

MODUL PRAKTIKUM ALGORITMA & PEMROGRAMAN I

C++

```
{ Context *  
if (COMPARE  
context->  
}  
//  
// Text handling  
//  
static void handle  
{  
if (context->addTit  
context->title.ap  
}  
..
```



SEKRETINGSIH NITA, S.Kom., MT
YESSI YUNITASARI, S.Kom., M.Cs
KELIK SUSSOLAICAH, S.Kom., M.Kom

LEMBAR PENGESAHAN

Tentang

MODUL PRAKTIKUM ALGORITMA & PEMROGRAMAN-1

Tim Penyusun :

**Sekreningsih Nita, S.Kom, MT
Yessi Yunitasari, S.Kom, M.Cs
Kelik Sussolaikah, S.Kom, M.Kom**

**Disyahkan dan ditanda-tangani di Madiun
Tanggal : Oktober 2021**

**Menyetujui,
Dekan Fakultas Teknik**

**Mengetahui,
Ketua Program Studi
Teknik Informatika**

**Ir. Sulistyaning Kartikawati, M.M, M.Pd
NIK : 110**

**Sri Anardani, S.Kom, MT
NIK : 1100.....**

KATA PENGANTAR

Puji syukur alhamdulillah kepada Tuhan Yang Maha Esa atas segala kasih karuniaNya sehingga dengan segala keterbatasan waktu, tenaga dan pikiran yang dimiliki penulis, akhirnya modul ini bisa tersusun.

Modul ini dibuat dengan tujuan untuk mempermudah pelaksanaan praktikum Algoritma & Pemrograman-1 dan mampu menjadi media akselerasi pemahaman serta keterampilan (*skill*) mahasiswa sesuai dengan kompetensi yang diharapkan.

Penulis sadar bahwa modul ini masih banyak memiliki kekurangan. Oleh karena itu saran dan masukan sangatlah diharapkan demi semakin baiknya kualitas dari penulisan modul ini. Tak lupa tim penyusun mrngucapkan terimakasih banyak atas bantuan dari berbagai pihak yang telah membantu tersusunnya modul ini.

Madiun, Oktober 2021

Tim Penyusun

PETUNJUK DAN TATA CARA PRAKTIKUM

Tujuan Instruksional Umum :

Mahasiswa diharapkan mampu mengimplementasikan algoritma menjadi susunan program sebagai perangkat lunak yang dipilih dalam melakukan pengolahan data dan penyediaan informasi yang tepat dan cepat.

Aturan Penilaian :

Tugas (dosen pengampu/asisten lab)	: 30%
Keberhasilan Program (dosen pengampu/asisten lab)	: 30%
Laporan Akhir Praktikum (mahasiswa)	: 20%
Kedisiplinan dan Kehadiran (dosen pengampu)	: 20%

Prosedur Pelaksanaan :

1. Mahasiswa wajib datang 15 menit sebelum praktikum dimulai
2. Mahasiswa wajib menyelesaikan tugas praktikum yang ada pada modul praktikum
3. Mahasiswa wajib membuat Laporan Akhir seluruh modul sesuai dengan ketentuan.
4. Mahasiswa harus mengikuti seluruh peraturan yang ada dengan sebaik-baiknya.
5. Mahasiswa yang tidak mengikuti kegiatan praktikum kurang dari 30% pelaksanaan praktikum keseluruhan akan dianggap tidak lulus dan harus mengulang pada semester depan.

Ketentuan Susunan Laporan Akhir Praktikum :

1. **Laporan Akhir Praktikum :** Laporan yang dikerjakan mahasiswa pada akhir kegiatan praktikum.
 - a. Halaman Judul (cover LAPORAN AKHIR)
 - b. Kata Pengantar
 - c. NIM, Nama
 - d. Daftar Isi
 - e. Daftar Gambar
 - f. Daftar Tabel
 - g. Daftar Pustaka
2. **Format Laporan Akhir Praktikum.**
 - a. Diketik pada kertas A4 (70 gram).
 - b. Cover sesuai ketentuan berwarna putih.
 - c. Berbentuk jilidan yang berisi Modul-1 s/d Modul-10

3. Contoh Cover Laporan Akhir Praktikum :

Cover Akhir Praktikum (Modul 1 s/d Modul 10) :

**PRODI TEKNIK INFORMATIKA –FAKULTAS TEKNIK
UNIVERSITAS PGRI MADIUN** *font size 14*



**LAPORAN AKHIR PRAKTIKUM
ALGORITMA & PEMROGRAMAN-I** *font size 16*

DISUSUN OLEH :

NAMA :
NIM :
KLAS :

} *font size 12*

font size 14 :

**LABORATORIUM BAHASA PEMROGRAMAN
PRODI TEKNIK INFORMATIKA-FAKULTAS TEKNIK
UNIVERSITAS PGRI MADIUN
2021/2022**

DAFTAR ISI

LEMBAR JUDUL	i
LEMBAR PENGESAHAN	ii
KATA PENGANTAR	iii
PETUNJUK DAN TATA CARA PRAKTIKUM	iv
DAFTAR ISI	v
 MODUL-1 PENGENALAN BAHASA PEMROGRAMAN C++	 1
1.1. Pengenalan IDE Borland C++	1
1.2. Struktur Program C++	2
1.3. Pengertian Variabel	3
1.4. Deklarasi Variabel	4
1.5. Rangkuman	7
1.6. Latihan	7
MODUL-2 INPUT OUTPUT STANDARD	9
2.1. Statement Input	9
2.2. Statement Output	13
2.3. Rangkuman	15
2.4. Latihan	16
 MODUL-3 STRUKTUR KONTROL KEPUTUSAN-1.....	 18
3.1. Pernyataan IF	18
3.2. Rangkuman	20
3.3. Latihan	21
 MODUL-4 STRUKTUR KONTROL KEPUTUSAN- 2	 23
4.1. Pernyataan IF Else.....	23
4.2. Pernyataan Nested IF	24
4.3. Rangkuman	26
4.4. Latihan	26
 MODUL-5 STRUKTUR KONTROL KEPUTUSAN- 3	 29
5.1. Pernyataan Switch Case	29
5.2. Rangkuman	32
5.3. Latihan	33

MODUL-6	MEMAHAMI PERULANGAN-1	36
6.1.	Pernyataan FOR	36
6.2.	Perulangan Tidak Berhingga	37
6.3.	Rangkuman	38
6.4.	Latihan	38
MODUL-7	MEMAHAMI PERULANGAN-2	41
7.1.	Pernyataan Nested FOR	41
7.2.	Rangkuman	42
7.3.	Latihan	42
MODUL-8	MEMAHAMI PERULANGAN-3	44
8.1.	Pernyataan WHILE	44
8.2.	Rangkuman	45
8.3.	Latihan	45
MODUL-9	MEMAHAMI PERULANGAN-4	47
9.1.	Pernyataan DO WHILE	47
9.2.	Rangkuman	48
9.3.	Latihan	48
MODUL-10	MEMAHAMI BREAK, CONTINUE, GOTO	50
10.1.	Pernyataan Break	50
10.2.	Pernyataan Continue	51
10.3.	Pernyataan Goto	51
10.4.	Rangkuman	52
10.5.	Latihan	53

PENGENALAN BAHASA PEMROGRAMAN BORLAND C++

1

Standar Kompetensi

1. Mahasiswa mampu membuat program sederhana dengan menggunakan bahasa pemrograman Borland C++

Kompetensi Dasar

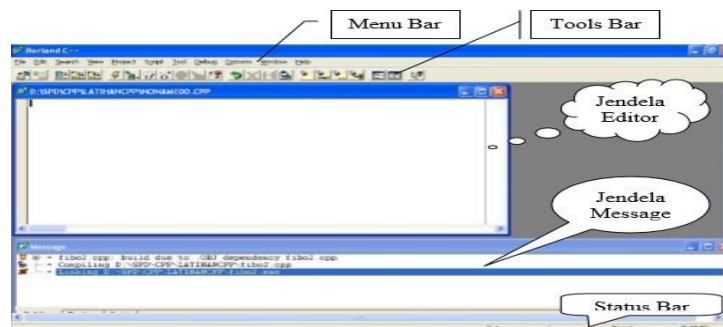
1. Mengetahui dan memahami Perkembangan Bahasa Pemrograman C++
2. Memahami IDE Borland C++
3. Mengetahui dan memahami Struktur Program C++.

1.1. Pengenalan IDE Borland C++:

IDE merupakan singkatan dari Integrated Development Environment, merupakan lembar kerja terpadu untuk pengembangan program.

IDE dari Borland C++, dapat digunakan untuk :

- Menulis Naskah Program.
- Mengkompilasi Program (*Compile*)
- Melakukan Pengujian Program (*Debugging*)
- Mengaitkan Object dan Library ke Program (*Linking*)
- Menjalankan Program (*Running*)



Gambar 1.1 : Window Utama Borland C++

IDE pada Borland C++, terbagi menjadi 4 (empat) bagian, yaitu :

a. Menu Utama (Menubar)

Menu utama terdiri dari ; File, Edit, Search, Run, Compile, Debug, Project, Options, Window dan Help

b. Baris Peralatan (Tools Bar)

Baris yang menampilkan shortcuts (icons) untuk mempermudah pengguna dalam pembuatan program-program C++, seperti icon open, save, compiler, run dan lain-lain.

c. Jendela Editor

Tempat untuk pengetikan program dan membuat program. Jika Pertama kali anda membuat program, namafile jendela editor adalah NONAME01.CPP

d. Jendela Message

Tempat untuk menampilkan pesan-pesan pada proses kompilasi dan Link program.

e. Baris Status

Baris dimana menampilkan keterangan-keterangan pada saat anda mengaktifkan menu bar dan submenu

1.2. Struktur Program C++

Struktur program C++, sama seperti struktur program C yang terdahulu. Struktur program C++ terdiri sejumlah blok fungsi, setiap fungsi terdiri dari satu atau beberapa pernyataan yang melaksanakan tugas tertentu.

