



# POLITEKNIK STATISTIKA STIS

*For Better Official Statistics*

# Dasar-Dasar Algoritma dan Pemrograman

Nori Wilantika, S.S.T., M.T.I.



- ▶ Jika seseorang ingin mengirim surat kepada kenalannya di tempat lain, langkah yang harus dilakukan adalah:

***Langkah :***

- ▶ Menulis surat
- ▶ Surat dimasukkan ke dalam amplop tertutup
- ▶ Amplop dikasih alamat penerima dan pengirim
- ▶ Amplop ditempeli perangko secukupnya.
- ▶ Pergi ke Kantor Pos terdekat untuk mengirimkannya



- Definisi :
  - Urutan langkah-langkah untuk memecahkan masalah yang disusun secara sistematis dan logis.
  - Kamus Besar Bahasa Indonesia:  
Algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah.
- Algoritma dibutuhkan untuk memerintah komputer mengambil langkah-langkah tertentu dalam menyelesaikan masalah.

## Definisi

- Kumpulan instruksi-instruksi tersendiri yang biasanya disebut *source code* yang dibuat oleh programmer (pembuat program).
- Program : Realisasi dari Algoritma.

Program = Algoritma + Bahasa

- “*computers do not solve problems they execute solutions*”
- Pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman manapun.
- Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman.
- Apapun bahasa pemrogramannya, output yang akan dikeluarkan sama karena algoritmanya sama.

- Tingkat kepercayaannya tinggi (*reliability*): hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.
- Pemrosesan yang efisien (*cost rendah*): proses harus diselesaikan secepat mungkin dan frekuensi kalkulasi yang sependek mungkin.
- Sifatnya general: bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.
- Bisa dikembangkan (*expandable*): haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.
- Mudah dimengerti: siapapun yang melihat, dia akan bisa memahami algoritma Anda. Susah dimengertinya suatu program akan membuat susah di-*maintenance* (kelola).

- Portabilitas yang tinggi (*portability*)

Bisa dengan mudah diimplementasikan di berbagai *platform komputer*.

- *Precise* (tepat, betul, teliti)

- Efektif: Tidak boleh ada instruksi yang tidak mungkin dikerjakan oleh pemroses yang akan menjalankannya.

- Harus *terminate*: jalannya algoritma harus ada kriteria berhenti.

- *Output* yang dihasilkan tepat.

## 1. Mendefinisikan masalah

- a. Kondisi awal, yaitu input yang tersedia.
- b. Kondisi akhir, yaitu output yang diinginkan.
- c. Data lain yang tersedia.
- d. Operator yang tersedia.
- e. Syarat atau kendala yang harus dipenuhi.

## 2. Buat Algoritma dan Struktur Cara Penyelesaian

Jika masalahnya kompleks, maka dibagi ke dalam modul-modul

## 3. Menulis program

Pilihlah bahasa yang mudah dipelajari, mudah digunakan, dan lebih baik lagi jika sudah dikuasai, memiliki tingkat kompatibilitas tinggi dengan perangkat keras dan platform lainnya.

## 4. Mencari Kesalahan

- a. Kesalahan sintaks (penulisan program)
- b. Kesalahan pelaksanaan: semantik, logika, dan ketelitian

## 5. Uji dan Verifikasi Program

## 6. Dokumentasi Program

## 7. Pemeliharaan Program

Setiap Algoritma akan selalu terdiri dari tiga bagian yaitu :

- Judul (Header)
- Kamus
- Algoritma

Judul adalah bagian teks algoritma yang digunakan sebagai tempat mendefinisikan nama dengan menentukan apakah teks tersebut adalah program, prosedur, fungsi.

```
Program Luas_Kubus      ← {Judul Algoritma}
{ Menghitung luas kubus untuk ukuran sisi yang dibaca dari piranti masukan lalu
mencetak hasilnya kepiranti keluaran}      ← {Spesifikasi Algoritma}
```

## Kamus

{Nama type, hanya untuk type yang bukan type dasar}

**type** jam : <hh,mm,ss :integer> {Type jam terdiri dari 3 masukan yaitu “hh” sebagai jam. “mm” sebagai menit dan “ss” sebagai detik}

{Nama konstanta, harus menyebutkan type dan nilai }

**constant** phi : real = 3,14159

**constant** nama : string = ‘Alex’

**constant** benar : boolean = true

{Nama Informasi, menyebutkan type}

x,y : integer {suatu nilai yang bertipe bilangan bulat}

NMax : real {nilai maksimum yang bertipe bilangan real}

Nama : string {suatu nilai yang merupakan kumpulan character}

P : point {suatu nilai pada bidang kartesian}

Cari : Boolean {suatu nilai logika}

Kamus adalah bagian teks algoritma sebagai tempat untuk mendefinisikan:

- Nama type
- Nama konstanta
- Nama variabel
- Nama fungsi
- Nama prosedur.

