

# Dasar Pemrograman Komputer (rev 1)

## **Tim Penyusun :**

Dr. Eko Mulyanto Yuniarno  
Arief Kurniawan, S.T., M.T.  
Dr. Diah Puspito Wulandari, S.T.,M.Sc.  
Eko Pramuntanto S.T.,M.T.  
Ahmad Zaini, S.T., M.Sc.  
Dion Hayu Fandiantoro, S.T., M.Eng.

28 Maret 2022

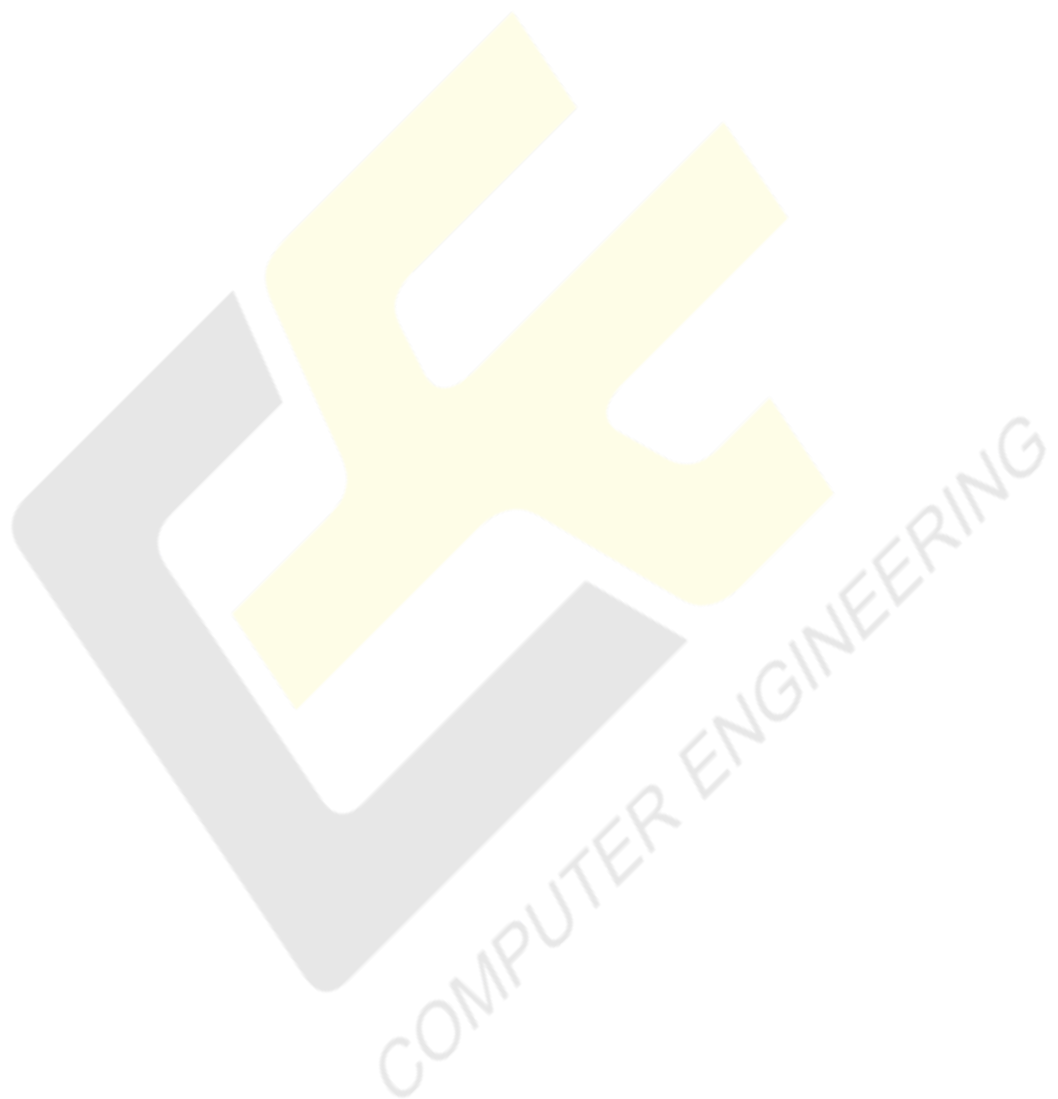


# Daftar Isi

<b>0</b>	<b>Cara Menggunakan Buku Ini</b>	<b>1</b>
0.1	Perangkat Lunak	1
0.2	OnlineGDB.com	1
0.2.1	Mencoba Contoh Program	1
0.3	CodeBlock	4
0.3.1	Code::Blocks	4
0.3.1.1	Instalasi Code::Blocks	4
<b>1</b>	<b>Sistem Komputer</b>	<b>7</b>
1.1	Perangkat Keras Komputer	7
1.1.1	Unit Fungsional Komputer	7
1.1.2	Unit Input	8
1.1.3	Unit Output	9
1.1.4	Unit Memori	10
1.1.5	Central Processing Unit	11
1.1.5.1	Arithmetic Logic Unit (ALU)	11
1.1.5.2	Unit Kontrol	11
1.2	Operasi Komputer	11
1.2.1	Program Komputer	11
1.2.2	Penterjemahan Bahasa Ke Bahasa Mesin	12
1.3	Perangkat Lunak Komputer	13
1.3.1	Perangkat Lunak Sistem	13
1.3.2	Perangkat Lunak Pemrograman	13
1.3.3	Aplikasi	14
<b>2</b>	<b>Variabel dan Ekspresi</b>	<b>15</b>
2.1	Variabel	15
2.2	Ekspresi	16
2.2.1	Ekspresi Integral dan Floating point	16
2.2.2	Ekspresi Relational	16
2.2.3	Ekspresi Logika	16
2.3	Contoh Soal	17
<b>3</b>	<b>Algoritma</b>	<b>19</b>
3.1	Pengertian Algoritma	19
3.2	Contoh Menyelesaikan masalah menggunakan algoritma	20
3.3	Penyajian Algoritma	21
3.3.1	Penyajian Algoritma secara Deskriptif	22
3.3.2	Penyajian Algoritma Menggunakan Pseudocode	22
3.3.2.1	Struktur Penulisan Algoritma Menggunakan Pseudocode	23
3.3.2.2	Perbandingan Algoritma dengan Pseudocode dengan bahasa pemrograman lain	24
3.3.3	Flowchart	25

3.3.3.1	Simbol Dasar Flowchart . . . . .	25
3.3.3.2	Implementasi Flowchart Untuk Menghitung Luas Segitiga . . . . .	26
3.3.3.3	Perbandingan FLOWchart, Pseudocode dan Bahasa pemrograman C untuk menghitung Luas segitiga . . . . .	27
<b>4</b>	<b>Aliran Kontrol dalam Bahasa Pemrograman</b>	<b>29</b>
4.1	Struktur Runtunan \ sekuensial . . . . .	29
4.2	Kontrol aliran pemilihan atau selection . . . . .	30
4.2.1	Kontrol aliran alternatif tunggal . . . . .	30
4.2.2	Alternatif Ganda . . . . .	33
4.3	Pengulangan(Repetition) . . . . .	35
4.4	Contoh Soal . . . . .	37
4.5	Latihan Soal . . . . .	39
<b>5</b>	<b>Pemrograman C</b>	<b>41</b>
5.1	Struktur Bahasa C . . . . .	41
5.2	Tipe Data Dalam C . . . . .	41
5.2.1	Tipe Data Integer . . . . .	42
5.2.2	Floating point atau bilangan pecahan . . . . .	42
5.3	Variabel Dalam C . . . . .	43
5.3.1	Structure . . . . .	43
5.4	Operator . . . . .	45
5.4.1	Operator Aritmetika . . . . .	45
5.4.2	Operator Penugasan . . . . .	46
5.4.3	Operator Perbandingan . . . . .	47
5.4.4	Operator Logika . . . . .	47
<b>6</b>	<b>Fungsi Output dan Input</b>	<b>51</b>
6.1	Format specifier . . . . .	51
6.1.0.1	title . . . . .	51
6.2	Fungsi printf() . . . . .	51
6.3	scanf . . . . .	53
6.4	Escape Sequence . . . . .	55
6.5	Tugas . . . . .	56
<b>7</b>	<b>Instruksi Pemilihan Dalam C</b>	<b>57</b>
7.1	Tujuan . . . . .	57
7.2	Statement if . . . . .	57
7.3	Statement else . . . . .	58
7.4	Statement else if . . . . .	58
7.5	Contoh Soal . . . . .	59
7.6	Latihan Soal . . . . .	61
<b>8</b>	<b>Operasi Pengulangan</b>	<b>63</b>
8.1	Loop atau pengulangan . . . . .	63
8.1.1	While Loop . . . . .	63
8.1.2	Do..while Loop . . . . .	64
8.1.3	For Loop . . . . .	65
8.2	break . . . . .	66
8.3	Loop Bersarang . . . . .	67
8.4	Nested Loop . . . . .	67
8.4.1	for loop bersarang . . . . .	67
8.5	Loop bersarang while loop . . . . .	68
8.6	Contoh Soal . . . . .	68

8.7	Latihan Soal . . . . .	69
8.8	Latihan . . . . .	69
<b>9</b>	<b>Array</b>	<b>71</b>
9.1	Konsep Dasar . . . . .	71
9.2	Variabel Array . . . . .	72
9.3	Array Satu Dimensi . . . . .	72
9.3.1	Deklarasi Array Satu Dimensi . . . . .	72
9.3.2	Inisialisasi Array Satu Dimensi . . . . .	73
9.3.3	Mengakses Elemen Array satu dimensi . . . . .	74
9.3.4	Contoh Soal . . . . .	74
9.3.5	Latihan Soal . . . . .	76
9.4	Array Multidimensi . . . . .	76
9.5	Array Dua Dimensi . . . . .	76
<b>10</b>	<b>Aplikasi Array</b>	<b>79</b>
10.1	Buble Sort . . . . .	79
10.2	Operasi Matriks . . . . .	79
10.2.1	Perkalian Dua Buah Matriks . . . . .	79
<b>11</b>	<b>String</b>	<b>81</b>
11.1	Deklarasi String . . . . .	81
11.2	Input Output String . . . . .	81
11.3	Operasi String . . . . .	82
11.3.1	fungsi strlen . . . . .	82
11.3.2	fungsi strcpy . . . . .	82
11.4	fungsi strcat . . . . .	83
11.5	fungsi strcmp . . . . .	83
11.6	Contoh Soal . . . . .	83



## Bab 0

# Cara Menggunakan Buku Ini

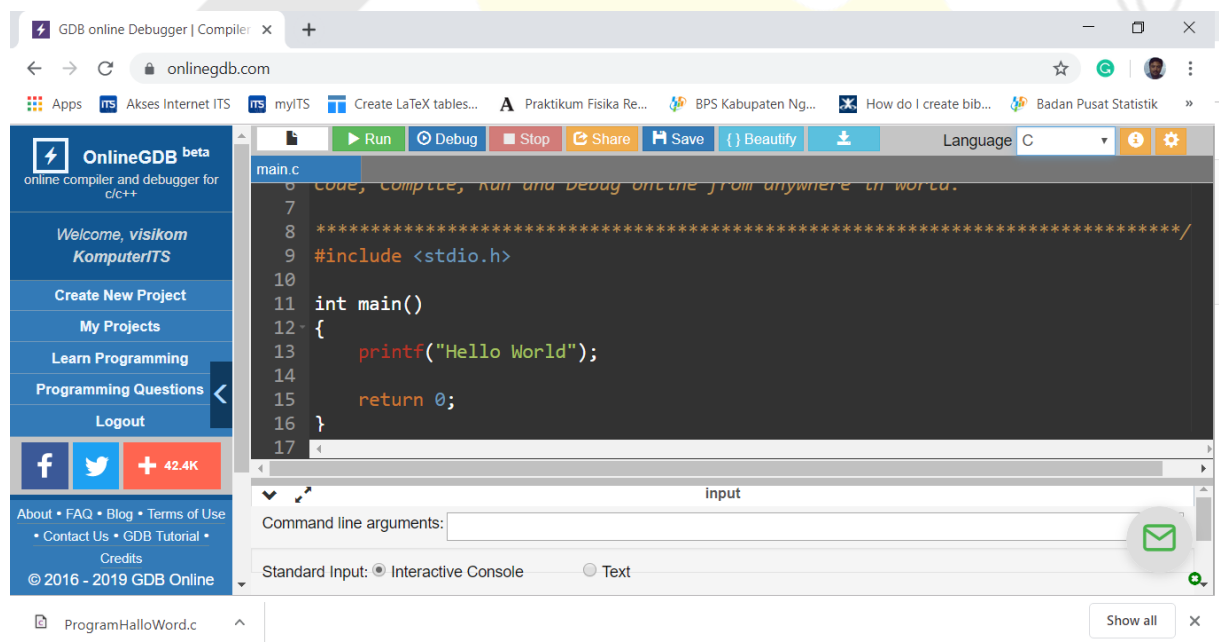
### 0.1 Perangkat Lunak

Contoh-contoh dalam buku ini dapat di jalankan dalam dua macam compiler

1. OnlineGDB
2. Code::Blocks

### 0.2 OnlineGDB.com

OnlineGDB.com adalah compiler dan debugger online untuk bahasa C / C ++.



Gambar 1

#### 0.2.1 Mencoba Contoh Program

Apabila pada suatu subbab terdapat contoh program seperti pada 1 maka akan disediakan cara contoh yang dapat dijalankan dengan cepat.

Contoh Kode program.

1. Kode Program mencetak "Hello world"

Listing 1: Contoh program menggunakan C

```

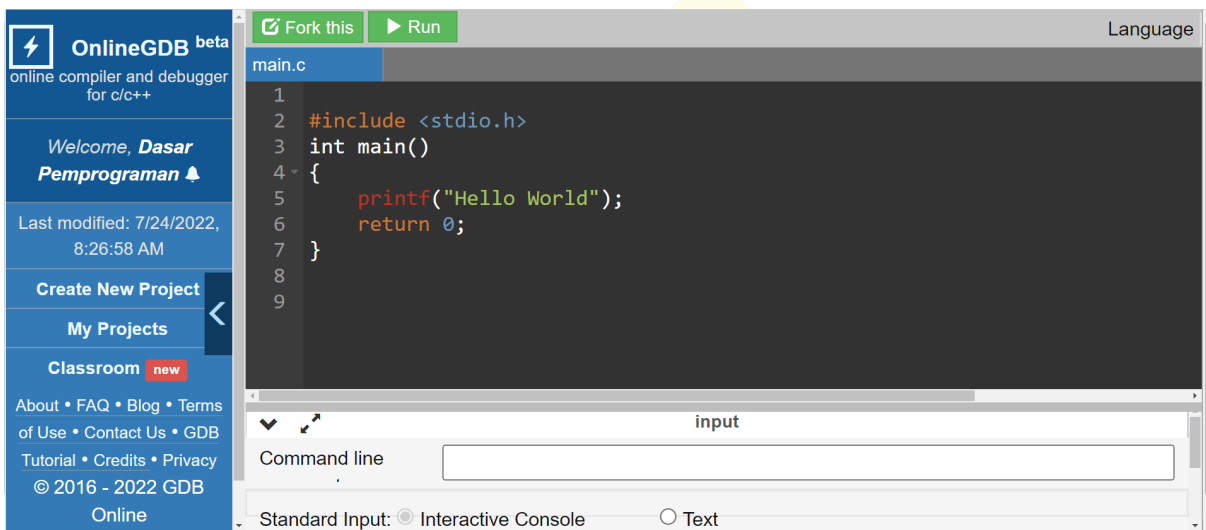
1  #include <stdio.h>
2  int main()
3  {
4      printf("Hello World");
5      return 0;
6  }
7

```

2. Setelah itu dibawahnya akan muncul perintah untuk mencoba sbb :

⇒ Untuk mencoba program Klik [di sini](#)

Maka akan di halaman WEB sebagai berikut :



Gambar 2: Halaman WEB untuk mencoba program

3. Menjalankan Program.

Pilih dan tekan tanda RUN



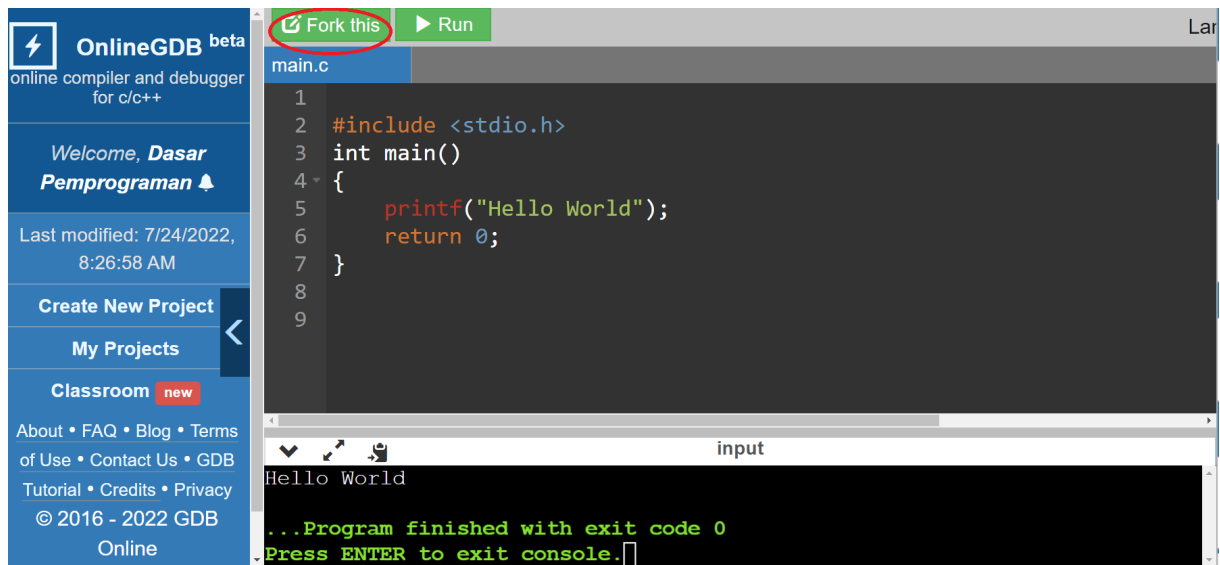
Gambar 3

Hasil cetak akan terlihat di bagian bawah yang di tandai dengan lingkaran hijau.



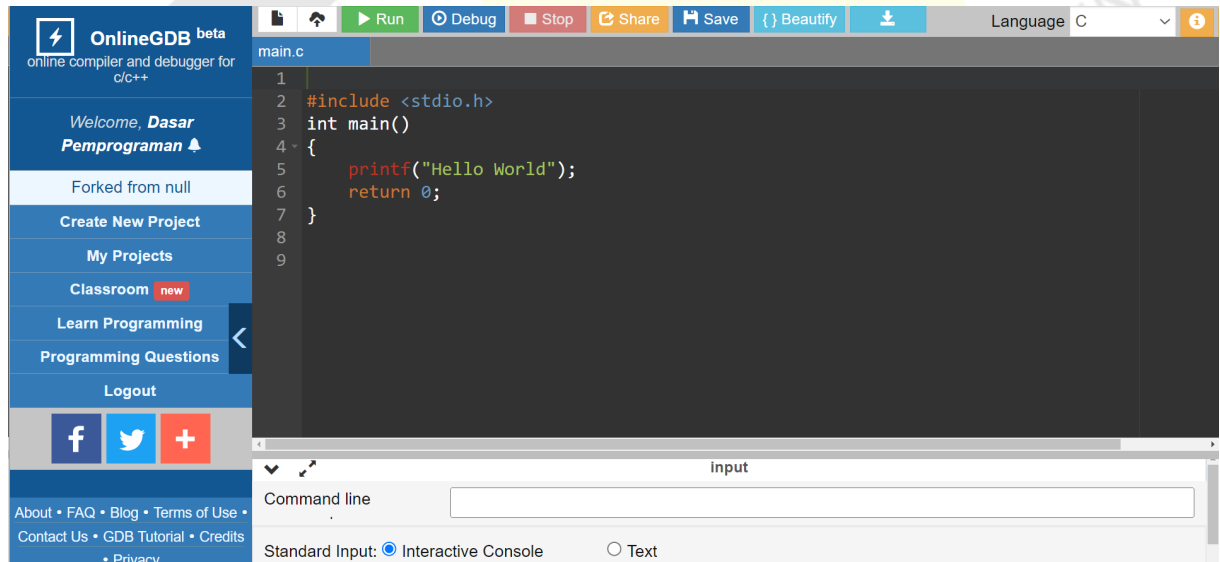
## 4. Memodifikasi Program.

Modifikasi program dilakukan dengan menekan tombol "Fork this" pada menu yang dilingkari dengan warna merah pada Gambar reffig:onlinegdb8.



Gambar 4: Modifikasi Program

Maka akan muncul halaman lengkap yang siap untuk di modifikasi.

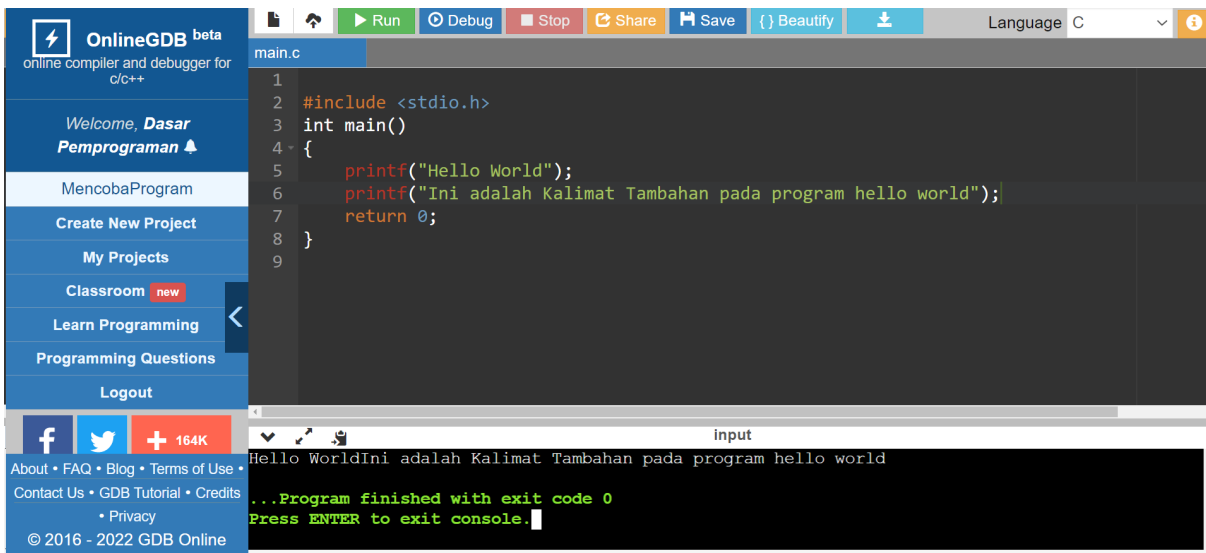


Gambar 5: Halaman program dengan menu lengkap

selanjutnya ditambahkan statement

```
printf ("Ini adalah kalimat tambahan pada hello world")
```

kenudian program kembali dijalankan dengan menekan tombol "Run".

