



**POLITEKNIK KAMPAR**  
**KOMPETEN DAN PROFESIONAL**

# PEMROGRAMAN DASAR

**PROGRAM STUDI**  
**TEKNIK INFORMATIKA**

Andri Nofiar.Am,M.Kom





## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	i
<b>DAFTAR ISI</b> .....	ii
<b>Modul 1</b> .....	1
<b>Menginstall MinGW</b> .....	1
<b>A. Langkah Kerja</b> .....	1
<b>B. Pembahasan</b> .....	4
<b>Modul 2</b> .....	5
<b>Membuat program menampilkan “Hallo Dunia”</b> .....	5
<b>A. Langkah kerja</b> .....	5
<b>B. Pembahasan</b> .....	7
<b>Modul 3</b> .....	8
<b>Program Aritmatika Sederhana</b> .....	8
<b>A. Langkah Kerja</b> .....	8
<b>B. Pembahasan</b> .....	9
<b>Modul 4</b> .....	15
<b>Membuat Program Menghitung Luas Segitiga dengan Nilai Alas dan Tinggi Merupakan Inputan dari Pengguna</b> .....	15
<b>A. Langkah Kerja</b> .....	15
<b>B. Pembahasan</b> .....	16
<b>Latihan 1</b> .....	17
<b>Modul 5</b> .....	18
<b>Increment dan Decrement</b> .....	18
<b>A. Langkah Kerja</b> .....	18
<b>B. Pembahasan</b> .....	19





<b>A. Langkah Kerja .....</b>	<b>36</b>
<b>B. Pembahasan .....</b>	<b>36</b>
<b>Modul 12 .....</b>	<b>38</b>
<b>Larik 2 Dimensi .....</b>	<b>38</b>
<b>A. Langkah Kerja .....</b>	<b>38</b>
<b>B. Pembahasan .....</b>	<b>39</b>
<b>DAFTAR PUSTAKA .....</b>	<b>40</b>