Bentuk umum

```
#include <file-header>
main()
{
    Pernyataan;
}
```

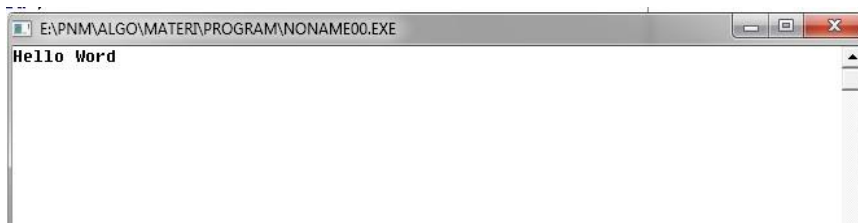
```
#include <iostream.h>
#include <conio.h>
main()
{
    cout<<"Hello Word";
    getch();
}
```

ALGO

Keterangan:

- `#include <file-header>`
merupakan preprocessor pada C++ untuk pemanggilan file header yang memuat beberapa perintah-perintah dari C++ (contoh, apabila ingin menggunakan perintah `cout` maka harus menggunakan file header `iostream.h`)
- `main()`
merupakan awal mula dari blok program utama
- tanda { dan }
sebagai punctuator sebagai awal blok program hingga akhir blok program
- `cout`
merupakan perintah keluaran pada C++
- `getch();`
apabila ditempatkan sebelum funtuator }, maka berfungsi sebagai penahan dari tampilan hasil , agar fungsi `getch` dapat dibaca maka ditambahkan header `conio.h`

Hasi Run:



1.3. Pengertian Variabel

Variable berfungsi untuk pemesanan memori dan memberikan tempat untuk menampung data atau data konstanta di dalam memori yang kita pesan. Data atau nilai tersebut dapat kita ubah-ubah selama program berjalan kecuali untuk variable konstanta.

Dalam peraturan pemberian nama (identitas) variabel dapat terdiri dari huruf, angka dan karakter (_). nama variabel harus dimulai dengan huruf atau garis bawah. dan juga bahwa huruf besar dan huruf kecil dalam pemrograman C++ itu sangat dianggap beda karena C++ adalah Bahasa Pemrograman Case-Sensitive.

Ada satu peraturan lainnya dalam pemberian nama pada variabel yaitu tidak boleh menggunakan spasi saat penamaan variabel, tapi jika nama tersebut memiliki dua kata atau lebih dan ingin tetap menamakanya dengan kata terpisah supaya mudah dibaca, anda dapat menulis namanya tanpa menggunakan spasi atau anda bisa menggantikan spasi dengan tanda (_). ini dapat mempermudah anda dalam pembacaan nama yang lebih dari 1 kata, peraturan kedua adalah nama variabel tidak boleh diawali dengan angka dan menggunakan operator aritmatika. Dalam mendirikan variabel kita membutuhkan jenis penyimpanan yang dapat dilakukan oleh variabel, hal itu dinamakan tipe data. Sebelum kita tuliskan nama dari variabel dibutuhkan jenis tipe data di depan nama variabel untuk menentukan jenis penyimpanan. tipe data variable secara umum bisa dibedakan menjadi tiga yaitu :

Variabel Numerik : Memungkinkan pengguna menyimpan data yang berbentuk nomer/bilangan. Variabel numerik dapat dibagi menjadi beberapa bagian yaitu : Bilangan Bulat (Integer “int”), Bilangan Desimal Berpresisi Tunggal (Floating Point “float”). Bilangan Desimal Berpresisi Ganda (Double Precision “double”).

Variabel Text : memungkinkan pengguna untuk menyimpan data berbentuk karakter. Bahasa pemrograman C++ menyediakan beberapa tipe data untuk jenis variabel seperti Char (Character / Karakter tunggal), string (kumpulan dari Karakter atau kata).

Boolean : dengan menuliskan “bool” merupakan jenis yang hanya dapat mewakili satu dari dua pilihan yaitu 1 (True) atau 0 (False).

1.4. Deklarasi Variabel

Deklarasi variabel adalah proses mendirikan atau pembuatan variabel dalam kode program. Deklarasi variabel diwajibkan didirikan dengan jenis variabel tersebut dengan menyertakan tipe data di awal deklarasi variabel. Deklarasi merupakan statement(pernyataan), Setiap kali pendeklarasian variabel harus diakhiri dengan tanda titik koma (;).

Bentuk Penulisan :

```
tipeData daftar_variabel = inisialisasi;
```

Inisialisasi merupakan opsional, kita dapat mendirikan variable tanpa atau menggunakan inisialisasi, kecuali variabel konstanta yang mengharuskan untuk mempunyai nilai awal (inisialisasi).

Contoh Penulisan :

```
int x;  
int y;
```

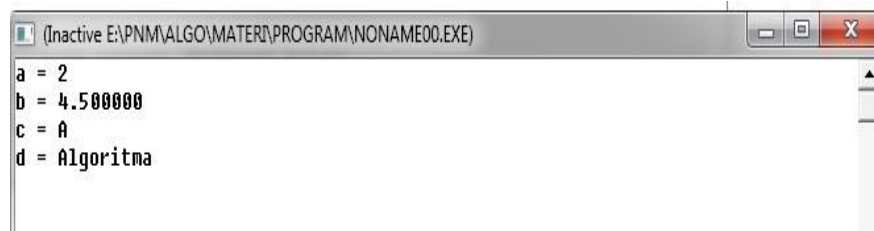
Diatas merupakan contoh pendeklarasian variabel yang valid. jika pendeklarasian jenis nilai variabel memiliki jenis (tipe data) yang sama dan merasa hal itu memakan banyak baris, maka bisa kita persingkat menjadi seperti ini.

```
int x,y;  
double a, b, c;  
float y;  
string nama ="Belajarcpp.om";  
char nama[20]="belajarcpp ", merek[10];  
string nama ="Belajarcpp";
```

Example :

```
#include <stdio.h>  
  
int main(void) {  
    int a;        // tipe integer (bilangan bulat)  
    float b;      // tipe float (bilangan real)  
    char c;       // tipe karakter  
  
    // tipe string, pendeklarasiannya  
    // harus langsung diinisialisasi  
    char d[10] = "Algoritma";  
  
    // inisialisasi variabel  
    a = 2;  
    b = 4.5;  
    c = „A“;  
  
    /*  
    * menampilkan isi variabel ke konsol  
    * perhatikan untuk aturan pemformatannya  
    * %d untuk integer, %f untuk float, %c untuk char, %s untuk  
    * string */  
    printf("a = %d\n", a);  
    printf("b = %f\n", b);  
    printf("c = %c\n", c);  
    printf("d = %s\n", d);  
  
    // boleh menambahkan baris berikut ini  
    // di setiap akhir main  
    return 0;  
}
```

Hasil Run:



```
(Inactive E:\PNM\ALGO\MATERI\PROGRAM\NONAME00.EXE)
a = 2
b = 4.500000
c = A
d = Algoritma
```

EXAMPLE 2:

```
#include <stdio.h>

int main(void) {
    // mendeklarasikan konstanta
    const float PI = 3.14;

    // mendeklarasikan variabel
    float luas;
    int a, b, c, r;

    // mengisikan nilai ke variabel (inisialisasi)
    a = 3;
    b = 4;
    c = 6;
    r = 7;

    // menampilkan ke konsol
    printf("a = %d b = %d c = %d\n\n", a, b, c);

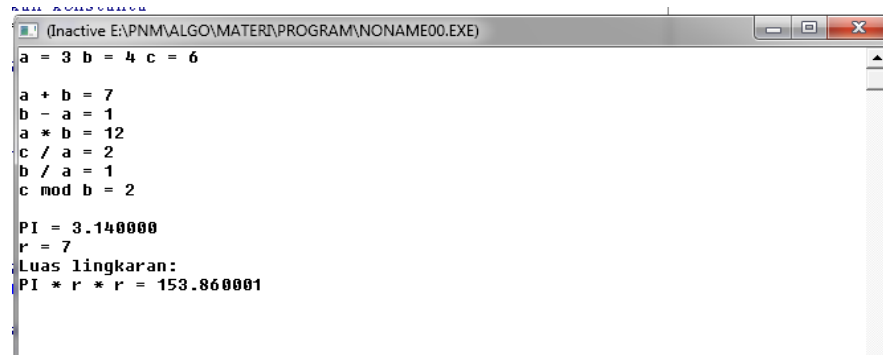
    // operasi aritmatik
    printf("a + b = %d\n", a + b);
    printf("b - a = %d\n", b - a);
    printf("a * b = %d\n", a * b);
    printf("c / a = %d\n", c / a);
    printf("b / a = %d\n", b / a);
    printf("c mod b = %d\n\n", c % b);

    printf("PI = %f\n", PI);
    printf("r = %d\n", r);

    // menghitung luas lingkaran
    luas = PI * r * r;
    printf("Luas lingkaran:\n");
    printf("PI * r * r = %f\n", luas);

    return 0;
}
```

HASIL RUN:



```
(Inactive E:\PNM\ALGO\MATERI\PROGRAM\NONAME00.EXE)
a = 3 b = 4 c = 6
a + b = 7
b - a = 1
a * b = 12
c / a = 2
b / a = 1
c mod b = 2

PI = 3.140000
r = 7
Luas lingkaran:
PI * r * r = 153.860001
```



1.5. Rangkuman

Bahasa C adalah merupakan bahasa pemrograman tingkat, menengah yaitu diantara bahasa tingkat rendah dan tingkat tinggi yang biasa disebut dengan bahasa tingkat menengah. Bahasa C mempunyai banyak kemampuan yang sering digunakan diantaranya kemampuan untuk membuat perangkat lunak, misalnya dBASE, Word Star dan lain-lain.

IDE dari Borland C++, dapat digunakan untuk :

- Menulis Naskah Program.
- Mengkompilasi Program (*Compile*)
- Melakukan Pengujian Program (*Debugging*)
- Mengaitkan Object dan Library ke Program (*Linking*)
- Menjalankan Program (*Running*)

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



1.6. Latihan

1. **KLAS-1A** : Buatlah program untuk menghitung rata – rata dari 3 buah bilangan:

a = 15
b = 25
c = 12

2. **KLAS-1B** : Buatlah program untuk menghitung Bilangan Terbesar dari dari 3 buah bilangan yang diinputkan.

Contoh Input :

Input Bilangan-1 : < 10 > ...

Input Bilangan-2 : < 23 > ...

Input Biangan-3 : < 15 > ...

Maka Output (hasil) :

JADI BILANGAN TERBSAR adalah : BILANGAN 3 yaitu 23

3. **KLAS-1C** : Buatlah program untuk menyebutkan ANGKA Genap atau Ganjil dari 3 ANGKA yang diinputkan :

Contoh Input :

Input ANGKA-1 : < 3 > ...

Input ANGKA-2 : < 20 > ...

Input ANGKA-3 : < 33 > ...

Maka Output (hasil) :

ANGKA-1 = 3 adalah GANJIL

ANGKA-2 = 20 adalah GENAP

ANGKA-3 = 33 adalah GANJIL

4. **KLAS-1D** : Buatlah program untuk mencari mana bilangan negatif/positif dari 3 buah bilangan yang dimasukkan.

Contoh Inputan :

Input Bilangan-1 : < 8 > ...