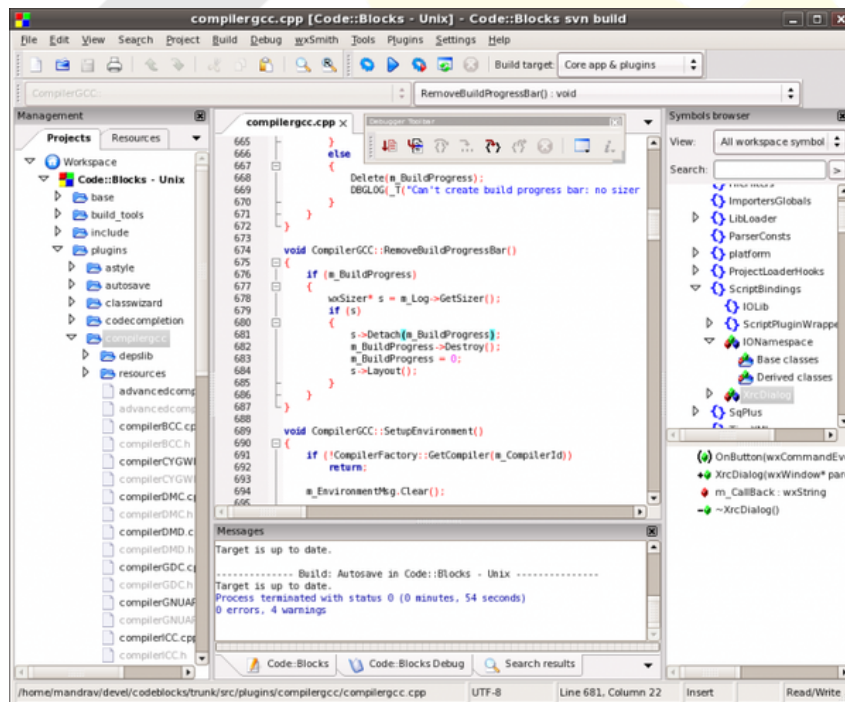


Gambar 6: Memodifikasi program Hello World

## 0.3 CodeBlock

### 0.3.1 Code:Blocks

Code::Blocks adalah IDE open-source cross-platform yang mendukung banyak compiler termasuk GCC, Clang dan Visual C++.

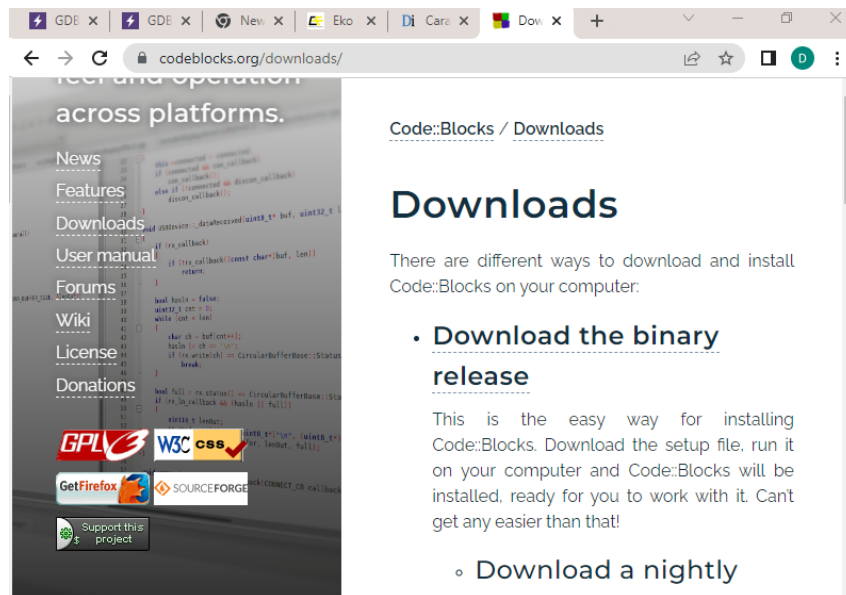


Gambar 7

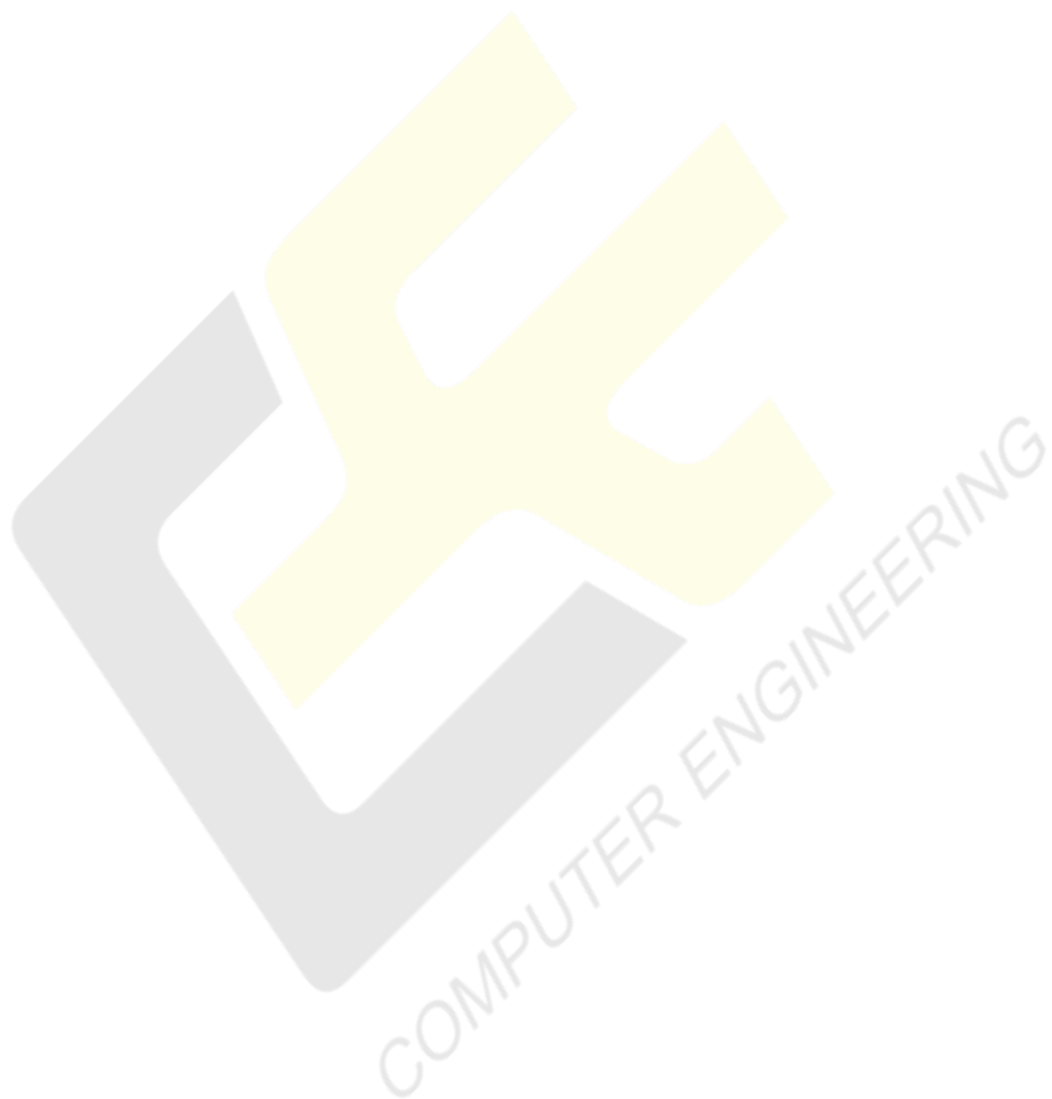
#### 0.3.1.1 Instalasi Code::Blocks

Code::Block dapat download di:

<https://www.codeblocks.org/downloads/>



Gambar 8



## Bab 1

# Sistem Komputer

Sistem Komputer Terdiri dari Perangkat Keras dan Perangkat Lunak

### 1.1 Perangkat Keras Komputer

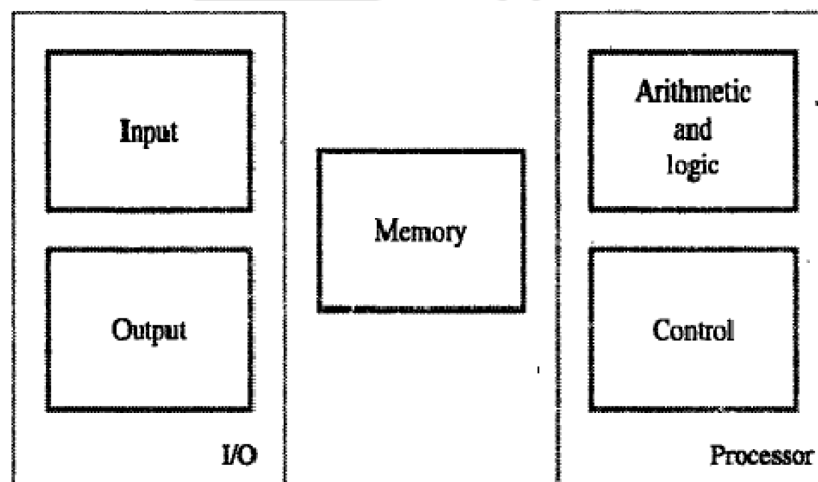
Kata komputer berasal dari kata compute. Menghitung artinya menghitung. Pada awalnya komputer didefinisikan sebagai kalkulator super cepat yang memiliki kapasitas untuk memecahkan masalah aritmatika dan ilmiah yang kompleks dengan kecepatan sangat tinggi. Tetapi pada saat ini komputer melakukan banyak tugas lain seperti menerima, mengurutkan, memilih, memindahkan, membandingkan berbagai jenis informasi. Informasi yang diberikan oleh user ke komputer ini adalah data input. Kemudian data input tersebut di proses oleh komputer untuk menghasilkan informasi dalam bentuk lain yang data output atau data keluaran.

Di dalam melaksanakan proses tersebut, komputer diberi serangkaian instruksi agar dapat melakukan berbagai macam operasi. Rangkaian instruksi tersebut disebut sebagai program komputer.



Gambar 1.1

#### 1.1.1 Unit Fungsional Komputer



Gambar 1.2

Every Digital computer systems consist of five distinct functional units. These units are as follows: Komputer digital mempunyai 5 buah unit fungsional yang berbeda.

1. unit Input : menerima informasi yang telah dikodekan dari operator manusia.
2. Memori : menyimpan informasi yang diterima untuk digunakan nanti.
3. Central Processing Unit (Processor) : mengeksekusi instruksi dari program yang disimpan dalam memori
4. Unit Output : bereaksi terhadap dunia luar.
5. Unit Kontrol: mengkoordinasikan semua tindakan ini.

%beginenumerate

### 1.1.2 Unit Input

Perangkat input adalah bagian dari instrumen atau perangkat keras yang memungkinkan pengguna untuk memberikan data, informasi, atau instruksi kontrol ke komputer. Data dimasukkan ke dalam komputer dalam format mentah, yang diubah menjadi bahasa yang dapat dimengerti komputer oleh perangkat input dan diproses oleh unit pemrosesan pusat (CPU) untuk menghasilkan output. Contoh :

- keyboard
- mouse
- mikrofon
- Usb disk
- kamera

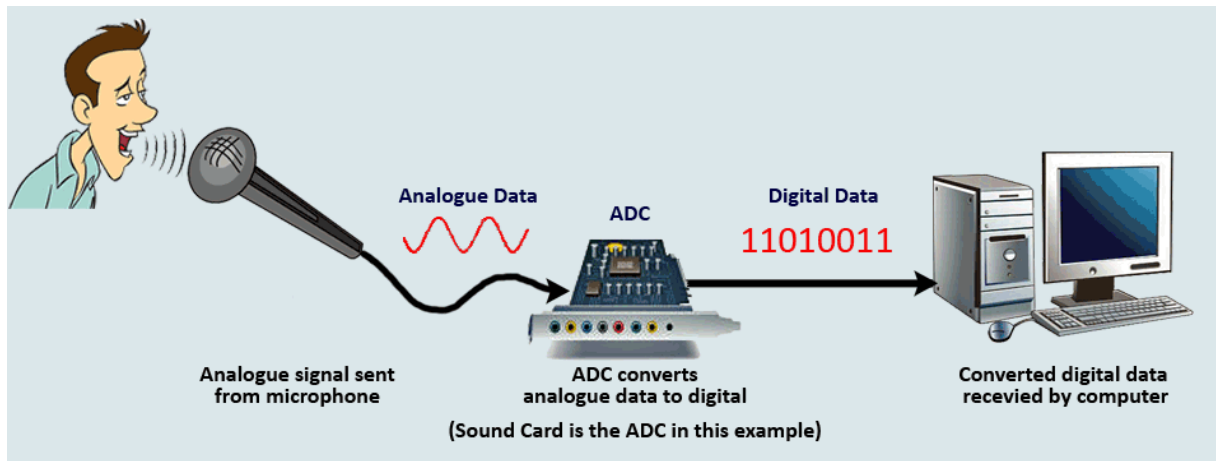


**Gambar 1.3:** Unit Input

Fungsi Unit input adalah :

- Menerima informasi (data) dari lingkungan luar komputer.

- Menterjemahkanyadata kedalam bentuk data digital yang dikenal oleh komputer.

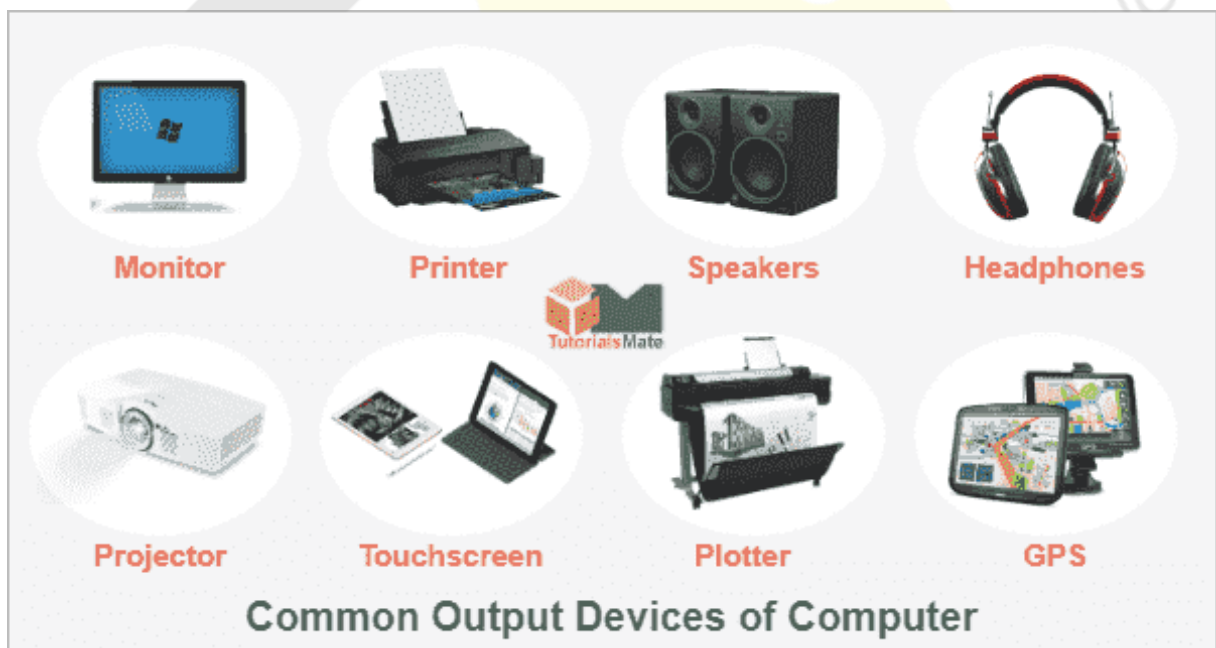


Gambar 1.4: Cara kerja microphone

### 1.1.3 Unit Output

Perangkat keluaran adalah perangkat keras komputer yang mengambil data dari sistem komputer dan selanjutnya menerjemahkan data yang diterima ke dalam bentuk yang dapat dimengerti oleh manusia seperti teks, visual, audio, atau salinan cetak (dicetak di atas kertas).

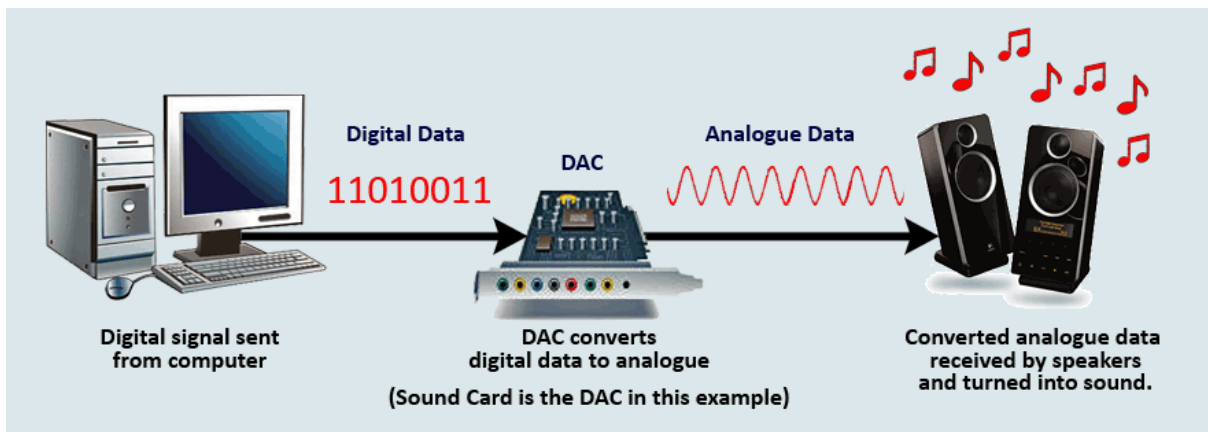
Dengan kata sederhana, perangkat output adalah jenis perangkat periferal yang membantu kita mendapatkan data yang diproses dari komputer.



Gambar 1.5: Unit Output

Contoh :

- Speaker.  
Data yang dihasilkan oleh komputer dalam bentuk data digital. Oleh perangkat soundcard data digital tersebut diterjemahkan ke dalam bentuk data analog yang kemudian diteruskan ke speaker. Output dari speaker tersebut adalah sinyal analog dalam bentuk suara yang dapat didengar oleh manusia.



Gambar 1.6

### 1.1.4 Unit Memori

Memori digunakan untuk menyimpan program dan data. Terdapat dua macam memory There are two classes of memory/storage:

- Primary Memory ( Memori Utama)
  - Memory yang bekerja dengan amat cepat.
  - Contoh:
    - \* Erandom access memory (RAM)
    - \* Cache Memory
 Memory lebih kecil dari RAM digunakan untuk menyimpan program atau data yang akan dijalankan oleh CPU.
- Secondary Storage (Memori Sekunder)
  - Lebih murah dari memory utama.
  - dapat menyimpan data dan program dengan jumlah besar.
  - contoh state drive (SSD), hard disk (HDD), CD, DVD, dsb.



Gambar 1.7: RAM





**Gambar 1.8:** Memory Sekunder

### 1.1.5 Central Processing Unit

Unit Processor terdiri dari Arithmetic Logic Unit dan Unit Kontrol

#### 1.1.5.1 Arithmetic Logic Unit (ALU)

Arithmetic Logic Unit (ALU) bertugas untuk menjalankan instruksi, melakukan semua perhitungan dan pengambilan keputusan.

- Melaksanakan perhitungan
  - Operasi aritmetika: penjumlahan, pengurangan, perkalian, pembagian, dsb.
  - Operasi Logika: AND, OR, NOT, dsb.

#### 1.1.5.2 Unit Kontrol

Unit Kontrol. mengontrol operasi seluruh sistem komputer. Unit kontrol mendapatkan instruksi dari program yang disimpan di unit penyimpanan utama menafsirkan instruksi ini dan selanjutnya mengarahkan unit lain untuk mengeksekusi instruksi. Dengan demikian ia mengelola dan mengkoordinasikan seluruh sistem komputer.

## 1.2 Operasi Komputer

### 1.2.1 Program Komputer

Program komputer digunakan untuk memberi instruksi ke komputer. Oleh karena komputer adalah perangkat elektronik maka instruksi tersebut harus berupa kode yang dipahami oleh komputer. Kode tersebut berupa kode biner yang bernilai 1 dan 0. Angka 1 merepresentasikan "MENYALA" dan "MATI". Sedangkan manusia sulit sekali untuk mengingat kombinasi dari kode biner dengan jumlah banyak. Oleh karena itu telah dikembangkan bahasa pemrograman untuk memberi instruksi kepada komputer yang menggunakan bahasa yang dikenal oleh manusia.

```
#include <stdio.h>

int main()
{
    int W, X, Y, Z;
    char B1, B2, B3, B4;
    //Operasi Logika AND terhadap Pernyataan1 dan pernyataan 2.

    B1 = W<X && Y<Z;
    printf (" Operasi W<X && Y<Z= %d\n", B1);
    B2 = W>X && Y<Z;
    printf (" Operasi W>X && Y<Z= %d\n", B2);
}
```

```
110111110110110100001010001001
01010100011110001010100101010
11110101010000010101101010101
110111110110110100001010001001
01010100011110001010100101010
11110101010000010101101010101
110111110110110100001010001001
01010100011110001010100101010
11110101010000010101101010101
```

**Bahasa Tingkat Tinggi**

**Penterjemah ke Bahasa Mesin**

**Bahasa Mesin**

- Mudah di pelajari manusia
- Bahasa manusia sehari-hari misalkan Bahasa Inggris

- Bahasa komputer
- Terdiri dari bilangan biner 1 dan 0

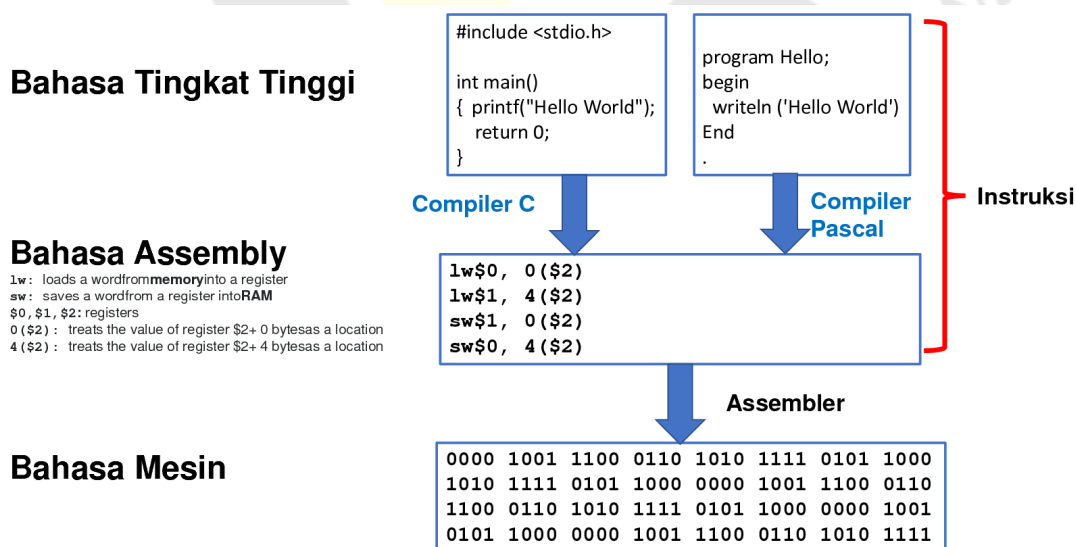
**Gambar 1.9:** Penterjemahan Bahasa Tingkat Tinggi Ke Bahasa mesin

Contoh bahasa tingkat tinggi yang banyak dikenal adalah :

- Bahasa C
- Bahasa C++
- Bahasa PASCAL
- Bahasa Basic
- Bahasa Visual Basic

Bahasa tingkat tinggi tersebut tidak dapat langsung digunakan untuk memberikan instruksi kepada komputer, untuk itu suatu penterjemah bahasa yang digunakan

### 1.2.2 Penterjemahan Bahasa Ke Bahasa Mesin

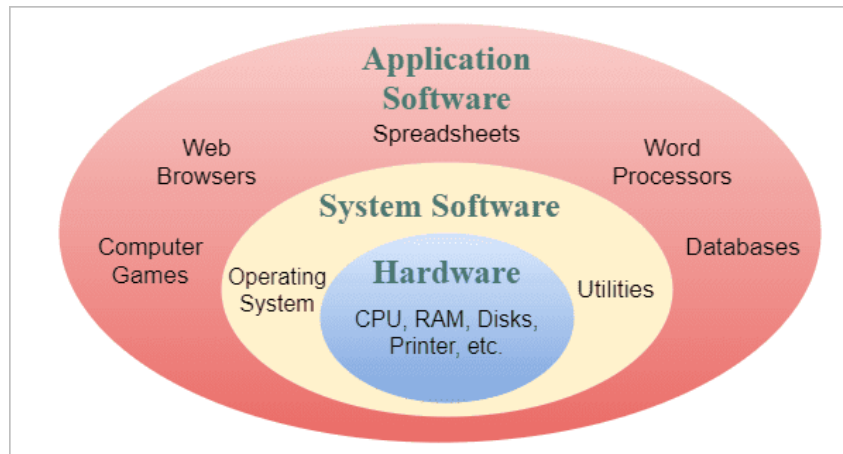


**Gambar 1.10**

1. Compiler menterjemahkan kode program dalam bahasa Tingkat Tinggi ke kode bahasa Assembly.
2. Kode program dalam bahasa assembly tersebut ke dalam bahasa mesin menggunakan assembler dalam bentuk kode biner.