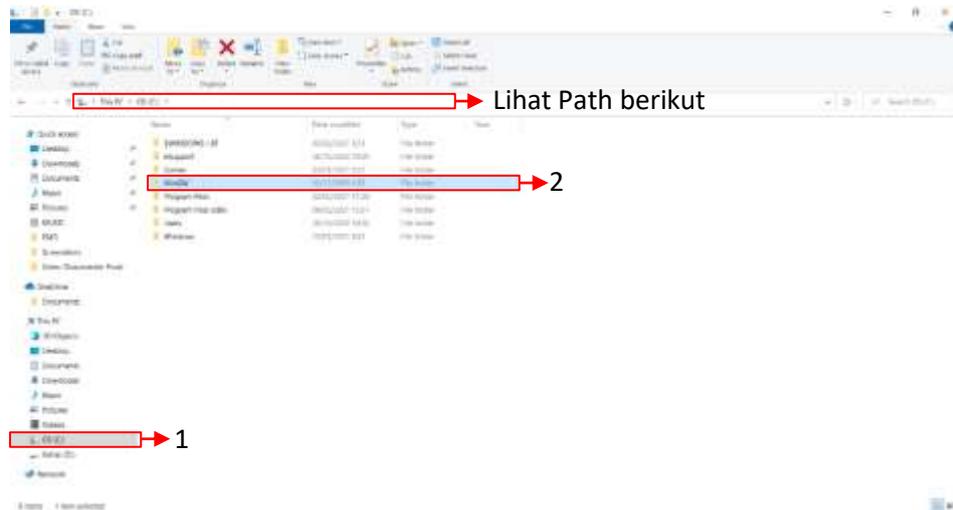


## Modul 1

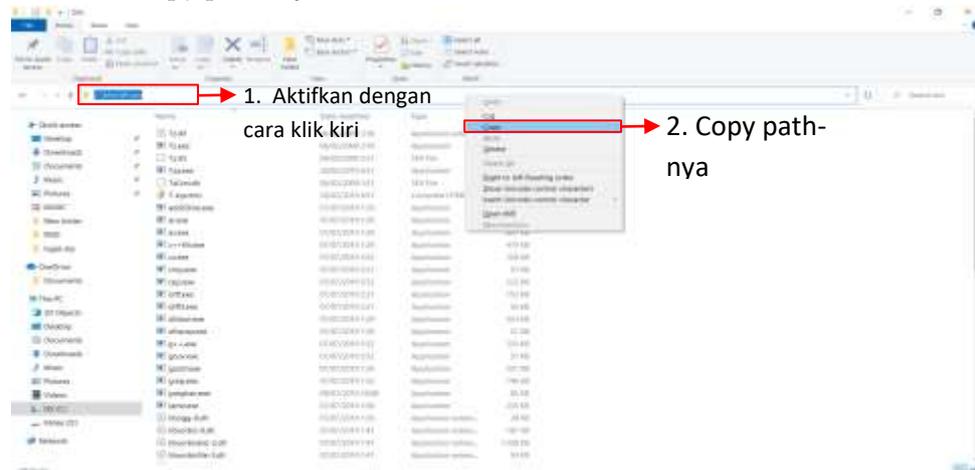
### Menginstall MinGW

#### A. Langkah Kerja

1. Ekstrak file MinGW ke partisi C:\ (Partisi Sistem)

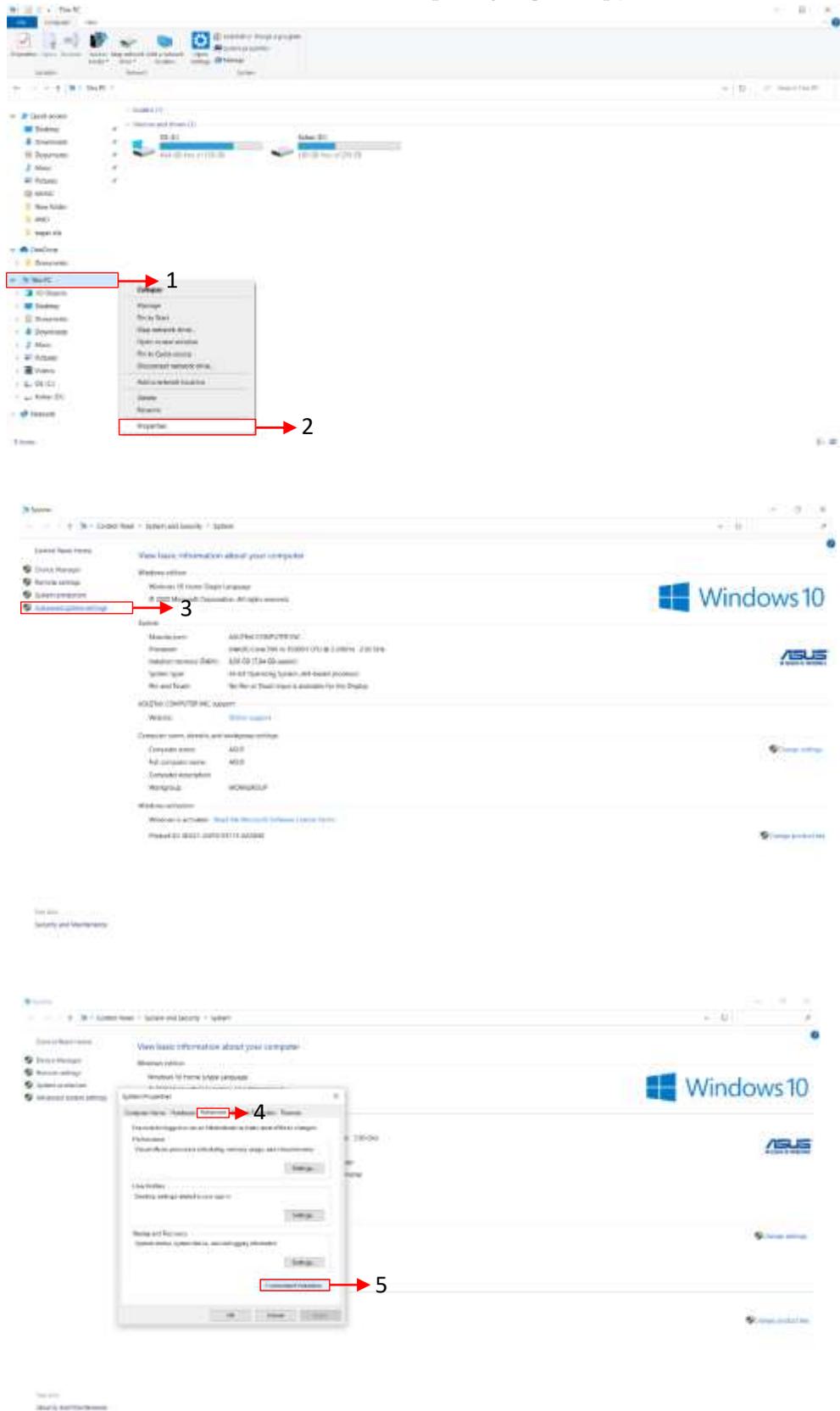


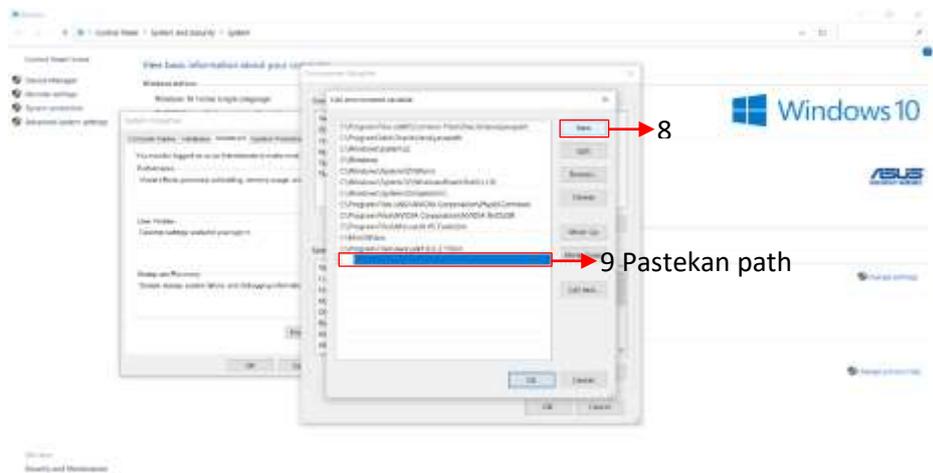
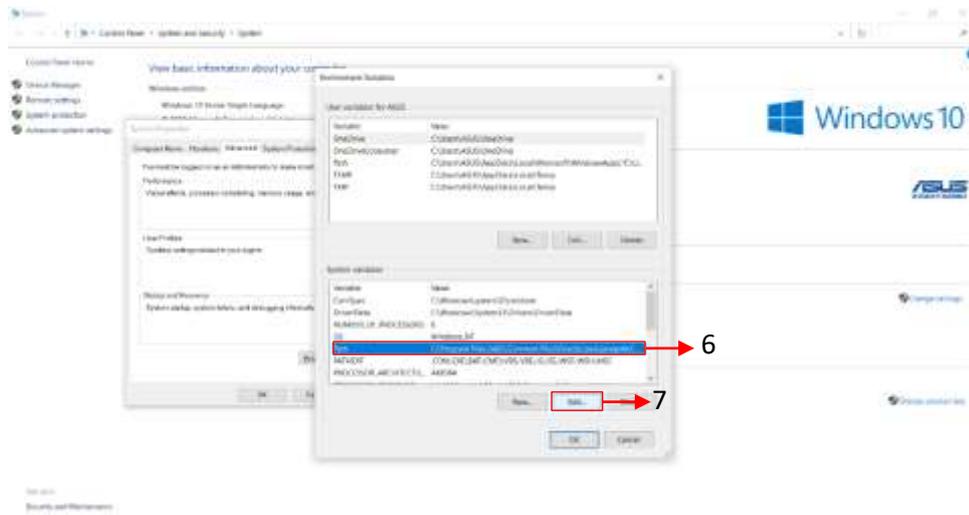
2. Atur *Environment Variable*-nya, Masuk ke folder C:\MinGW\bin kemudian *copy path*-nya





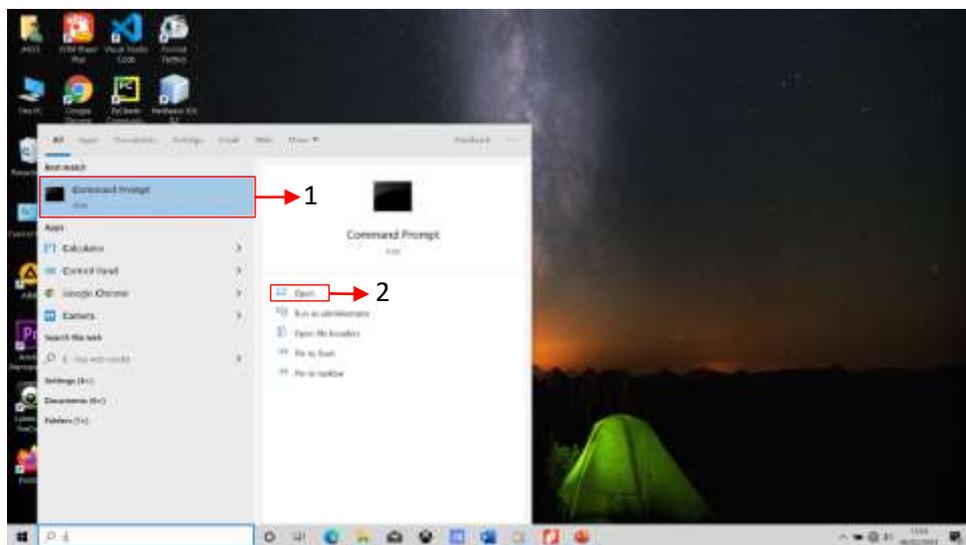
3. Buka *Environment Variable* Paste-kan path yang di-copy





