

## PERTEMUAN 1 PENGANTAR ALGORITMA DAN PEMROGRAMAN

### A. Tujuan Pembelajaran

1. mahasiswa mampu menuliskan algoritma dalam bentuk *flowchart*, *pseudo code*, dan kalimat deskriptif.

### B. Uraian Materi

#### 1.1 Pengertian Algoritma

Dalam pengertian Algoritma merupakan urutan atau alur dan langkah-langkah dalam menyelesaikan dan mengetahui masalah yang bersifat logis dan juga sistimatis. Dalam kehidupan sehari-hari, apabila kita amati, sebenarnya kita telah melakukan kaidah-kaidah algoritma. Sebagai contoh adalah saat kita memasak mie instan. Pada bagian belakang bungkus mie instan apabila kita lihat, disitu terdapat langkah demi langkah bagaimana cara memasak dan juga menyajikannya. Apabila langkah-langkah yang dijabarkan tidaklah logis maka kita akan mendapatkan hasil yang pasti tidak akan sesuai seperti yang diharapkan.

Dan juga algoritma tidak harus mengikuti langkah-langkah baku seperti perhitungan dimatematika. Algoritma mengajarkan bagaimana memecahkan masalah dengan berbagai solusi dan memilih mana solusi yang terbaik.

Berikut ini adalah ciri-ciri yang dimiliki dalam sebuah algoritma yaitu :

#### 1) Kepastian

Langkah - langkah yang dijabarkan harus pasti dan tidak bermakna ganda.

#### 2) Batasan

Batasan dipakai agar algoritma berakhir setelah menjalankan sejumlah proses dan langkah-langkah.

#### 3) Efektif

Efektif yang dimaksud disini adalah Instruksi yang dijalankan dengan efektif.

#### 4) Masukan

Algoritma tidak harus memiliki satu saja masukan, tapi algoritma bisa memiliki nol atau lebih masukan..

#### 5) Keluaran

Keluaran yang dimiliki paling tidak menghasilkan satu keluaran.

Algoritma memiliki sifat-sifat sebagai berikut :

- 1) Symbol dan sintaks tidak harus baku dan tidak harus dari suatu bahasa pemrograman tertentu..
- 2) Tidak bergantung kepada suatu bahasa pemrograman tertentu.
- 3) Urutan dan notasi-notasinya dapat digunakan untuk ditranslasikan pada bahasa pemrograman manapun.
- 4) Algoritma ini dapat diterapkan disemua kejadian sehari-hari dan dapat dipakai untuk mewakili suatu urutan kejadian secara logis .

Tiga struktur dasar dari algoritma :

- 1) Runtunan (Sequence)  
Setiap instruksi dalam algoritma dijalankan secara berurutan (step by step).
- 2) Pemilihan (Selection)  
Instruksi akan dijalankan apabila persyaratan terpenuhi atau bernilai benar (true), jika instruksi bernilai salah (false), maka instruksi ini tidak akan dijalankan.
- 3) Pengulangan (repetition)  
Pengerjaan instruksi yang berulang sesuai dengan kondisi yang telah ditentukan.

Dalam penulisan algoritma atau yang biasa disebut notasi algoritmik, tidak ada hal baku didalam menuliskannya. Suatu notasi Algoritma bukanlah notasi pada bahasa didalam pemrograman sehingga siapapun itu orangnya dapat membuat notasi algoritma yang berbeda-beda. Akan tetapi aturan-aturan dan kaidah-kaidah harus ditaati untuk menghindari kekeliruan.

Dibawah ini adalah beberapa notasi yang sering digunakan dalam menulis dan mendeskripsikan algoritma diantaranya yaitu :

- 1) Suatu kalimat deskriptif
- 2) Pseudo Code
- 3) Flow chart

Dalam dunia komputer, algoritma, bahasa pemrograman dan juga program sangatlah berhubungan dengan erat. Bahasa didalam Pemrograman adalah bahasa yang digunakan komputer untuk penulisan suatu program. Sedangkan program adalah beberapa atau sekumpulan instruksi yang berupa pernyataan yang kemudian ditulis dengan menggunakan bahasa computer atau bahasa pemrograman yang melibatkan pemilihan struktur data. Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, demikian juga sebaliknya.