### 1.3 Perangkat Lunak Komputer

- Perangkat lunak komputer atau software adalah kumpulan program komputer dan data terkait, yang menyediakan instruksi untuk memberi tahu komputer tentang apa yang harus dilakukan dan bagaimana cara melakukannya.
- Serangkaian instruksi tersebut akan memandu perangkat keras untuk menyelesaikan setiap tugas.



Gambar 1.11

#### 1.3.1 Perangkat Lunak Sistem

- Perangkat lunak sistem adalah perangkat lunak komputer yang dirancang untuk mengoperasikan perangkat keras komputer untuk menyediakan fungsionalitas dasar dan menyediakan platform untuk menjalankan perangkat lunak aplikasi.
- Refers to the operating system and all utility programs that manage computer resources at a low level

#### 1.3.2 Perangkat Lunak Pemrograman

- Programming software include tools in the form of programs or applications that software developers use to create, debug, maintain, or otherwise support other programs and applications.
- The term usually refers to relatively simple programs such as compilers, debuggers, interpreters, linkers, and text editors,
- Contoh :
  - C
  - C++
  - BASIC
  - JAVA
  - Visual Basic
  - Python
  - HTML
  - PHP

### 1.3.3 Aplikasi

- A program or group of programs designed for end users
- Allows end users to accomplish one or more specific (non-computer related) tasks.



## Bab 2

# Variabel dan Ekspresi

### 2.1 Variabel

Variabel (peubah) merupakan suatu wadah untuk menampung suatu nilai. Nilai pada variabel dapat diubah dengan operasi tertentu. Sebelum memberikan nilai pada variabel, perlu didefinisikan terlebih dulu jenis variabel tersebut.

Beberapa jenis variabel yang umum digunakan antara lain:

- Integer (bilangan bulat).

Variabel berjenis bilangan bulat hanya dapat menampung bilangan bulat seperti -4, 2, 1, 100, 1240, dan sebagainya. Jika ada operasi pembagian terhadap bilangan bulat, maka hasilnya akan selalu bilangan bulat.

Beberapa bahasa pemrograman cara mendeklarasikan variabel berjenis bilangan bulat adalah dengan memberikan nilai bilangan bulat pada variabel tersebut. Contoh :

```
A = 10
B = 4
C= A / B
```

Oleh karena  $A = 10$  adalah bilangan bulat dan  $B = 4$  adalah bilangan bulat maka  $C = 2$

- Float (bilangan pecahan).

Variabel berjenis float digunakan untuk menyimpan nilai pecahan. Operasi pembagian pada variabel ini akan menghasilkan bilangan pecahan.

```
A = 10.0
B = 4.0
C= A / B
```

Oleh karena  $A = 10.0$  adalah bilangan pecahan dan  $B = 4.0$  adalah bilangan pecahan maka  $C = 2.5$ .

- Untai (string).

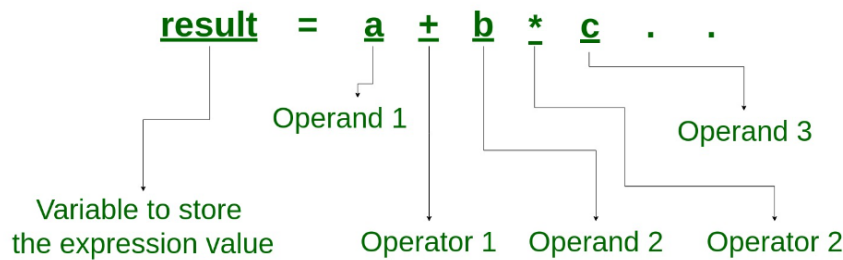
Variabel bertipe string dapat digunakan untuk menampung suatu teks. Nilai pada variabel berjenis string harus diapit dengan tanda kutip (kutip tunggal atau ganda).

```
A = "10"
B = "4"
C= B+A
```

Oleh karena  $A = "10"$  adalah berjenis string dan  $B = "4"$  adalah berjenis string maka  $C = A+B = "104"$ .

## 2.2 Ekspresi

Ekspresi adalah kumpulan operator dan operan yang mewakili nilai tertentu. Suatu ekspresi dapat terdiri dari satu atau lebih operand, tidak ada atau lebih operator untuk menghasilkan suatu nilai.



Gambar 2.1: Bentuk Ekspresi

Jenis Ekspresi

### 2.2.1 Ekspresi Integral dan Floating point

Operator	Simbol	Ekspresi Bilangan Bulat	Hasil	Ekspresi Bilangan pecahan	Hasil
Penjumlahan	+	12+5	17	12.0+5.0	17.0
Pengurangan	-	12-5	7	12.0-5.0	7.0
Perkalian	*	12*5	60	12.0*5.0	60.0
Pembagian	/	12/5	2	12.0/5.0	2.4
Modulus	%	12 % 5	2	Tidak ada	Tidak Ada

### 2.2.2 Ekspresi Relational

Ekspresi Relasional adalah membandingkan antara suatu nilai apakah sama dengan, lebih kecil dari, atau lebih besar dari nilai lain.

Operator	Operator	Ekspresi	Hasil
Kurang dari	<	10<20	True
Kurang dari atau sama dengan	<=	10<=20	True
Lebih besar dari	>	10>20	False
Lebih besar dari atau sama dengan	>=	10>=20	False
Sama dengan	==	10==20	False
Tidak Sama dengan	!=	10!=20	True

### 2.2.3 Ekspresi Logika

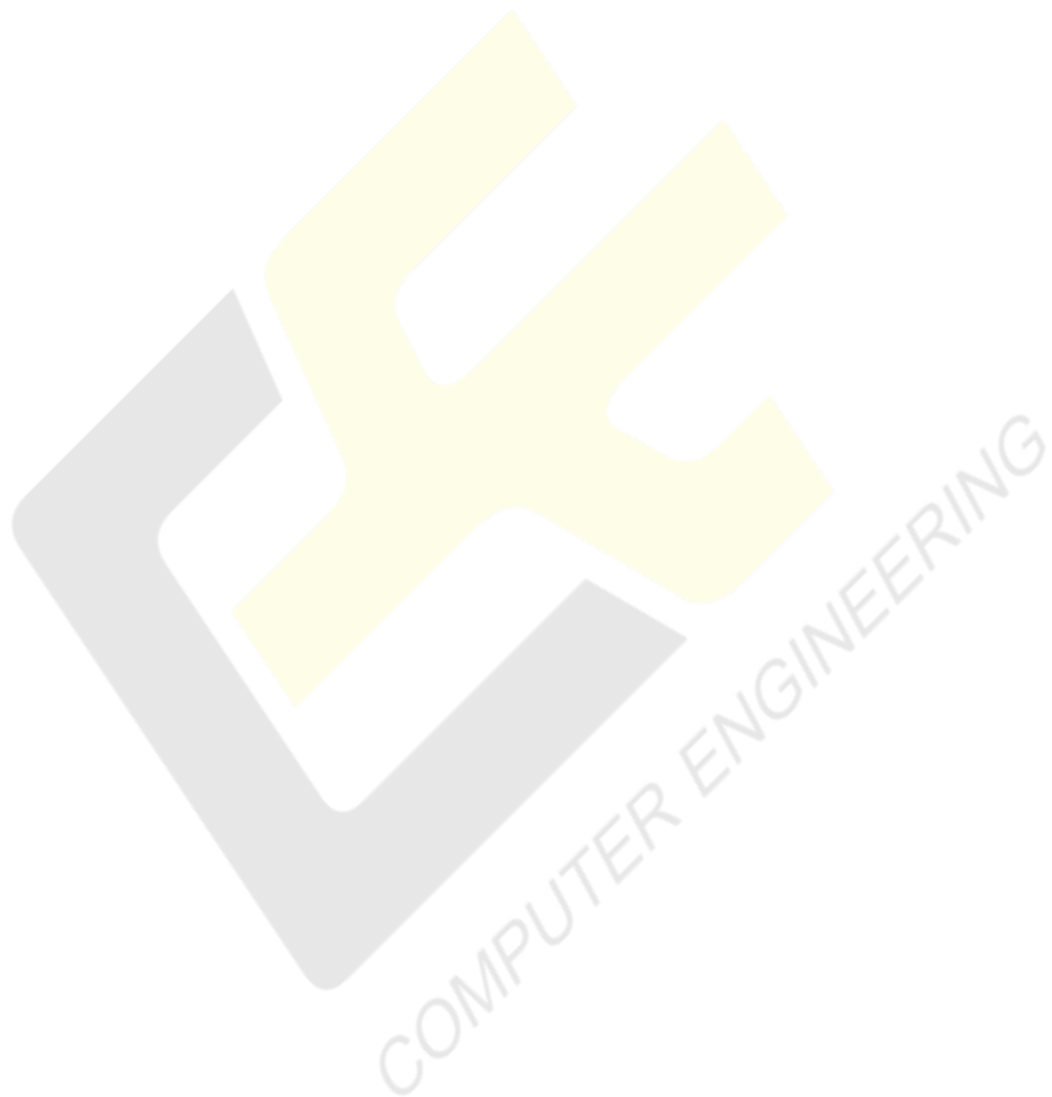
Output dari ekspresi logika adalah True atau False.

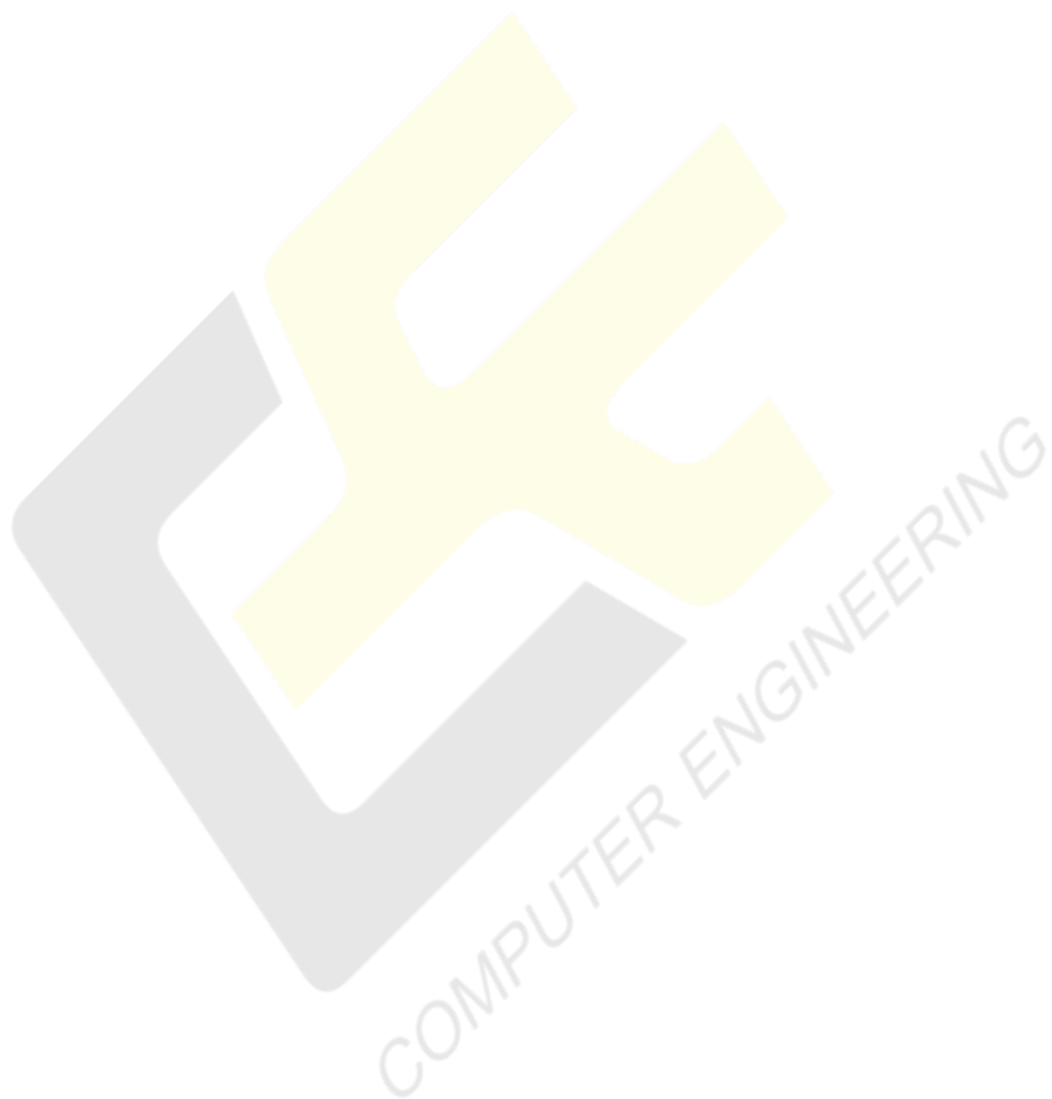
- Operator Logika AND.  
Akan bernilai benar apabila kedua operand bernilai benar.
- Ekspresi Logika OR.  
Akan bernilai benar apabila salah satu dari operan benar.
- Operator Logika NOT  
Expresi NOT memberikan nilai kebalikan dari ekspresi yang disebutkan. Jika nilai yang disebutkan bernilai BENAR maka akan menghasilkan nilai SALAH, begitu pula sebaliknya.

**Tabel 2.1:** Tabel kebenaran logika AND, OR dan NOT

I1	I2	I1 And I2	I1 OR I2	Not I1
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

## 2.3 Contoh Soal







## Bab 3

# Algoritma

### 3.1 Pengertian Algoritma

**Algoritma :**

Algoritma merupakan rangkaian instruksi, perintah atau statment yang dijalankan secara terurut untuk menyelesaikan suatu permasalahan.

Contoh permasalahan sehari-hari yang dapat diselesaikan menggunakan algoritma.

#### 1. Algoritma Membuat kopi

Langkah 1 : Ambil cangkir  
Langkah 2 : Ambil kopi instan  
Langkah 3 : Masukkan bubuk kopi ke dalam cangkir sesuai takaran  
Langkah 4 : Siapkan air panas. Kalau belum tersedia, panaskan air.  
Langkah 5 : Tuangkan air panas ke dalam cangkir  
Langkah 6 : Aduk cangkir hingga kopi larut  
Langkah 7 : Tambahkan gula atau krimer sesuai selera. Aduk.  
Langkah 8 : Minum kopi

**Gambar 3.1:** Algoritma Membuat kopi

#### 2. Manajemen Penjualan Kamu pemilik bisnis kerajinan yang memperoleh pesanan.

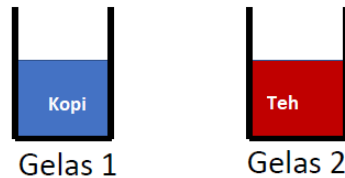
Langkah 1 : Memperoleh pesanan  
Langkah 2 : Mencetak detail pesanan yang berisikan informasi lengkap  
Langkah 3 : Cek apakah valid dan layak diproses  
Langkah 4 : Jika pesanan tidak valid maka batalkan pesanan dan pergi ke Langkah 16  
Langkah 5 : Jika pesanan valid maka setuju pesanan dan langsung diteruskan ke Langkah 6  
Langkah 6 : Cek stok  
Langkah 7 : Jika Stok mencukupi maka laksanakan langkah 9  
Langkah 8 : Beli bahan baku dan buat produk sesuai jumlah pesanan  
Langkah 9 : Masukkan produk yang sudah jadi ke gudang stok  
Langkah 12 : Pesanan siap dikirimkan  
Langkah 13 : Pembayaran penuh diterima  
Langkah 14 : Pesanan dikirimkan ke alamat pemesan  
Langkah 15 : Pemesan menerima paket  
Langkah 16 : Selesai

**Gambar 3.2:** Algoritma Manajemen penjualan

### 3.2 Contoh Menyelesaikan masalah menggunakan algoritma

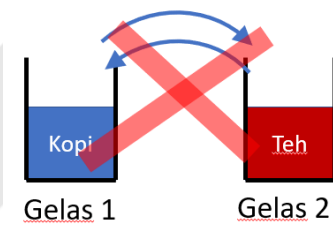
Contoh Algoritma : memecahkan masalah menukar isi dua gelas

- Permasalahan yang akan diselesaikan :  
Menukar isi dari dua buah gelas yaitu Gelas 1 dan Gelas 2 yang masing berisi kopi dan teh seperti yang diperlihatkan pada Gambar 3.3.



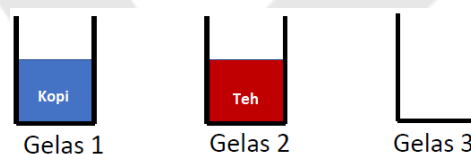
**Gambar 3.3:** Dua buah gelas masing-masing berisi kopi dan teh

- Dengan syarat :  
Isi dari Gelas 1 tidak dapat dituangkan langsung ke Gelas 2 atau sebaliknya agar isi dari kedua gelas tersebut tidak bercampur.



**Gambar 3.4:** Isi Gelas 1 dan Gelas 2 tidak boleh tercampur

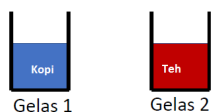
- Pemecahan masalah :  
Diperlukan gelas ketigayang kosong untuk digunakan sebagai pemenampung sementara isi Gelas 1 atau isi Gelas 2.



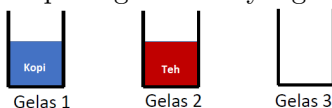
**Gambar 3.5:** Gelas ke 3 yang kosong sebagai penampung sementara isi Gelas 1 dan Gelas 2

Berdasarkan hal tersebut maka dapat dibuat langkah-langkah untuk menukar isi dua gelas berisi kopi dan teh sebagai berikut :

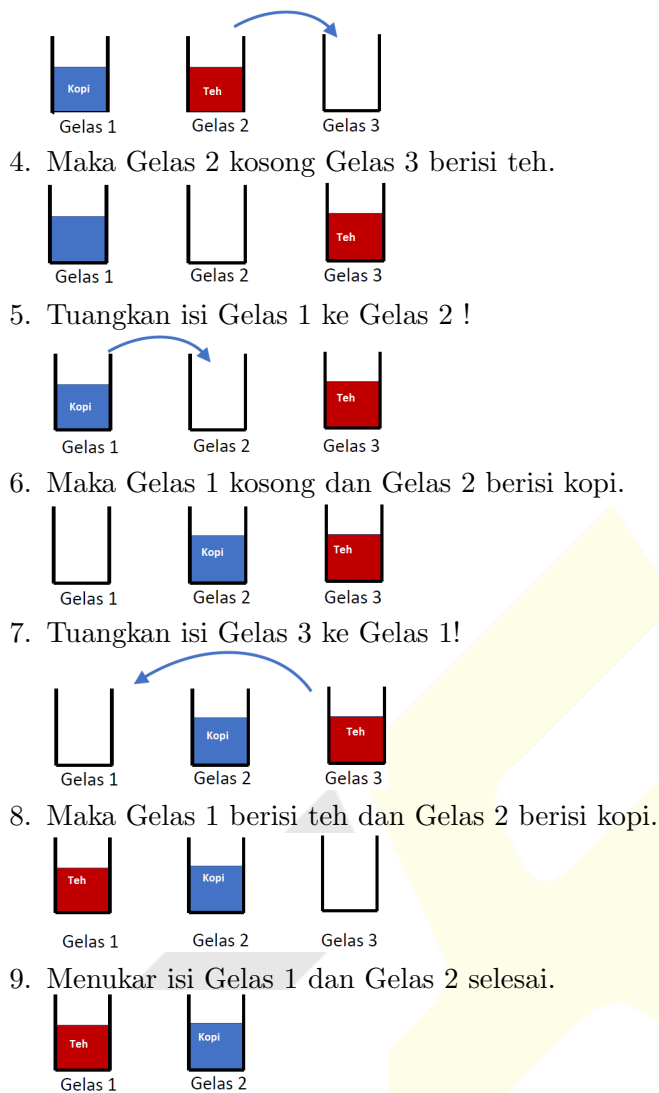
1. Mulai menukar isi dua gelas. Gelas 1 berisi kopi dan Gelas 2 berisi teh



2. Siapkan gelas ke 3 yang kosong!.



3. Tuangkan isi Gelas 2 ke Gelas 3 yang kosong !



Demo Klik [di sini](#)

Dari penjelasan tersebut maka dapat dibuat langkah-langkah cara menukar isi dua gelas adalah sbb :

- Langkah 1. Untuk Gelas 1 berisi Kopi dan Gelas 2 berisi Teh.
- Langkah 2. Siapkan Gelas ke 3 yang kosong!
- Langkah 3. Tuangkan isi Gelas ke 2 ke Gelas ke 3!
- Langkah 4. Tuangkan isi Gelas ke 1 ke Gelas ke 3!
- Langkah 5. Tuangkan isi gelas ke 3 ke Gelas ke 1 !
- Langkah 6 Menukar isi dua gelas selesai

**Gambar 3.6:** Algoritma menukar isi dua gelas dengan Gelas 1 berisi kopi dan Gelas 2 berisi teh

Nah langkah-langkah untuk menukar isi dua gelas seperti pada Gambar 3.6 disebut sebagai Algoritma menukar isi dua gelas.

### 3.3 Penyajian Algoritma

Terdapat beberapa cara menyajikan algoritma.

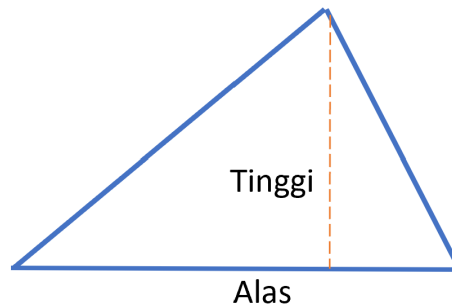
- Deskriptif
- Pseudo Code
- flowchart

### 3.3.1 Penyajian Algoritma secara Deskriptif

Algoritma dituliskan dengan menggunakan cerita atau bahasa umum seperti dalam bahasa sehari-hari.

Gambar 3.1 dan Gambar 3.2 adalah contoh cara menyajikan algoritma secara deskriptif.

1. Contoh algoritma untuk memecahkan masalah untuk menghitung luas segitiga.



**Gambar 3.7:** Segitiga

Luas segitiga dapat dihitung dengan menggunakan persamaan :

$$Luas = \frac{1}{2} \times Alas \times Tinggi \quad (3.1)$$

Apabila kita akan menghitung luas segitiga dengan panjang alas =4 dan tinggi =5 maka dengan menggunakan persamaan diatas akan diperoleh Luas segitiga adalah 10.