Klik **OK** sampai semua jendela tertutup

#### 4. Buka *Command Prompt*





5. Ketik `g++ --version` untuk melihat apakah proses instalasi MinGW berhasil

```
Microsoft Windows [Version 10.0.19041.208]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>g++ --version 1
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.

C:\Users\ASUS>
```

MinGW berhasil di install

## B. Pembahasan

Bagaimanapun ceritanya mesin tetaplah mesin, komputer tidak akan pernah bisa mengerti langsung perintah manusia, maka dari itu agar komputer dapat mengerti perintah manusia diperlukan sebuah pengubah bahasa atau biasa disebut *compiler*. Yang barusan kita lakukan adalah meng-*install* pengubah tersebut agar program yang kita buat dapat dimengerti oleh komputer dan dapat dijalankan.

Untuk membuka *Environment variables* dapat menggunakan shortcut sebagai berikut, pertama tekan logo *windows* + **R** pada *keyboard* secara bersamaan, kemudian ketikkan *sysdm.cpl* setelah itu *klik enter*.

Untuk menjalankan Command Prompt pada windows dapat digunakan *shortcut* atau jalan pintas sebagai berikut, pertama tekan logo *windows* + **R** pada *keyboard* secara bersamaan, kemudian ketikkan *cmd* kemudian tekan *enter*.

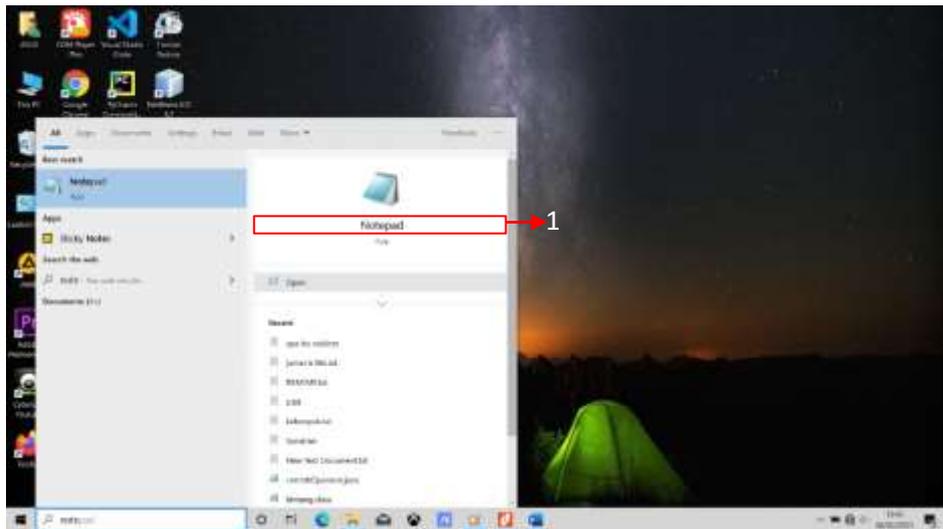


## Modul 2

### Membuat program menampilkan “Hallo Dunia”

#### A. Langkah kerja

##### 1. Buka *Notepad*



##### 2. Ketikkan *SourceCode* berikut :

```
/*
 * Program 1 helloworld.c
 * Nama : "buat nama Anda Disini"
 * Nim  : "Masukkan Nim Anda Disini"
 * Baris ini adalah komentar tidak akan dieksekusi
 */
#include <stdio.h>
#include <stdlib.h>
int main(void) //Fungsi yang harus ada pada tiap program C (ini merupakan
komentar)
{ // Kurung Kurawal pembuka Program C

printf("Hallo Dunia \n"); //Menampilkan tulisan "Hallo Dunia" dan membuat
karakter enter

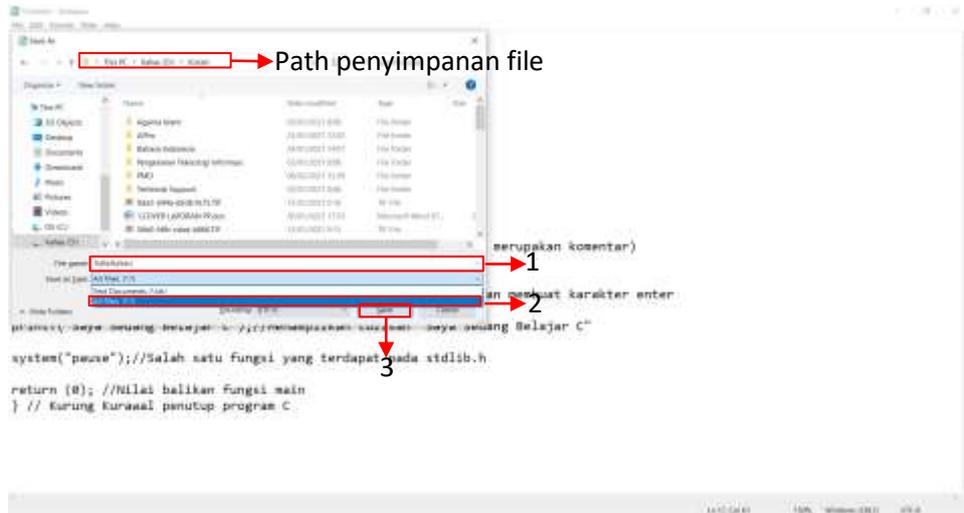
printf("Saya Sedang Belajar C");//Menampilkan tulisan "Saya Sedang Belajar
C"

system("pause");//Salah satu fungsi yang terdapat pada stdlib.h

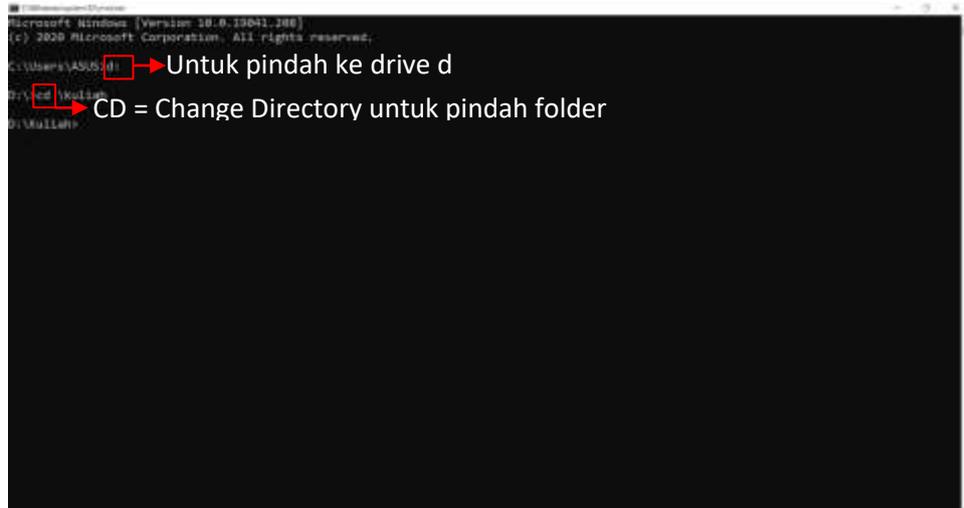
return (0); //Nilai balikan fungsi main
} // Kurung Kurawal penutup program C
```



