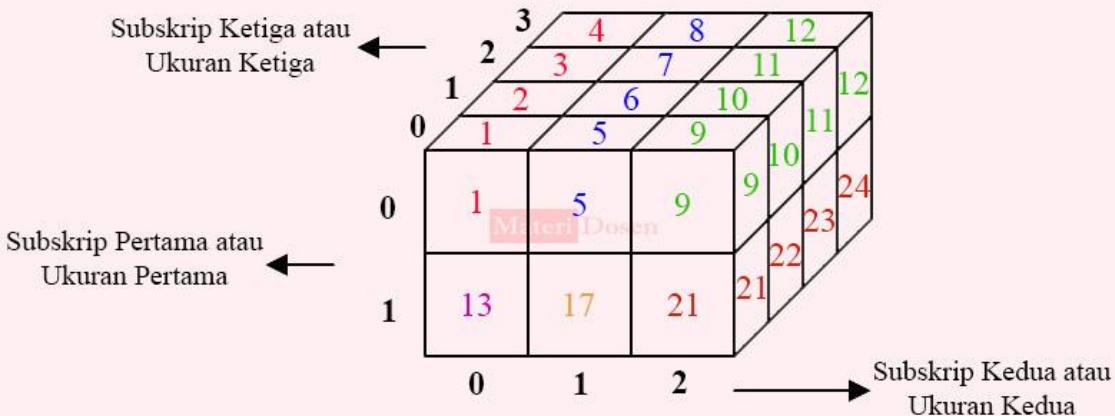
```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int a[2][3]; //membuat ordo matrik 2x3
6     int b[2][3]; //membuat ordo matrik 2x3
7
8     //mengisi nilai matrik A
9     cout<<"Inputkan nilai matrik A"<<endl;
10    for (int i=0; i<=1; i++) {
11        for (int j=0; j<=2; j++) {
12            cout<<"Inputkan nilai index ke-["<<i<<""] ["<<j<<"] : ";
13            cin>>a[i][j];
14        }
15    }
16
17    //mengisi nilai matrik B
18    cout<<"\nInputkan nilai matrik B"<<endl;
19    for (int i=0; i<=1; i++) {
20        for (int j=0; j<=2; j++) {
21            cout<<"Inputkan nilai index ke-["<<i<<""] ["<<j<<"] : ";
22            cin>>b[i][j];
23        }
24    }
25
26    //menampilkan matrik A
27    cout<<"\nNilai matrik A";
28    for (int i=0; i<=1; i++) {
29        cout<<endl;
30        for (int j=0; j<=2; j++) {
31            cout<<a[i][j]<<" ";
32        }
33    }
34
35    //menampilkan matrik B
36    cout<<"\n\nNilai matrik B";
37    for (int i=0; i<=1; i++) {
38        cout<<endl;
39        for (int j=0; j<=2; j++) {
40            cout<<b[i][j]<<" ";
41        }
42    }
43
44    //menampilkan penjumlahan matrik A+B
45    cout<<"\n\nHasil penjumlahan matrik A+B";
46    for (int i=0; i<=1; i++) {
47        cout<<endl;
48        for (int j=0; j<=2; j++) {
49            cout<<a[i][j]+b[i][j]<<" ";
50        }
51    }
52 }
```

➤ Latihan Array 2 Dimensi (7)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string data[3][3];
6
7     for (int i=0; i<=2; i++) {
8         for (int j=0; j<=2; j++) {
9             if (j==0) {
10                 cout<<"Nama Mahasiswa : ";
11                 cin>>data[i][j];
12             } else if (j==1) {
13                 cout<<"Nilai UTS : ";
14                 cin>>data[i][j];
15             } else if (j==2) {
16                 cout<<"Nilai UAS : ";
17                 cin>>data[i][j];
18             }
19         }
20     }
21
22     cout<<"\nNama\tUTS\tUAS";
23     for (int i=0; i<=2; i++) {
24         cout<<endl;
25         for (int j=0; j<=2; j++) {
26             cout<<data[i][j]<<"\t";
27         }
28     }
29 }
```

➤ Ilustrasi Array 3 Dimensi

```
Int LARIK [2][3][4] = {{{1,2,3,4},{5,6,7,8},{9,10,11,12}},{ {13,14,15,16},{17,18,19,20},{21,22,23,24}}};
```



Ilustrasi Array Tiga dimensi dengan 24 elemen

➤ Latihan Array 1 Dimensi (1)