## 1.2 Kalimat Deskriptif

Sebuah algoritma dapat dituliskan dalam berbagai cara, bahkan dengan bahasa sehari-hari. Algoritma dapat dituliskan dalam kalimat deskriptif yang menjelaskan instruksi atau langkah-langkah yang akan dilakukan dengan jelas. Penulisan algoritma dengan kalimat deskriptif adalah yang paling mudah dan sederhana untuk dibuat namun memiliki beberapa kekurangan. Kalimat deskriptif dapat bersifat ambigu, dan oleh karena itu, kalimat deskriptif tidak memiliki karakteristik yang pasti. Suatu algoritma harus jelas dan tidak boleh memiliki lebih dari satu makna. Kalimat deskriptif jarang sekali digunakan dalam menyelesaikan masalah-masalah yang kompleks.

Teks algoritma kalimat deskriptif meliputi :

### 1) Head

Head atau judul memberikan nama atau apa yang akan kita buat pada algoritma; biasanya nama yang kita buat sudah dapat memberikan gambaran kepada aturan dari penyelesaian masalah dan juga masalah yang akan diselesaikan.

### 2) Deklarasi

Menyatakan jenis dari setiap elemen data (variabel) yang akan digunakan dalam algoritma.

### 3) Deskripsi

Adalah inti dari prosedur untuk penyelesaian masalah, meliputi pernyataan atau operasi, fungsi, penjelasan, dll.

Berikut contoh algoritma dengan kalimat deskriptif :

### Algoritma pada Luas Lingkaran

{Program menghitung luas lingkaran dan menampilkannya ke layar dengan inputan berupa jari-jari lingkaran.}

Deklarasi :

jarling = real {jari jari lingkaran dengan tipe data bil. pecahan}

luas = real {luas lingkaran dengan tipe data bil. pecahan}

phi = 3.14

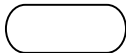
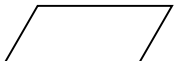
Deskripsi:


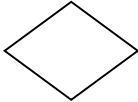
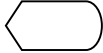







1. Baca jarling
2. Hitung luas = phi \* jarling \* jarling
3. Tampilkan luas ke layar
4. Selesai

### 1.3 Flow Chart

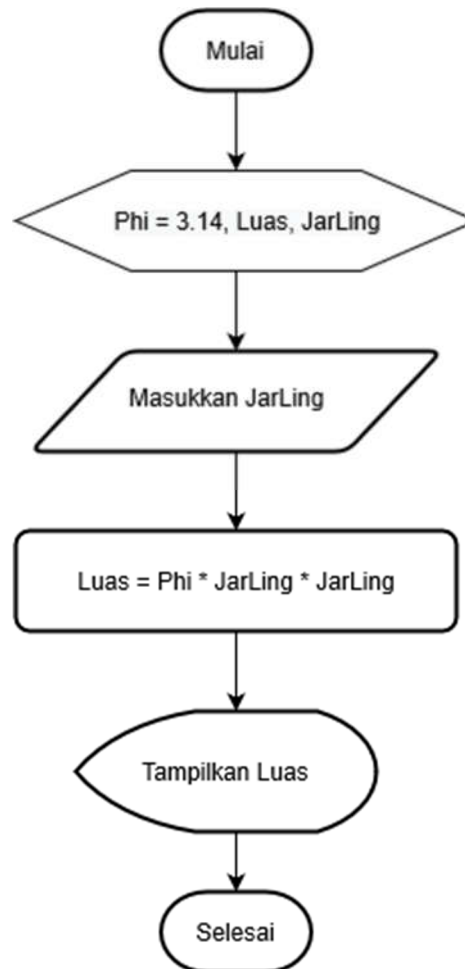
Flow chart adalah penggambaran secara grafis dari notasi algoritma. Dengan menggunakan flow chart kita dapat menggambarkan urutan atau langkah-langkah yang berisi pernyataan dalam penulisan algoritma. Flow chart berisi sekumpulan simbol-simbol yang menggambarkan proses tertentu. Suatu algoritma yang ditulis menggunakan flow chart dapat menggunakan simbol-simbol sebagai berikut :