Algoritma adalah bagian inti dari suatu algoritma yang berisi instruksi atau pemanggilan aksi yang telah didefinisikan.

**Algoritma**

**input** (c,d) {menerima masukan 2 bilangan c dan d}

**if** c < d **then** {operasi kondisional}

    e  $\leftarrow$  a + b {e di *assignment* oleh nilai a dan b}

**else**

    e  $\leftarrow$  a - b

**output** (e) {hasil keluaran berupa bilangan e}

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

- Algoritma dengan struktur Bahasa Indonesia
- Algoritma dengan *Pseudocode*
- Algoritma dengan *Flowchart*

## Sifat: Umum

- Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
- Tidak tergantung pada suatu bahasa pemrograman.
- Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.

***Contoh : Menghitung rata-rata tiga buah data***

Algoritma dengan struktur bahasa Indonesia :

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

Penyajian algoritma dengan pseudocode berarti menggunakan kode yang mirip dengan kode pemrograman yang sebenarnya. Pseudocode lebih rinci dari struktur Bahasa Indonesia/Inggris.

***Contoh : Menghitung rata-rata tiga buah data***

Algoritma dengan struktur pseudocode :

- 1) input (a, b, c)
- 2) Jml = a+b+c
- 3) Rerata = Jml/3
- 4) Output (Rerata)

- ❖ Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program.
- ❖ Flowchart menolong analis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan
- ❖ menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Kegunaan:

- Untuk mendesain program
- Untuk merepresentasikan program

Maka, flowchart harus dapat merepresentasikan komponen-komponen dalam bahasa pemrograman.

## a. Relationship

Flowchart dapat memberikan gambaran yang efektif, jelas, dan ringkas tentang prosedur logic. Teknik penyajian yang bersifat grafis jelas akan lebih baik daripada uraian-uraian yang bersifat teks khususnya dalam menyajikan logika-logika yang bersifat kompleks.

## b. Analysis

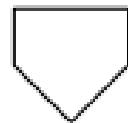
Dengan adanya pengungkapan yang jelas dalam model atau chart, maka para pembaca dapat dengan mudah melihat permasalahan atau memfokuskan perhatian pada area-area tertentu sistem informasi.

## c. Communication

Karena simbol-simbol yang digunakan mengikuti suatu standar tertentu yang sudah diakui secara umum, maka flowchart dapat merupakan alat bantu yang sangat efektif dalam mengkomunikasikan logika suatu masalah atau dalam mendokumentasikan logika tersebut.