```

1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     // membuat array 3 dimensi kosong
6     int a[2][3][4];
7
8     // mengisi nilai array 3 dimensi
9     a[0][0][0] = 1;    a[1][0][0] = 13;
10    a[0][0][1] = 2;   a[1][0][1] = 14;
11    a[0][0][2] = 3;   a[1][0][2] = 15;
12    a[0][0][3] = 4;   a[1][0][3] = 16;
13    a[0][1][0] = 5;   a[1][1][0] = 17;
14    a[0][1][1] = 6;   a[1][1][1] = 18;
15    a[0][1][2] = 7;   a[1][1][2] = 19;
16    a[0][1][3] = 8;   a[1][1][3] = 20;
17    a[0][2][0] = 9;   a[1][2][0] = 21;
18    a[0][2][1] = 10;  a[1][2][1] = 22;
19    a[0][2][2] = 11;  a[1][2][2] = 23;
20    a[0][2][3] = 12;  a[1][2][3] = 24;
21
22
23     // mencetak salah satu isi array
24     cout<<"Nilai index ke-[1][2][3]: "<<a[1][2][3]<<endl;
25 }
```

➤ Latihan Array 2 Dimensi (2)

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     // membuat dan memberi nilai array 3 dimensi
6     int a[2][3][4] = {{{1,2,3,4}, {5,6,7,8},{9,10,11,12}},
7     | | | | {{13,14,15,16}, {17,18,19,20},{21,22,23,24}}};
8
9     // mencetak salah satu isi array
10    cout<<"Nilai index ke-[1][2][3]: "<<a[1][2][3]<<endl;
11 }
```

➤ Latihan Array 3 Dimensi (3)

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     // membuat dan memberi nilai array 3 dimensi
6     int a[2][3][4] = {{{1,2,3,4}, {5,6,7,8},{9,10,11,12}},
7     | | | | {{13,14,15,16}, {17,18,19,20},{21,22,23,24}}};
8
9     // mencetak nilai array 3 dimensi
10    for (int i=0; i<=1; i++) {
11        for (int j=0; j<=2; j++) {
12            for (int k=0; k<=3; k++) {
13                cout<<"Nilai index ke-["<<i<<" ] ["<<j<<" ] ["<<k<<" ] : "<<a[i][j][k]<<endl;
14            }
15        }
16    }
17 }
```

Good Luck !

MODUL PRAKTIKUM #11 – Search and Sort

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Memahami konsep pencarian data
- 2) Dapat mengembangkan konsep pencarian data
- 3) Dapat menggunakan konsep pencarian data dengan benar
- 4) Memahami konsep mengurutkan data
- 5) Dapat mengembangkan konsep mengurutkan data
- 6) Dapat menggunakan konsep mengurutkan data dengan benar

B. Materi

- **Search (mencari)**
 - Linear/Sequential Search
 - Mampu untuk data yang acak
 - Secara umum lebih lambat dari binary search
 - Binary Search
 - Hanya untuk data yang sudah diurutkan
 - Secara umum lebih cepat dari linear search
- **Sort (mengurutkan)**

➤ Contoh Linear/Sequential Search (1)

```
1 #include <iostream>
2 using namespace std;
3
4 int main () {
5     int data[5]={20,37,42,77,89};
6     int cari, ditemukan=0;
7
8     cout<<"Data Array\n";
9     for (int i=0; i<5; i++){
10         cout<<data[i]<< " ";
11     }
12
13     cout<<"\nMasukan nilai yg anda cari : ";
14     cin>>cari;
15
16     for (int i=0; i<5; i++) {
17         if (data[i] == cari) {
18             ditemukan = 1;
19             cout<<"Nilai yg dicari terdapat pada data ke "<<i+1;
20             break;
21         }
22     }
23     if (ditemukan == 0) {
24         cout<<"Nilai yang dicari tidak ditemukan";
25     }
26 }
```

➤ Contoh Linear/Sequential Search (2)

```
1 #include <iostream>
2 using namespace std;
3
4 int main () {
5     int data[10]={20,37,42,20,89,20,37,11,77,92};
6     int cari, ditemukan=0, hasilcari[10];
7     cout<<"Data Array\n";
8     for (int i=0; i<10; i++){
9         cout<<data[i]<< " ";
10    }
11
12    cout<<"\nMasukan nilai yg anda cari : ";
13    cin>>cari;
14    for (int i=0; i<10; i++) {
15        if (data[i] == cari) {
16            ditemukan++;
17            hasilcari[ditemukan] = i;
18        }
19    }
20    if (ditemukan == 0) {
21        cout<<"Nilai yang dicari tidak ditemukan";
22    } else {
23        cout<<"Nilai yg dicari terdapat pada data ke "<<endl;
24        for (int i=1; i<=ditemukan; i++){
25            cout<<hasilcari[i]+1<< " ";
26        }
27    }
28 }
```

➤ Contoh Binary Search