**Program Menghitung Luas Segitiga**  
**Algoritma :**  
 Langkah 1 : Mulai menghitung luas segitiga  
 Langkah 2 : Masukan Nilai Alas = 4  
 Langkah 3 : Masukan Nilai Tinggi =5  
 Langkah 4 : Menghitung Luas Segitiga = Alas x Tinggi x 0.5  
 Langkah 5 : Mencetak hasil Luas Segitiga  
 Langkah 6 : Selesai menghitung luas segitiga.

**Gambar 3.8:** Algoritma menghitung luas segitiga disajikan secara deskriptif

⇒ Demo Klik [di sini](#)

Algoritma yang ditunjukkan pada Gambar 3.8 menjelaskan langkah-langkah menghitung luas segitiga.

Dari Gambar tersebut jelas bahwa perhitungan luas segitiga dilaksanakan dalam enam

### 3.3.2 Penyajian Algoritma Menggunakan Pseudocode

Pseudocode

- Pseudo = semu \ mirip.
- code = kode program.

Pseudocode adalah tiruan atau imitasi dari kode bahasa pemrograman.

#### 3.3.2.1 Struktur Penulisan Algoritma Menggunakan Pseudocode

Struktur penulisan menggunakan pseudocode adalah sebagai berikut. Seperti yang telah dijelaskan bahwa pseudocode adalah imitasi dari bahasa pemrograman sesungguhnya oleh karena itu struktur pseudocode akan mengikuti bahasa pemrograman yang sesungguhnya. Contoh struktur pseudocode mirip bahasa C adalah sebagai berikut.

1. Bagian Kepala (Header)  
berisi Nama algoritma dan penjelasan algoritma.  
Contoh :

```
Algoritma Menentukan_nilai_terbesar_dari_3_bilangan  
(Mencari dan mencetak bilangan terbesar yang tersimpan dalam tiga variabel a,b dan c)
```

2. Bagian Deklarasi  
Bagian deklarasi berisikan semua nama variabel yang dipakai di dalam algoritma. Variabel adalah tempat menampung data sementara. Setiap variabel harus didefinisikan menurut jenisnya. Jenis-jenis data tersebut meliputi :

- Integer (bilangan bulat).
- Float (bilangan pecahan desimal).
- Boolean
- Karakter dan string

Contoh :

```
Deklarasi:  
a,b,c, Terbesar : integer
```

3. Bagian Deskripsi.  
Memuat langkah-langkah penyelesaian masalah, contoh :

```
Deskripsi:  
Read(a,b,c)  
Terbesar = a  
  
If (b>Terbesar) then  
    Terbesar = b;  
end if  
If c>Terbesar then  
    Terbesar= c  
end if  
Print Terbesar
```

Selanjutnya secara keseluruhan algoritma untuk mencari bilangan terbesar dapat dilihat pada Gambar 3.9.

```

Algoritma Menentukan_nilai_terbesar_dari_3_bilangan
(Mencari dan mencetak bilangan terbesar yang tersimpan
dalam tiga variabel a,b dan c)

Deklarasi:
a,b,c, Terbesar : integer

Deskripsi:
Read(a,b,c)
Terbesar = a

If (b>Terbesar) then
    Terbesar = b;
end if

If c>Terbesar then
    Terbesar= c
end if
Print Terbesar

```

**Gambar 3.9:** Algoritma menggunakan pseudocode untuk mencari nilai terbesar diantara a,b dan c

### 3.3.2.2 Perbandingan Algoritma dengan Pesudocode dengan bahasa pemrograman lain

Selanjutnya akan dibandingkan antara algoritma dengan pseudocode, bahasa C dan PASCAL. Algoritma pada Gambar 3.10 dengan variabel Alas,Tinggi dan Luas bertipe data float atau pecahan.

```

Algoritma Menghitung Luas Segitiga

Deklarasi :
Alas,Tinggi,Luas :float

Deskripsi:
Alas = 4
Tinggi = 5
Luas = 0.5*Alas*Tinggi
print Luas

```

**Gambar 3.10:** Pseudocode menghitung luas segitiga

selanjutnya mari kita lihat implementasi algoritma tersebut pada dua bahasa pemrograman C dan Pascal.

Selanjutnya dengan menyandingkan algoritma kita buat kita bandingkan dengan implementasinya dengan menggunakan bahas bahasa pemrograman C dan bahasa pemrograman pascal.

**Pseudocode :**

```

Algoritma Menghitung Luas
Segitiga

Deklarasi :
Alas,Tinggi,Luas :float

Deskripsi:
Alas = 4
Tinggi = 5
Luas = 0.5*Alas*Tinggi
print Luas

```

(a)

**Bahasa C :**

```

//Program menghitung luas
segitiga
#include <stdio.h>

int main()
{
    //Deklarasi Variabel
    float Alas, Tinggi, Luas;

    //Deskripsi Program
    Alas = 4;
    Tinggi = 5;
    Luas = 0.5*Alas*Tinggi;
    printf("Luas = %f",Luas);
    return 0;
}

```

(b)

**Bahasa Pascal :**

```

program
Menghitung_Luas_Segitiga;
//Deklarasi
var
Alas,Tinggi,Luas :real;

Begin
    //Deskripsi
    Alas := 4;
    Tinggi :=5;
    Luas := 0.5*Alas*Tinggi;
    writeln (Luas);

end.

```

(c)

**Gambar 3.11:** Menghitung luas segitiga dengan menggunakan (a).Algoritma Pesudocode, (b) Bahasa C dan (c) Bahasa pascal

Dari Gambar 3.11 terlihat bahwa urutan langkah

**Tabel 3.1:** Perbandingan instruksi pada pseudocode, Bahasa C dan Bahasa Pascal untuk menghitung luas segitiga

Baris	Pseudocode	Bahasa C	Bahasa Pascal
1	Menghitung Luas Segitiga	//Menghitung Luas Segitiga	program Menghitung_Luas_Segitiga;
2			//Deklarasi
3		#include <stdio.h>	var
4		int main()	Alas,Tinggi,Luas :real;
5		{	begin
6	Deklarasi :	//Deklarasi Variabel	
7	Alas,Tinggi,Luas :float	float Alas, Tinggi, Luas;	
8			
9	Deskripsi:	//Deskripsi Program	//Deskripsi Program
10	Alas = 4	Alas = 4;	Alas := 4;
11	Tinggi = 5	Tinggi = 5;	Tinggi :=5;
12	Luas = 0.5*Alas*Tinggi	Luas = 0.5*Alas*Tinggi;	Luas := 0.5*Alas*Tinggi;
13	print Luas	printf("Luas = f",Luas);	writeln ('Luas Segitiga',Luas);
14		return 0;	
15		}	end.




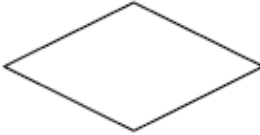

Dengan membandingkan

### 3.3.3 Flowchart

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis berupa bagan untuk menunjukkan langkah-langkah berisi urutan proses untuk memecahkan masalah. Setiap proses berisi pernyataan yang diwakili dengan simbol tertentu. Hubungan antar proses di tunjukan oleh garis penghubung.

#### 3.3.3.1 Simbol Dasar Flowchart

Terdapat banyak simbol dalam flowchart. Di dalam buku ini akan digunakan lima simbol dasar seperti yang ditunjukkan pada Gambar 3.12 yang terdiri dari : terminator, proses, input\output,decesion dan flow line.

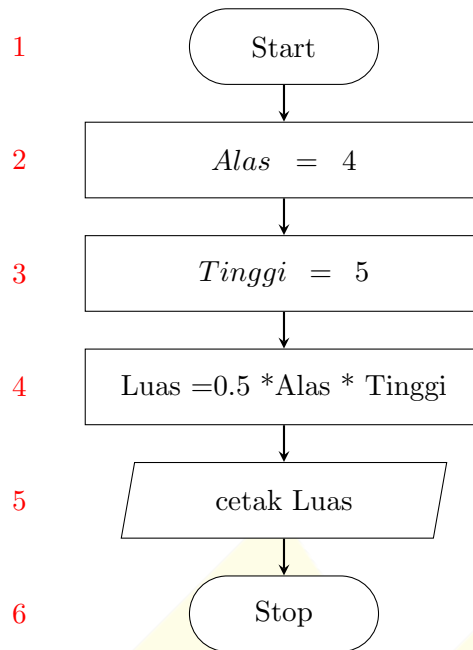
No	Simbol	Nama	Fungsi
1		Terminator	Simbol Awal (Start) Simbol Akhir (End)
2		Proses	Perhitungan / pengolahan
3		Input / Output Data	Membaca atau menulis data
4		Decision	Simbol pernyataan pilihan, berisi suatu kondisi yang selalu menghasilkan 2 nilai keluaran yaitu benar atau salah
5		Flow Line	Simbol aliran / penghubung

Gambar 3.12: Simbol Dasar Flowchart

### 3.3.3.2 Implementasi Flowchart Untuk Menghitung Luas Segitiga

Contoh flowchart untuk menghitung luas segitiga diperlihatkan pada Gambar 3.13

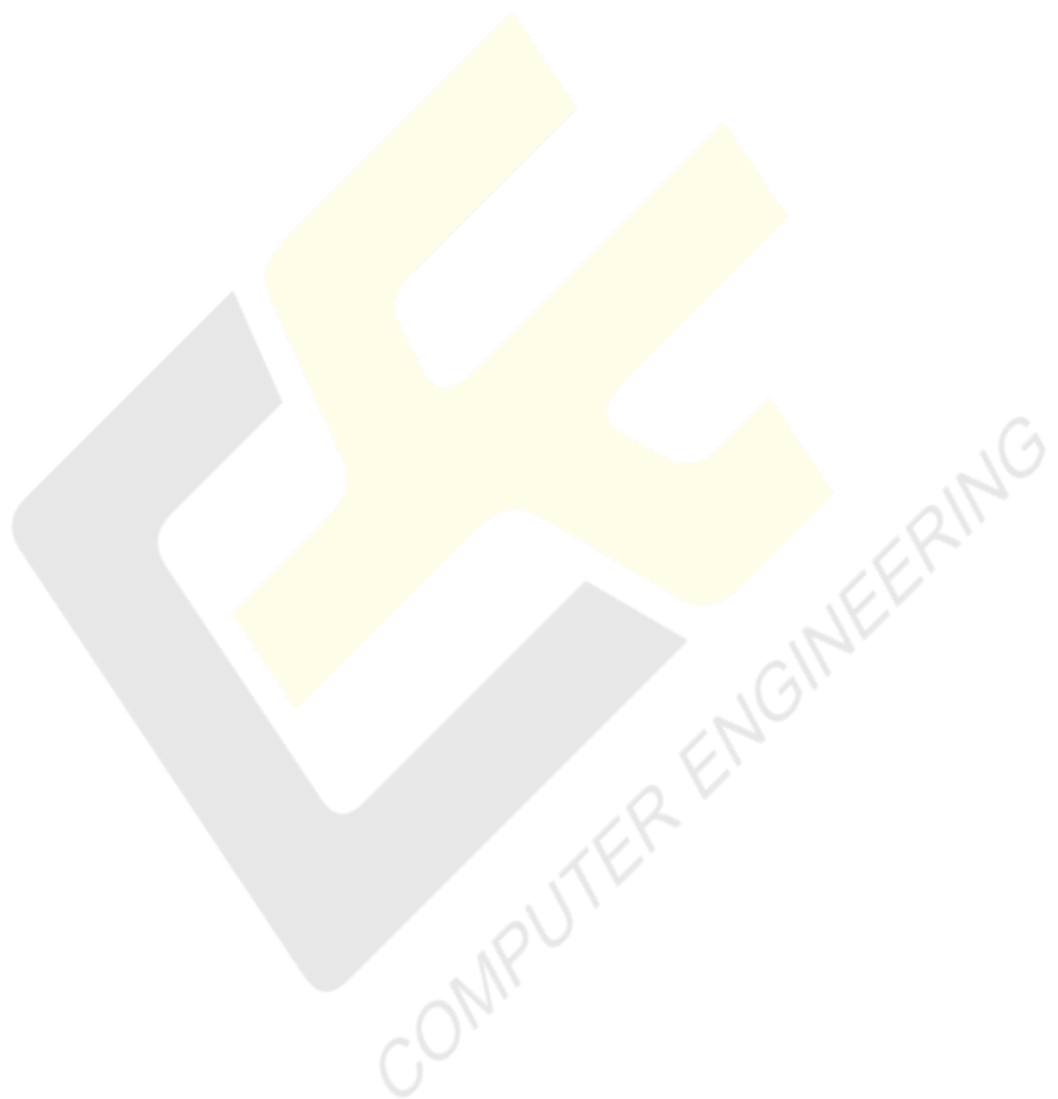




**Gambar 3.13:** Flowchart menghitung luas segitiga

⇒ Demo Klik [di sini](#)

#### 3.3.3.3 Perbandingan FLOWchart, Pseudocode dan Bahasa pemrograman C untuk menghitung Luas segitiga



## Bab 4

# Aliran Kontrol dalam Bahasa Pemrograman

<https://docplayer.info/31259823-Chapter-2-tipe-data-identifier-operator-dan-control-statement.html>  
Aliran kontrol pada program adalah bagaimana setiap instruksi pada algoritma atau program dikerjakan. Arah aliran kontrol tersebut dapat dipilih berdasarkan kondisi yang telah ditetapkan terlebih dahulu

Struktur Kontrol adalah cara untuk menentukan aliran kontrol dalam program. Algoritma atau program dapat memilih ke arah statement dan instruksi apa yang dikerjakan berdasarkan parameter dan kondisi tertentu.

Berdasarkan urutan langkah yang digunakan dalam menjalankan instruksi untuk memecahkan masalah, maka struktur algoritma dapat dibagi menjadi :

1. runtunan (sekuensial)
2. seleksi kondisi atau percabangan.
3. iterasi atau perulangan.

Untuk memecahkan suatu masalah yang kompleks maka ketiga struktur algoritma tersebut dapat digunakan secara bergantian.

### 4.1 Struktur Runtunan \ sekuensial

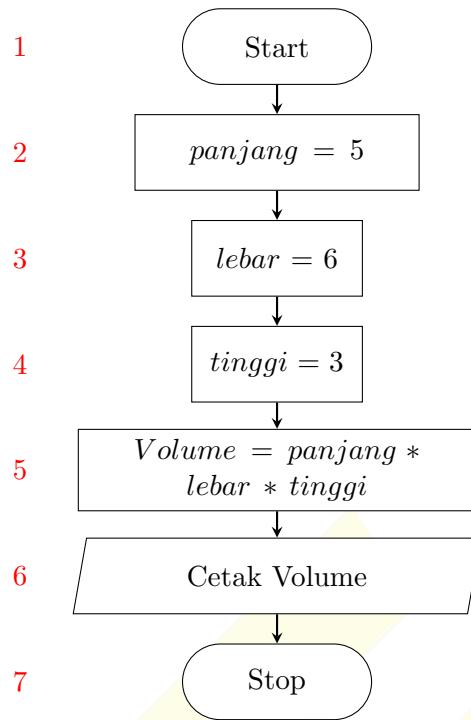
Runtunan adalah algoritma menjalankan instruksi secara berurutan sesuai dengan urutan instruksi yang telah diberikan.

```
Instruksi 1  
Instruksi 2  
Instruksi 3  
..  
Instruksi N
```

**Gambar 4.1:** Instruksi dengan jumlah N dikerjakan satu persatu sampai selesai

Gambar 4.1 adalah serangkaian instruksi mulai dari Instruksi 1 sampai dengan Instruksi N.

Contoh runtunan diperlihatkan oleh flowchart Gambar 4.2. Didalam menghitung volume balok, instruksi dijalankan satu persatu mulai dari memasukkan nilai panjang balok, diikuti dengan memasukkan nilai lebar balok dan yang terakhir memasukkan tinggi balok.



Gambar 4.2: Flowchart menghitung volume balok

⇒ Demo Klik [di sini](#)

Algoritma Menghitung Volume balok

Deklarasi :  
panjang, lebar, tinggi : float

Deskripsi :  
panjang = 5  
lebar = 6  
tinggi = 3  
Volume = panjang\*lebar\*tinggi  
print Volume

Gambar 4.3

## 4.2 Kontrol aliran pemilihan atau selection

Algoritma dapat memilih suatu rangkaian instruksi untuk dijalankan apabila suatu kondisi terpenuhi.

Struktur kontrol aliran pemilihan ini dapat dibagi menjadi dua yaitu:

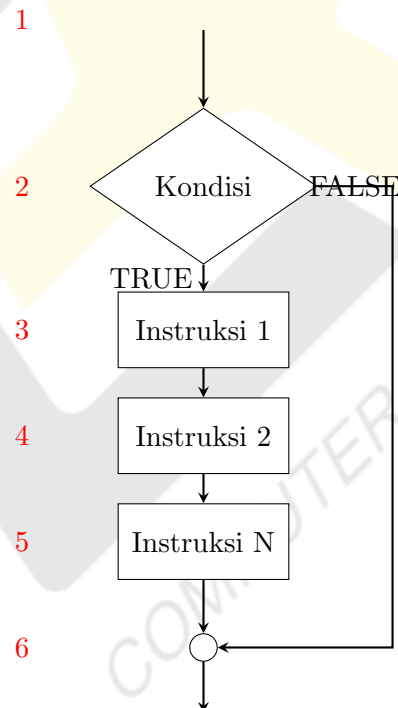
### 4.2.1 Kontrol aliran alternatif tunggal

Pada kontrol aliran alternatif tunggal ini suatu rangkaian instruksi dijalankan apabila suatu kondisi telah dipenuhi. Bentuk struktur kontrol ini ditunjukkan pada Gambar oleh Gambar 4.4. Suatu blok instruksi yang dibatasi oleh **Then** dan **Endif** yang terdiri dari Instruksi 1, Instruksi 2 sampai dengan Instruksi N akan dijalankan apabila suatu kondisi di penuhi.

```
If (Kondisi) Then  
  Instruksi 1  
  Instruksi 2  
  .  
  .  
  Instruksi N  
EndIf
```

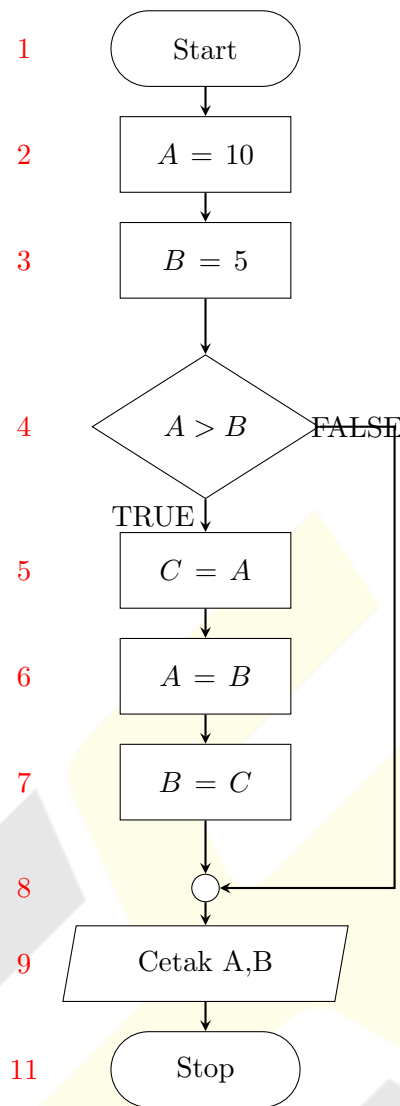
**Gambar 4.4:** Struktur kontrol aliran alternatif tunggal

dari Gambar 4.5 terlihat jelas bagaimana aliran proses yang dijalankan pada langkah sebelumnya (1) menuju ke blok (2) untuk diperiksa kondisinya. Jika kondisi benar maka instruksi 1 sampai dengan Instruksi N akan dijalankan, apabila tidak kondisi tidak dipenuhi Instruksi 1 sampai dengan Instruksi N akan dilewati.



**Gambar 4.5:** Flowchart Struktur kontrol aliran alternatif tunggal

**Contoh 4.2.1** Pada contoh ini akan diurutkan dari kecil ke besar isi dua variabel A dan B dengan nilai A lebih besar dari B dengan nilai A = 10 dan B = 5. Langkah-langkah untuk mengurutkan isi dua variabel tersebut diperlihatkan pada Gambar 4.2.1.



**Gambar 4.6:** Flowchart Mengurutkan isi dua variabel A dan B dengan nilai  $A > B$

⇒ Demo Klik [di sini](#)

Algoritma Mengurutkan dua bilangan A dan B

Deklarasi :

A,B,C :Integer

Deskripsi :

A = 10

B = 5

if (A>B) Then

    C=A

    A=B

    B=C

EndIf

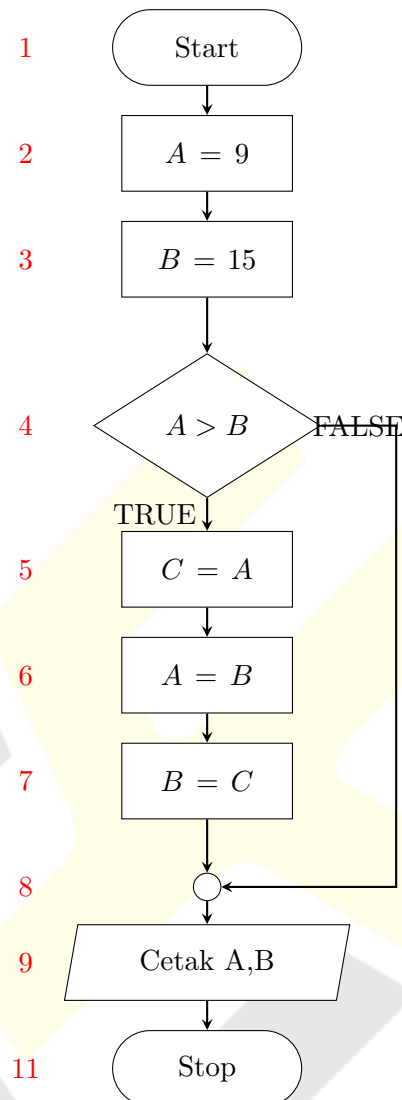
Print A,B

**Gambar 4.7:** Algoritma Mengurutkan dua bilangan A dan B dengan dimana  $A > B$ .

**Contoh 4.2.2** Contoh Kasus mengurutkan isi dua variabel A dan B dengan nilai A lebih kecil

dari B

Selanjutnya kita akan mempelajari aliran proses untuk mengurutkan isi dari dua variabel dengan nilai A lebih kecil dari B.