**Tabel 1. 1 Simbol-Simbol Flowchart**

Simbol	Maksud
	Terminal (START, END)
	Input/Output (READ, WRITE)

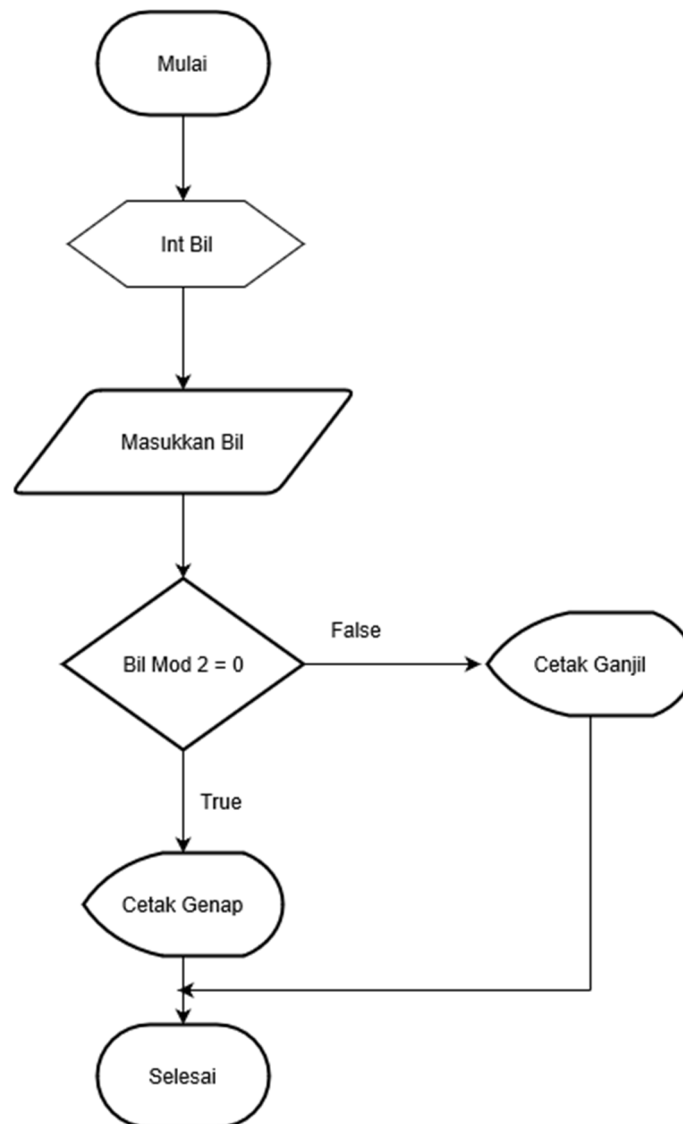
	Proses
	Pemilihan atau pencabangan (YES, NO)
	Menampilkan
	Alur
	Titik sambungan pada halaman yang sama
	Titik konektor yang berada pada halaman lain
	Call (Memanggil subprogram)
	Dokumen
	Stored Data (Simpanan Data)
	Preparation (Pemberian nilai awal suatu variabel)

Dibawah ini adalah suatu contoh flow chart dalm algoritma untuk menghitung luas lingkaran dimana rumus luas lingkaran adalah  $\text{phi} * \text{jari-jari} * \text{jari-jari}$ . Sedangkan nilai jari-jari lingkaran akan di input melalui keyboard, dan hasil perhitungan luas akan ditampilkan di layar.