```
1 #include <iostream>
2 using namespace std;
3 main(){
4     int target;
5     int data[10]={11,22,33,44,55,66,77,88,99,100};
6     int low, mid, high, ditemukan=0;
7     cout<<"Data Array\n";
8     for (int i=0; i<10; i++){
9         cout<<data[i]<< " ";
10    }
11    cout<<"\nMasukan angka yang dicari: ";
12    cin>>target;
13    low=0;
14    high=9;
15    while(low<=high){
16        mid=(low+high)/2;
17        if(target==data[mid]){
18            ditemukan = 1;
19            cout<<"Nilai yg dicari terdapat pada data ke "<<mid+1;
20            break;
21        } else if(target>data[mid]){
22            low=mid+1;
23        } else if(target<data[mid]){
24            high=mid-1;
25        }
26    }
27    if (ditemukan == 0) {
28        cout<<"Nilai yang dicari tidak ditemukan";
29    }
30 }
```

➤ Contoh Sort

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     int data[10], i, j, tmp;
5     cout<<"Program Mengurutkan Angka Kecil Ke Besar"=><endl;
6     for(i=0; i<10; i++) {
7         cout<<"Masukkan Angka ke "<<(i+1)<<" : ";
8         cin>>data[i];
9     }
10    cout<<"Data sebelum diurutkan : "<<endl;
11    for(i=0; i<10; i++) {
12        cout<<data[i]<<" ";
13    }
14
15    for(i=0; i<9; i++) {
16        for(j=i+1; j<10; j++) {
17            if(data[i]>data[j]) {
18                tmp = data[i];
19                data[i] = data[j];
20                data[j] = tmp;
21            }
22        }
23    }
24
25    cout<<"\nData setelah diurutkan : "<<endl;
26    for(i=0; i<10; i++) {
27        cout<<data[i]<<" ";
28    }
29 }
```

Good Luck !

MODUL PRAKTIKUM #12 – Fungsi

A. Tujuan

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Memahami konsep fungsi dalam pemrograman C++ secara benar.
- 2) Mengenal bentuk fungsi dalam pemrograman C++ secara benar.
- 3) Dapat membuat fungsi sendiri dalam aplikasinya pada pembuatan program secara tepat.
- 4) Dapat mengembangkan bentuk-bentuk fungsi dalam pemrograman secara benar.

B. Materi

- Latar Belakang
 - Ketika program kita sudah sangat besar dan kompleks maka akan semakin sulit membacanya?
 - Oleh karena itu dibutuhkan fungsi yang digunakan untuk membungkus program menjadi bagian-bagian kecil.
 - Tujuannya agar program tidak menumpuk pada fungsi main() saja.
- Fungsi
 - Fungsi adalah suatu teknik pemrograman di mana program yang biasanya cukup besar dibagi-bagi menjadi beberapa bagian program yang lebih kecil sehingga akan mudah dipahami dan dapat digunakan kembali, baik untuk program itu sendiri maupun program lain yang memiliki proses yang sama.
 - Fungsi yang selalu ada pada program C++ adalah fungsi main() karena fungsi tersebutlah yang pertama kali akan dieksekusi/dijalankan.
- Kelebihan Fungsi
 - Program lebih pendek
 - Mudah dibaca dan dimengerti
 - Mudah didokumentasi
 - Mengurangi kesalahan dan mudah mencari kesalahan
 - Kesalahan yang terjadi bersifat “local”

➤ Standard Library Function

- Yaitu fungsi-fungsi yang telah disediakan oleh C++ dalam file-file header atau librarynya.
- Misalnya: clrscr()
- Untuk function ini kita harus mendeklarasikan terlebih dahulu library yang akan digunakan, yaitu dengan menggunakan preprosesor.
- Misalnya: #include <conio.h>

➤ Programmer-Defined Function

- Adalah function yang dibuat oleh programmer sendiri.
- Function ini memiliki nama tertentu yang unik dalam program, letaknya terpisah dari program utama (fungsi main)
- Terdiri dari fungsi void dan non void

➤ Fungsi yang tidak mengembalikan nilai (void)

- Disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut.
- Tidak memiliki nilai kembalian fungsi
- Ciri:
 - tidak adanya keyword return.
 - tidak adanya tipe data di dalam deklarasi fungsi.
 - menggunakan keyword void.