**Gambar 4.8:** Flowchart Mengurutkan isi dua variabel A dan B dengan nilai  $A < B$

Dengan membandingkan maka dapat dilihat :

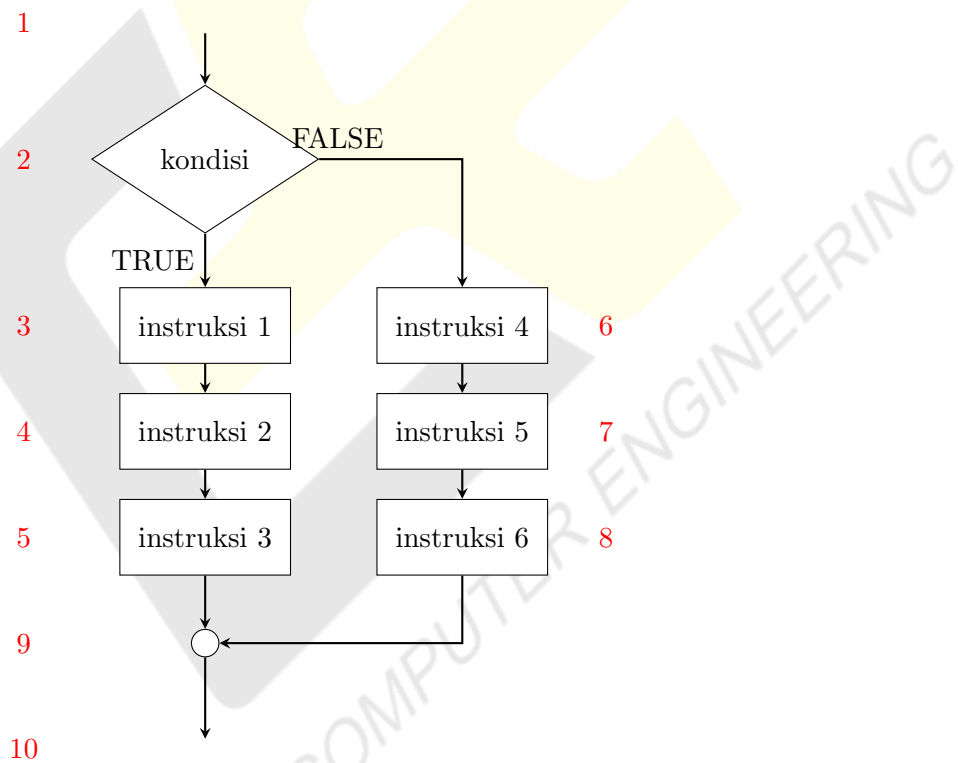
- contoh tersebut terlihat bahwa apabila nilai A lebih besar dari B maka kondisi  $A > B$  pada baris empat akan bernilai benar, sehingga algoritma menjalankan langkah 5,6,7 untuk menukar isi dua A dan B.
- Tetapi ketika nilai A lebih kecil dari B seperti pada contoh 2 maka baris 5,6,7 akan dil

#### 4.2.2 Alternatif Ganda

Selection dengan alternatif ganda memberikan dua pilihan berdasarkan kondisi untuk menjalankan dua instruksi yang berbeda. Bentuk kontrol struktur ini diperlihatkan pada Gambar 4.9. Aliran proses ditentukan oleh dua macam **kondisi**. Apabila kondisi tersebut benar maka akan dijalankan alternatif pertama untuk menjalankan Instruksi 1, Instruksi 2 dan Instruksi 3. Apabila kondisi tidak dipenuhi maka menjalankan Instruksi 4, Instruksi 5 dan Instruksi 6.

```
If (Kondisi) Then
  Instruksi 1
  Instruksi 2
  instruksi 3
Else
  Instruksi 4
  Instruksi 5
  Instgruksi 6
EndIf
```

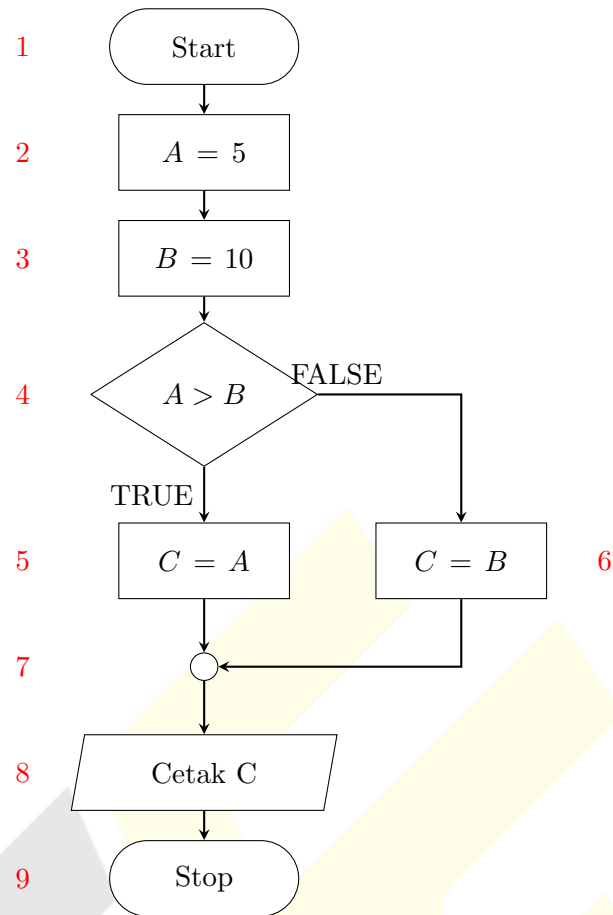
**Gambar 4.9:** Struktur pemilihan alternatif ganda



**Gambar 4.10:** Flowchart untuk struktur pemilihan alternatif ganda

4.11 adalah mencari dan mencetak nilai terbesar isi dari dua variabel A dan B. Instruksi  $C=A$  akan di laksanakan apabila kondisi  $A>B$  terpenuhi sedangkan instruksi  $C=B$  akan dilaksanakan apabila kondisi tersebut tidak terpenuhi.





**Gambar 4.11:** Mencari nilai terbesar dari dua variabel A dan B

⇒ Demo Klik [di sini](#)

Algoritma Mencari dan Mencetak isi bilangan terbesar

Deklarasi :  
A, B, C : float

Deskripsi :

```

A = 5
B = 10
If A>B Then
    C = A
Else
    C = B
EndIf
print C

```

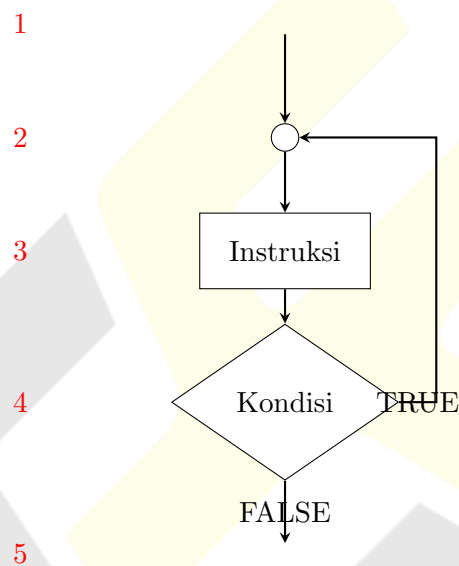
**Gambar 4.12:** Algoritma mencari dan mencetak bilangan terbesar diantara dua variabel A dan B

## 4.3 Pengulangan(Repetition)

Sekumpulan instruksi yang sama dijalankan beberapakali.

Bentuk struktur pengulangan diperlihatkan oleh Gambar 4.13.

1. Langkah untuk menjalankan instruksi berasal dari baris 1 akan menuju ke simpul pada baris 2.
2. Kemudian dilanjutkan menuju ke baris ke 3 dimana terdapat beberapa instruksi yang akan dijalankan secara berulang.
3. Pada langkah selanjutnya, dengan menempatkan struktur pemilihan pada baris ke empat maka diberikan dua kondisi.
  - (a) Apabila kondisi BENAR maka langkah untuk menjalankan instruksi akan diarahkan ke simpul pada baris ke 2 untuk menjalankan instruksi pada baris ke 3 lagi.
  - (b) Apabila kondisi SALAH maka arah langkah menjalankan instruksi akan diarahkan ke baris 5 untuk menjalankan langkah selanjutnya. .



Gambar 4.13: Struktur Pengulangan

Dari penjelasan tersebut maka terlihat bahwa pengulangan melibatkan struktur pemilihan yang digunakan untuk membelokkan arah aliran proses untuk kembali mengulang instruksi pada baris ke 3

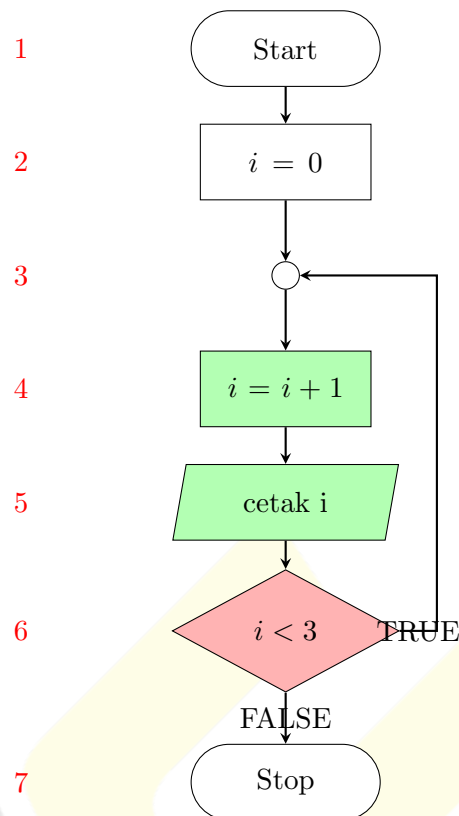
```

loop:
  Instruksi 1
  Instruksi 2
  If Kondisi Then
    goto Loop
  EndIf
  
```

**Contoh 4.3.1** Membuat flowchart dan algoritma untuk mencetak deret angka sebagai berikut :

1,2,3

Jawab : Flowchart untuk mencetak bilangan 1,2,3



**Gambar 4.14:** Pengulangan dengan menambahkan seleksi untuk membatasi jumlah pengulangan

⇒ Demo Klik [di sini](#)

Untuk lebih jelas diberikan contoh flowcart yang diguna Gambar 4.13 diperlihatkan dua kondisi. Kondisi yang pertama Selanjutnya akan dicoba struktur pengulangan Algoritma untuk menceta bilangan 1,2,3

Algoritma Mencetak bilangan dari 1 sampai 3

Deklarasi:  
i : integer

Deskripsi :

i=1

loop:

i=i+1

cetak i

if i<3 then

goto loop

endif

## 4.4 Contoh Soal

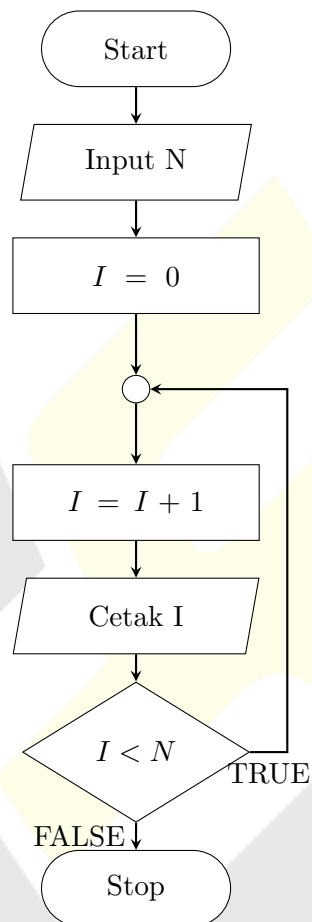
.

4.4.1 Buatlah flowchart untuk membuat deret angka embuat deret angka dari 1 sampai ! N.

input N = 10

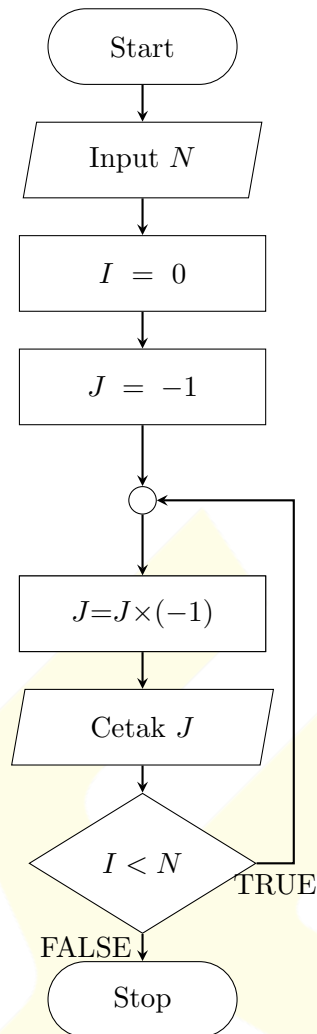
Output : 1,2,3,4,5,6,7,8,9,10

Jawab :



4.4.2 Buatlah flowchart untuk membuat deret angka membuat deret angka dengan pola  $1, -1, 1, -1, \dots$  sampai suku ke  $N$ .

Jawab :



## 4.5 Latihan Soal

4.5.1 Buatlah flowchart untuk menampilkan deret angka sampai suku ke N dengan ketentuan sebagai berikut.

- Apabila input jumlah suku N positif maka bilangan genap bernilai positif dan ganjil bernilai negatif.
- Apabila input jumlah suku N negatif maka bilangan genap bernilai negatif dan ganjil bernilai positif.

Contoh :

Input Jumlah suku N = 10

Output : 1,-2,3,-4,5,-6,7,-8,9,-10

Input Jumlah suku N = -10

Output : -1,2,-3,4,-5,6,-7,8,-9,10

4.5.2 Apabila bilangan fibuncaci untuk suku ke  $n$  megikuti persamaan

$$f(n) = \begin{cases} 0 & x = 0 \\ 1 & x = 1 \\ f(n-1) + f(n-2) & x \geq 2 \end{cases}$$

Maka buatlah deret angka bilangan fibonaci sampai suku ke  $n$

0,1,1,2,3,5,8,13,...

Dengan persamaan untuk mencari bilangan fibonaci suku ke  $n$  adalah :

Apabila :

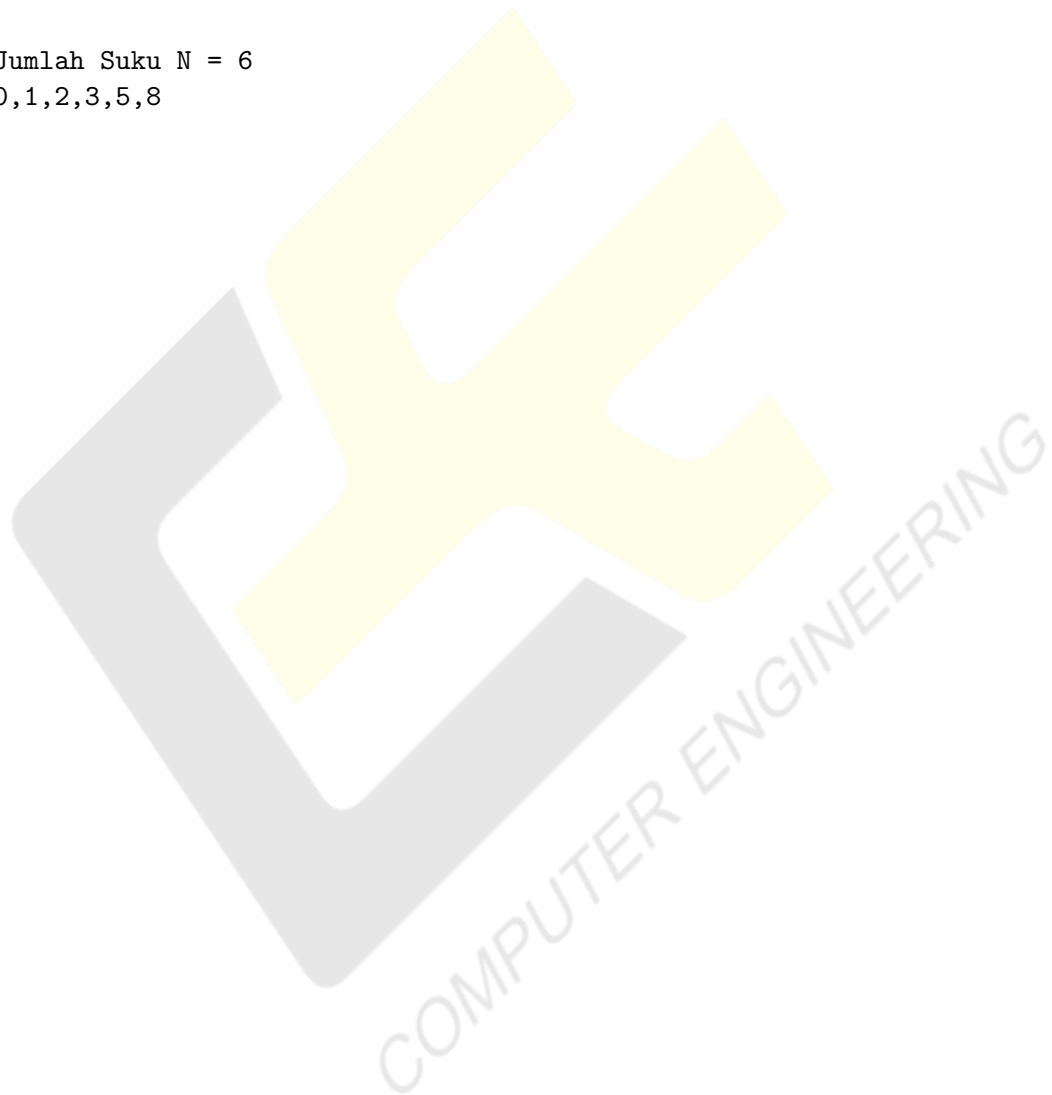
$$\begin{aligned}a &= f(n-2) \\b &= f(n-1) \\c &= b + a\end{aligned}\tag{4.1}$$

maka buatlah flowchart untuk mencari deret angka bilangan fibonaci sampai suku ke  $N$ .

Contoh :

Input : Jumlah Suku  $N = 6$

Output : 0,1,2,3,5,8



## Bab 5

# Pemrograman C

### 5.1 Struktur Bahasa C

Program pada Listing 5.1 merupakan program sederhana untuk mencetak ke layar.

**Listing 5.1:** Contoh program sederhana dalam bahasa C

```
1  #include <stdio.h>
2
3  int main()
4  {
5      //Mencetak Kelayar
6      printf("Halo Dunia");
7      return 0;
8  }
```

⇒ Untuk mencoba program Klik [di sini](#)

Keterangan program adalah sbb.

- Baris 1 : `#include <stdio.h>`  
header file library untuk fungsi input dan output seperti `printf()` (seperti yang digunakan pada baris ke 6)
- Baris 2 : Baris Kosong.
- Baris 3 : `int main()`  
adalah fungsi utama yang akan di jalankan terlebih dahulu.
- Baris 4 : `{`  
Memulai blok statement fungsi `main()`
- Baris 5 : `//Mencetak Ke Layar`  
Komentar untuk menjelaskan program, Komentar akan diabaikan oleh program.
- Baris 6 : `printf("Halo Dunia");`  
Mencetak tulisan mencetak "Halo Dunia" ke layar.
- Baris 7 : `return 0;`  
Mengakhir fungsi `main()`
- Baris 8 : `}`  
Menutup block `main()`

### 5.2 Tipe Data Dalam C

Setiap variabel dalam C memiliki tipe data terkait. Setiap tipe data membutuhkan jumlah memori yang berbeda dan memiliki beberapa operasi khusus yang dapat dilakukan di atasnya.

### 5.2.1 Tipe Data Integer

**Tabel 5.1:** Tipe Data Integer atau bilangan bulat

Type	Ukuran Penyimpanan	Rentang Nilai
char	1 byte	-128-127 atau 0-255
unsigned char	1 byte	0 - 255
signed char	1 byte	-128 - 127
int	2 atau 4 bytes	-32,768 - 32,767
unsigned int	2 or 4 bytes	0 - 65,535 atau 0 - 4,294,967,295
short	2 bytes	-32,768-32,767
unsigned short	2 bytes	0-65,535
long	8 bytes	-9223372036854775808-9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

**Listing 5.2:** Contoh Bilangan Integer

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <limits.h>
4 #include <float.h>
5
6 int main(int argc, char** argv) {
7
8     printf("CHAR_BIT      :   %d\n", CHAR_BIT);
9     printf("CHAR_MAX      :   %d\n", CHAR_MAX);
10    printf("CHAR_MIN      :   %d\n", CHAR_MIN);
11    printf("INT_MAX       :   %d\n", INT_MAX);
12    printf("INT_MIN      :   %d\n", INT_MIN);
13    printf("LONG_MAX     :   %ld\n", (long) LONG_MAX);
14    printf("LONG_MIN     :   %ld\n", (long) LONG_MIN);
15    printf("SCHAR_MAX    :   %d\n", SCHAR_MAX);
16    printf("SCHAR_MIN    :   %d\n", SCHAR_MIN);
17    printf("SHRT_MAX     :   %d\n", SHRT_MAX);
18    printf("SHRT_MIN     :   %d\n", SHRT_MIN);
19    printf("UCHAR_MAX    :   %d\n", UCHAR_MAX);
20    printf("UINT_MAX     :   %u\n", (unsigned int) UINT_MAX);
21    printf("ULONG_MAX    :   %lu\n", (unsigned long) ULONG_MAX);
22    printf("USHRT_MAX    :   %d\n", (unsigned short) USHRT_MAX);
23
24    return 0;
25 }
26

```

⇒ Untuk mencoba program Klik [di sini](#)

### 5.2.2 Floating point atau bilangan pecahan

Type	Ukuran Penyimpanan	Rentang Nilai	Ketelitian
float	4 byte	1.2E-38 - 3.4E+38	6 angka desimal
double	8 byte	2.3E-308 - 1.7E+308	15 angka desimal
long double	10 byte	3.4E-4932 - 1.1E+4932	19 angka desimal



**Listing 5.3:** Contoh Bilangan float atau pecahan

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <limits.h>
4 #include <float.h>
5
6 int main(int argc, char** argv) {
7
8     printf("Storage size for float : %d \n", sizeof(float));
9     printf("FLT_MAX      : %g\n", (float) FLT_MAX);
10    printf("FLT_MIN      : %g\n", (float) FLT_MIN);
11    printf("-FLT_MAX     : %g\n", (float) -FLT_MAX);
12    printf("-FLT_MIN     : %g\n", (float) -FLT_MIN);
13    printf("DBL_MAX      : %g\n", (double) DBL_MAX);
14    printf("DBL_MIN      : %g\n", (double) DBL_MIN);
15    printf("-DBL_MAX     : %g\n", (double) -DBL_MAX);
16    printf("Precision value: %d\n", FLT_DIG );
17
18    return 0;
19 }

```

⇒ Untuk mencoba program Klik [di sini](#)

## 5.3 Variabel Dalam C

- Variabel adalah nama untuk tempat pada area penyimpanan data pada memori agar program dapat di manipulasi. Aturan pemberian nama variabel adalah sebagai berikut :
  - Setiap variabel pada C harus dinyatakan secara khusus untuk menentukan ukuran dan layout dari memory dari variabel.
  - Nama dari variabel terdiri dari huruf angka dan karakter underscore.
  - Nama Variabel harus dimulai dengan huruf atau underscore.

- Deklarasi Variabel Dalam C.  
`type variable_list;`

- Contoh :

```

1 float Luas, Lebar , Tinggi;
2 int  JumlahHari;

```

### 5.3.1 Structure

1. Definisi :

Didalam pemrograman C, structure adalah sekumpulan variabel dengan tipe yang dapat berbeda dibawah satu nama.