**Gambar 1. 1** Flowchart Menghitung Luas Segitiga

Sebagai contoh lain adalah flow chart algoritma untuk mengecek apakah sebuah bilangan yang di inputkan melalui keyboard adalah bilangan genap atau bilangan ganjil. Dimana untuk mengetahui sebuah bilangan tersebut adalah bilangan genap atau ganjil, adalah dengan cara membagi bilangan tersebut dengan angka 2, apabila bilangan tersebut habis dibagi 2 atau dengan kata lain bilangan modulus 2 sama dengan 0 maka bilangan tersebut adalah bilangan genap, jika sisa pembagiannya adalah 1 maka bilangan tersebut adalah bilangan ganjil.



**Gambar 1. 2** Flowchart menentukan bilangan genap atau ganjil

#### 1.4 Pseudo Code

Pseudo code adalah notasi algoritma yang menyerupai notasi bahasa pemrograman tingkat tinggi. Keuntungan menggunakan pseudocode adalah kemudahan dalam mentranslasi ke notasi bahasa pemrograman, karena terdapat kemiripan dengan notasi bahasa pemrograman. Seorang programmer yang ingin menerapkan algoritma tertentu, terutama yang kompleks atau algoritma baru, biasanya akan memulainya dengan membuat deskripsi dalam bentuk pseudocode. Setelah pseudocode tersebut jadi, maka langkah selanjutnya hanya tinggal menterjemahkannya ke bahasa pemrograman tertentu.

Penulisan algoritma menggunakan pseudo code dapat menggunakan notasi-notasi sebagai berikut :

**Tabel 1. 2** Tabel Notasi Pseudocode

Pernyataan	Notasi algoritmik	Maksud
Penulisan	Cout (x)	Nilai x dicetak di piranti keluaran
	Cout (x,y)	Nilai x dan y dicetak di piranti keluaran
	Cout ("Hello")	Text <b>Hello</b> dicetak di piranti keluaran
Pembacaan	read(a)	Baca nilai a
	read(a,b)	Baca nilai a,b
Penugasan	bil←x	Isikan nilai variabel x kedalam variabel bil
Komentar	// komentar	Komentar untuk 1 (satu) baris dimulai dengan //
	/* Komentar baris 1 Komentar baris 2 --- Komentar baris n */	Komentar untuk banyak baris dimulai dengan /* dan diakhiri dengan */
Ekspresi	a > b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>nilai a lebih besar dari nilai b</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a >= b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>a &gt;= b</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a < b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>nilai a &lt; nilai b</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a <= b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>nilai a &lt;= b</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a == b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>nilai a = nilai b</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a != b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>nilai a tidak sama dengan nilai b</b> , apabila tidak maka nilai kembaliannya



		<b>false</b>
	a AND b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>kedua kondisi ( a dan b ) bernilai true</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	a OR b	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>salah satu kondisi ( a dan b ) bernilai true</b> , apabila tidak maka nilai kembaliannya <b>false</b>
	Not a	Ekspresi boolean yang akan memiliki nilai kembalian <b>true</b> jika <b>hasil evaluasi nilai a adalah false</b> , apabila tidak maka nilai kembaliannya <b>false</b>
Kondisi	If <kondisi> then <pernyataan>	Jika kondisi true / benar maka pernyataan akan dijalankan
	if <condi> then <case 1> else <case 2>	apabila kondisi true / benar maka pernyataan 1 akan dijalankan sebaliknya apabila kondisi false / salah maka pernyataan 2 yang akan dijalankan.
Pengulangan	while <cond> do { case }	Pengulangan pernyataan akan dijalankan selama kondisi true / bernar apabila kondisi false / salah maka pengulangan akan dihentikan
	repeat { case } until <cond>	Pengulangan pernyataan akan dijalankan selama kondisi true / bernar apabila kondisi false / salah maka pengulangan akan dihentikan
	for variable = nilai1 to nilai2  { pernyataan }	Pengulangan pernyataan akan dijalankan dari variable nilai1 sampai dengan nilai2

Berisi uraian penyelesaian masalah, meliputi pernyataan atau operasi, fungsi, penjelasan, dll.