Process



Off-page  
reference



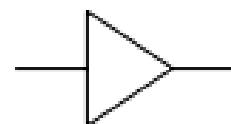
Decision



Predefined  
process



Input/Output  
data



Control  
transfer



Document



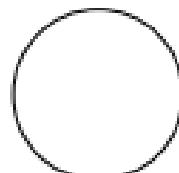
Terminator



Preparation



Stored data



Loop Connector



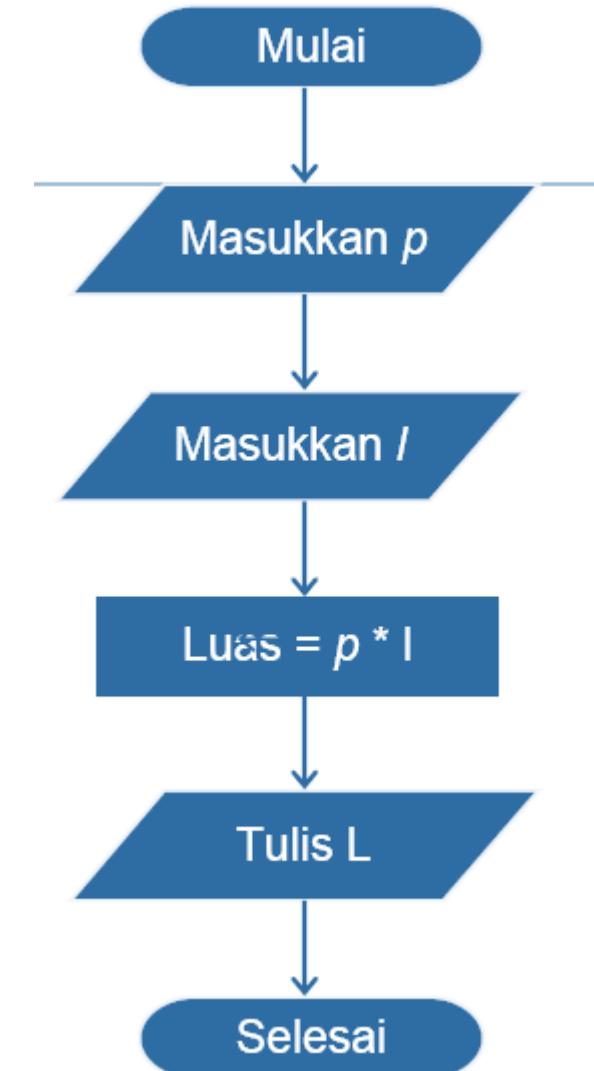
Line Connector



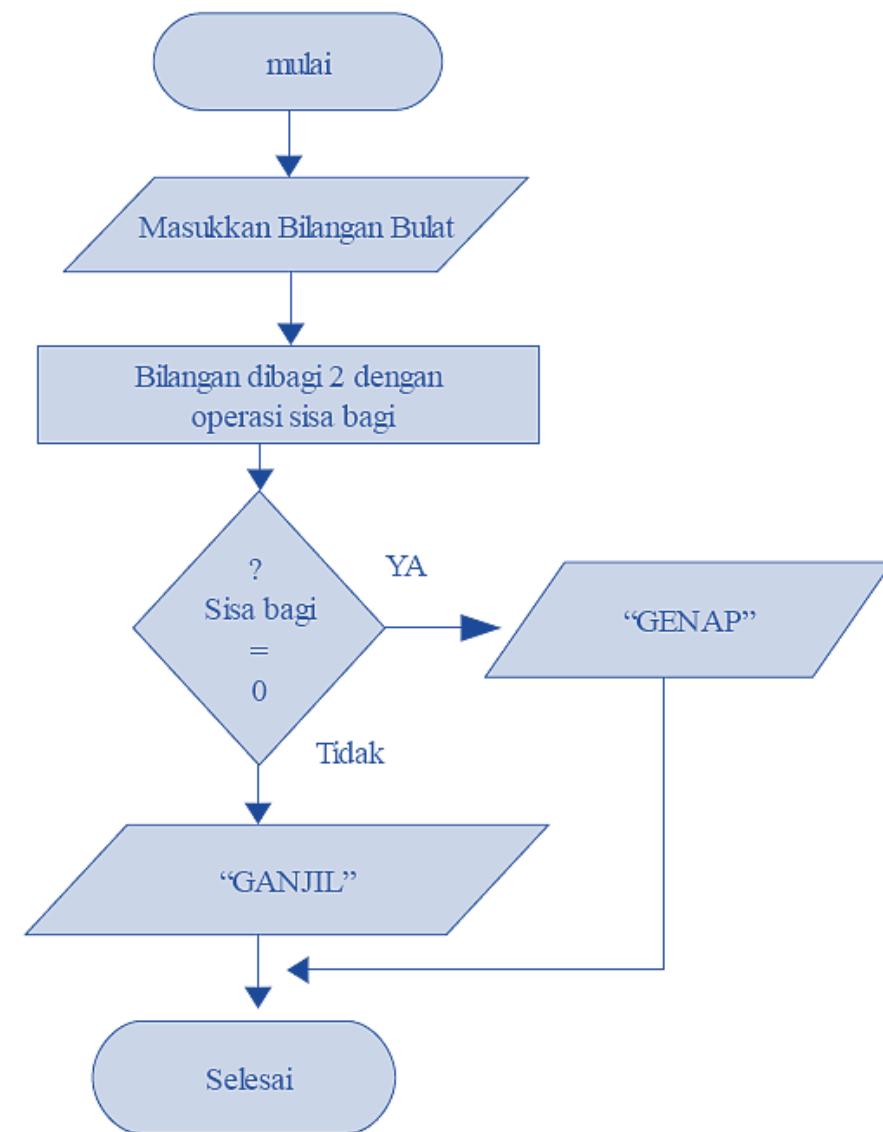
Masalah: Menghitung luas persegi panjang