2. Deklarasi

```

struct structureName
{
    TipeData Member1;
    TipeData Member2;
    ...
};

```

contoh :

```

1 struct Bangun2D
2 {
3     float panjang;
4     float tinggi;
5     int Jenis;
6 }

```

### 3. Membuat Variable Sturct;

```

struct NamaStruktur
{
    TipeData Member1;
    TipeData Member2;
    TipeData Member3;
};

int main()
{ // Mendeklarasikan variabel struct
    struct NamaStruktur NamaVariabel;
    .
    .
    .
}

```

contoh

```

1 #include <stdio.h>
2 // Mendefinisikan struktur untuk bangun Bangun2D
3 struct Bangun2D
4 {
5     float Luas;
6     float tinggi;
7     float alas;
8     int Jenis;
9     // Jenis merupakan flag
10    // Segitiga : Jenis = 1
11    // Persegi Panjang : Jenis = 2;
12 };
13 int main()
14 {
15     // Mendeklarasikan variabel V1 bertipe struct Bangun2D
16     struct Bangun2D V;
17     // Inisialisasi parameter V sebagai segitiga
18     V.alas = 10;
19     V.tinggi = 5;
20     V.Jenis = 1;
21     // Mengitung Luas berdasar Parameter Jenis
22     if (V.Jenis==1)
23     {
24         V.Luas = 0.5*V.alas*V.tinggi;
25     }
26     if (V.Jenis==2)
27     {
28         V.Luas = V.alas*V.tinggi;
29     }
30     printf("Luas = %f",V.Luas);
31     return 0;
32 }

```

⇒ Untuk mencoba program Klik [di sini](#)

## 5.4 Operator

### 5.4.1 Operator Aritmetika

**Tabel 5.2:** Operator Aritmetika

Operator	Nama	Contoh
+	Penjumlahan	$x + y$
-	Pengurangan	$x - y$
*	Perkalian	$x * y$
/	Pembagian	$x / y$
%	Modulus	$x \% y$
++	Increment	$++x$
--	Decrement	$--x$

Contoh :

```
1 #include <stdio.h>
2
3 main() {
4
5     int a = 21;
6     int b = 10;
7     int c ;
8
9     c = a + b;
10    printf("Line 1 - Value of c is %d\n", c );
11
12    c = a - b;
13    printf("Line 2 - Value of c is %d\n", c );
14
15    c = a * b;
16    printf("Line 3 - Value of c is %d\n", c );
17
18    c = a / b;
19    printf("Line 4 - Value of c is %d\n", c );
20
21    c = a % b;
22    printf("Line 5 - Value of c is %d\n", c );
23
24    c = a++;
25    printf("Line 6 - Value of c is %d\n", c );
26
27    c = a--;
28    printf("Line 7 - Value of c is %d\n", c );
29 }
```

⇒ Untuk mencoba program Klik [di sini](#)

### 5.4.2 Operator Penugasan

**Tabel 5.3:** Operator Penugasan

Operator	Contoh	Sama Seperti
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
-=	x -= 3	x = x - 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

```

1 #include <stdio.h>
2
3 main() {
4
5     int a = 21;
6     int c ;
7
8     c = a;
9     printf("Line 1 - = Operator Example, Value of c = %d\n", c );
10
11    c += a;
12    printf("Line 2 - += Operator Example, Value of c = %d\n", c );
13
14    c -= a;
15    printf("Line 3 - -= Operator Example, Value of c = %d\n", c );
16
17    c *= a;
18    printf("Line 4 - *= Operator Example, Value of c = %d\n", c );
19
20    c /= a;
21    printf("Line 5 - /= Operator Example, Value of c = %d\n", c );
22
23    c = 200;
24    c %= a;
25    printf("Line 6 - %= Operator Example, Value of c = %d\n", c );
26
27    c <<= 2;
28    printf("Line 7 - <<= Operator Example, Value of c = %d\n", c );
29
30    c >>= 2;
31    printf("Line 8 - >>= Operator Example, Value of c = %d\n", c );
32
33    c &= 2;
34    printf("Line 9 - &= Operator Example, Value of c = %d\n", c );
35
36    c ^= 2;
37    printf("Line 10 - ^= Operator Example, Value of c = %d\n", c );
38
39    c |= 2;
40    printf("Line 11 - |= Operator Example, Value of c = %d\n", c );
41 }

```

⇒ Untuk mencoba program Klik [di sini](#)

### 5.4.3 Operator Perbandingan

**Tabel 5.4:** Operator Perbandingan

Operator	Name	Example
==	Sama Dengan	x == y
!=	Tidak Sama Dengan	x != y
>	Lebih Besar	x > y
<	Kurang Dari	x < y
>=	Lebih besar sama atau sama dengan dari	x >= y
<=	kurang atau sama dengan dari	x <= y

- Contoh 1: Mencari Nilai terbesar antara dua buah variabel X=6 dan Y=5

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int X,Y;
6      char B1,B2,B3,B4,B5,B6;
7      X=5;
8      Y=7;
9      B1 = X<Y;
10     printf("(%d < %d) = %d\n",X,Y,B1);
11     B2 = X<=Y;
12     printf("(%d <= %d) = %d\n",X,Y,B2);
13     B3 = X==Y;
14     printf("(%d == %d) = %d\n",X,Y,B3);
15     B4 = X>Y;
16     printf("(%d > %d) = %d\n",X,Y,B4);
17     B5 = X>=Y;
18     printf("(%d >= %d) = %d\n",X,Y,B5);
19     B6 = X!=Y;
20     printf("(%d != %d) = %d\n",X,Y,B6);
21     return 0;
22 }
23

```

⇒ Untuk mencoba program Klik [di sini](#)

```

(5 < 7) = 1
(5 <= 7) = 1
(5 == 7) = 0
(5 > 7) = 0
(5 >= 7) = 0
(5 != 7) = 1

```

**Gambar 5.1:** figure

### 5.4.4 Operator Logika

- menggabungkan lebih dari satu pernyataan.

Operator	Nama	Contoh
&&	AND	$x < 5 \ \&\& \ x < 10$
	OR	$x < 5 \    \ x < 4$
!	NOT	$!(x < 5 \ \&\& \ x < 10)$

- Tabel Kebenaran Logika AND.

Untuk  $I_1$  adalah pernyataan ke satu dan  $I_2$  adalah pernyataan ke dua. Gabungan dari kedua pernyataan dengan operasi AND mempunyai kebenaran sebagai berikut.

No	I1	I2	AND
1	False	False	False
2	False	True	False
3	True	False	False
4	True	True	True

Contoh :

Untuk : W= 5, X = 10, Y = 7 dan Z=12			
No	Pernyataan 1 (I1)	Pernyataan 2 (I2)	I1 && I2
1	W<X =True	Y<Z= True	True
2	W>X=False	Y<Z=True	False
3	W<X=True	Y>Z=False	False
4	W>X=Flase	Y>Z=False	False

```

1  #include <stdio.h>
2
3  int main()
4  {
5      int W, X,Y,Z;
6      char B1,B2,B3,B4;
7      W = 5;
8      X= 10;
9      Y = 7;
10     Z = 12;
11     printf("W=%d,X=%d,Y=%d,Z=%d\n",W,X,Y,Z);
12     //Operasi Logika AND terhadap Pernyataan1 dan pernyataan 2.
13
14     B1 = W<X && Y<Z;
15     printf ("Operasi W<X && Y<Z= %d\n",B1);
16     B2 = W>X && Y<Z;
17     printf ("Operasi W>X && Y<Z= %d\n",B2);
18     B3 = W<X && Y>Z;
19     printf ("Operasi W<X && Y>Z= %d\n",B3);
20     B4 = W>X && Y>Z;
21     printf ("Operasi W>X && Y>Z= %d\n",B4);
22     return 0;
23 }
24

```

⇒ Untuk mencoba program Klik [di sini](#)

```

W=5,X=10,Y=7,Z=12
Operasi W<X && Y<Z= 1
Operasi W>X && Y<Z= 0
Operasi W<X && Y>Z= 0
Operasi W>X && Y>Z= 0

```

Gambar 5.2

- Tabel Kebenaran Logika OR

No	I1	I2	OR
1	False	False	False
2	False	True	True
3	True	False	True
4	True	True	True

Contoh Program :

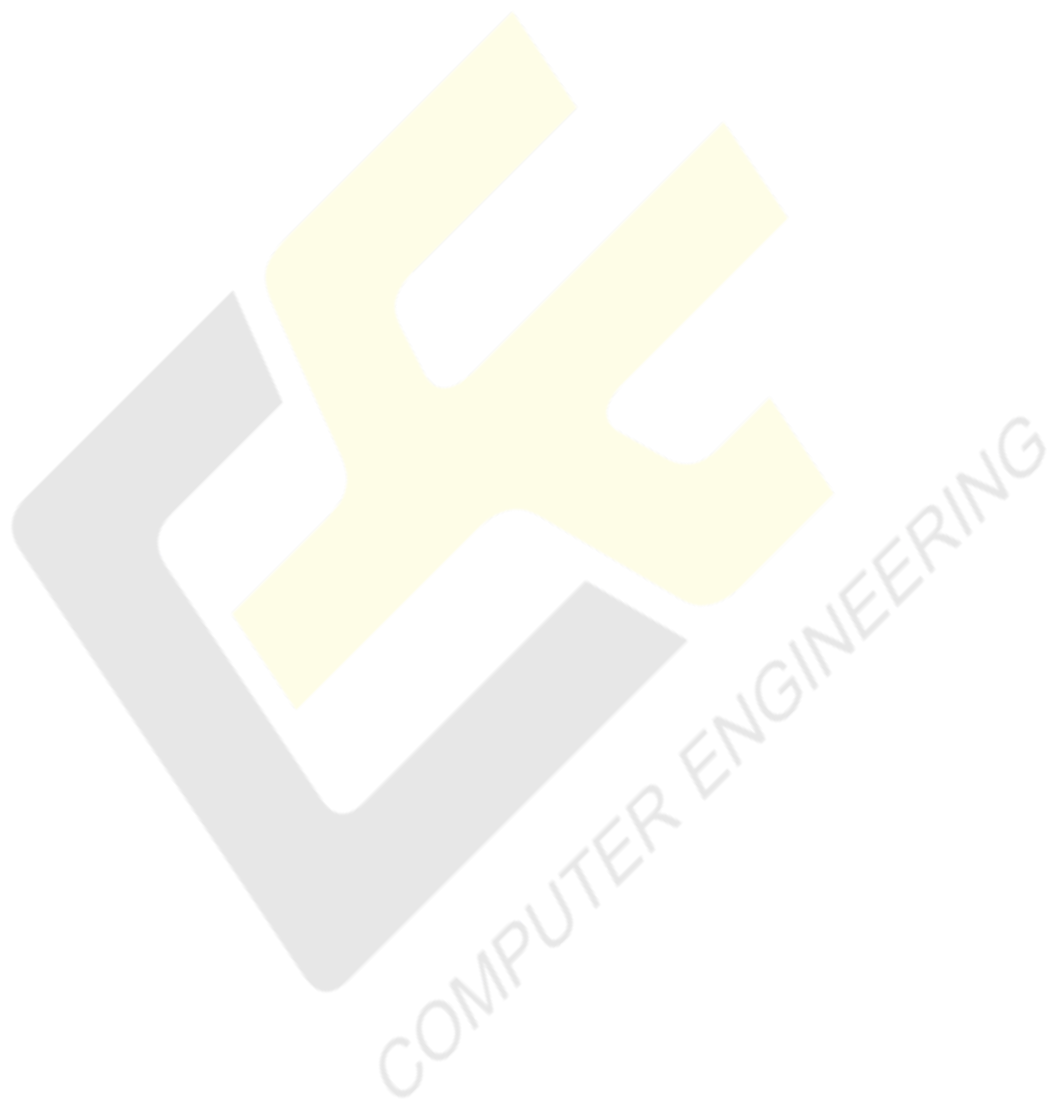
```

1 #include <stdio.h>
2
3 int main()
4 {
5     int W, X,Y,Z;
6     char B1,B2,B3,B4;
7     W = 5;
8     X= 10;
9     Y = 7;
10    Z = 12;
11    printf("\n");
12    printf(" W=%d,X=%d,Y=%d,Z=%d\n",W,X,Y,Z);
13    //Operasi Logika OR terhadap Pernyataan1 dan pernyataan 2.
14    B1 = W<X || Y<Z;
15    printf (" Operasi W<X || Y<Z= %d\n",B1);
16    B2 = W>X || Y<Z;
17    printf (" Operasi W>X || Y<Z= %d\n",B2);
18    B3 = W<X || Y>Z;
19    printf (" Operasi W<X || Y>Z= %d\n",B3);
20    B4 = W>X || Y>Z;
21    printf (" Operasi W>X || Y>Z= %d\n",B4);
22    return 0;
23 }
```

⇒ Untuk mencoba program Klik [di sini](#)

- Tabel Kebenaran Logika NOT

No	I1	Not
1	True	False
2	False	True





## Bab 6

# Fungsi Output dan Input

### 6.1 Format specifier

Format specifiers digunakan untuk mendefinisikan jenis data yang akan dicetak dengan menggunakan fungsi `printf()` atau menerima input dengan menggunakan fungsi `scanf()`.

**Tabel 6.1:** Format specifier

Format Specifier	Jenis Data
%d or %i	int
%f	float
%lf	double
%c	char
%s	Untuk string

#### 6.1.0.1 title

### 6.2 Fungsi printf()

Fungsi `printf` pada C digunakan untuk mencetak string ke output yang dilengkapi dengan format specifier yang dimulai dengan % pada string.

Sintaks :

```
printf(const char *format,v1,v2,...,vn)
```

**Contoh 6.2.1** Mencetak text ke layar.

**Listing 6.1:** Mencetak Tulisan "Pemrograman C Ke layar

```
1  #include <stdio.h>
2  int main()
3  {
4      // Menampilkan tulisan yang terletak diantara tanda petik dua
5      printf("C Programming");
6      return 0;
7  }
8
```

- Seluruh program C harus berisi fungsi `main()` tempat program memulai menjalankan kode.
- Fungsi `printf()` adalah library untuk mengirim output yang telah diformat ke layar. Fungsi `printf()` mencetak string dalam tanda dua tanda petik.
- Untuk menggunakan fungsi `printf()` dalam program harus disertakan file header `stdio.h` dengan menggunakan statement `#include <stdio.h>`.

- Statement `return 0;` dalam fungsi `main()` menunjukkan status "Exit" dari program.

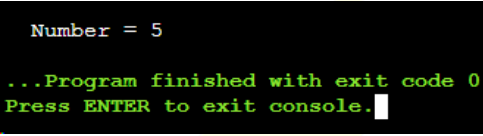
### Contoh 6.2.2 Mencetak bilangan bulat.

```

1  #include <stdio.h>
2  int main()
3  {
4      int testInteger = 5;
5      printf("Number = %d", testInteger);
6      return 0;
7  }
8

```

⇒ Untuk mencoba program Klik [di sini](#)



```

Number = 5
...Program finished with exit code 0
Press ENTER to exit console.

```

Gambar 6.1

Pada contoh ini digunakan format specifier `%d` untuk mencetak tipe data `int`. `%d` pada teks akan digantikan oleh isi dari `testInteger`.

### Contoh 6.2.3 Output bilangan pecahan (float atau double)

- Alas : mempunyai tipe data float
- Tinggi : mempunyai tipe data float
- Luas : mempunyai tipe data float
- Persamaan menghitung luas segitiga:

$$Luas = \frac{1}{2} \times Alas \times Tinnnggi \quad (6.1)$$

```

1  #include <stdio.h>
2
3  int main()
4  {
5      //Mendeklarasikan variabel
6      float Alas;
7      float Tinggi;
8      float Luas;
9      //Inisialisasi variabel
10     Alas = 10;
11     Tinggi = 5;
12     //MEnghitung Luas Segitiga
13     Luas = 0.5*Alas*Tinggi;
14     //Mencetak luas segitiga ke layar
15     printf("Luas = %f",Luas);
16     return 0;
17 }
18
19

```

⇒ Untuk mencoba program Klik [di sini](#)

Penjelasan program :

**Baris 1:** Menggunakan fungsi yang terdapat pada `stdio.h` agar dapat menggunakan fungsi `printf()`

**Baris 3** Deklarasi fungsi utama `main()` untuk memulai program.

**Baris 6-8** Mendeklarasikan variabel `Alas`, `Tinggi` dan `Luas` bertipe data `float` untuk menyimpan data parameter luas segitiga.

**Baris 10 dan 11** Memberi nilai ke Variabel `Alas=10` dan `Tinggi=5`

**Baris 13** Menghitung luas alas sesuai dengan persamaan 6.2

**Baris 15** Mencetak `Luas` ke layar dengan menggunakan perintah `printf`. Untuk mencetak bilangan pecahan bertipe `float` perintah `printf` memerlukan *format specifier*. dalam contoh tersebut digunakan format `%f`.

## 6.3 scanf

Fungsi `scanf(const char *format, ...)` membaca input dengan format.

### 1. Sintaks

```
int scanf(const char *format, ...)
```

### 2. Prameter

format string pada C yang terdiri dari satu atau lebih yang terdiri dari Karakter Whitespace, Karakter Non-whitespace dan Format specifiers.

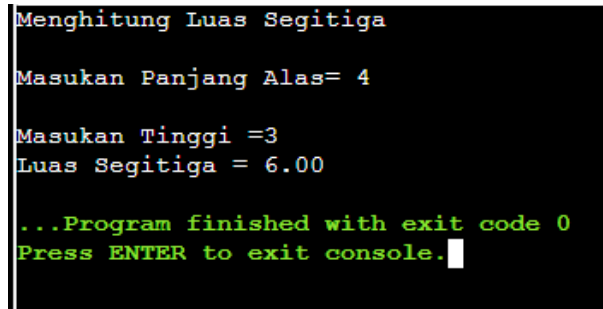
### 3. Return Value

Ketika berhasil maka fungsi mengembalikan jumlah item dari argumen yang berhasil di baca.

#### 6.3.1 Menghitung luas segitiga dengan parameter `Alas` bertipe data dan `Tinggi` yang diinputkan.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float Alas ,Tinggi,Luas;
6
7     printf("Menghitung Luas Segitiga\n");
8     printf("\nMasukan Panjang Alas= ");
9     scanf("%f",&Alas);
10    printf("\nMasukan Tinggi =");
11    scanf("%f",&Tinggi);
12    Luas = 0.5*Alas *Tinggi;
13    printf("Luas Segitiga = %.2f", Luas);
14    return 0;
15 }
16
```

⇒ Untuk mencoba program Klik [di sini](#)



```
Menghitung Luas Segitiga
Masukan Panjang Alas= 4
Masukan Tinggi =3
Luas Segitiga = 6.00
...Program finished with exit code 0
Press ENTER to exit console.
```

Gambar 6.2

**Baris 1** `#include <stdio.h>`

Menggunakan file header `stdio.h`

**Baris 4** `float Alas, Tinggi, Luas;`

Mendeklarasikan variabel `Alas`, `Tinggi` dan `Luas` bertipe data `float`. Tipe data yang digunakan adalah `float` karena input dan output parameter luas segitiga adalah sembarang bilangan pecahan positif.

**Baris 7** `printf("Menghitung Luas Segitiga\n");`

Mencetak informasi aplikasi.

**Baris 8** `printf("\nMasukan Panjang Alas= ");`

Memberikan informasi ke user bahwa parameter yang dimasukan adalah panjang alas.

**Baris 9** `scanf("%f",&Alas);`

**Baris 10** `printf("\nMasukan Tinggi =");`

**Baris 11** `scanf("%f",&Tinggi);`

**Baris 12** `Luas = 0.5*Alas *Tinggi;`

**Baris 13** `printf("Luas Segitiga = %2.2f", Luas);`

**Contoh 6.3.2** Program memasukan Nama dan Alamat email dari keyboard.

Pada contoh ini dipelajari bagaimana cara menginputkan string atau text dari keyboard dan mencetak kelayar. Input dari contoh program ini ada dua yang terdiri dari `sNama` dan `sAlamatEmail`. Oleh karena text berisi banyak karakter maka masing-masing variabel dideklarasikan sebagai kumpulan karakter dengan jumlah karakter untuk `sNama=20` dan `sAlamatEmail=30`.

```
1  #include <stdio.h>
2
3  int main ()
4  {
5      char sNama[20], sAlamatEmail[30];
6
7      printf("Masukan Nama: ");
8      scanf("%19s", sNama);
9
10     printf("Masukan Alamat email : ");
11     scanf("%29s", sAlamatEmail);
12
13     printf("Nama : %s\n", sNama);
14     printf("Alamat Email:%s", sAlamatEmail);
15     return(0);
16 }
17
```

Penjelasan program :

**Contoh 6.3.2** Program menghitung luas segitiga dengan panjang Alas dan Tinggi yang diinputkan melalui keyboard.

## 6.4 Escape Sequence

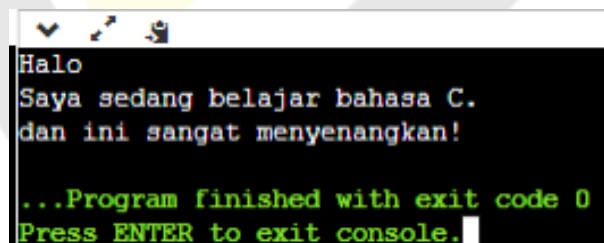
Escape Sequence adalah urutan karakter yang digunakan untuk memformat output dan tidak ditampilkan ketika dicetak ke layar. Setiap karakter mempunyai fungsi tertentu.

Escape sequence	Fungsi	
\a	bell, alarm	
\b	Backspace	
\f	Ganti halaman	
\n	Ganti baris	
\r	Carriage return	
\t	tab horisontal	
\v	tab vertikal	
\'	Petik tunggal	
\"	Petik Ganda	
\?	Tanda tanya	
\\	Backslash	

**Contoh 6.4.1** Mencetak kelayar dan ganti baris menggunakan escape squance \n untuk ganti baris.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Halo \nSaya sedang belajar bahasa C.\ndan ini sangat menyenangkan
6         !");
7     return 0;
8 }
```

⇒ Untuk mencoba program Klik [di sini](#)

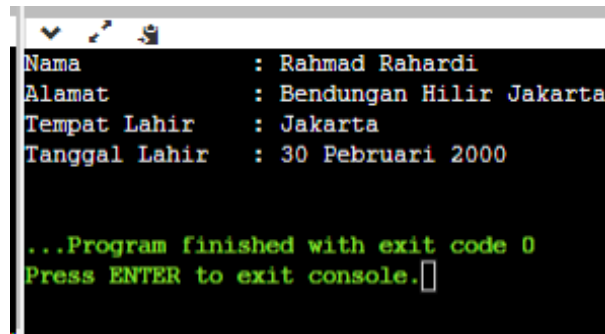


Gambar 6.3

**Contoh 6.4.1** Mencetak kelayar dan ganti baris menggunakan escape squance \t untuk mengatur tab.

```
1 #include <stdio.h>
2 int main(void)
3 {
4     printf("Nama \t\t: Rahmad Rahardi\n");
5     printf("Alamat \t\t: Bendungan Hilir Jakarta\n");
6     printf("Tempat Lahir \t: Jakarta\n");
7     printf("Tanggal Lahir \t: 30 Pebruari 2000\n");
8
9     return (0);
10 }
```

⇒ Untuk mencoba program Klik [di sini](#)



```

Nama      : Rahmad Rahardi
Alamat    : Bendungan Hilir Jakarta
Tempat Lahir : Jakarta
Tanggal Lahir : 30 Pebruari 2000

...Program finished with exit code 0
Press ENTER to exit console.

```

Gambar 6.4

## 6.5 Tugas

**6.5.1** Buatlah program dalam bahasa C untuk menghitung volume tabung dengan parameter yang diinputkan adalah :

- Radius (R)
- Tinggi (T)

dengan persamaan volume tabung (V):

$$V = \pi \times R^2 \times T$$

**6.5.2** Buatlah program dalam bahasa C untuk menghitung dan mencetak Nilai Total dari komponen penilaian matakuliah Dasar Pemrograman Komputer yang diinputkan dengan bobot untuk masing-masing komponen tersebut adalah :

- Nilai Tugas (NT) dengan bobot  $w1 = 20$  persen.
- Nilai Praktikum (NP) dengan bobot  $w2 = 30$  persen.
- Nilai Ujian Tengah (NUT) Semester dengan bobot  $w3 = 25$  persen.
- Nilai Ujian Akhir Semester (NUAS) dengan bobot  $w4 = 25$  persen.

Total Nilai diperoleh dengan cara menjumlahkan Nilai tiap komponen dan bobotnya.

$$Nilai = NT \times w1 + NP \times w2 + NUT \times w3 + NUAS \times w4$$

## Bab 7

# Instruksi Pemilihan Dalam C

### 7.1 Tujuan

### 7.2 Statement if

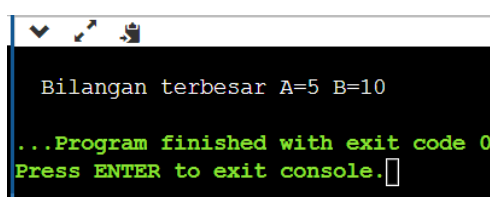
Statement if digunakan untuk menentukan blok kode C yang dijalankan apabila kondisi benar (TRUE),

```
if (Kondisi)
{
    // blok kode yang akan dijalankan ketika kondisi benar.
}
```

```
1  include <stdio.h>
2
3  int main()
4  {
5      //Deklarasi variabel
6      int a,b,c;
7      a= 5;
8      b =10;
9
10     if (a>b)
11     {
12         c =a;
13         a = b;
14         b = c;
15     }
16     //Mencetak Hasil Ke layar
17     printf("Bilangan terbesar A=%d B=%d",a,b);
18     return 0;
19 }
```

**Listing 7.1:** Mengurutkan dua bilangan A dan B dari kecil ke besar

⇒ Untuk mencoba program Klik [di sini](#)



```
Bilangan terbesar A=5 B=10
...Program finished with exit code 0
Press ENTER to exit console.
```

Gambar 7.1

### 7.3 Statement else

Statement else digunakan untuk menentukan blok kode yang di jalankan apabila kondisi salah.