➤ Fungsi yang mengembalikan nilai (nonvoid)

- Disebut non-void karena mengembalikan nilai kembalian yang berasal dari keluaran hasil proses function tersebut.
- Memiliki nilai kembalian
- Ciri:
 - ada keyword return
 - ada tipe data yang mengawali deklarasi fungsi
 - tidak ada keyword void

➤ Contoh Void (1)

```
1 #include <iostream>
2 using namespace std;
3
4 //membuat fungsi hello()
5 void hello() {
6     cout<<"Hello Selamat Datang!\n";
7 }
8
9 int main() {
10    //memanggil fungsi hello()
11    hello();
12 }
```

➤ Contoh Void (2)

```
1 #include <iostream>
2 using namespace std;
3
4 //membuat fungsi hello()
5 void hello() {
6     cout<<"Hello Selamat Datang!\n";
7 }
8
9 int main() {
10    //memanggil fungsi hello()
11    hello();
12    hello();
13    hello();
14    hello();
15    hello();
16 }
```

➤ Contoh Void (3)

```
1 #include <iostream>
2 using namespace std;
3
4 //membuat fungsi hello() yg memiliki parameter nilai
5 void hello(string nama){
6     cout<<"Hello Selamat Datang "<<nama<<"!\n";
7 }
8
9 int main(){
10    //memanggil fungsi hello() dgn memberi nilai parameter
11    hello("Andi");
12    hello("Ani");
13    hello("Ana");
14    hello("Ali");
15    hello("Adi");
16 }
```

➤ Contoh Void (4)

```
1 #include <iostream>
2 using namespace std;
3
4 //membuat fungsi yg memiliki parameter nilai
5 void fungsiluas(int p, int l){
6     int luas = p*l;
7     cout<<"Hasil Luasnya adalah "<<luas;
8 }
9
10 int main(){
11    //memanggil fungsi dgn memberi nilai parameter
12    fungsiluas(4,5);
13 }
```

➤ Contoh Void (5)

```
1 #include <iostream>
2 using namespace std;
3
4 //membuat fungsi yg memiliki parameter nilai
5 void fungsiluas(int p, int l){
6     int luas = p*l;
7     cout<<"Hasil Luasnya adalah "<<luas;
8 }
9
10 int main(){
11     int panjang, lebar;
12     cout<<"Inputkan nilai panjang = ";
13     cin>>panjang;
14     cout<<"Inputkan nilai lebar = ";
15     cin>>lebar;
16
17     //memanggil fungsi dgn memberi nilai parameter
18     fungsiluas(panjang,lebar);
19 }
```

➤ Contoh Void (6)

```
1 #include <iostream>
2 using namespace std;
3
4 //deklarasi fungsi
5 void fungsiluas(int p, int l);
6
7 int main(){
8     int panjang, lebar;
9     //memanggil fungsi dgn memberi nilai parameter
10    cout<<"Inputkan nilai panjang = ";
11    cin>>panjang;
12    cout<<"Inputkan nilai lebar = ";
13    cin>>lebar;
14    fungsiluas(panjang,lebar);
15 }
16
17 //definisi fungsi
18 void fungsiluas(int p, int l){
19     int luas = p*l;
20     cout<<"Hasil Luasnya adalah "<<luas;
21 }
```

➤ Contoh Non Void

```

1 #include <iostream>
2 using namespace std;
3
4 //deklarasi variabel global
5 int luas;
6
7 //membuat fungsi yg memiliki parameter nilai
8 int fungsiluas(int p, int l){
9     luas = p*l;
10    return luas;
11 }
12
13 int main(){
14     //deklarasi variabel lokal
15     int panjang, lebar;
16
17     cout<<"Inputkan nilai panjang = ";
18     cin>>panjang;
19     cout<<"Inputkan nilai lebar = ";
20     cin>>lebar;
21
22     //memanggil fungsi dgn memberi nilai parameter
23     fungsiluas(panjang,lebar);
24     cout<<"Hasil Luasnya adalah "<<luas;
25 }
```

➤ Contoh Rekursif

```

1 #include <iostream>
2 using namespace std;
3
4 int rekursiffaktorial(int f){
5     if (f==1){
6         return 1;
7     } else {
8         return f*rekursiffaktorial(f-1);
9     }
10 }
11
12 int main(){
13     int faktorial;
14
15     cout<<"Inputkan nilai faktorial = ";
16     cin>>faktorial;
17
18     cout<<"Hasil Faktorialnya adalah "<<rekursiffaktorial(faktorial);
19 }
```