Input Bilangan-2 : < -5 > ...

Input Biangan-3 : < 1 > ...

Maka Output (hasil) :

Bilangan-1 : Bilangan POSITIF

Bilangan-2 : Bilangan NEGATIF

Bilangan-3 : Bilangan POSITIF

5. **KLAS-1E** : Buatlah program untuk mencari mana bilangan ganjil dan positif, genap dan positif dan bilangan negative dari bilangan yang dimasukkan.

Contoh Inputan :

Input Bilangan-1 : <10 > ...

Maka Output (hasil) :

Bilangan genap dan positif

INPUT OUTPUT STANDART

2

Standar Kompetensi

1. Mahasiswa mampu membuat program sederhana dengan menggunakan bahasa pemrograman C++ untuk operasi I/O standar

Kompetensi Dasar

1. Dapat melakukan operasi I/O standar melalui keyboard dan screen
2. Dapat menggunakan format specifier untuk masing – masing tipe data
3. Mengerti penggunaan backslash character code

2.1 Statement Input

Beberapa fungsi / statement input yang dapat digunakan yaitu :

1. *Scanf*

Fungsi *scanf()* digunakan untuk memasukkan **berbagai jenis data**. Bentuk Umum dari fungsi ini adalah :

`scanf("penentu format", &nama-variabel);`

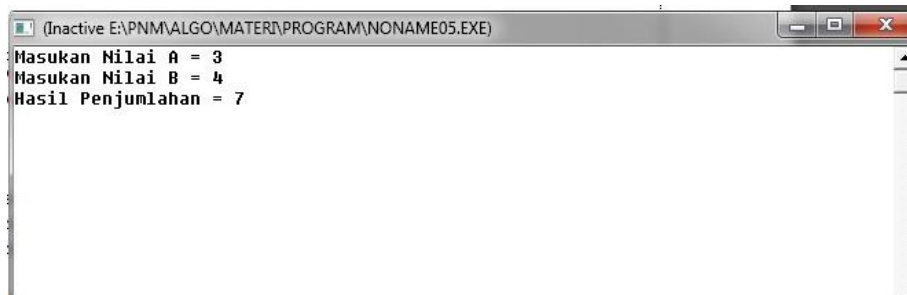
Keterangan

simbol **&** merupakan pointer yang digunakan untuk menunjuk kealamat variabel memori yang dituju.

Example:

```
# include <stdio.h>
# include <conio.h>
main()
{
    a
    int a, b, c = 0 ;
    clrscr();
    printf("Masukan Nilai A = "); scanf("%d",&a);
    printf("Masukan Nilai B = "); scanf("%d",&b);
    c = a + b;
    printf("Hasil Penjumlahan = %d",c);
}
```


Hasil Run:



2. Gets

Fungsi **gets()** digunakan untuk memasukkan **data string**. Bentuk Umum dari fungsi ini adalah :

gets(nama-variabel-array);

Example:

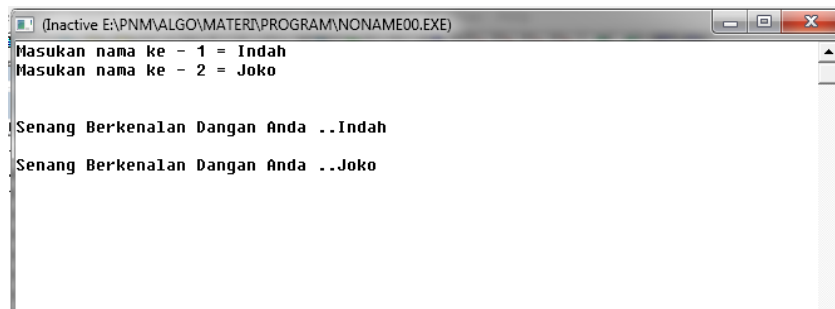
```
# include <stdio.h>
# include <conio.h>

main()
{
    char nm1[20];
    char nm2[20];
    clrscr();

    printf ("Masukan nama ke - 1 = ");
    gets(nm1);
    printf("Masukan nama ke - 2 = ");
    gets(nm2);
    printf("\n\n");

    printf("Senang Berkenalan Dangan Anda ..%s", nm1);
    printf("\n\n");
    printf("Senang Berkenalan Dangan Anda ..%s", nm2);
}
```

Hasil Run



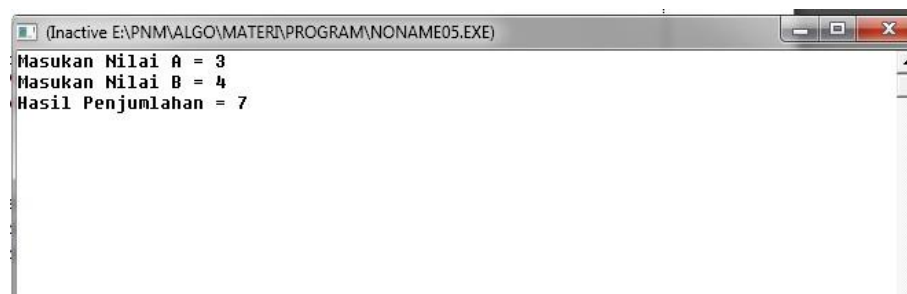
3. Cin

Fungsi ***cin()*** merupakan sebuah object didalam C++ digunakan untuk memasukkan suatu data. Untuk menggunakan fungsi ***cin()*** ini, harus menyertakan file header ***iostream.h*** .

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

main()
{
float a, b, c;
clrscr();
cout<<"Masukan Nilai A : ";
cin>>a;
cout<<"Masukan Nilai B : ";
cin>>b;
c = a + b;
cout<<"Hasil Penjumlahan Nilai C : "<<c<<endl;
}
```

Hasil Run:



4. Getch

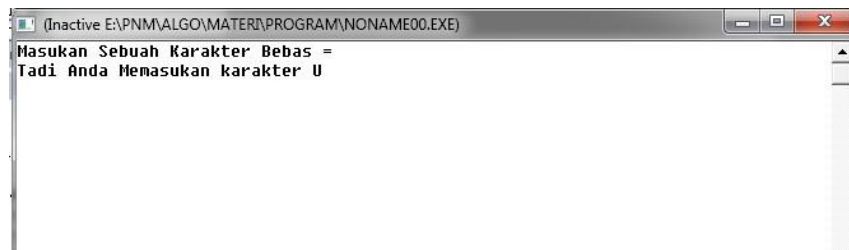
Fungsi ***getch()*** (*get character and echo*) dipakai untuk membaca sebuah karakter dengan sifat karakter yang dimasukkan tidak perlu diakhiri dengan menekan tombol

ENTER, dan karakter yang dimasukan tidak akan ditampilkan di layar. *File header* yang harus disertakan adalah **conio.h**.

Example:

```
# include <stdio.h>
# include <conio.h>
main()
{
    char kar;
    clrscr();
    printf("Masukan Sebuah Karakter Bebas = ");
    kar = getch();
    printf("\nTadi Anda Memasukan karakter %c", kar);
    getch();
}
```

Hasil RUN :



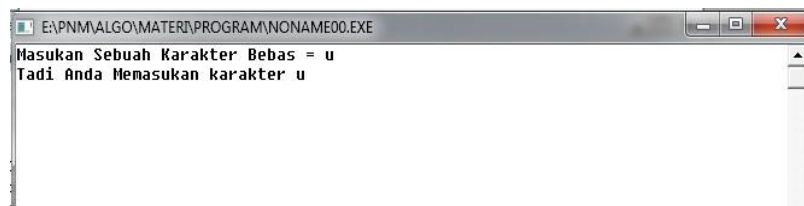
5. Getche

Fungsi **getche()** dipakai untuk membaca sebuah karakter dengan sifat karakter yang dimasukkan tidak perlu diakhiri dengan menekan tombol ENTER, dan karakter yang dimasukkan ditampilkan di layar. File header yang harus disertakan adalah **conio.h**.

Example:

```
# include <stdio.h>
# include <conio.h>
main()
{
    char kar;
    clrscr();
    printf("Masukan Sebuah Karakter Bebas = ");
    kar = getche();
    printf("\nTadi Anda Memasukan karakter %c", kar);
    getch ();
}
```

Hasil Run:



2.2 Statement Output

1. Printf

Fungsi ***printf()*** merupakan fungsi keluaran yang paling umum digunakan untuk menampilkan informasi ke layar.

Example:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a = 7;
    char b = 'G';
    clrscr();
    printf("%c Merupakan Abjad Yang Ke - %d", b, a);
}
```

Hasil Run:



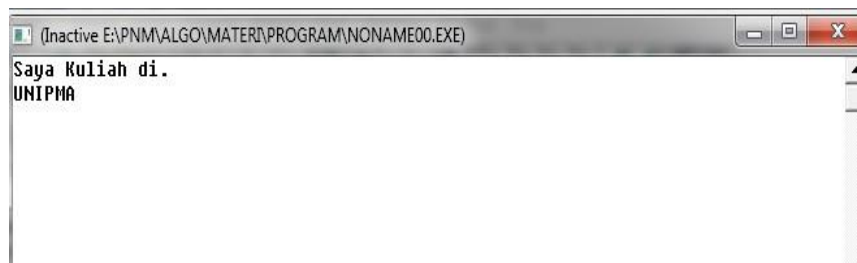
2. Puts

Perintah ***puts()*** sebenarnya sama dengan ***printf()***, yaitu digunakan untuk mencetak string ke layar. ***puts()*** berasal dari kata ***PUT STRING***.

Example:

```
#include <stdio.h>
#include <conio.h>
main()
{
    char a[4] = "UNIPMA";
    clrscr();
    puts("Saya Kuliah di. ");
    puts(a);
}
```

Hasil Run:



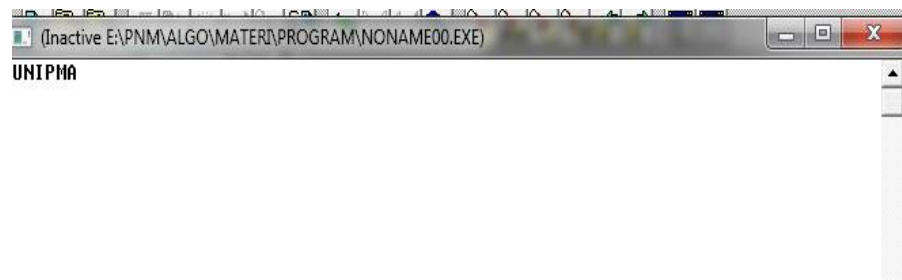
3. Putchar

Perintah ***putchar()*** digunakan untuk menampilkan **sebuah karakter ke layar**. Penampilan karakter tidak diakhiri dengan pindah baris.