```
if (Kondisi)
{
    // Blok kode yang dijalankan apabila kondisi benar.
} else
{
    // Blok kode yang dijalankan apabila kondisi salah.
}
```

Contoh :

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int Jam;
6     Jam = 5;
7     if (Jam <10)
8     {
9         printf("Selamat pagi");
10    }else
11    {
12        printf("Selamat siang");
13    }
14    return 0;
15 }
16
17
```

⇒ Untuk mencoba program Klik [di sini](#)

### 7.4 Statement else if

Pernyataan else if digunakan untuk menentukan kondisi baru jika kondisi pertama salah.

```
if (Kondisi1)
{
    // Blok kode yang dijalankan apabila kondisi1 benar.
} else if (Kondisi2) {
    // Blok kode yang dijalankan apabila kondisi1 salah dan
    Kondisi 2 benar.
} else
{
    // Blok kode yang dijalankan apabila kondisi1 salah dan
    Kondisi 2 salah.
}
```

contoh :

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int Jam;
6     Jam = 16;
7     if (Jam <10)
```



```

8  {
9      printf("\n Selamat pagi");
10 }else if(Jam<15)
11 {
12     printf("\n Selamat siang");
13 }else
14 {
15     printf("\n Selamat Sore");
16 }
17 return 0;
18 }

```

⇒ Untuk mencoba program Klik [di sini](#)

## 7.5 Contoh Soal

**Contoh 7.5.1** Menentukan apakah suatu bilangan yang diinputkan merupakan bilangan genap atau ganjil.

- Pengertian bilangan genap :

Bilangan genap adalah bilangan bulat yang kalau dibagi dengan 2 mempunyai sisa 0.

- Pengertian bilangan ganjil.

Bilangan ganjil adalah bilangan bulat yang kalau dibagi dengan 2 mempunyai sisa 1.

Didalam bahasa C untuk mencari sisa bagi dinamakan sebagai modulus dengan lambang %. Contoh :

A = 4 % 2;  
B = 5 % 1;

Maka hasilnya adalah A= 0 dan B=1.

**Contoh 7.5.2** Menghitung biaya pemakaian air secara progresif.

**Tabel 7.1:** Tarif Progresif Air PDAM

No	Pemakaian Air ( $m^3$ )	Harga
1	0 – 10	Rp 350/m <sup>3</sup>
2	11 – 20	Rp 600/m <sup>3</sup>
3	21 – 30	Rp 900/m <sup>3</sup>
4	30	Rp 1.800/m <sup>3</sup>

- Pemakaian Air = 25
- Biaya progresif
  - Biaya progresif pertama  
 $Biaya1 = 10 \times 350 = 3500$
  - Biaya progresif kedua  
 $Biaya2 = 10 \times 600 = 6000$
  - Biaya progresif ke tiga  
 $Biaya3 = 5 \times 900 = 4500$

- Biaya progresif ke empat

$$Biaya4 = 0 \times 1800 = 0$$

- $Total = Biaya1 + Biaya2 + Biaya3 + Biaya4$

$$Total = 3500 + 6000 + 4500 + 0$$

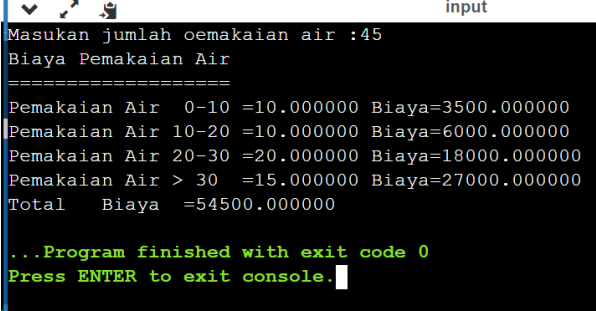
$$Total = 14000$$

```

1 #include <stdio.h>
2
3 int main()
4 {
5     float JumlahPemakaianAir, TotalBiaya;
6     float PemakaianAir1, PemakaianAir2;
7     float PemakaianAir3, PemakaianAir4;
8     float Biaya1, Biaya2, Biaya3, Biaya4;
9
10    printf("Masukan jumlah pemakaian air :");
11    scanf("%f", &JumlahPemakaianAir);
12
13    PemakaianAir1 = 0;
14    PemakaianAir2 = 0;
15    PemakaianAir3 = 0;
16    PemakaianAir4 = 0;
17
18    if (JumlahPemakaianAir <= 10)
19    {
20        PemakaianAir1 = JumlahPemakaianAir;
21    } else if (JumlahPemakaianAir <= 20)
22    {
23        PemakaianAir1 = 10;
24        PemakaianAir2 = JumlahPemakaianAir - 10;
25    } else if (JumlahPemakaianAir <= 30)
26    {
27        PemakaianAir1 = 10;
28        PemakaianAir2 = 10;
29        PemakaianAir3 = JumlahPemakaianAir - 20;
30    } else
31    {
32        PemakaianAir1 = 10;
33        PemakaianAir2 = 10;
34        PemakaianAir3 = 20;
35        PemakaianAir4 = JumlahPemakaianAir - 30;
36    }
37    Biaya1 = PemakaianAir1 * 350;
38    Biaya2 = PemakaianAir2 * 600;
39    Biaya3 = PemakaianAir3 * 900;
40    Biaya4 = PemakaianAir4 * 1800;
41    TotalBiaya = Biaya1 + Biaya2 + Biaya3 + Biaya4;
42    printf("Biaya Pemakaian Air \n");
43    printf("=====\n");
44    printf("Pemakaian Air 0-10 =%f Biaya=%f\n", PemakaianAir1, Biaya1);
45    printf("Pemakaian Air 10-20 =%f Biaya=%f\n", PemakaianAir2, Biaya2);
46    printf("Pemakaian Air 20-30 =%f Biaya=%f\n", PemakaianAir3, Biaya3);
47    printf("Pemakaian Air > 30 =%f Biaya=%f\n", PemakaianAir4, Biaya4);
48    printf("Total    Biaya    =%f", TotalBiaya);
49    return 0;
50 }

```

⇒ Untuk mencoba program Klik [di sini](#)



```
input
Masukan jumlah pemakaian air :45
Biaya Pemakaian Air
=====
Pemakaian Air 0-10 =10.000000 Biaya=3500.000000
Pemakaian Air 10-20 =10.000000 Biaya=6000.000000
Pemakaian Air 20-30 =20.000000 Biaya=18000.000000
Pemakaian Air > 30 =15.000000 Biaya=27000.000000
Total Biaya =54500.000000

...Program finished with exit code 0
Press ENTER to exit console.
```

Gambar 7.2

## 7.6 Latihan Soal

- 7.6.1** Buatlah program untuk menentukan hari dari tanggal yang di inputkan untuk bulan Juli 2022 apabila tanggal 1 Juli 2022 adalah hari Jumat.



## Bab 8

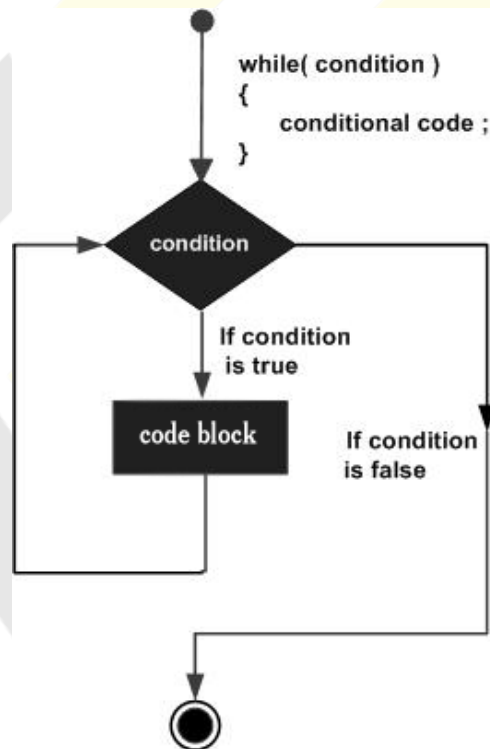
# Operasi Pengulangan

### 8.1 Loop atau pengulangan

#### 8.1.1 While Loop

Pengulangan `while` akan mengulang blok kode selama kondisi yang telah ditentukan BENAR.

1. Diagram alir dari Loop While adalah



Gambar 8.1

2. Sintaks.

```
while (kondisi)
{
    // blok kode yang dijalankan
}
```

Blok kode dapat berisi statement tunggal atau banyak. Kondisi adalah ekspresi dan benar apabila bernilai bukan 0. Pengulangan akan dilakukan selama kondisi `true`.

Ketika kondisi menjadi `FALSE` maka program akan keluar dari loop.

## 3. Contoh :

```

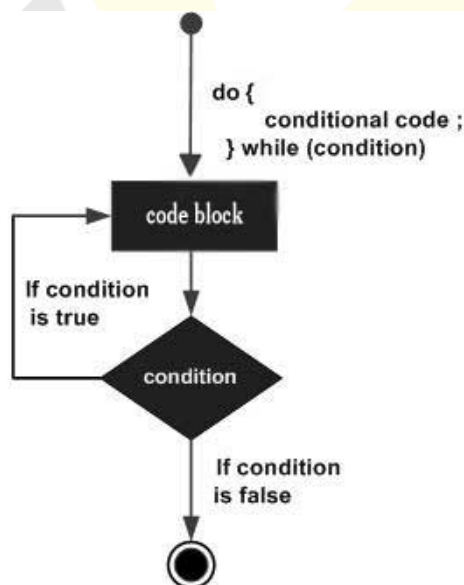
1  #include <stdio.h>
2
3  int main ()
4  {
5      /* definisi variabel lokal*/
6      int a = 5;
7
8      /* Eksekusi while loop */
9      while( a < 20 ) {
10         printf("Nilai dari a a: %d\n", a);
11         a++;
12     }
13
14     return 0;
15 }
16

```

## 8.1.2 Do..while Loop

Pengulangan dengan The `do/while` adalah varian dari pengulangan `while`. Pengulangan ini akan menjalankan kode blok sedikitnya sekali sebelum kondisi diuji apakah benar.

## 1. Diagram Alir



Gambar 8.2

## 2. Sintaks

```

do
{
    //Kode blok yang akan dijalankan
}while(Kondisi)

```

## 3. Contoh.

```

1  #include <stdio.h>
2
3  int main () {
4

```

```

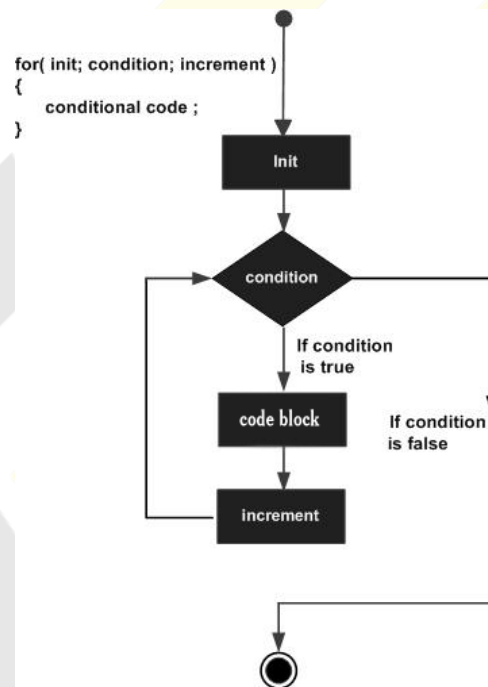
5 // Definisi variabel
6 int a = 10;
7
8 //Eksekusi loop
9 do {
10     printf("Nikai a: %d\n", a);
11     a = a + 1;
12 }while(a< 20 );
13
14 return 0;
15 }
16

```

### 8.1.3 For Loop

Pengulangan dengan `for` digunakan ketika jumlah pengulangan terhadap kode blok diketahui secara pasti.

#### 1. Diagram Alir



Gambar 8.3

#### 2. Sintaks.

```

for ( init; kondisi; increment )
{
    statement;
}

```

- Langkah ini dilakukan hanya sekali untuk menginisialisasi variabel kontrol.
- Langkah selanjutnya adalah evaluasi kondisi, jika TRUE maka isi dari loop di kerjakan dan apabila FALSE maka akan melompat ke statement setelah loop for.
- Setelah blok statement dari loop for di jalankan maka kontrol dari flow akan meloncat ke stement icrement. Statement ini di kemudian mengupdate variabel kontrol.
- Stelah itu apabila hasil evaluasi kondisi TRUE maka loop akan kembali akan menjalankan blok statement loop for.

## 3. Contoh :

```

1 #include <stdio.h>
2
3 int main ()
4 {
5     int a;
6
7     //Eksekusi for loop
8     for( a = 10; a < 20; a = a + 2 )
9     {
10        printf("nilai a: %d\n", a);
11    }
12    return 0;
13 }

```

Penjelasan baris demi baris adalah sebagai berikut

- Baris 1 `#include <stdio.h>`  
Deklarasi file header
- Baris 3 `int main()`  
Memulai fungsi utama
- Baris 4 `{`
- Baris 5 `int a`  
Mendeklarasikan variabel kontrol
- Baris 7 `for( a = 10; a < 20; a = a + 2 )`  
Mengeksekusi blok loop `for` dengan menggunakan tiga statement
- i. `a = 10`  
Menginisialisasi variabel kontrol `a` dengan nilai 10
  - ii. `a < 20`  
Menentukan kondisi untuk mengulang loop.
  - iii. `a = a + 2`  
increment `a` setiap mengulang loop
- Baris 9 `{`  
Memulai blok loop `for`
- Baris 10 `printf("nilai a: %d\n", a);`  
badan blok loop `for` mencetak nilai `a` ke layar
- Baris 11 `}`  
Mengakhiri blok loop `for`
- Baris 12 `return 0;`  
Mengembalikan nilai ke fungsi `main`
- Baris 13 `}`  
Mengakhiri fungsi `main`

## 8.2 break

Statement `break` digunakan untuk keluar dari loop.

**Contoh 8.2.1** Contoh penggunaan `break` untuk keluar dari loop `for` ketika nilai `i` telah mencapai 4

```

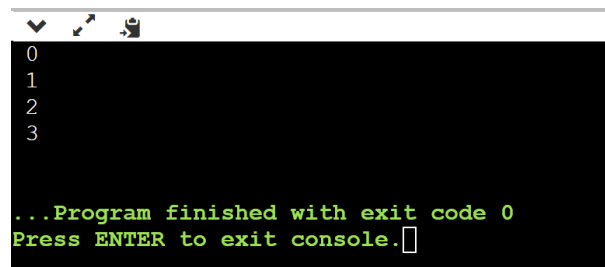
1 #include <stdio.h>
2
3 int main()
4 {

```



```
5  int i;
6  for (i = 0; i < 10; i++)
7  {
8      if (i == 4)
9      {
10         break;
11     }
12     printf("%d\n", i);
13 }
14 }
```

⇒ Untuk mencoba program Klik [di sini](#)



```
0
1
2
3
...Program finished with exit code 0
Press ENTER to exit console.
```

Gambar 8.4

Penjelasan program adalah sebagai berikut.

**Baris 5 :** Loop for akan melakukan iterasi dari  $i=0$  sampai  $i=9$

**Baris 6 :** Ketika  $i$  mencapai 4 maka pengulangan akan diputus dan akan langsung mengeksekusi baris 14

**Output :** Hasil output seperti diperlihatkan pada Gambar 8.4. Berupa deret angka yang menurun dari 0 sampai dengan 3. Nilai  $i=4$  tidak dicetak karena perintah **break** dilaksanakan sebelum fungsi **printf** pada baris 12.

## 8.3 Loop Bersarang

### 8.4 Nested Loop

Pemrograman C memungkinkan untuk menggunakan satu loop di dalam loop lain.

#### 8.4.1 for loop bersarang

Sintaks :

```
for ( init; condition; increment ) {

    for ( init; condition; increment ) {
        statement;
    }
    statement;
}
```

## 8.5 Loop bersarang while loop

```
while(condition) {
    while(condition) {
        statement(s);
    }
    statement(s);
}
```

## 8.6 Contoh Soal

### 1. Bilangan faktorial.

Faktorial bilangan asli  $n$  adalah perkalian semua bilangan asli yang kurang atau sama dengan  $n$ . Faktorial dilambangkan dengan tanda  $!$ . Jadi jika  $n!$ , maka dibaca  $n$  faktorial

$$n! = 1 \times 2 \times 3 \times 4 \times \dots \times (n-1) \times n$$

Contoh :

$$\begin{aligned} 1! &= 1 \\ 2! &= 1 \times 2 = 2 \\ 3! &= 1 \times 2 \times 3 = 6 \\ 4! &= 1 \times 2 \times 3 \times 4 = 24 \end{aligned}$$

```
1 #include <stdio.h>
2 int main()
3 {
4     int num,i;
5     long int fact;
6
7     printf("Masukan bilangan bulat : ");
8     scanf("%d",&num);
9
10    /*perkalian bilangan faktorial mulai dari 1*/
11    fact=1;
12    for(i=1; i<=num; i=i+1)
13    {
14        fact=fact*i;
15    }
16    printf("\nFaktorial dari %d adalah = %ld",num,fact);
17
18    return 0;
19 }
```

⇒ Untuk mencoba program Klik [di sini](#)

### 2. Dengan menggunakan loop for bersarang buatlah program dalam bahasa C untuk menampilkan perkalian dari 1 sampai dengan 100 sebagai berikut.

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
.
.
.
```

10 \* 8 = 80  
10 \* 9 = 90  
10 \* 10 = 100

Jawab :

```
1 #include <stdio.h>
2 int main(void)
3 {
4     for (int i = 1; i <= 10; i++) {
5         for (int j = 1; j <= 10; j++) {
6             printf("%i * %i = %i\n", i, j, j*i);
7         }
8         printf("\n");
9     }
10    return 0;
11 }
```

⇒ Untuk mencoba program Klik [di sini](#)

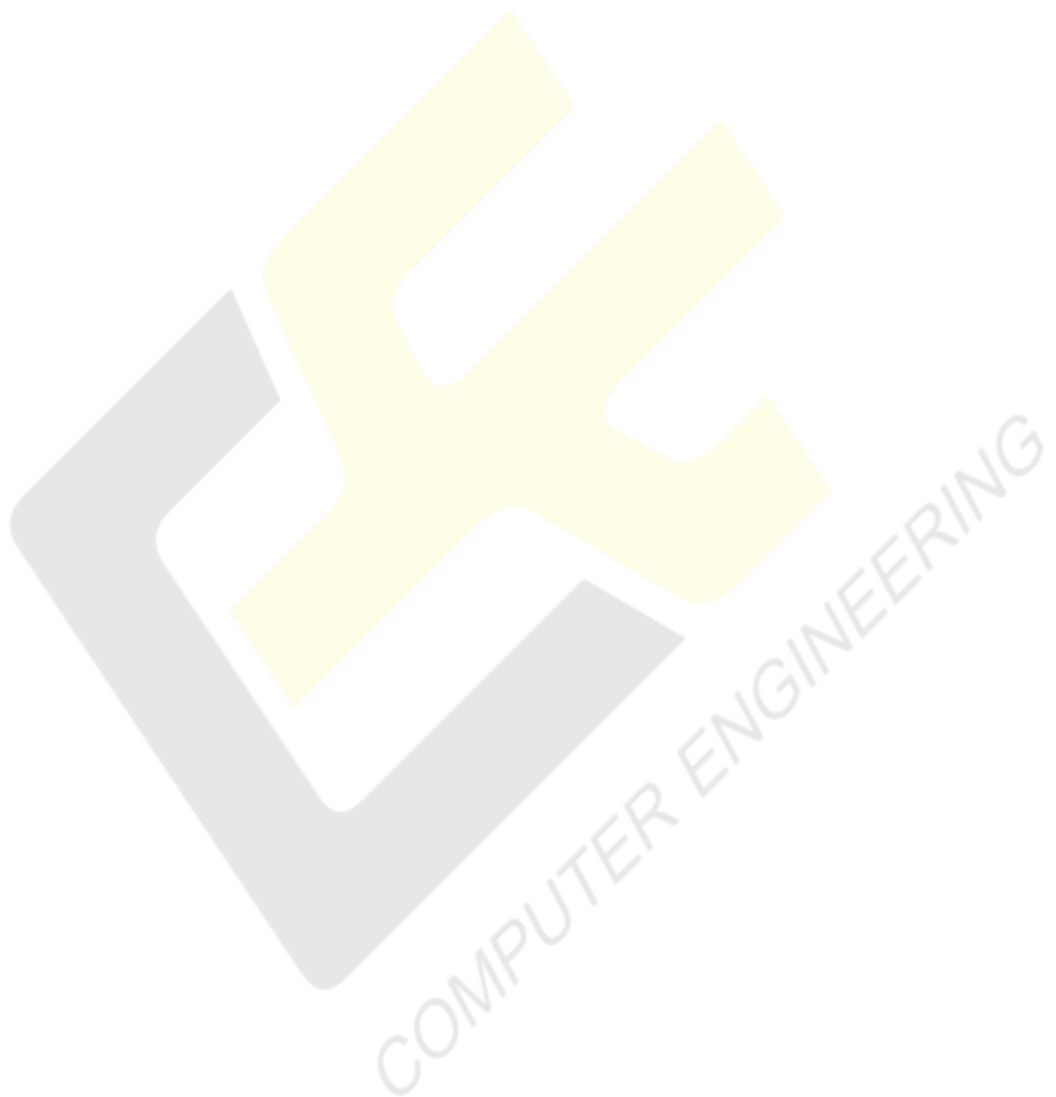
## 8.7 Latihan Soal

- Mencetak bilangan genap dengan menggunakan while.

## 8.8 Latihan

1. Buatlah algoritma, flowchart, dan program untuk mencetak bilangan genap dengan menggunakan `do_while`.  
Contoh :  
Input : Jumlah bilangan Genap yang akan dicetak (N) = 10  
Output : 2, 4, 6, 8, 10, 12, 14, 16, 18, 20
2. Buatlah Algoritma, flowchart, dan program untuk menampilkan deret fibonacci sejumlah n.  
Contoh :  
Input : Jumlah deret fibonacci yang akan dicetak (N) = 10  
Output : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
3. Buatlah algoritma, flowchart, dan program untuk mencetak bilangan prima hingga n angka menggunakan `while`. Contoh :  
Input : Jumlah bilangan Prima yang akan dicetak (N) = 5  
Output : 2, 3, 5, 7, 11
4. Buatlah Algoritma, flowchart, dan program untuk menampilkan segitiga pascal sejumlah n baris menggunakan `for`. Contoh :  
Input : Jumlah Baris segitiga Pascal (N) = 5  
Output :

				1						
			1		1					
		1		2		1				
	1		3		3		1			
	1		4		6		4		1	
1		5		10		10		5		1



# Bab 9

## Array

So far we have used only single variable name for storing one data item.

### 9.1 Konsep Dasar

Banyak aplikasi membutuhkan banyak data dengan karakteristik yang sama. Contoh :

**Tabel 9.1:** Data Suhu harian berdasarkan waktu dari jam 06:00 sampai 12:00

Waktu(Jam)	06:00	08:00	10:00	12:00	14:00	16:00	18:00
Suhu	25°C	28°C	29°C	30°C	29°C	28°C	26°C

Tabel 9.1 berisi data suhu harian yang diukur setiap dua jam mulai jam 06 : 00 sampai dengan jam 16 : 00. Sampai saat ini kita masih mempelajari bagaimana menyimpan satu data dalam satu variabel.