Berikut ini adalah contoh suatu algoritma menggunakan pseudo code untuk mencetak pesan lulus apabila nilai ujian adalah lebih besar dari 60.

### Algoritma Lulus

```
{ mencetak pesan Lulus jika nilai ujian >=60.}
```

#### Deklarasi :

nilai : integer

#### Deskripsi:

```
read(nilai)
if nilai >= 60 then
    print('Lulus' )
else
    print('Tida k Lulus')
endif
```

Algoritma Euclidean adalah salah satu algoritma untuk mencari FPB (Faktor Persekutuan Terbesar) dari 2 buah bilangan. Langkahnya adalah dengan membagi bilangan yang lebih besar (misalkan a) dengan bilangan yang lebih kecil (b), pembagian tersebut akan menghasilkan hasil bagi h1 dan sisa s1. Jika s1 = 0, maka FPB nya adalah b. Tetapi bila s1 tidak sama dengan 0, maka prosesnya diulang lagi dengan membagi b dengan s1. Pembagian tersebut akan menghasilkan hasil bagi h2 dan sisa s2. Jika s2 = 0, maka FPB nya adalah s1 (sisa pembagian sebelumnya), jika s2 bukan 0, maka lakukan pembagian s1 dengan s2.

Berikut ini adalah contoh kalimat deskriptif, flow chart dan pseudo code untuk algoritma euclidean :

- a. Kalimat deskriptif algoritma euclidean

### Algoritma Euclidean

```
{ Diberikan dua buah bilangan bulat positif a dan b ( $a \geq b$ ). Algoritma Euclidean mencari pembagi bersama terbesar dari kedua bilangan tersebut, yaitu bilangan positif terbesar yang habis membagi a dan b}
```

**Deskripsi :**

1. Jika  $a = 0$  maka  
    b adalah jawabannya;  
    Stop.  
    Tetapi jika  $b \neq 0$ ,  
    Lanjutkan ke langkah 2.
2. Bagilah a dengan b dan misalkan s adalah sisanya
3. Ganti nilai a dengan nilai b, nilai b dengan nilai s,  
    lalu ulang kembali ke langkah 1.