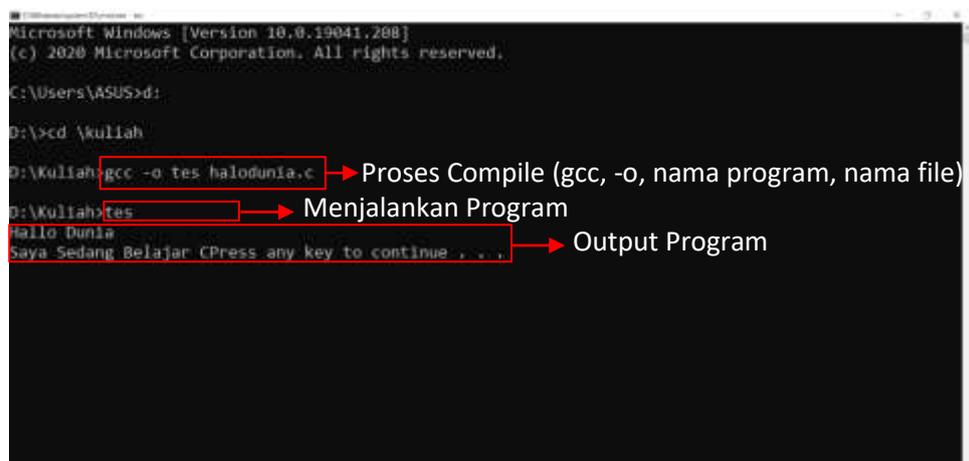
3. Simpan *SourceCode* diatas



4. Buka *Command Prompt* arahkan ke tempat menyimpan *file*



5. *Compile* program kemudian jalankan





## B. Pembahasan

Program yang baru saja kita buat adalah Langkah awal ke program berikutnya, kita harus tau bagaimana menampilkan kata “Hallo Dunia” ke layar. Perlu kita ketahui bahwa Bahasa C adalah Bahasa terstruktur, dimana baris program dibaca terurut dari atas sampai kebawah.

Pada pemrograman Bahasa C yang pertama kali harus kita tuliskan ialah *header*. *Header* ini berisi fungsi-fungsi yang akan kita gunakan. Sebagai contoh kita ingin menggunakan fungsi **printf**, nah, fungsi **printf** ini terdapat dalam header **#include <stdio.h>**. Jadi jika kita ingin menggunakan fungsi **printf** maka terlebih dahulu kita harus menuliskan **#include <stdio.h>** baru bisa menuliskan fungsi **printf** dibawah.

Kemudian setelah menuliskan *header*, kita harus menuliskan fungsi utama atau main method. Kita bisa mengetikkan **int main** untuk program yang memiliki nilai balikan, dan **void main** untuk program yang tidak memiliki nilai balikan. Fungsi utama ini berfungsi seperti gerbang untuk masuk kedalam program.

Setelah itu, tuliskan kurung kurawal. Nah, didalam kurung kurawal ini lah baris fungsi yang ingin kita gunakan dituliskan. Setiap selesai menuliskan satu baris fungsi, wajib dituliskan tanda titik koma (;) sebagai penutup. Apabila tidak maka yang terjadi program akan error. Contoh pada program yang barusan kita jalankan disana terdapat fungsi **printf()**, maka setelah tanda tutup kurung kita harus menuliskan titik koma. Pada fungsi diatas terdapat “\n” yang berfungsi untuk enter satu kali. Apabila tidak ada maka program akan ditampilkan kesamping.

Agar program yang dibuat dapat dijalankan pada computer maka kita harus meng-compile program yang baru saja dibuat. Untuk meng-compile pertama-tama kita buka cmd, kemudian arahkan alamat penyimpanan sourcecode (program) menggunakan fungsi **CD** (change directory). Untuk melihat isi folder yang sedang aktif bisa menggunakan fungsi **DIR**, kemudian untuk Kembali satu tingkat path sebelumnya bisa mengetikan **CD..**.(contoh sekarang berada di D:\Program, untuk ke D:\ kita ketikkan **CD..**).



## Modul 3

### Program Aritmatika Sederhana

#### A. Langkah Kerja

1. Pertama buka *Notepad*
2. Kemudian ketikkan *SourceCode* berikut

```
#include <stdio.h>

void main()
{
int A,B,T;
A=5;
B=2;
T=A+B;
printf("%i",T);
}
```

3. Simpan *sourcecode* diatas dengan *format .c*
4. Buka *cmd* kemudian arahkan direktori penyimpanan ketempat menyimpan *sourcecode* tadi

```
Microsoft Windows [Version 10.0.19041.288]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd
D:\>cd \kuliah
D:\>gcc -o tes tes.c
D:\>tes
7
D:\>
```

→ Mengarahkan ke direktori penyimpanan

→ Proses Compile

→ hasil



## B. Pembahasan

Program yang baru saja dibuat adalah program perhitungan aritmatika sederhana. Didalam program digunakan 3 variable yaitu variable **A**, **B**, dan **T**. Variable sendiri adalah sebuah wadah untuk menampung nilai dimana nilai dalam sebuah variable dapat diubah selama program berjalan. Penulisan variable bebas selain dari fungsi fungsi yang terdapat dalam Bahasa C. apabila variable terdiri dari 2 atau lebih suku kata maka penamaan variable dijadikan satu (IniAdalahVariable) atau di beri tanda garis bawah (Ini\_Adalah\_Variable). Untuk menggunakan variable dalam Bahasa C, variable harus dideklarasikan terlebih dahulu. Pendeklarasian variable harus disertai dengan tipe data yang digunakan. Karena dalam program diatas program melakukan perhitungan maka tipe data yang digunakan adalah tipe data **Integer**. Tipe data integer dalam Bahasa C di dekarasikan sebagai **int**.