Apabila kita akan membuat program dalam bahasa C untuk mencari suhu tertinggi berdasarkan data tersebut maka kita perlu mendeklarasikan tujuh buah variabel untuk menampung setiap suhu pada tabel tersebut dan membandingkan ketujuh variabel tersebut untuk mencari variabel yang menampung suhu tertinggi.

Sebagai contoh :

```
1 #include <stdio.h>
2
3 int main()
4 {
5     float v1,v2,v3,v4,v5,v6,v7 ;
6     float Tertinggi;
7     v1 = 25; v2 =28; v3=29; v4=30;
8     v5= 29; v6=28; v7=26;
9     Tertinggi = v1;
10    if (v1>Tertinggi)
11    { Tertinggi = v1;
12    }
13    if (v2>Tertinggi)
14    { Tertinggi = v2;
15    }
16    if (v3>Tertinggi)
17    {Tertinggi = v3;
18    }
19    if (v4>Tertinggi)
20    {Tertinggi = v4;
21    }
22    if (v5>Tertinggi)
23    {Tertinggi = v5;
24    }
25    if (v6>Tertinggi)
26    {Tertinggi = v6;
```

```

27 }
28 if (v7>Tertinggi)
29 {Tertinggi = v7;
30 }
31 printf("Suhu tertinggi =%f",Tertinggi);
32 return 0;
33 }

```

Listing 9.1: Menari suhu tertinggi harian

⇒ Untuk mencoba program Klik [di sini](#)

## 9.2 Variabel Array

Array atau biasa disebut larik adalah koleksi data dimana setiap elemen mempunyai nama yang sama dan bertipe sama. Setiap elemen diakses berdasarkan indeks elemennya

Dari contoh sebelumnya untuk menyimpan 7 data suhu kita harus mendekklarasikan 7 variabel :

v1,v2,v3,v4,v5,v6,v7

Dengan demikian apabila di perlukan variabel yang digunakan untuk menyimpan data sejenis dengan jumlah sebanyak  $N$ , maka perlu dideklarasikan variabel dengan jumlah  $N$ .

Dengan menggunakan variabel array maka cukup mendeklarasikan satu variabel yang dapat menampung data yang banyak yang disebut sebagai array.

float v[7]

Dengan menggunakan variabel array maka program 9.1 akan dapat mudah diolah menjadi

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int indeks;
6     float v[10] ;
7     float Tertinggi;
8     v[1] = 25; v[2] =28; v[3]=29; v[4]=30;
9     v[5]= 29; v[6]=28; v[7]=26;
10    Tertinggi = v[1];
11    for (indeks = 1;indeks<7;indeks=indeks+1)
12    { if (v[indeks]>Tertinggi)
13        {
14            Tertinggi = v[indeks];
15        }
16    }
17    printf("Suhu tertinggi =%f",Tertinggi);
18    return 0;
19 }

```

⇒ Untuk mencoba program Klik [di sini](#)

## 9.3 Array Satu Dimensi

### 9.3.1 Deklarasi Array Satu Dimensi

Variabel array dimensi satu dideklarasikan dengan menentukan jenis elemen dan jumlah elemen yang di perlukan oleh array.

Sintaks :

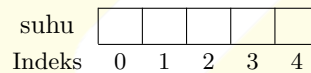
```
TipeData NamaVariabel [UkuranArray ];
```

1. **TipeData.**  
Jenis elemen data elemen array :float,int,char dsb
2. **NamaVariabel**  
Namariabel mengikuti aturan pemberian nama variabel,
3. **UkuranArray**  
konstanta integer lebih besar dari 0.

1. Contoh :Mendeklarasikan variabel array dengan nama **suhu** , bertipe data **float** dengan jumlah elemen 5.

```
float suhu[5];
```

hasil deklarasi suhu tersebut akan menyediakan lima buah tempat pada memory seperti yang diperlihatkan pada Gambar 9.1



**Gambar 9.1:** Hasil deklarasi variabel array suhu dengan jumlah elemen 5

Dari contoh tersebut maka

- indeks array dimulai dari 0
- indeks terakhir mempunyai nilai satu lebih kecil dari ukuran array.  
Dengan rentang indeks mulai dari 0 sampai 4. .
- Setiap elemen array berisi data dengan jenis **float**.

### 9.3.2 Inisialisasi Array Satu Dimensi

Inisialisasi array ketika deklarasi mengikuti aturan berikut

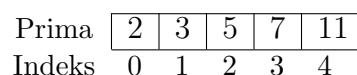
1. Sintaks : .

```
JenisData NamaArray[JumlahElemen] = {daftar dengan jumlah < JumlahElemen };
```

2. Contoh :

- (a) Inisialisasi variabel prima untuk menyimpan 5 bilangan prima pertama.

```
int prima[5]={2,3,5,7,11}
```



**Gambar 9.2:** Inisialisasi array satu dimensi

Hasil dari inisialisasi tersebut adalah :

- 2 disimpan di **prima[0]**
- 3 disimpan di **prima[1]**

- 5 disimpan di `prima[2]`
- 6 disimpan di `prima[3]`
- 11 disimpan di `prima[4]`

(b) Sehingga program 9.1 dapat diubah menjadi :

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int indeks;
6     float Tertinggi;
7     // inisialisasi array 1 dimensi
8     float v[7]={25,28,29,30,29,28,26} ;
9
10    Tertinggi = v[1];
11    for (indek = 1;indek<7;indek=indek+1)
12    { if (v[indek]>Tertinggi)
13      {   Tertinggi = v[indek];
14      }
15    }
16    printf("Suhu tertinggi =%f",Tertinggi);
17    return 0;
18 }
```

### 9.3.3 Mengakses Elemen Array satu dimensi

Mengakses elemen array dilakukan dengan memberi indeks variabel array.

1. Sintaks :

```
NamaArray[Indeks]
```

2. Contoh :

(a) contoh :seperti pada Program ?? baris ke 10.

```
Tertinggi = v[1];
```

atau seperti pada baris ke 13.

```
Tertinggi = v[indeks];
```

(b) Contoh :

```
float Tertinggi;
float Suhu[5]={25,25,26,27,28,25};
Tertinggi = Suhu[3];
```

### 9.3.4 Contoh Soal

9.3.4.1 Diketahui himpunan bilangan bulat :

$$v = \{1, 4, 7, 3, 2, 1\} \quad (9.1)$$

(a) Dengan menggunakan persamaan 9.2 hitunglah jumlah seluruh bilangan dalam himpunan  $v$  tersebut.

$$Tot = \sum_{i=0}^{N-1} v_i \quad (9.2)$$

Untuk  $N = 6$  adalah jumlah data yang disimpan pada  $v$ ,  $Tot$  adalah jumlah total.



(b) Hitunglah rata-rata

$$Rat = \frac{1}{N} \sum_{i=0}^{N-1} v_i \quad (9.3)$$

Dengan *Rat* adalah rata-rata.

Jawab :

(a) Menghitung Jumlah dalam array

```

1 #include <stdio.h>
2
3 int main()
4 {
5     float v[6]={1,4,7,3,2,1};
6     float Tot;
7     int N=6;
8     for (int i=0;i<N;i++)
9     {
10         Tot = Tot + v[i];
11     }
12     printf("Jumlah = %2.2f",Tot);
13     return 0;
14 }
```

⇒ Untuk mencoba program Klik [di sini](#)

(b) Program Rata- rata

```

1 #include <stdio.h>
2
3 int main()
4 {
5     float v[6]={1,4,7,3,2,1};
6     float Rat;
7     int N=6;
8     for (int i=0;i<N;i++)
9     {
10         Rat = Rat + v[i];
11     }
12     Rat =Rat /N;
13     printf("Rat = %2.2f",Rat);
14     return 0;
15 }
```

⇒ Untuk mencoba program Klik [di sini](#)

9.3.4.2 Buatlah program dalam bahasa C untuk menyimpan data yang diinputkan

```

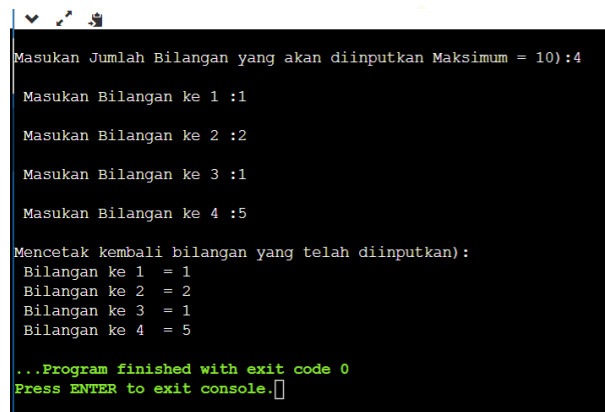
1 #include <stdio.h>
2
3 int main()
4 {
5     int Bilangan[20];
6     int i,JumlahBilangan;
7     int BilanganInput;
8     do
9     {
10         printf("\nMasukan Jumlah Bilangan yang akan diinputkan Maksimum = 10):"
11         );
12         scanf("%d",&JumlahBilangan);
13     }while(JumlahBilangan>10|| JumlahBilangan<1);
14     for (i = 0;i<JumlahBilangan;i=i+1)
15     {
16         printf("\n Masukan Bilangan ke %d :",i+1);
```

```

17     scanf("%d",&BilanganInput);
18     Bilangan[i] = BilanganInput;
19 }
20
21 printf("\nMencetak kembali bilangan yang telah diinputkan:");
22 for (i = 0;i<JumlahBilangan;i=i+1)
23 {
24     printf("\n Bilangan ke %d = %d",i+1,Bilangan[i]);
25 }
26 return 0;
27 }
28

```

⇒ Untuk mencoba program Klik [di sini](#)



```

Masukan Jumlah Bilangan yang akan diinputkan Maksimum = 10):4
Masukan Bilangan ke 1 :1
Masukan Bilangan ke 2 :2
Masukan Bilangan ke 3 :1
Masukan Bilangan ke 4 :5
Mencetak kembali bilangan yang telah diinputkan:
Bilangan ke 1 = 1
Bilangan ke 2 = 2
Bilangan ke 3 = 1
Bilangan ke 4 = 5
...Program finished with exit code 0
Press ENTER to exit console.

```

Gambar 9.3

### 9.3.5 Latihan Soal

9.3.5.1 adf adf adf adf adf dasf

## 9.4 Array Multidimensi

### 1. Deklarasi

```
data_type array_name[size1][size2]....[sizeN];
```

- **data\_type**: Jenis data yang disimpan dalam array.
- **array\_name**: Nama array
- **size1 , size2,... ,sizeN**: Dimensi array.

```

1 float Kecepatan[3][4][5];
2 int D[10][3];

```

## 9.5 Array Dua Dimensi

### 1. Deklarasi Array Dua dimensi dengan dimensi **size1** dan **size2**

```
type name[size1][size2];
```

Contoh :

```
1 float Suhu[7][7];
```

## 2. Inisialisasi Array Dua Dimensi

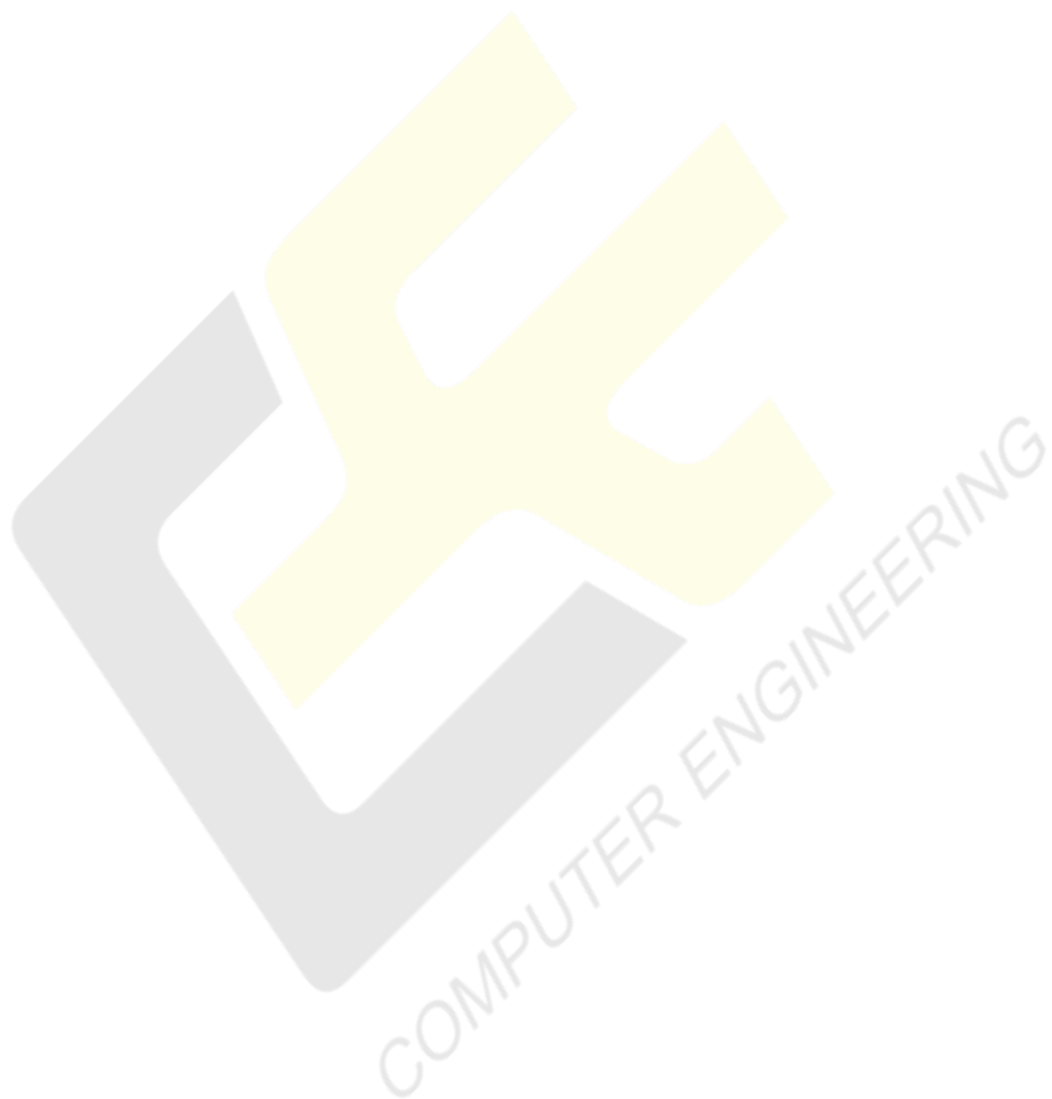
**Tabel 9.2:** Daftar suhu harian selama satu minggu

	Waktu						
	06:00	08:00	10:00	12:00	14:00	16:00	18:00
Senin	25	25	26	30	28	27	26
Selasa	25	25	26	30	28	27	26
Rabu	24	25	26	30	28	27	25
Kamis	26	26	26	29	28	26	26
Jumat	23	25	28	30	29	27	26
Sabtu	25	25	26	30	28	27	27
Minggu	25	25	28	29	29	27	26

```

1 #include <stdio.h>
2
3 int main()
4 {
5     int i,j;
6     //deklarasi dan inisialisasi variabel array 2d
7     // dengan dimensi 7 x 7
8     float Suhu[7][7]=
9     {
10         {25, 25, 26, 30, 28, 27, 26},
11         {25, 25, 26, 30, 28, 27, 26},
12         {24, 25, 26, 30, 28, 27, 25},
13         {26, 26, 26, 29, 28, 26, 26},
14         {23, 25, 28, 30, 29, 27, 26},
15         {25, 25, 26, 30, 28, 27, 27},
16         {25, 25, 28, 29, 29, 27, 26}
17     };
18     for(i=0;i<7;i++)
19     {
20         for (j=0;j<7;j++)
21         {
22             printf("%f ",Suhu[i][j]);
23         }
24         printf("\n");
25     }
26     return 0;
27 }
```

⇒ Untuk mencoba program Klik [di sini](#)



## Bab 10

# Aplikasi Array

### 10.1 Buble Sort

### 10.2 Operasi Matriks

#### 10.2.1 Perkalian Dua Buah Matriks

diketahui dua matriks

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{12} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M2} & a_{M2} & \cdots & a_{MK} \end{bmatrix} \quad (10.1)$$

dan

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{12} & b_{22} & \cdots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K2} & b_{K2} & \cdots & b_{KN} \end{bmatrix} \quad (10.2)$$

dengan

- Untuk matriks  $A$  : Jumlah baris =  $M$  dan Jumlah Kolom =  $K$
- Untuk matriks  $B$  : Jumlah baris  $K$  dan Jumlah Kolom =  $N$

untuk perkalian antara  $A$  dan  $B$

$$C = A * B \quad (10.3)$$

maka diperoleh matriks

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1N} \\ c_{12} & c_{22} & \cdots & c_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M2} & c_{M2} & \cdots & c_{MN} \end{bmatrix} \quad (10.4)$$

dimana

$$\begin{aligned} c_{11} &= a_{11}b_{11} + a_{12}b_{21} + \cdots + a_{1K}b_{K1} \\ c_{12} &= a_{11}b_{12} + a_{12}b_{22} + \cdots + a_{1K}b_{K2} \\ c_{13} &= a_{11}b_{13} + a_{12}b_{23} + \cdots + a_{1K}b_{K3} \\ &\vdots \\ c_{MN} &= a_{M1}b_{11} + a_{M2}b_{22} + \cdots + a_{MK}b_{KN} \end{aligned} \quad (10.5)$$

atau

$$c_{ij} = \sum_{k=1}^M a_{ik} b_{kj} \quad (10.6)$$

implementasi program

```

1 #include <stdio.h>
2
3 int main()
4 {
5     float A[10][10], B[10][10], C[10][10];
6     int N, M, K, i, j, k;
7     float v;
8     printf("\n Jumlah Baris Matriks A (Maks 10) : M = ");
9     scanf("%d", &M);
10    printf("\n Jumlah Kolom Matriks A dan Baris Matriks B (Maks 10) : K = ");
11    scanf("%d", &K);
12    printf("\n Jumlah Kolom Matriks B (Maks 10) : N = ");
13    scanf("%d", &N);
14    //Input Matriks B
15    for (i=0; i<M; i++)
16    {
17        for (j=0; j<K; j++)
18        {
19            printf("\nInput A(%d,%d)= ", i, j);
20            scanf("%f", &v);
21            A[i][j]=v;
22        }
23    }
24
25    //Input Matriks B
26    for (i=0; i<K; i++)
27    {
28        for (j=0; j<N; j++)
29        {
30            printf("\nInput B(%d,%d)= ", i, j);
31            scanf("%f", &v);
32            B[i][j]=v;
33        }
34    }
35
36    //Perkalian dua matriks
37    for (i = 0; i<M; i++)
38    {
39        for (j=0; j<N; j++)
40        {
41            C[i][j]=0;
42            for (k=0; k<K; k++)
43            {
44                C[i][j]=C[i][j]+A[i][k]*B[k][j];
45            }
46        }
47    }
48
49    // Mencetak hasil perkalian matriks
50    for (i=0; i<M; i++)
51    {
52        printf("[");
53        for (j=0; j<N; j++)
54        {
55            printf(" %f", C[i][j]);
56        }
57        printf("]\n");
58    }
59    return 0;
60 }

```

# Bab 11

## String

### 11.1 Deklarasi String

1. definisi

String adalah array karakter satu dimensi yang diakhiri oleh karakter NULL \0.

2. Deklarasi

```
char Nama_Str[size];
```

3. Inisialisasi String

```
1 char first_name[15] = "Komputer";
2 char first_name[15] = {'K','o','m','p','u','t','e','r','\0'};
3 // Karakter NULL '\0' diperlukan di akhir deklarasi
4 char string1 [5] = "halo";
5 /* Ukuran String = 'h'+'a'+'l'+'o'+"NULL" = 5 */
6 char string2 [] = "dunia";
7 /* Ukuran String = 'd'+'u'+'n'+'i'+'a'+"NULL" = 6 */
8 char string3[5] = {'h', 'a', 'l', 'o', '\0'} ;
9 /*Deklarasi dengan ukuran array 5*/
```

### 11.2 Input Output String

1. Input String dari user adalah menggunakan fungsi scanf

```
char str[20];
scanf("%s",str);
```

contoh :

```
1 #include <stdio.h>
2 int main()
3 {
4     char nama[20];
5     printf("Input nama: ");
6     scanf("%s", nama);
7     printf("\nNama adalah %s.", nama);
8     return 0;
9 }
```

⇒ Untuk mencoba program Klik [di sini](#)

2. Output String menggunakan fungsi printf

Contoh :

```
char nama[20]="EKO";
printf(" %s",nama);
```

### 11.3 Operasi String

Operasi string melibatkan fungsi string didefinisikan dalam library `string.h`

No.	Fungsi	Deskripsi	Contoh Program
1)	<code>strlen(string_name)</code>	mengembalikan panjang string	<a href="#">Klik di sini</a>
2)	<code>strcpy(destination, source)</code>	mengkopi string dari source ke destination	<a href="#">Klik di sini</a>
3)	<code>strcat(Str1,Str2)</code>	Menggabungkan dua string Str1 dan Str2	<a href="#">Klik di sini</a>
4)	<code>strcmp(Str1, Str2)</code>	Membandingkan Str1 dan Str2, Apabila kedua sama maka akan mengembalikan nilai 0.	<a href="#">Klik di sini</a>

#### 11.3.1 fungsi strlen

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     int n;
6     char str []="Komputer ITS";
7     //Menghitung panjang str
8     n = strlen(str);
9     printf("Panjang String : %d",n);
10    return 0;
11 }
```

⇒ Untuk mencoba program [Klik di sini](#)

#### 11.3.2 fungsi strcpy

Fungsi string tidak mempunyai operator penugasan seperti pada tipe data lain. Seperti contoh pada tipe data `int` seperti contoh

```
1 int a;
2 a = 5; //penugasan untuk inisialisasi
```

untuk memberi penugasan pada string menggunakan fungsi `strcpy`.  
Contoh Mengcopy isi `Str1` ke `Str2`

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char Str1[40]="ITS Surabaya";
6     char Str2[40];
7     // Mengcopy isi dari Str1 ke Str2
8     // Str2 = Str1
9     strcpy(Str2,Str1);
10    printf("Isi Dari Str2 : %s",Str2);
11    return 0;
12 }
```

⇒ Untuk mencoba program [Klik di sini](#)



## 11.4 fungsi strcat

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char str1[40]="ITS ";
6     char str2[40]="Surabaya";
7     //Menggabungkan dua string
8     //Str1 = Str1 + Str2
9     strcat(str1,str2);
10    printf("Isi dari str1: %s",str1);
11    return 0;
12 }
```

⇒ Untuk mencoba program Klik [di sini](#)

## 11.5 fungsi strcmp

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char str1[20],str2[20];
5     int n;
6     printf("Masukan String pertama : ");
7     scanf("%s",str1); //membaca string dari konsol
8     printf("Masukan String ke dua: ");
9     scanf("%s",str2);
10    //Membandingkan isi dari str1 dan str2
11    n = strcmp(str1,str2);
12    printf("Hasil perbandingan = %d \n",n);
13    // Apabila n==0 maka str1 sama dengan str2
14    if (n==0)
15    {
16        printf("%s'==%s",str1,str2);
17    }
18    // Apabila n<0 maka str1 lebih kecil dari str2
19    if (n<0)
20    {
21        printf("%s<%s",str1,str2);
22    }
23    // Apabila n>0 maka str1 lebih besar dari str2
24    if (n>0)
25    {
26        printf("%s>%s",str1,str2);
27    }
28
29    return 0;
30 }
```

⇒ Untuk mencoba program Klik [di sini](#)

## 11.6 Contoh Soal