Algoritma:

1. Masukkanpanjang( $p$ )
2. Masukkanlebar( $l$ )
3. Hitungluas( $L$ ), yaitu panjang kali lebar
4. Cetakluas( $L$ )



## Masalah: Menentukan bilangan ganjil atau genap



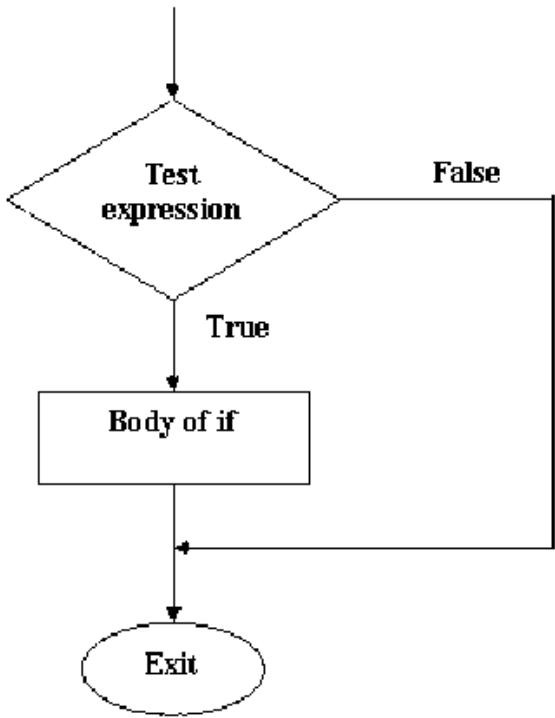
Struktur dasar algoritma :

1. Struktur Runtunan (Sequence Proses)
2. Struktur Pemilihan (Selection Proses)
3. Struktur Pengulangan (Iteration Proses)

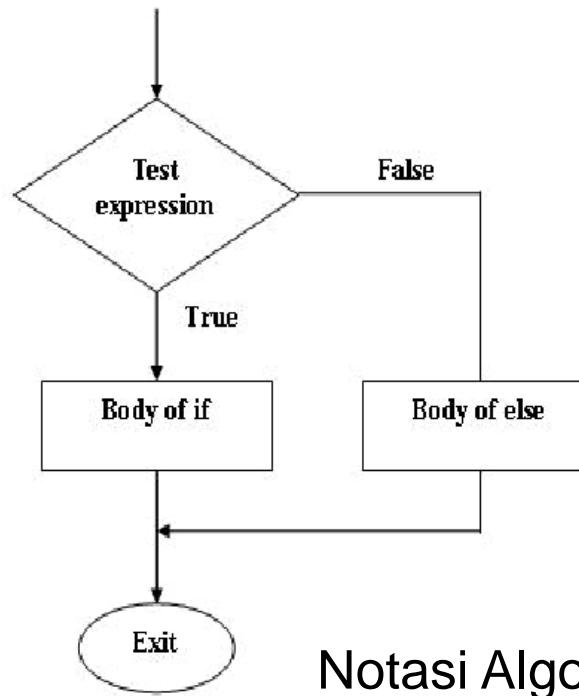
- Sebuah runtunan terdiri dari satu atau lebih ‘instruksi’;
- Tiap-tiap instruksi dilaksanakan secara berurutan sesuai dengan urutan penulisannya;
- sebuah instruksi baru bisa dilaksanakan setelah instruksi sebelumnya selesai dilaksanakan.

Pada struktur ini, jika kondisi terpenuhi maka salah satu aksi akan dilaksanakan dan aksi yang ke dua diabaikan.

***Kondisi adalah persyaratan yang dapat dinilai benar atau salah sehingga akan memunculkan ‘aksi’ yang berbeda dengan ‘kondisi’ yang berbeda.***



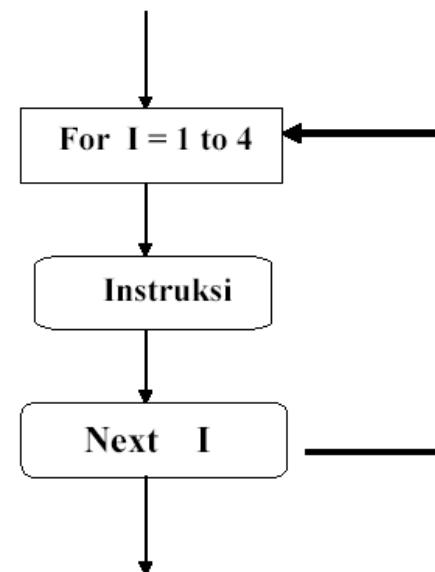
Notasi algoritmik:  
**IF** Syarat **THEN**  
 Aksi {True}  
**ENDIF** {False}