Dalam Bahasa C, nilai dari sebuah variable bisa langsung dideklarasikan **int A =5;** atau seperti program diatas, nilai dari variable **A** baru diinisialisasi atau diisikan dibawahnya. Apabila sesudah operasi **T = A+B;** selesai dijalankan kemudian kita menginisialisasi nilai **A** yang baru menjadi **7** misalnya, hal ini sah-sah saja, inilah yang dimaksud nilai suatu variable dapat diubah-ubah. Untuk contoh nilai variable dapat diubah-ubah akan kita temui di modul-modul selanjutnya.

Kemudian, untuk perintah output dari Bahasa C pada modul ke-2 kita hanya menampilkan teks saja. Lain halnya dengan modul ke-3, kita menampilkan variable dimana variable tersebut memiliki tipe data. Untuk menampilkannya ke layar ada sedikit perbedaan dimana kita harus menuliskan tipe data dari variable yang ingin ditampilkan kemudian baru variabelnya. Contoh **printf(“%i”, T);** variable **T** akan tampil sebagai integer (**%i** menandakan **integer**). Kemudian kita ingin menampilkan tulisan sebelum dan sesudah variable maka akan jadi seperti ini **printf(“hasil perhitungannya adalah %i dengan penjumlahan %i dan %i”, T, A, B);**. Variable **T** akan ditampilkan di **%i** sesudah kata ‘adalah’, Variable **A** akan ditampilkan di **%i** sesudah kata ‘penjumlahan’ dan variable **B** akan di tampilkan pada **%i** sesudah kata ‘dan’.



Operasi aritmatika yang dilakukan pada program diatas adalah operasi penjumlahan. Untuk melakukan penjumlahan dalam Bahasa C digunakan tanda tambah (+), operasi pengurangan menggunakan tanda kurang (-), operasi perkalian menggunakan tanda bintang (\*), operasi pembagian menggunakan tanda garis miring(/), serta operasi sisa pembagian menggunakan tanda persen(%).

Tipe data adalah jenis data. Menurut Wikipedia: “Tipe data atau kadang disingkat dengan ‘tipe’ saja adalah sebuah pengelompokan data untuk memberitahu *compiler* atau *interpreter* bagaimana programmer ingin mengolah data tersebut”. Ada delapan jenis tipe data pada C:

1. Char
2. Integer
3. Float
4. Array
5. Structure
6. Pointer
7. Enum
8. Void

Dari delapan tipe data tersebut kita hanya akan mempelajari 4 tipe data sebagai berikut:

1. **Char**: adalah tipe data yang berisi 1 huruf atau 1 karakter;
2. **Integer**: adalah tipe data yang berupa angka;
3. **Float**: adalah tipe data yang berupa bilangan pecahan
4. **Double**: adalah tipe data sama seperti **float**, namun double memiliki ukuran penyimpanan yang lebih besar dibandingkan **float**.

Setiap tipe data memiliki panjang dan ukuran, perhatikan tabel berikut



Tipe Data	Ukuran	Panjang	Contoh
char	1 byte	-128 — 127 atau 0 — 255	'A', 'a', '3'
int	2 byte	-32,768 — 32,767 atau -2,147,483,648 — 2,147,483,647	32, 1, 4
float	4 byte	1.2E-38 — 3.4E+38	4.3, 2.2, 6.0
double	8 byte	2.3E-308 — 1.7E+308	4.2, 4.22, 3.2

Format penulisan koma untuk tipe data **float** dan **double** menggunakan tanda titik ”.”, mengikuti format internasional. Lalu untuk tipe data char kita harus menuliskannya dalam tanda petik, karena spasi termasuk dalam karakter

Setelah kita mengetahui tentang tipe data, selanjutnya kita harus mengetahui operator-operator dalam Bahasa C.

### 1. Operator Aritmatika

Operator aritmatika adalah operator yang biasa kita temukan untuk operasi matematika. Aritmatika sendiri merupakan cabang ilmu matematika yang membahas perhitungan sederhana, seperti kali, bagi, tambah dan kurang (kabataku)

Tabel berikut merangkum operator aritmatika dalam bahasa pemrograman C:

Operator	Penjelasan	Contoh
+	Penambahan	a = 5 + 2
-	Pengurangan	a = 5 - 2
*	Perkalian	a = 5 * 2
/	Pembagian (real/pecahan)	a = 5 / 2
%	Sisa hasil bagi (modulus)	a = 5 % 2



## 2. Operator Increment dan Decrement

Operator **increment** dan **decrement** adalah sebutan untuk operasi seperti `a++`, dan `a--`. Ini sebenarnya penulisan singkat dari operasi `a = a + 1` serta `a = a - 1`. **Increment** digunakan untuk menambah nilai variabel sebanyak 1 angka, sedangkan **decrement** digunakan untuk mengurangi nilai variabel sebanyak 1 angka.

Penulisannya menggunakan tanda tambah 2 kali untuk **increment**, dan tanda kurang 2 kali untuk **decrement**. Penempatan tanda tambah atau kurang ini boleh di awal seperti `++a` dan `--a`, atau di akhir variabel seperti `a++` dan `a--`.

Dengan demikian terdapat 4 jenis **increment** dan **decrement** dalam bahasa C:

Operator	Contoh	Penjelasan
Pre-increment	<code>++a</code>	Tambah a sebanyak 1 angka, lalu tampilkan hasilnya
Post-increment	<code>a++</code>	Tampilkan nilai a, lalu tambah a sebanyak 1 angka
Pre-decrement	<code>--a</code>	Kurangi a sebanyak 1 angka, lalu tampilkan hasilnya
Post-decrement	<code>a--</code>	Tampilkan nilai a, lalu kurangi a sebanyak 1 angka

## 3. Operator Perbandingan / Relasional

Operator perbandingan dipakai untuk membandingkan 2 buah nilai, apakah nilai tersebut sama besar, lebih kecil, lebih besar, dll. Hasil dari operator perbandingan ini adalah boolean **True** atau **False**, namun karena bahasa C tidak memiliki tipe data boolean bawaan, maka hasilnya adalah integer 1 atau 0.



Operator	Penjelasan	Contoh	Hasil
==	Sama dengan	5 == 5	1 (true)
!=	Tidak sama dengan	5 != 5	0 (false)
>	Lebih besar	5 > 6	0 (false)
<	Lebih kecil	5 < 6	1 (true)
>=	Lebih besar atau sama dengan	5 >= 3	1 (true)
<=	Lebih kecil atau sama dengan	5 <= 5	1 (true)

#### 4. Operator Logika / Boolean

Operator logika dipakai untuk menghasilkan nilai boolean true atau false dari 2 kondisi atau lebih. Operator ini akan dipakai pada fungsi logika if else.