Example:

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    putchar('U');
    putchar('N');
    putchar('I');
    putchar('P');
    putchar('M');
    putchar('A');
}
```

Hasil Run:



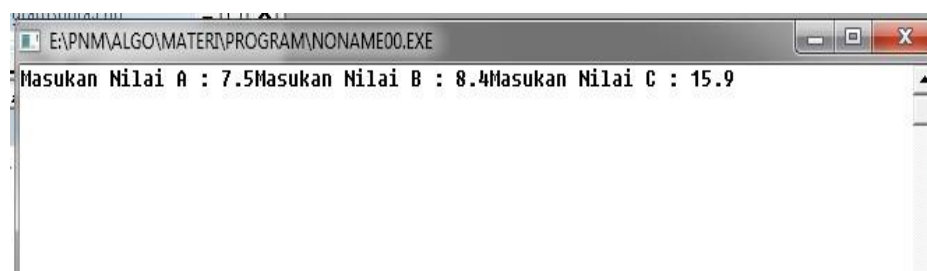
4. Cout

Fungsi *cout()* merupakan sebuah object didalam Borland C++ digunakan untuk menampilkan suatu data kelayar. Untuk menggunakan fungsi *cout()* ini, harus menyertakan file header ***iostream.h*** .

Example:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
main()
{
    float a, b, c;
    a=7.5; b=8.4; c=0
    clrscr();
    cout<<"Masukan Nilai A : "<<a;
    cout<<"Masukan Nilai B : "<<b;
    c = a + b;
    cout<<"Masukan Nilai C : "<<c;
    getch();
}
```

Hasil Run:





2.3. Rangkuman

Bab ini membahas tentang operasi input dan output standar dengan keyboard dan screen. Selain itu juga akan membahas tentang penggunaan format specifier untuk performatan input dan output sesuai tipe data yang digunakan. Perintah keluaran merupakan kumpulan perintah yang digunakan untuk menampilkan sedangkan perintah inputan merupakan kumpulan perintah yang digunakan untuk member nilai atau input data



2.4. Latihan

1. **KLAS-1A** : Buatlah program untuk outputnya DATA BIOGRAFI anda , contoh berikut ini:

MY BIODATA	
NAME	: < input >
PLACE	: < input >
DATE BORN	: < input >
ADDRESS	: < input >
STATUS	: < input >

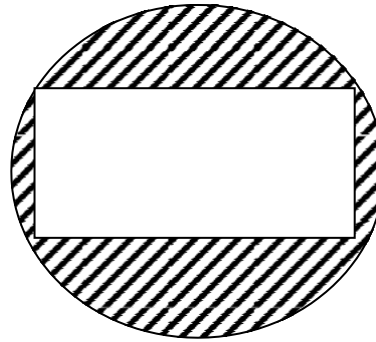
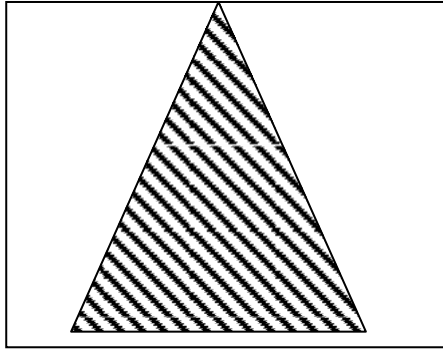
2. **KLAS-1B** : Buatlah program sehingga outputnya seperti berikut ini:

MY FAMILIY :

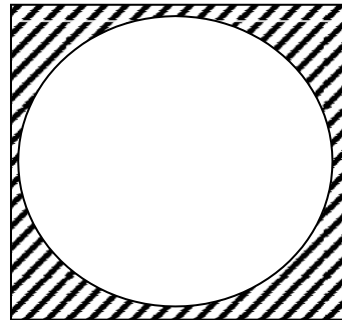
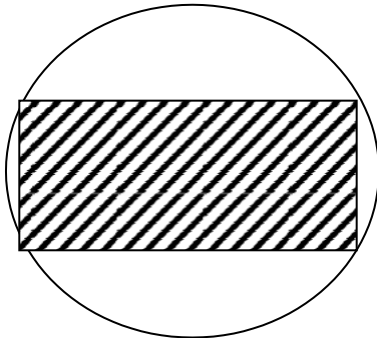
Hello everyone ... I want to introduction for my family. First, my mother Is ALCE and my father is JACK. I Have two family , ones

my big brother is ERIC and My little brother is FADIL. So I am is 'ME'

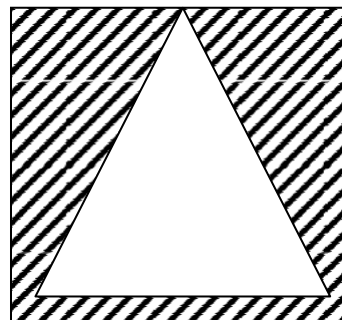
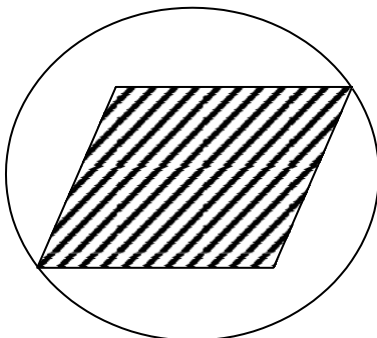
3. **KLAS-1C** : Buatlah program untuk menghitung Luas gambar yang TIDAK DIARSIR :



4. **KLAS-1D** : Buatlah program untuk menghitung luas gambar YANG DIARSIR :



5. **KLAS-1E** : Buatlah program untuk menghitung luas gambar YANG DIARSIR :



STRUKTUR KONTROL KEPUTUSAN 1

3

Standar Kompetensi

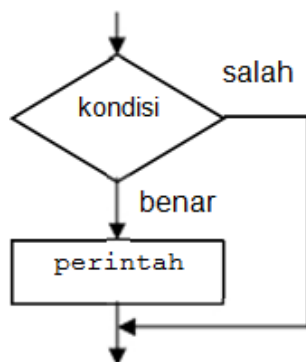
1. Mahasiswa mampu memahami struktur keputusan yang ada di C++

Kompetensi Dasar

1. Dapat memahami struktur kontrol (if, if-else, if-else-if, switch) untuk memilih alur blok yang akan dieksekusi.
2. Dapat membedakan penggunaan masing – masing struktur kontrol (antara if dan switch)

3.1 Pernyataan IF

Pernyataan *if* mempunyai pengertian, “ *Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan*”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut:



Gambar 3.1 Diagram Alir IF

Bentuk Umum Pernyataan IF

if (*kondisi*)

```
{  
    pernyataan;  
    .....  
}
```

Penulisan *kondisi* harus didalam tanda kurung dan berupa ekspresi relasi dan penulisan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk

atau pernyataan kosong. Jika pemakaian *if* diikuti dengan pernyataan majemuk, bentuk penulisannya sebagai berikut :

Menentukan besarnya discount dari pembelian barang yang diberikan seorang pembeli, dengan kriteria :

- Tidak ada discount jika total pembelian kurang dari Rp. 50.000,-
- Jika total pembelian lebih dari atau sama dengan Rp. 50.000,- discount yang diterima sebesar 20% dari total pembelian.

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

main()
{
    double tot_beli, discount=0, jum_bayar=0;
    clrscr();

    cout<<"Total Pembelian Rp. ";
    cin>>tot_beli;

    if (tot_beli >= 50000)
        discount = 0.2 * tot_beli;

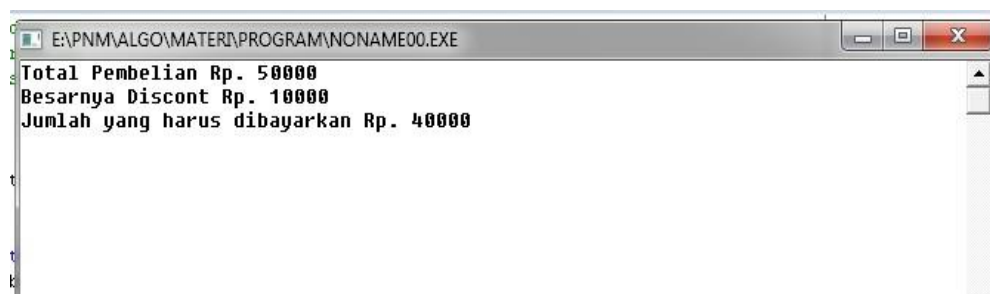
    cout<<"Besarnya Discount Rp. "<<discount<<endl;

    jum_bayar = tot_beli - discount;

    cout<<"Jumlah yang harus dibayarkan Rp. "
    "<<jum_bayar;

    getch();
}
```

Hasil Run:

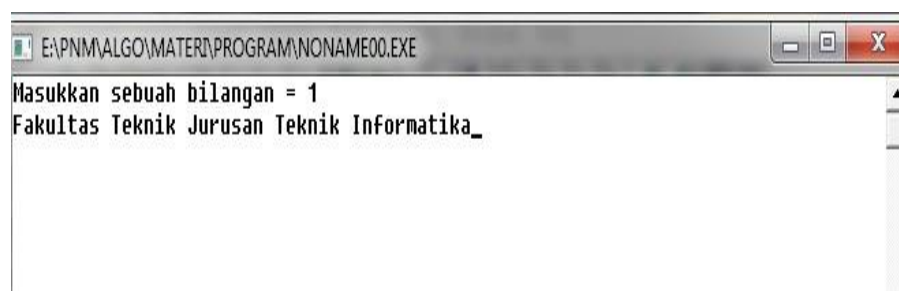


```
E:\PNM\ALGO\MATERI\PROGRAM\NONAME00.EXE
Total Pembelian Rp. 50000
Besarnya Discount Rp. 10000
Jumlah yang harus dibayarkan Rp. 40000
```

Example 2:

```
#include <stdio.h>
#include <conio.h>
int main(int argc, char* argv[])
{
    int x;
    printf("Masukkan sebuah bilangan = ");
    scanf("%d", &x);
    if(x%2!=0)
    {
        printf("Fakultas      Teknik      Jurusan      Teknik
Informatika");
    }
    getch();
    return 0;
}
```

Hasil Run:



3.2. Rangkuman

Pernyataan Percabangan digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada. Pernyataan *if* mempunyai pengertian, “ *Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan*”.



3.3. Latihan

KLAS-1A :

1. Buatlah program menggunakan C++ dengan tampilan sebagai berikut:

```
*****
Penerimaan Mahasiswa Baru UNIPMA
*****
Nim                1805101125
Nama mahasiswa    : Juan Oscar Brilian
Alamat mahasiswa  : Jl. Salak 9 Madiun
Belajar           : Algoritma & Pemrograman-I
Biaya Belajar     : Rp. 200.000
Cicilan Pertama   : Rp. 150.000
Cicilan Kedua     : Rp. 50.000
Tunggakan         : Rp. 0      <output>
*****
```

KLAS-1B :

- 2.. Buatlah program menggunakan C++ dengan tampilan sebagai berikut:

```
*****
          NOTA PENJUALAN KOMPUTER
*****
Kode Barang       : MO101      < input >
Nama Barang       : PRINTER   < input >
Jumlah Barang     : 5          < input >
Harga Satuan      : Rp. 1.200.000 < input >
-----
Total Harga       : Rp. 4.500.000 < output >
Bayar             : Rp. 5.000.000 < input >
Kembali           : Rp. 500.000 < output >
*****
```

KLAS-1C :

3. Buatlah program menggunakan C++ untuk menentukan harga barang setelah di diskon dengan tampilan sebagai berikut:

```
-----
Masukkan harga barang = Rp. 150.000
Masukkan discount     =      10 %
-----
Harga barang          = Rp. 150.000
Besar diskon          = Rp. 15.000
-----
Hargabersih           = Rp. 135.000
-----
```

KLAS-1D :

4.. Buatlah program menggunakan C++ dengan tampilan sebagai berikut:

```
*****
Penerimaan Angsuran Nasabah
*****
No Rek          081335123456
Nama nasabah    : SAIFUL
Alamat nasabah  : Jl. Mawar No. 5
Jenis Kredit    : Dagang
Dana            : Rp. 5.000.000
Cicilan Pertama : Rp. 750.000
Cicilan Kedua   : Rp. 750.000
Tunggakan       : Rp. 3.500.000 <output>
*****
```

KLAS-1E :

5.. Buatlah program menggunakan C++ dengan tampilan sebagai berikut:

```
*****
Penerimaan Angsuran Nasabah
*****
Nota          1
Nama nasabah   : SUSI
Alamat nasabah : Jl. Madiun
Jumlah barang  5
Harga          : Rp. 750.000
Diskon         : Rp. 15%
Total          : Rp. 3.750.000 <output>
Total Diskon   : Rp. 562.500 <output>
Total Bayar    : Rp. 3.187.500 <output>
*****
```

STRUKTUR KONTROL KEPUTUSAN 2

4

Standar Kompetensi

1. Mampu memahami struktur keputusan yang ada di C++

Kompetensi Dasar

1. Dapat memahami struktur kontrol (IF ELSE, IF ELSE IF, SWITCH) untuk memilih alur blok yang akan dieksekusi.
2. Dapat membedakan penggunaan masing – masing struktur kontrol (antara if dan switch)

4.1 Pernyataan IF - ELSE

Pernyataan *if* mempunyai pengertian, “ *Jika kondisi bernilai benar, maka perintah-1 akan dikerjakan dan jika tidak memenuhi syarat maka akan mengerjakan perintah-2*”.

Bentuk umum dari pernyataan *if*

```
if (kondisi)  
    perintah-1;  
  
else  
  
    perintah-2;
```

Perintah-1 dan perintah-2 dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Jika pemakaian *if-else* diikuti dengan pernyataan majemuk, bentuk penulisannya sebagai berikut :

```
if (kondisi)  
{  
    perintah-1;  
    ...  
}  
else  
{  
    perintah-2;  
    ...  
}
```

Example 1:

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
main()
{
    double tot_beli, discount=0, jum_bayar=0;
    clrscr();

    cout<<"Total Pembelian Rp. ";
    cin>>tot_beli;

    if (tot_beli >= 50000)
        discount = 0.3 * tot_beli;
    else
        discount = 0.05 * tot_beli;

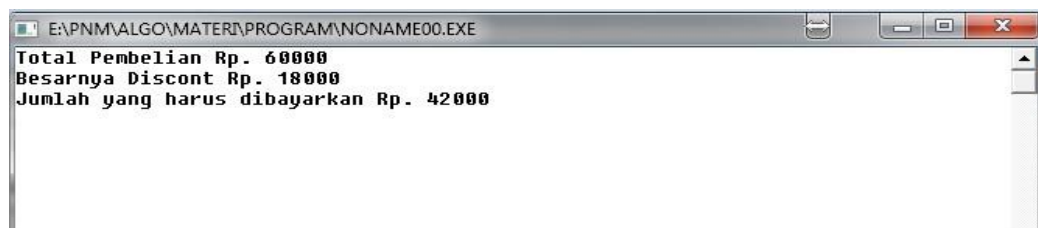
    cout<<"Besarnya Discont Rp. "<<discount<<endl;

    jum_bayar = tot_beli - discount;

    cout<<"Jumlah yang harus dibayarkan Rp. "<<jum_bayar;

    getch();
}
```

HASIL RUN:



4.2 Pernyataan NESTED IF

Nested if merupakan pernyataan if berada didalam pernyataan if yang lainnya.

Bentuk penulisan pernyataan Nested if adalah :

```
if(syarat)
    if(syarat)
        ... perintah;
    else
        ... perintah;
else
    if(syarat)
        ... perintah;
    else
        ... perintah;
```

EXAMPLE:

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
main()
{
    float penghsl, jasa=0, komisi=0, total=0;
    clrscr();

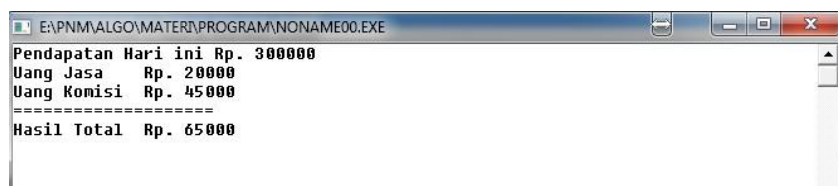
    cout<<"Pendapatan Hari ini Rp. ";
    cin>>penghsl;

    if (penghsl >= 0 && penghsl <= 200000)
    {
        jasa=10000;
        komisi=0.1*penghsl;
    }
    else
    {
        if (penghsl<=500000)
        {
            jasa=20000;
            komisi=0.15*penghsl;
        }
        else
        {
            jasa=30000;
            komisi=0.2*penghsl;
        }
    }

    /* menghitung total */
    total = komisi+jasa;

    cout<<"Uang Jasa      Rp. "<<jasa<<endl;
    cout<<"Uang Komisi   Rp. "<<komisi<<endl;
    cout<<"===== "<<endl;
    cout<<"Hasil Total   Rp. "<<total<<endl;
    getch();
}
```

HASIL RUN:



```
E:\PNM\ALGO\MATERI\PROGRAM\NONAME00.EXE
Pendapatan Hari ini Rp. 300000
Uang Jasa      Rp. 20000
Uang Komisi   Rp. 45000
=====
Hasil Total   Rp. 65000
```




5.3. Rangkuman

Pernyataan Percabangan digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada. Pernyataan Perintah-1 dan perintah-2 dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong. Jika pemakaian *if-else* diikuti dengan pernyataan majemuk,



5.4. Latihan

KLAS-1A :

1. Buatlah program untuk menentukan range nilai dari inputan data sbb :

Ketentuan Soal : TABEL NILAI

Nilai Angka	Nilai Huruf
100 – 85	A
84 – 70	B
69 – 50	C
49 – 40	D
0 – 39	E

Tampilan di layar seperti di bawah ini :

DAFTAR NILAI MAHASISWA SEMESTER-I

NO	NAMA MHS	MATA KULIAH	NILAI ANGKA	NILAI HURUF
1	< input >	< input >	< input >	<output>
2	< input >	< input >	< input >	<output>
3	< input >	< input >	< input >	<output>
4	< input >	< input >	< input >	<output>
5	< input >	< input >	< input >	<output>

KLAS-1B :

2. Buatlah program untuk memasukkan data nilai mahasiswa (ketentuan nilai sama dengan no.1 di atas). Tampilan layout seperti di bawah ini :

Nama : <INPUT >
Nim : <INPUT >
Kelas : <INPUT >

MATA KULIAH	NILAI ANGKA	NILAI HURUF	SKS
< input >	< input >	< output >	< input >
< input >	< input >	< output >	< input >
< input >	< input >	< output >	< input >
< input >	< input >	< output >	< input >
Total SKS			<output>

Kelas 1C

3. Buatlah program untuk menentukan range nilai dari inputan data sbb :
NIM, Nama, Nilai1, Nilai2, Nilai3, Nilai4 dan Nilai5, Kemudian Hitung total nilai dan rata nilai.

Jika Rata nilai:

90 - 100	:	A
85 - 89	:	AB
76 - 84	:	B
68 - 74	:	BC
55 - 67	:	C
45 - 54	:	D
0 - 44	:	E

Kelas 1D :

4. Buatlah program untuk memasukkan nama barang, jumlah, harga satuan, total harga suatu barang dagangan Toko Manis. Tampilan (layout) seperti di bawah ini :

TOKO MANIES

Jl. Setia Budi no. 10

Madiun

NO	NAMA BARANG	JML BRG	HARGA SATUAN	JUMLAH HARGA
1	<input>	<input>	<input>	<output>
2	<input>	<input>	<input>	<output>
3	<input>	<input>	<input>	<output>
4	<input>	<input>	<input>	<output>
5	<input>	<input>	<input>	<output>
TOTAL HARGA				<output>

Kelas 1E

1. Buatlah program untuk menentukan range nilai dari inputan data sbb :
NIM, Nama, Nilai Tugas, Nilai Keaktifan, Nilai UTS, Nilai UAS,
Nilai Tugas mempunyai komposisi 20%, nilai keaktifan mempunyai komposisi 10%, nilai UTS mempunyai komposisi 30%, nilai UAS mempunyai komposisi 40%, Kemudian Hitung total nilai dan rata nilai.
Jika Rata nilai:

90 – 100	:	A
85 – 89	:	A-
76 – 84	:	B
68 - 74	:	B-
55 – 67	:	C
45 – 54	:	D
0 - 44	:	E

STRUKTUR KONTROL KEPUTUSAN 3

5

Standar Kompetensi

5. Mampu memahami struktur keputusan yang ada di C++

Kompetensi Dasar

1. Dapat memahami struktur kontrol (if, if-else, if-else-if, switch) untuk memilih alur blok yang akan dieksekusi.
2. Dapat membedakan penggunaan masing – masing struktur kontrol (SWITCH-CASE)

5.1. Pernyataan switch – case

Bentuk dari **switch - case** merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif penyelesaian. Pernyataan **switch - case** ini memiliki kegunaan sama seperti if – else bertingkat, tetapi penggunaannya untuk memeriksa data yang bertipe karakter atau integer.

Bentuk penulisan perintah ini sebagai berikut :