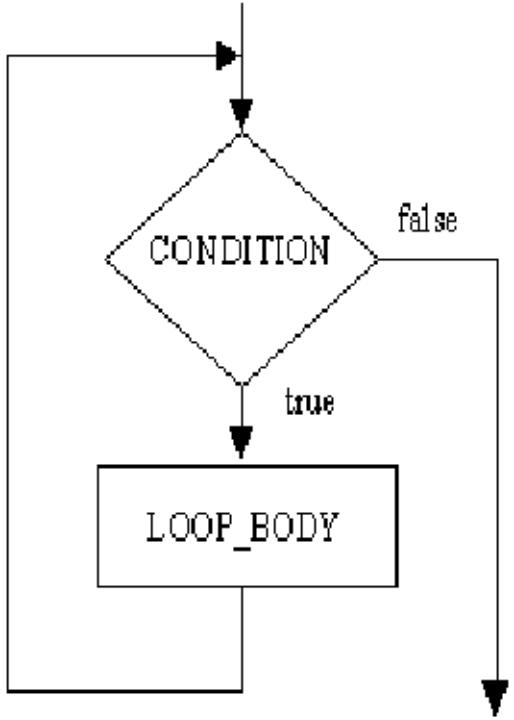
Notasi Algoritma,  
**IF** syarat **THEN**  
 aksi-1 {true}  
**ELSE**  
 aksi-2 {false}  
**ENDIF**

Digunakan untuk program yang pernyataannya akan dieksekusi berulang-ulang. Instruksi dikerjakan selama memenuhi suatu kondisi tertentu.

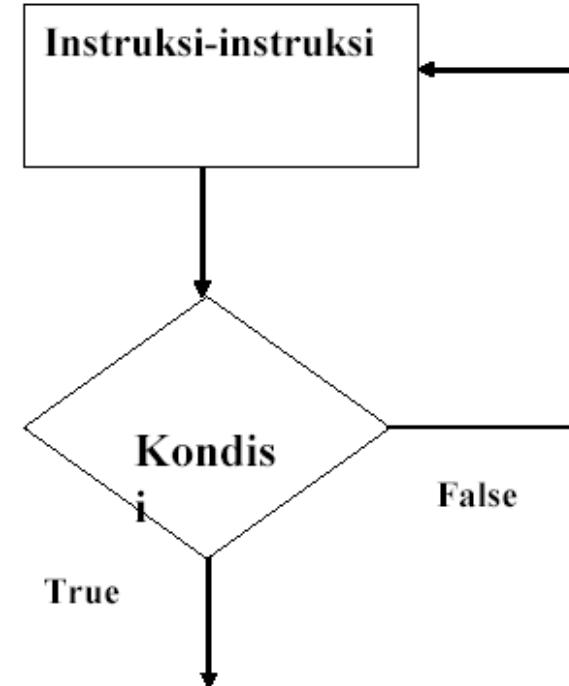
Jika syarat (kondisi) masih terpenuhi maka pernyataan (aksi) akan terus dilakukan secara berulang.



**For var=awal to akhir**  
.....  
instruksi-instruksi  
.....  
**Next var**



**While** {kondisi} do  
 .....  
 instruksi-instruksi  
 .....  
**Endwhile**



**Repeat**  
 .....  
 Instruksi  
 .....  
**Until** (kondisi)



# LATIHAN



1. Ada 2 gelas kosong berukuran: 5 liter dan 3 liter.  
Bagaimana cara mendapatkan air berukuran 2 liter?  
Bagaimana cara kita mendapatkan air berukuran 4 liter?

2. Diketahui sebuah algoritma berikut ini :

**Deklarasi :**

*i, m : integer*

**Deskripsi :**

*i = 0*

*m = 0*

*while i < 9 then*

*m = i \* i*

*cetak m*

*i = i + 1*

*endwhile.*

Tulis output yang dihasilkan algoritma di atas !





# POLITEKNIK STATISTIKA STIS

*For Better Official Statistics*

# TERIMA KASIH