Operator	Nama	Penjelasan	Contoh
&&	And	Akan menghasilkan 1 jika kedua operand 1	1 && 0, hasilnya: 0
	Or	Akan menghasilkan 1 jika salah satu operand 1	1    0, hasilnya: 1
!	Not	Akan menghasilkan 1 jika operand 0	!0, hasilnya: 1

#### 5. Operator Bitwise

**Bitwise** adalah operator khusus untuk menangani operasi logika bilangan biner dalam bentuk bit. **Bilangan biner** sendiri merupakan jenis bilangan yang hanya terdiri dari 2 jenis angka, yakni 0 dan 1. Jika nilai asal yang dipakai bukan bilangan biner, akan dikonversi secara otomatis oleh compiler C menjadi bilangan biner. Misalnya 7 desimal = 0111 dalam bilangan biner.

Bahasa C mendukung 6 jenis operator bitwise. Daftar lengkapnya dapat dilihat pada tabel berikut:



Operator	Nama	Contoh	Biner	Hasil (biner)	Hasil (desimal)
&	AND	10 & 12	1010 & 1100	1000	8
	OR	10   12	1010   1100	1110	14
^	XOR	10 ^ 1	1010 ^ 1100	0110	6
~	NOT	~ 5	~0101	1011	-11 (Two's complement)
<<	Left shift	10 << 1	1010 << 1	10100	20
>>	Right shift	10 >> 1	1010 >> 1	101	5

## 6. Operator Assignment

Operator assignment adalah operator yang digunakan untuk memberikan nilai ke dalam suatu variabel. Di dalam bahasa C, operator assignment menggunakan tanda sama dengan " = ". Nantinya juga terdapat operator assignment gabungan, seperti " += ", " -= ", dsb. <sup>1</sup>

<sup>1</sup> <https://www.duniaikom.com/tutorial-belajar-c-jenis-jenis-operator-dalam-bahasa-c/> diakses pada tanggal 12 April 2021



## Modul 4

### Membuat Program Menghitung Luas Segitiga dengan Nilai Alas dan Tinggi Merupakan Inputan dari Pengguna

#### A. Langkah Kerja

1. Buka *notepad*
2. Salin *SourceCode* berikut

```
/*
 * Program 2 Untuk Menghitung Luas Segitiga
 * Nama : "buat nama Anda Disini"
 * Nim : "Masukkan Nim Anda Disini"
 * Baris ini adalah komentar tidak akan dieksekusi
 */
/*Penulisan komentar juga dapat dibuat di antara tanda awal
dan
akhir seperti contoh ini */
//atau dapat juga menggunakan tanda slash seperti disamping
#include <stdio.h> //Mengakses file library stdio.h
#include <stdlib.h>
#define dua 2 //Deklarasi konstanta
int main(void) //Fungsi yang harus ada pada setiap program C
{
    float a, t, LuasSegitiga; /*Deklarasi variabel. Ini
berarti
a, t dan LuasSegitiga merupakan
variabel bertipe float*/
printf("Perhitungan Luas Segitiga \n"); //Perintah
output
printf("Alas = ");
scanf("%f" &a); /*Perintah input berupa variabel
bertipe
float*/
printf("Tinggi = ");
scanf("%f" &t);
LuasSegitiga = a*t/dua; /*Perintah berupa penugasan
perhitungan perkalian*/
printf("Luas Segitiga = %f \n", LuasSegitiga);
system("pause"); /*Salah satu fungsi yang ada pada
library
stdlib.h*/
```

3. Jalankan program tersebut dan pahami fungsinya



4. Hasilnya akan seperti ini

```
D:\>tes
Perhitungan Luas Segitiga → Judul Program (output dari fungsi printf)
Alas = 5 → Input dari user(menggunakan fungsi scanf)
Tinggi = 6 → Input dari user(menggunakan fungsi scanf)
Luas Segitiga = 15.000000 → Hasil operasi aritmatika
Press any key to continue . . . → Output fungsi dari "system ('cls')"
```

## B. Pembahasan

Program yang baru saja dibuat menggunakan nilai inputan dari pengguna. Program yang baru saja dibuat memiliki prinsip kerja yang hampir sama dengan kalkulator dimana kita tinggal memasukkan angka yang ingin dioperasikan dan kita mendapatkan hasil.

Fungsi yang digunakan adalah fungsi scanf dimana fungsi scanf terdapat pada header `#include <stdio.h>`. Pada fungsi scanf kita harus menyediakan satu variable sebagai wadah untuk menampung nilai inputan yang akan diterima. Kemudian struktur penulisan fungsi scanf mirip dengan fungsi printf dimana kita menuliskan tipe data inputan dari pengguna. Pada variable yang digunakan untuk menampung nilai inputan ditambahkan symbol '&'.

Tipe data yang digunakan dalam program adalah tipe data float, dimana tipe data float merupakan tipe data bilangan berkoma. Selain menggunakan tipe data float, kita bisa juga menggunakan tipe data double. Perbedaan keduanya terdapat pada Panjang koma yang dapat ditampung. Tipe data float dapat menampung banyak angka di belakang koma dibandingkan dengan tipe data double.



## Latihan 1

Buatlah program untuk hasil seperti gambar berikut.

```
D:\>soal
Program menghitung volume balok
Masukkan panjang balok : 4
Masukkan lebar balok : 3
Masukkan tinggi balok : 6
Volume balok adalah 72
D:\>
```



## Modul 5

### Increment dan Decrement

#### A. Langkah Kerja

1. Buka *notepad*
2. Salinlah *SourceCode* program berikut ini

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a, b;
    a = 5;
    b = 7;
    //post increment
    printf("%i \n",a);
    printf("%i \n",a++);
    printf("%i \n",a++);
    system("pause");
    //pre increment
    printf("%i \n",a);
    printf("%i \n",++a);
    printf("%i \n",++a);
    system("pause");
    //post decrement
    printf("%i \n",b);
    printf("%i \n",b--);
    printf("%i \n",b--);
    system("pause");
    //pre decrement
    printf("%i \n",b);
    printf("%i \n",--b);
    printf("%i \n",--b);
    system("pause");
    return (0);
}
```

3. Jalankan program tersebut dan pahami fungsinya



## B. Pembahasan

Increment dan decrement merupakan operator dalam pemrograman untuk mengubah nilai dari suatu variable. Berbeda dengan konstanta yang nilainya tidak diubah-ubah. Untuk penjelasan operator increment dan decrement dapat dilihat pada modul 3 Program Aritmatika Sederhana. Berikut ini hasil compile dari program yang telah kita buat:

```
D:\>gcc -o tes 12.c
D:\>tes
5
5
6
Press any key to continue . . .
7
8
9
Press any key to continue . . .
7
7
6
Press any key to continue . . .
5
4
3
Press any key to continue . . .
```





## B. Pembahasan

Fungsi if else merupakan salah satu fungsi seleksi kondisi yang akan kita gunakan pada Bahasa C. Jika kita ingin menyeleksi satu kondisi saja maka kita hanya menggunakan if saja Atanpa menggunakan else. Kemudian, jika kita ingin menyeleksi dua kondisi maka kita menggunakan fungsi if untuk kondisi pertama dan else untuk kondisi lainnya. Dan jika kita punya lebih dari dua kondisi yang ingin diseleksi maka kita menambahkan elif atau else if antara fungsi if dan fungsi else. Jika ada tiga kondisi maka fungsi if diikuti fungsi else if dan terakhir fungsi else. Jika ada empat kondisi maka fungsi if diikuti fungsi if else sebanyak 2 kali kemudian terakhir fungsi else. Pada fungsi if else menggunakan operator Perbandingan/relasional yang bisa dilihat pada modul 3. Berikut ini hasil *compile* dari program yang telah kita buat:

```
D:\>tes
Program Seleksi Kecukupan Nilai
=====
Masukkan Nilai Anda : 70
Silahkan datang Dosen Anda untuk perbaikan Nilai

Press any key to continue . . .

D:\>tes
Program Seleksi Kecukupan Nilai
=====
Masukkan Nilai Anda : 60
Maaf Anda gagal

Press any key to continue . . .

D:\>tes
Program Seleksi Kecukupan Nilai
=====
Masukkan Nilai Anda : 80
Anda Lulus

Press any key to continue . . .
```



## Modul 7

### Seleksi Kondisi 2

#### A. Langkah Kerja

1. Buka *notepad* atau teks editor manapun yang Anda sukai
2. Salin *SourceCode* dibawah ini

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a, b;
    printf("Program Seleksi Kecukupan Nilai\n");
    printf("=====\n");
    printf("Masukkan Nilai PMD Anda : ");
    scanf("%i", &a);
    printf("Masukkan Nilai ELD Anda : ");
    scanf("%i", &b);
    if((a>70)&&(b>70)){
        printf("Anda Lulus\n");
    }
    else if (((a>70)&&(b<70))||((a<70)&&(b>70))){
        printf("Silahkan datang Dosen Anda untuk perbaikan
        Nilai\n");
    }
    else{
        printf("Maaf Anda gagal\n");
    }
    printf("\n");
    system("pause");
    return (0);
}
```

3. Jalankan program tersebut dan pahami fungsinya



## B. Pembahasan

Seleksi kondisi yang baru saja dibuat sama dengan modul 6, perbedaannya terletak pada operator Boolean yang ditambahkan. Penjelasan mengenai operator Boolean dapat dilihat pada modul 3. Berikut ini hasil *compile* dari program yang telah kita buat:

```
D:\>tes
Program Seleksi Kecukupan Nilai
=====
Masukkan Nilai PMD Anda : 70
Masukkan Nilai ELD Anda : 80
Maaf Anda gagal

Press any key to continue . . .

D:\>tes
Program Seleksi Kecukupan Nilai
=====
Masukkan Nilai PMD Anda : 80
Masukkan Nilai ELD Anda : 80
Anda Lulus

Press any key to continue . . .
```



## Modul 8

### Seleksi Kondisi 3

#### A. Langkah Kerja

1. Buka *notepad* atau teks editor manapun yang anda suka
2. Salin *Source Code* dibawah

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int pilih, a, b, hasil;
    printf("Program Menu Kalkulator\n");
    printf("=====\n");
    printf("1. Menu Penjumlahan \n");
    printf("2. Menu Perkalian \n");
    printf("3. Keluar \n");
    printf("Masukkan pilihan Anda berupa angka :");
    scanf("%i", &pilih);

    switch(pilih){
        case 1:{
            printf("Menu Penjumlahan \n");
            printf("=====\n");
            printf("Masukkan angka pertama:");
            scanf("%i", &a);
            printf("Masukkan angka kedua :");
            scanf("%i", &b);
            hasil = a + b;
            printf("Hasil Penjumlahan = %i", hasil);
            break;
        }
        case 2:{
            printf("Menu Perkalian \n");
            printf("=====\n");
            printf("Masukkan angka pertama:");
            scanf("%i", &a);
            printf("Masukkan angka kedua :");
            scanf("%i", &b);
            hasil = a * b;
            printf("Hasil Perkalian = %i", hasil);
            break;
        }
        case 3:{
            exit(0);
        }
        default:
            printf("Maaf tidak ada menu untuk pilihan Anda\n");
            exit(0);
    }
    printf("\n");
    system("pause");
}
```

3. Jalankan program tersebut dan pahami setiap fungsinya



## B. Pembahasan

Seleksi kondisi yang baru dibuat adalah seleksi kondisi switch case. Seleksi kondisi switch case tidak seperti seleksi kondisi if else. Pada seleksi kondisi switch case kondisi yang dibandingkan itu hanya satu saja. Satu case untuk satu kondisi. Maksudnya, pada seleksi kondisi if else kita bisa membuat kondisi kurang dari 50 ( $a < 50$ ). Namun pada switch case tidak bisa. Kondisinya hanya satu jika 50 apa yang terjadi, tidak bisa kurang atau lebih. Jika kondisi selain 50 dan kondisi tersebut tidak terdapat dalam kondisi manapun di case kita maka digunakan fungsi default. Berikut ini hasil compile dari program yang telah kita buat:

```
D:\>tes
Program Menu Kalkulator
=====
1. Menu Penjumlahan
2. Menu Perkalian
3. Keluar
Masukkan pilihan Anda berupa angka :1
Menu Penjumlahan
=====
Masukkan angka pertama:2
Masukkan angka kedua :3
Hasil Penjumlahan = 5
Press any key to continue . . .

D:\>tes
Program Menu Kalkulator
=====
1. Menu Penjumlahan
2. Menu Perkalian
3. Keluar
Masukkan pilihan Anda berupa angka :2
Menu Perkalian
=====
Masukkan angka pertama:3
Masukkan angka kedua :4
Hasil Perkalian = 12
Press any key to continue . . .
```



## Modul 9

### Perulangan

#### A. Langkah Kerja

1. Buka *notepad* atau teks *editor* manapun yang Anda sukai
2. Salin *Sourcecode* berikut