## b. Pseudo code algoritma euclidean

**Algoritma Euclidean**

{ Program mencari pembagi bilangan terbesar, a dan b bilangan bulat positif }

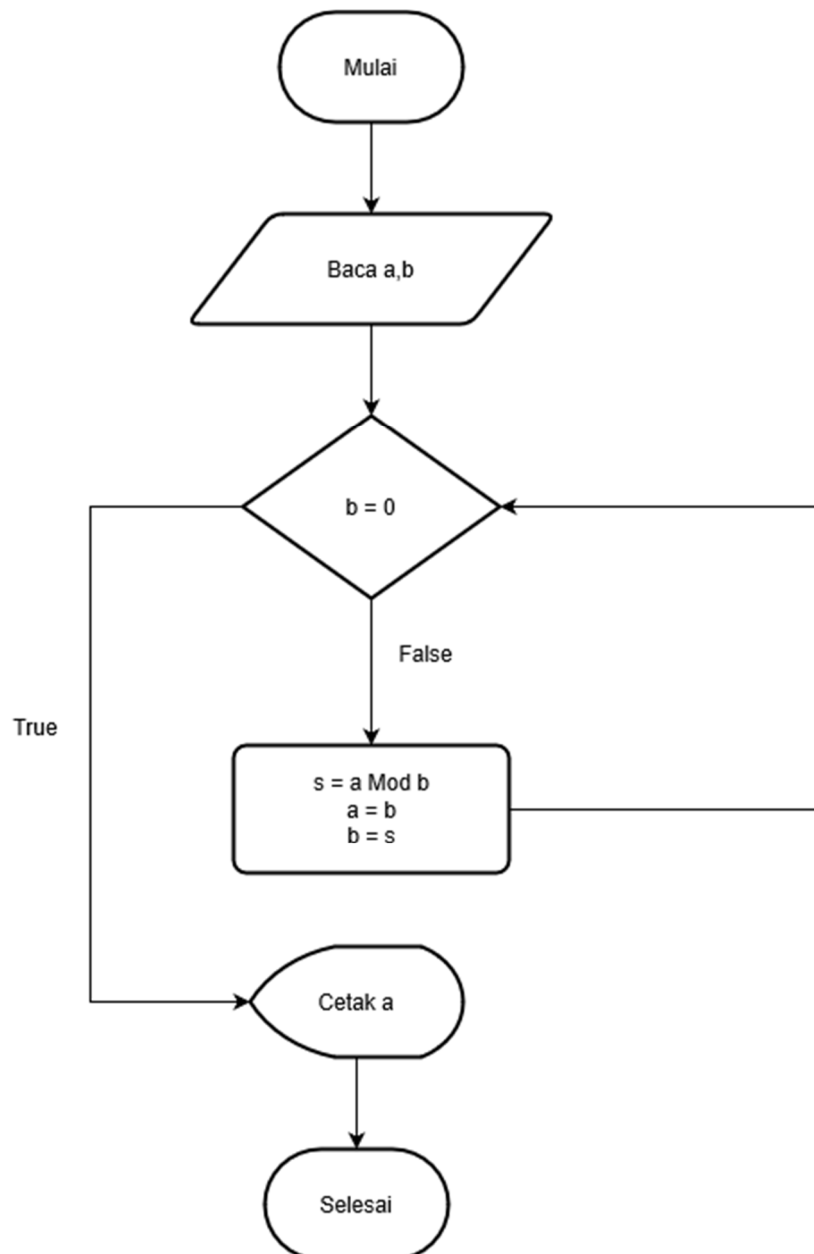
**Deklarasi :**

a, b : integer { bil yg akan dicari pbt-nya }  
s : integer { sisa hasil bagi }

**Deskripsi :**

```
read(a,b)    {a ≥ b}
while b ≠ 0 do
  s <-- a Mod b  {hitung sisa hasil bagi}
  a <-- b
  b <-- s
endwhile {kondisi selesai pengulangan: b=0, maka pbt=a}
print(a)
```

## c. Flow chart algoritma euclidean



**Gambar 1. 3** Flow chart algoritma euclidean

### C. Soal Latihan / Tugas

1. Buatlah algoritma ( dalam bentuk kalimat deskriptif, flow chart dan pseudo code ) untuk menukar dua buah bilangan.
2. Buatlah algoritma untuk mencetak deret angka bilangan ganjil dari 1 sampai 20.

3. Buatlah algoritma ( dalam bentuk kalimat deskriptif, flow chart dan pseudo code ) untuk menginput 3 buah bilangan, kemudian tentukan bilangan terbesar, terkecil, dan rata-ratanya.

#### D. Referensi

A.S, R. (2018). *LOGIKA ALGORITMA dan PEMROGRAMAN DASAR*. BANDUNG: MODULA.

Charibaldi, N. (2004). *Modul Kuliah Algoritma Pemrograman II Edisi Kedua*. Yogyakarta.

Davis, S. R. (2014). *C++ For Dummies* (7th ed.). John Wiley & Sons, Inc.

Deitel, P., & Deitel, H. (2014). *C++ How To Program* (9th ed.). United State of America: Pearson.

Munir, R. (2005). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.

Sjukani, M. (2014). *Algoritma dan Struktur Data 1 dengan C, C++ dan Java* (Edisi 9 ed.). Jakarta: Mltra Wacana Media.