```
switch (ekspresi integer atau karakter )
{
    case konstanta-1 :
        ... perintah;
        ... perintah;
        break;
    case konstanta-2 :
        ... perintah;
        ... perintah;
        break;
    .....
    .....
    default :
        ... perintah;
        ... perintah;
}
```

Setiap cabang akan dijalankan jika syarat nilai konstanta tersebut dipenuhi dan **default** akan dijalankan jika semua cabang di atasnya tidak terpenuhi.

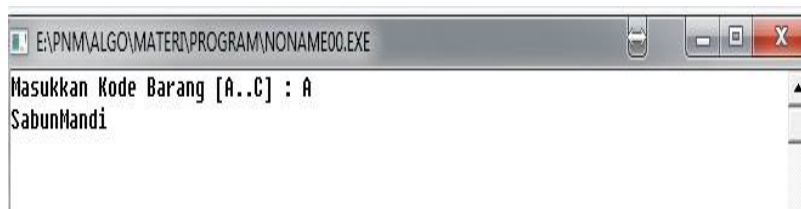
Pernyataan *break* menunjukkan bahwa perintah siap keluar dari *switch*. Jika pernyataan ini tidak ada, maka program akan diteruskan ke cabang – cabang yang lainnya.

Example:

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
main()
{
    char kode;
    clrscr();
    cout<<"Masukkan Kode Barang [A..C] : ";
    cin>>kode;

    switch(kode)
    {
        case 'A' :
            cout<<"SabunMandi";
            break;
        case 'B' :
            cout<<"SabunCuci";
            break;
        case 'C' :
            cout<<"Deterjent";
            break;
        default:
            cout<<"Anda Salah Memasukan kode";
            break;
    }
    getch();
}
```

HASIL RUN:

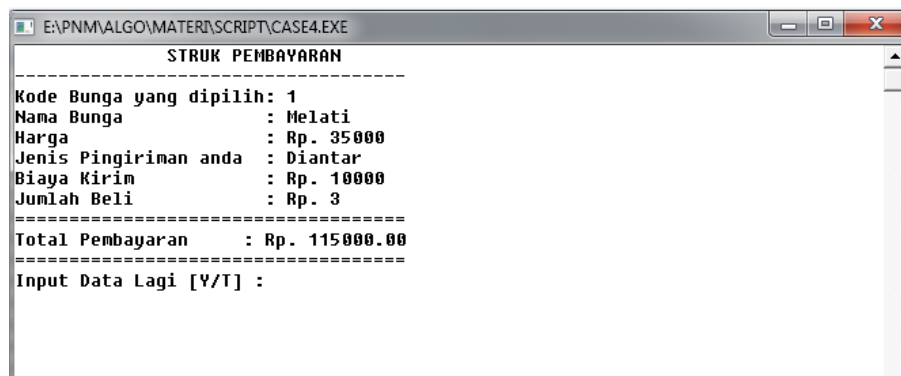


EXAMPLE 2:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <string.h>
main()
{
char pil,nb[10],ket[20],jwb;
float kode, h,biaya,jb,total;
lagi:
    clrscr();    // CONTOH LAT CASE DAN IF
    <<"TOKO BUNGA LESTARI"<<endl;
    cout<<"1. Melati    Rp. 35000/ Bungkus"<<endl;
    cout<<"2. Mawar     Rp. 25000/ Tangkai"<<endl;
    cout<<"3. Tulip      Rp. 45000/ Tangkai"<<endl;
    cout<<"Pilih Kode Bunga      : ";cin>>kode;
    cout<<endl;
    cout<<"Pilih Jenis Pengiriman"<<endl;
    cout<<"A. Diantar"<<endl;
    cout<<"B. Tidak Diantar"<<endl;
    cout<<"Pilihan Anda          : ";cin>>pil;
    switch (kode==1)
    {
    case 1:
    {
    strcpy(nb,"Melati");
    h=35000;
    }break;
    case 2:
    {
    strcpy(nb,"Mawar");
    h=25000;
    }break;
    default:
    {
    strcpy(nb,"Tulip");
    h=45000;
    }break;
    }
    if (pil=='A' || pil=='a')
    {
    strcpy(ket,"Diantar");
    biaya=10000;
    }else
    {
    strcpy(ket,"Tidak Diantar");
    biaya=0;
    }
}
```

```
cout<<"Jumlah Beli                : ";cin>>jb;
total=(jb*h)+biaya;
clrscr();
cout<<"                        STRUK PEMBAYARAN"<<endl;
cout<<"-----"<<endl;
cout<<"Kode Bunga yang dipilih: "<<kode<<endl;
cout<<"Nama Bunga                : "<<nb<<endl;
cout<<"Harga                      : Rp. "<<h<<endl;
cout<<"Jenis Pingiriman anda      : "<<ket<<endl;
cout<<"Biaya Kirim                : Rp. "<<biaya<<endl;
cout<<"Jumlah Beli                : Rp. "<<jb<<endl;
cout<<"===== "<<endl;
printf("Total Pembayaran          : Rp. %2.2f\n",total);
cout<<"===== "<<endl;
cout<<"Input Data Lagi [Y/T] : ";jwb=getche();
if (jwb=='Y' || jwb=='y')
goto lagi;
else
goto keluar;
getch();
keluar:
}
```

HASIL RUN;



5.2. Rangkuman

SWITCH - CASE merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif penyelesaian.



5.3. Latihan

1. **KLAS-1A** : Buatlah program untuk hasil berikut ini dengan pemakaian struktur case:

```
Menentukan hari

1 = SENIN   3 = RABU   5 = JUM'AT   7 = MINGGU
2 = SELASA  4 = KAMIS  6 = SABTU

Masukkan kode hari (1..7): 5
Hari Jum'at
```

2. **KLAS-1B** : Buatlah program untuk hasil berikut ini dengan pemakaian struktur case:

```
Masukkan kode Bulan (1..12) = 12
Bulan Desember_
```

3. **KLAS-1C** : Buatlah program untuk hasil berikut ini dengan pemakaian struktur case:

```
Masukan Nama Anda: dimas
Selamat Datang dimas Di KOPMA UNIPMA
macam macam barang :
1. Buku Tulis
2. Pensil
3. penghapus 2
4. Spidol
5. Penggaris
Pilih barang yang ingin dibeli (1-5) : 5

Barang Yang Dibeli : penggaris
Harga Satuan: Rp 4000
```

4. **KLAS-1D** : Buatlah program untuk hasil berikut ini dengan pemakaian struktur case:

```
Masukan Nama Anda: samid
Selamat Datang samid Di Bengkel Yamaha
macam macam barang :
1. Velg racing
2. Spion
3. Knalpot 2
4. Busi
5. oli
Pilih barang yang ingin dibeli (1-5) : 4

Barang Yang Dibeli : Busi
Harga Satuan: Rp 35000
_
```


5. **KLAS-1E** : Buatlah program untuk hasil berikut ini dengan pemakaian struktur case:
Menentukan jumlah hari setiap bulan dengan ketentuan sebagai berikut:
Bulan 1,3,5,7,8,10,12 jumlah hari 31
Bulan 4,6,9,11 jumlah hari 30
Bulan 2 jumlah hari 28, apabila tahun kabisat 29

MEMAHAMI PERULANGAN 1

6

Standar Kompetensi

2. Mahasiswa mampu membuat program perulangan dengan Borland C++

Kompetensi Dasar

1. Mengetahui dan memahami Perulangan dalam Bahasa Pemrograman C++
2. Memahami penulisan dan Deklarasi Perulangan
3. Memahami tujuan dan penggunaan Perulangan

6.1 Pernyataan FOR

Perulangan yang pertama adalah *for*. Bentuk umum pernyataan *for* sebagai berikut :

FOR (inisialisasi; syarat pengulangan; pengubah nilai pencacah)

Bila pernyataan didalam *for* lebih dari satu maka pernyataan-pernyataan tersebut harus diletakan didalam tanda kurung.

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )  
{  
    pernyataan / perintah;  
    pernyataan / perintah;  
    pernyataan / perintah;  
}
```

Kegunaan dari masing-masing argumen *for* diatas adalah :

- Inisialisasi : merupakan bagian untuk memberikan nilai awal untuk variabel-variabel tertentu.
- Syarat Pengulangan : memegang kontrol terhadap pengulangan, karena bagian ini yang akan menentukan suatu perulangan diteruskan atau dihentikan.
- Pengubah Nilai Pencacah : mengatur kenaikan atau penurunan nilai pencacah.

EXAMPLE1:

```
/* ----- */
/* Program for - bilangan naik */
/* ----- */
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
main()
{
    int a;
    clrscr();
    for(a = 1; a <= 10; ++a)
        cout<<a;

    getch();
}
```

HASIL RUN:



EXAMPLE 2:

```
/* ----- */
/* Program for - bilangan turun */
/* ----- */
# include <stdio.h>
# include <conio.h>
#include<iostream.h>
main()
{
    int a;
    clrscr();
    for(a = 10; a >= 1; --a)
        cout<<a;

    getch();
}
```

HASIL RUN:

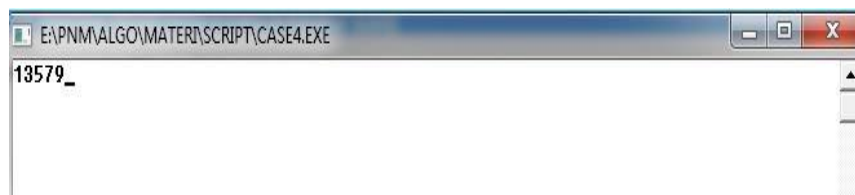


EXAMPLE 3:

```
/* Program for - bilangan ganjil */
/* ----- */
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
main()
{
    int a;
    clrscr();
    for(a = 1; a <= 10; a+=2)
        cout<<a;

    getch();
}
```

HASIL RUN;



6.2 PERULANGAN TIDAK BERHINGGA

Perulangan tak berhingga merupakan perulangan (loop) yang tak pernah berhenti atau mengulang terus, hal ini sering terjadi disebabkan adanya kesalahan penanganan kondisi yang dipakai untuk keluar dari loop.