```
#include <stdio.h>
int main(){
    for (int i = 0; i < 5; i++){
        printf("%i. Hallo Politeknik
Kampar\n", (i+1));
    }
    printf("\n");

    int a=0;
    while(a<5){
        printf("%i. Hallo Mahasiswa TIF Politeknik
Kampar\n", (a+1) );
        a++;
    }
    printf("\n");

    int b=0;
    do{
        printf("%i. Selamat Datang Mahasiswa
Politeknik Kampar\n", (b+1));
        b++;
    }
    while(b<5);
    return 0;
}
```

3. Jalankan kemudian pahami fungsi-fungsinya.



## B. Pembahasan

Perulangan (Looping) merupakan fungsi untuk melakukan suatu perintah secara berulang ulang. Contoh kita ingin menampilkan angka dari 1-5, dari pelajaran sebelumnya kita bisa saja menggunakan fungsi printf sebanyak 5 kali atau menggunakan fungsi printf sebanyak 1 kali namun menuliskan 1-5. Kalau hanya 5 hal ini sah-sah saja, lalu bagaimana jika yang ingin ditampilkan 1-100? Akan terasa tidak efisien hal ini dilakukan. Oleh karena itu kita menggunakan fungsi perulangan/looping. Fungsi perulangan sendiri terdiri dari 3, yaitu:

1. Fungsi perulangan for.

Struktur Perulangan For

```
for(inisiasi; terminasi; iterasi){  
    Proses;  
}
```

2. Fungsi perulangan while.

Struktur Perulangan For

```
inisiasi;  
While(terminasi){  
    Proses;  
    Iterasi;  
}
```

3. Fungsi perulangan do-while.

Struktur Perulangan For

```
inisiasi;  
{  
    Proses;  
    Iterasi;  
} While(terminasi;)
```



Perbedaan penggunaan ketiga perulangan tersebut terdapat pada struktur fungsi. Kemudian perbedaan lainnya pada perulangan for, kita tau berapa banyak perulangan yang kita inginkan, namun pada perulangan while dan do-while tidak. Berikut ini hasil *compile* dari program yang telah kita buat:

```
D:\>gcc -o tes 12.c
D:\>tes
1. Hallo Politeknik Kampar
2. Hallo Politeknik Kampar
3. Hallo Politeknik Kampar
4. Hallo Politeknik Kampar
5. Hallo Politeknik Kampar

1. Hallo Mahasiswa TIF Politeknik Kampar
2. Hallo Mahasiswa TIF Politeknik Kampar
3. Hallo Mahasiswa TIF Politeknik Kampar
4. Hallo Mahasiswa TIF Politeknik Kampar
5. Hallo Mahasiswa TIF Politeknik Kampar

1. Selamat Datang Mahasiswa Politeknik Kampar
2. Selamat Datang Mahasiswa Politeknik Kampar
3. Selamat Datang Mahasiswa Politeknik Kampar
4. Selamat Datang Mahasiswa Politeknik Kampar
5. Selamat Datang Mahasiswa Politeknik Kampar
D:\>_
```



## Modul 10

### Perulangan Bersarang

#### A. Langkah Kerja

1. Buka *notepad* atau teks editor manapun yang Anda sukai
2. Salin *SourceCode* dibawah ini

```
#include <stdio.h>
int main(){
    for (int i = 0; i < 5; i++){
        for (int j = 0; j < 5; j++){
            printf("%i,%i; ",i,j);
        }
        printf("\n");
    }
    printf("\n");

    int a=0;
    while(a<5){
        int b=0;
        while(b<5){
            printf("%i,%i; ", a, b);
            b++;
        }
        printf("\n");
        a++;
    }
    printf("\n");

    int b=0;
    do{
        int c=0;
        do{
            printf("%i,%i; ", b, c);
            c++;
        }while(c<5);
        printf("\n");
        b++;
    }
    while(b<5);
    return 0;
}
```

3. Jalankan program tersebut, kemudian pahami fungsi-fungsinya.





## Modul 11

### Membuat Pola Bintang

#### A. Langkah Kerja

1. Buka *notepad* atau teks editor manapun yang Anda sukai
2. Salin *SourceCode* Berikut

```
#include <stdio.h>
int main(){
    for (int i = 0; i < 5; i++){
        for (int j = 0; j < i; j++){
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

3. Jalankan program tersebut kemudian pahami alur kerjanya.

#### B. Pembahasan

Berikut ini adalah hasil program diatas

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











## Modul 12

### Larik 1

#### A. Langkah Kerja

1. Buka notepad atau teks editor manapun yang Anda sukai
2. Salin Sourcecode Berikut

```
#include <stdio.h>
int main(){
    int data[3]={1,2,3};

    printf("%i \n", data[0]);
    printf("%i \n", data[1]);
    printf("%i \n", data[2]);

    for (int i = 0; i < 3; i++){
        printf("%i \n", data[i]);
    }
    return 0;
}
```

3. Jalankan program tersebut kemudian fahami fungsinya.

#### B. Pembahasan

Larik atau array adalah sebuah metode untuk menampung beberapa nilai ke dalam satu variable. Sederhananya kita ibaratkan variable adalah sebuah rumah. Dan nilai dari variable adalah kita sebagai penghuni. Maka untuk menempatkan 1 orang dibutuhkan sebuah rumah, kalau 1 sampai lima orang mungkin tidak masalah tapi jika ada 100 orang, berapa banyak lahan yang digunakan? Begitu juga sebuah variable dalam memori, jika 1 sampai 5 variable saja mungkin tidak masalah tapi jika ada 100 variabel, berapa banyak ruang yang dibutuhkan.

Kembali lagi ke permasalahan rumah tadi, kalau satu rumah satu orang maka akan banyak ruang yang tidak terpakai ini akan menjadi tidak efektif. Maka itu digunakanlah larik/array. Larik ini sendiri seperti kamar dalam rumah. Jadi, satu rumah disekat(kamar) dan dapat diisikan banyak orang



dalam satu rumah. Jadi sedikit ruang kosong yang tidak terpakai. Sama seperti memori atau penyimpanan dalam Komputer.

Yang baru saja kita buat merupakan array 1 dimensi. Pengalamatan (indeks) pada array dimulai dari-0. Contoh pada program diatas, data[0], berarti array variable data indeks ke-0. Indeks ke-0 sendiri adalah 1. Jadi data[0] pada array yang dibuat adalah 0. Berikut ini hasil compile dari program yang telah kita buat:

```
D:\>gcc -o tes 12.c
D:\>tes
1
2
3
1
2
3
D:\>_
```





## B. Pembahasan

Program diatas merupakan larik 2 dimensi, dimana larik 2 dimensi sendiri terdiri dari baris dan kolom. Apabila di deklarasikan larik[2][3]. Maka larik tersebut memilik 2 baris dan 3 kolom. Berikut ini hasil *compile* dari program yang telah kita buat:

```
D:\>gcc -o tes 13.c
D:\>tes
1, 2, 3
4, 5, 6

1, 2, 3,
4, 5, 6,
D:\>
```



## DAFTAR PUSTAKA

Andre. 2018. *Tutorial Belajar C Part 26: Jenis-jenis Operator dalam Bahasa C.* <https://www.duniailkom.com/tutorial-belajar-c-jenis-jenis-operator-dalam-bahasa-c/> (diakses tanggal 12 April 2021)



ISBN 978-623-96017-9-9 (PDF)