Example:

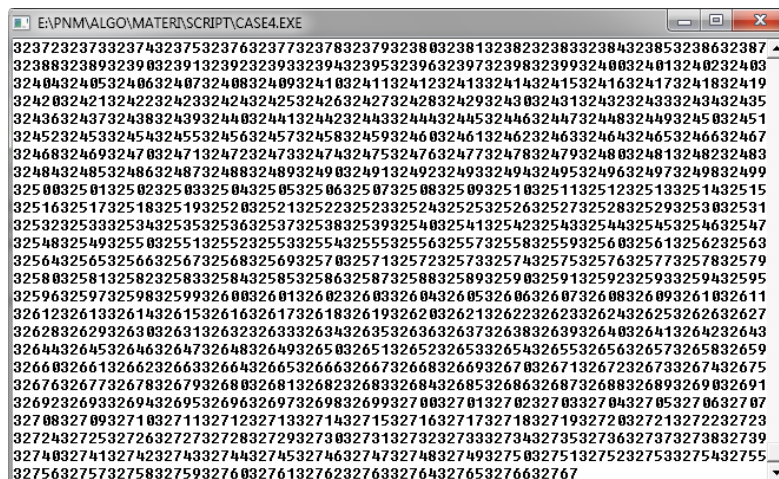
```

/* ----- */
/* Program for Tdk Berhingga */
/* ----- */
#include<stdio.h>
#include<conio.h>
main()
{
    int bil;
    clrscr();
    for (bil = 60; bil >=10; bil++)
        printf("%d", bil);

    getch();
}

```

Hasil Run;



```

32372323733237432375323763237732378323793238032381323823238332384323853238632387
32388323893239032391323923239332394323953239632397323983239932400324013240232403
32404324053240632407324083240932410324113241232413324143241532416324173241832419
32420324213242232423324243242532426324273242832429324303243132432324333243432435
32436324373243832439324403244132442324433244432445324463244732448324493245032451
32452324533245432455324563245732458324593246032461324623246332464324653246632467
32468324693247032471324723247332474324753247632477324783247932480324813248232483
32484324853248632487324883248932490324913249232493324943249532496324973249832499
32500325013250232503325043250532506325073250832509325103251132512325133251432515
32516325173251832519325203252132522325233252432525325263252732528325293253032531
32532325333253432535325363253732538325393254032541325423254332544325453254632547
32548325493255032551325523255332554325553255632557325583255932560325613256232563
32564325653256632567325683256932570325713257232573325743257532576325773257832579
32580325813258232583325843258532586325873258832589325903259132592325933259432595
32596325973259832599326003260132602326033260432605326063260732608326093261032611
32612326133261432615326163261732618326193262032621326223262332624326253262632627
32628326293263032631326323263332634326353263632637326383263932640326413264232643
32644326453264632647326483264932650326513265232653326543265532656326573265832659
32660326613266232663326643266532666326673266832669326703267132672326733267432675
32676326773267832679326803268132682326833268432685326863268732688326893269032691
32692326933269432695326963269732698326993270032701327023270332704327053270632707
32708327093271032711327123271332714327153271632717327183271932720327213272232723
32724327253272632727327283272932730327313273232733327343273532736327373273832739
32740327413274232743327443274532746327473274832749327503275132752327533275432755
327563275732758327593276032761327623276332764327653276632767

```



6.3. Rangkuman

Perulangan adalah proses pengulangan kata atau unsur kata. Reduplikasi juga merupakan proses penurunan kata dengan perulangan utuh maupun sebagian.



6.4. Latihan

1. **KLAS-1A** : Buatlah program C++ dengan perulangan FOR.sebanyak 5x.
Sehingga hasil seperti di bawah ini

```
Tetap semangat belajar C++  
Tetap semangat belajar C++  
Tetap semangat belajar C++  
Tetap semangat belajar C++  
Tetap semangat belajar C++  
-
```

2. **KLAS-1B** : Buatlah program C++ untuk menggunakan FOR.sebanyak 10x secara ascending. Sehingga hasil seperti di bawah ini

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

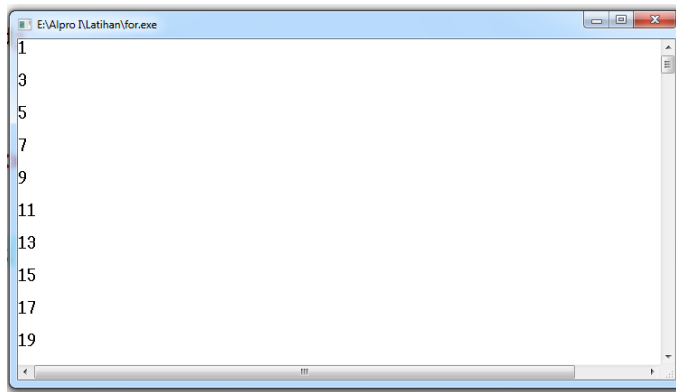
3. **KLAS-1C** : Buatlah program C++ untuk menggunakan FOR. Sehingga hasil seperti di bawah ini

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

4. **KLAS-1D** : Buatlah program C++ untuk menggunakan FOR. Sehingga hasil seperti di bawah ini

```
Pengulangan  
1  
5  
9  
13  
17  
21  
25
```

5. **KLAS-1E** : Buatlah program C++ untuk menggunakan FOR. Sehingga hasil seperti di bawah ini



MEMAHAMI PERULANGAN 2

7

Standar Kompetensi

3. Mahasiswa mampu membuat program perulangan dengan Borland C++

Kompetensi Dasar

1. Mengetahui dan memahami Perulangan dalam Bahasa Pemrograman C++
2. Memahami penulisan dan Deklarasi Perulangan
3. Memahami tujuan dan penggunaan Perulangan

7.1 PERNYATAAN NESTED – FOR

Pernyataan Nested *for* adalah suatu perulangan *for* didalam perulangan *for* yang lainnya. Bentuk umum pernyataan *Nested for* sebagai berikut :

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
{
    for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
    {
        pernyataan / perintah;
    }
}
```

Didalam penggunaan nested-for, perulangan yang didalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang diluar diselesaikan

EXAMPLE:

```
/* ----- */
/* Program for - Nested for */
/* ----- */
#include<stdio.h>
#include<conio.h>
main()
{
    int a, b;
    clrscr();
    for(a = 1; a <= 5; a++)
    {
        printf("\n");
        for(b = a; b <= 5; b++)
            printf(" %d ",a);
    }
    getch();
}
```


HASIL RUN :



```
E:\PNM\ALGO\MATERI\SCRIPT\CASE4.EXE
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5

```



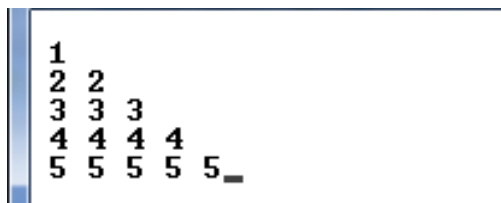
7.2. Rangkuman

Perulangan adalah proses pengulangan kata atau unsur kata. Reduplikasi juga merupakan proses penurunan kata dengan perulangan utuh maupun sebagian.



7.3. Latihan

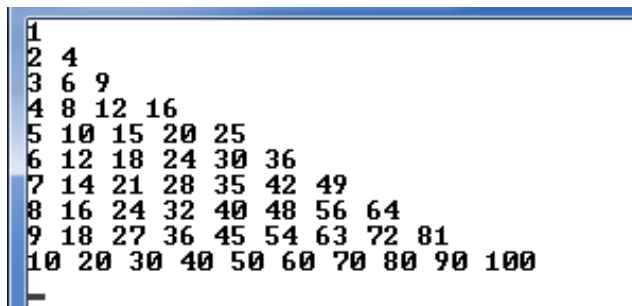
1. **KLAS-1A** : Buatlah program C++ dengan menggunakan NESTED FOR. sehingga Hasil seperti di bawah ini



```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

```

2. **KLAS-1B** : Buatlah program C++ dengan menggunakan NESTED FOR. sehingga Hasil seperti di bawah ini



```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
9 18 27 36 45 54 63 72 81
10 20 30 40 50 60 70 80 90 100

```

3. **KLAS-1C** : Buatlah program C++ dengan menggunakan NESTED FOR. sehingga Hasil seperti di bawah ini

```
123456
23456
3456
456
56
6
```

4. **KLAS-1D** : Buatlah program C++ dengan menggunakan NESTED FOR. sehingga Hasil seperti di bawah ini

```
Program Segitiga
Segitiga Siku-siku Rata Kanan

Masukkan Tinggi Segitiga : 5

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

5. **KLAS-1E** : Buatlah program C++ dengan menggunakan NESTED FOR. sehingga Hasil seperti di bawah ini

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

MEMAHAMI PERULANGAN 3

8

Standar Kompetensi

4. Mahasiswa mampu membuat program perulangan dengan Borland C++

Kompetensi Dasar

1. Mengetahui dan memahami Perulangan dalam Bahasa Pemrograman C++
2. Memahami penulisan dan Deklarasi Perulangan
4. Memahami tujuan dan penggunaan Perulangan WHILE

8.1 PERNYATAAN WHILE

Pernyataan perulangan *while* merupakan instruksi perulangan yang mirip dengan perulangan *for*. Bentuk perulangan *while* dikendalikan oleh syarat tertentu, yaitu perulangan akan terus dilaksanakan selama syarat tersebut terpenuhi.

Bentuk umum perulangan *while*, sebagai berikut :

```
while ( syarat )
```

```
    Pernyataan / perintah ;
```

Bentuk umum perulangan *while*, dengan lebih dari perintah / pernyataan, sebagai berikut :

```
while ( syarat )  
{  
    Pernyataan / perintah ;  
    Pernyataan / perintah ;  
}
```

Example:

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int i; // Sebagai variabel pencacah yang menyatakan
           // jumlah tulisan C++ yang harus ditampilkan
    clrscr(); // Hapus layar
    i = 0;    // Mula-mula diisi sama dengan nol
    while ( i < 10 )
    {
        cout << " C++ " << endl;
        i ++ ;    // Menaikkan pencacah sebesar 1
    }
}
```

HASIL RUN:



8.2. Rangkuman

while merupakan instruksi perulangan yang mirip dengan perulangan **for**. Bentuk perulangan **while** dikendalikan oleh syarat tertentu



8.3. Latihan

1. **KLAS-1A** : Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

MAIN MENU

1. Input Data
2. Edit Data
3. Delete Data
4. Exit

Your Choise (1/2/3/4) ? ...

Apabila dipilih menu no 1, maka akan tampil teks “Anda memilih menu 1”. Demikian pula untuk menu 2 dan 3. Kemudian setelah itu muncul teks “Tekan ENTER untuk kembali ke menu utama”. Artinya begitu kita tekan ENTER menu pilihan akan muncul kembali, dst. Akan tetapi bila yang dipilih menu 4 (EXIT), program langsung berhenti.

2. **KLAS-1B** : Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

```
=====
Latihan Perulangan C++ dengan While
=====
1.Menampilkan deret angka 1 sampai 10      : 012345678910
2.Menampilkan deret bilangan genap dari angka 1 sampai 10 : 246810
```

3. **KLAS-1C** : Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

```
=====
Latihan Perulangan C++ dengan While
=====
1.Menampilkan deret angka 1 sampai 10      : 012345678910
3.Menampilkan deret bilangan ganjil dari angka 1 sampai 10 : 13579
```

4. **KLAS-1D** : Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

```
=====
Latihan Perulangan C++ dengan While
=====
1.Menampilkan deret angka 1 sampai 10      : 012345678910
```

5. **KLAS-1E** : Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE

```
Masukkan Banyak Data =10
2
4
6
8
10
```

MEMAHAMI PERULANGAN 4

9

Standar Kompetensi

5. Mahasiswa mampu membuat program perulangan dengan Borland C++

Kompetensi Dasar

1. Mengetahui dan memahami Perulangan dalam Bahasa Pemrograman C++
2. Memahami penulisan dan Deklarasi Perulangan
3. Memahami tujuan dan penggunaan Perulangan DO WHILE

9.1. PERNYATAAN DO – WHILE

Pernyataan perulangan **do - while** merupakan bentuk perulangan yang melaksanakan perulangan terlebih dahulu dan pengujian perulangan dilakukan dibelakang.

Bentuk umum perulangan **do - while**, sebagai berikut :

```
do
    pernyataan / perintah ;
while ( syarat );
```

Bentuk umum perulangan **do - while**, dengan lebih dari perintah / pernyataan, sebagai berikut :

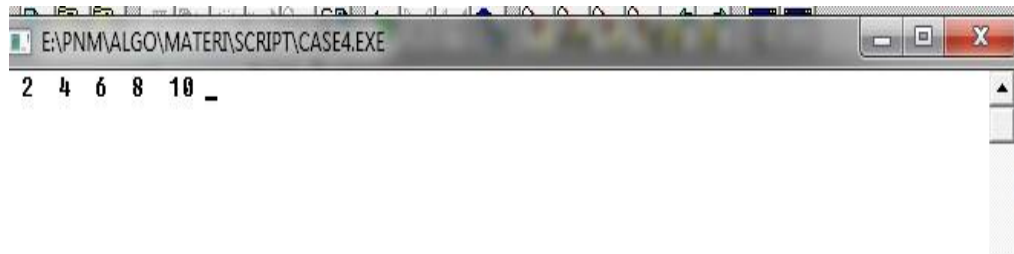
```
do
{
    Pernyataan / perintah ;
    Pernyataan / perintah ;
}

while ( syarat );
```

Example:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int bil=2;
    clrscr();
    do
    {
        printf(" %d ",bil);
        bil+=2;
    }
    while (bil<=10);
    getch();
}
```

HASIL RUN:



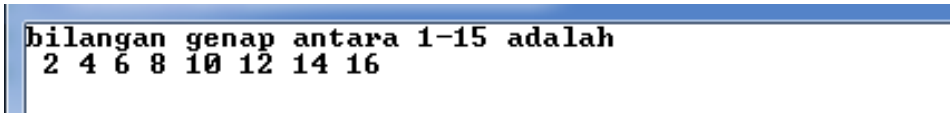
9.2. Rangkuman

Pernyataan perulangan **do - while** merupakan bentuk perulangan yang melaksanakan perulangan terlebih dahulu dan pengujian perulangan dilakukan dibelakang.



9.3. Latihan

1. **KLAS-1A** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan Do While



```
bilangan genap antara 1-15 adalah
2 4 6 8 10 12 14 16
```

2. **KLAS-1B** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan Do While

```
Nilai ke 1 = 5
Nilai ke 2 = 10
Nilai ke 3 = 15
Nilai ke 4 = 20
Nilai ke 5 = 25
Nilai ke 6 = 30
Nilai ke 7 = 35
Nilai ke 8 = 40
Nilai ke 9 = 45
Nilai ke 10 = 50
```

3. **KLAS-1C** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan Do While.

```
Input banyak nilai = 2
Input Nilai 1 = 12
Input Nilai 2 = 32

Total nilai = 44
Rata nilai = 22
```

4. **KLAS-1D** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan Do While

```
banyak barang yang dimasukkan = 2
Barang ke 1 = meja
Barang ke 2 = kursi

anda sudah memasukkan semua barang_
```

5. **KLAS-1E** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan Do While

```
Masukkan Banyaknya Data =2
Pelanggan ke 1
Masukkan Nama =daniel
Masukkan Harga =20000
Masukkan Jumlah =3
Masukkan Diskon =10
Bayar =54000
Total Semua =54000
Pelanggan ke 2
Masukkan Nama =danu
Masukkan Harga =30000
Masukkan Jumlah =2
Masukkan Diskon =5
Bayar =57000
Total Semua =111000
```


MEMAHAMI BREAK, CONTINUE, GOTO

10

Standar Kompetensi

6. Mahasiswa mampu membuat program memakai BREAK, CONTINUE, GOTO dengan Borland C++

Kompetensi Dasar

1. Mengetahui dan memahami 3 PERINTAH PENTING dalam Bahasa Pemrograman C++
2. Memahami penulisan dan Deklarasi BREAK, CONTINUE, GOTO
3. Memahami tujuan dan penggunaan BREAK, CONTINUE, GOTO

10.1. PERNYATAAN BREAK

Pernyataan *break* telah dibahas pada pernyataan pengambilan keputusan *switch*. Pernyataan *break* ini berfungsi untuk keluar dari struktur *switch*. Selain itu pernyataan *break* berfungsi keluar dari perulangan (*for*, *while* dan *do-while*). Jika pernyataan *break* dikerjakan, maka eksekusi akan dilanjutkan ke pernyataan yang terletak sesudah akhir dari badan perulangan (*loop*).

Example:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int bil = 1;
    clrscr();
    do
    {
        if (bil >= 6)
            break;
        printf(" %d ",bil);
    }
    while(bil++);
    getch();
}
```

Hasil Run:



10.2. PERNYATAAN CONTINUE

Pernyataan *continue* digunakan untuk mengarahkan eksekusi ke iterasi (proses) berikutnya pada loop yang sama, dengan kata lain mengembalikan proses yang sedang dilaksanakan ke-awal loop lagi, tanpa menjalankan sisa perintah dalam loop tersebut.

EXAMPLE:

```
#include <stdio.h>
#include <conio.h>
main()
{
    int bil;
    clrscr();

    for(bil=1; bil<=10; ++bil)
    {
        if(bil==6)
            continue;

        printf(" %d ",bil);
    }
    getch();
}
```

HASIL RUN:



10.3. PERNYATAAN GOTO

Pernyataan **goto** merupakan instruksi untuk mengarahkan eksekusi program ke-pernyataan yang diawali dengan suatu label. Label merupakan suatu pengenalan (*identifier*) yang diikuti dengan tanda titik dua (:).

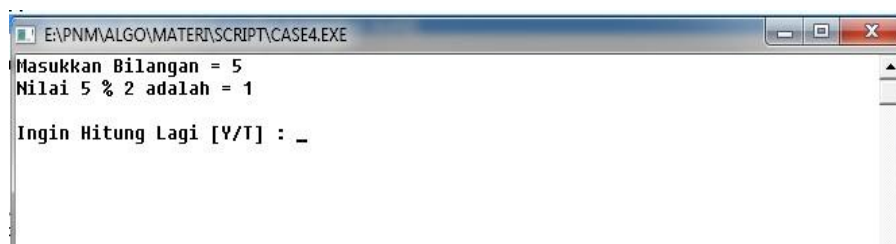
Bentuk pemakaian **goto** sebagai berikut :

goto label;

EXAMPLE:

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
main()
{
    int a, b;
    char lagi;
    atas:
        clrscr();
        cout<<"Masukkan Bilangan = ";
        cin>>a;
        b = a % 2;
        printf("Nilai %d %% 2 adalah = %d",a, b);
        printf("\n\nIngin Hitung Lagi [Y/T] : ");
        lagi = getch();
        if (lagi == 'Y' || lagi == 'y')
            goto atas;
        getch();
}
```

HASIL RUN:



```
E:\PNM\ALGO\MATERI\SCRIPT\CASE4.EXE
Masukkan Bilangan = 5
Nilai 5 % 2 adalah = 1
Ingin Hitung Lagi [Y/T] : _
```



10.4. Rangkuman

break berfungsi untuk keluar dari struktur *switch*, Selain itu pernyataan *break* berfungsi keluar dari perulangan (*for*, *while* dan *do-while*). *continue* digunakan untuk

mengarahkan eksekusi ke iterasi (proses) berikutnya pada loop yang sama . **goto** merupakan instruksi untuk mengarahkan eksekusi program ke-pernyataan yang diawali dengan suatu label



10.5. Latihan

1. **KLAS-1A** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan **switch case & Break**

```
Kode Barang <1..3> : 3
Nomer barang tersedia dan stok tersisa 30.
```

2. **KLAS-1B** : Dengan menggunakan **Go to** silahkan buat simulasi program dimana program akan berhenti ketika dimasukkan angka 5.

```
Masukan Angka : 1
Masukan Angka : 2
Masukan Angka : 3
Masukan Angka : 4
Masukan Angka : 1
```

3. **KLAS-1C** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan **continue**

```
Jumlah : 10
1
2
3
4
6
7
8
9
10
```

4. **KLAS-1D** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan **continue**

```
Jumlah : 5
```

```
1  
2  
4  
5
```

5. **KLAS-1E** : Buatlah simulasi program dengan tampilan di bawah ini menggunakan **goto**

```
Masukkan Angka [1..10] =2
```

```
2
```

```
Masukkan Angka [1..10] =3
```

```
3
```

```
Masukkan Angka [1..10] =10
```

```
10
```

```
Masukkan Angka [1..10] =11
```

```
11
```

MODUL PRAKTIKUM ALGORITMA & PEMROGRAMAN I

```
    }  
  
    code = curl_easy_setopt(conn, CURLOPT_URL, url);  
    if (code != CURLE_OK)  
    {  
        fprintf(stderr, "Failed to set URL [%s]\n", errorBuffer);  
        return false;  
    }  
  
    code = curl_easy_setopt(conn, CURLOPT_FOLLOWLOCATION, 1L);  
    if (code != CURLE_OK)  
    {  
        fprintf(stderr, "Failed to set redirect option [%s]\n", errorBuffer);  
        return false;  
    }
```

PROGRAM STUDI INFORMATIKA
UNIVERSITAS PGRI MADIUN
2021/2022