



# ALGORITMA & PEMROGRAMAN

ESTER AYUK PUSVITA, M.KOM. ~ STMIK PESAT NABIRE

3/6/2023

1

**“ALGORITMA ADALAH URUTAN LANGKAH-LANGKAH LOGIS PENYELESAIAN MASALAH YANG DISUSUN SECARA SISTEMATIS DAN LOGIS”. KATA LOGIS DISINI BERARTI BENAR SESUAI DENGAN LOGIKA MANUSIA. UNTUK MENJADI SEBUAH ALGORITMA, URUTAN LANGKAH YANG DITEMPUH UNTUK MENYELESAIKAN MASALAH HARUS MEMBERIKAN HASIL YANG BENAR.**

- Ex :
- Algoritma MENUKAR\_ISI\_BEJANA
  - ???

- Masalah dapat berupa apa saja, dengan catatan untuk setiap masalah, ada syarat kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Konsep algoritma sering kali disetarakan dengan sebuah resep.
- Sebuah resep biasanya memiliki daftar bahan atau bumbu yang akan digunakan, urutan pengerjaan dan bagaimana hasil dari urutan pengerjaan tersebut. Apabila bahan yang digunakan tidak tertera (tidak tersedia) maka resep tersebut tidak akan dapat dikerjakan. Demikian juga jika urutan pengerjaannya tidak beraturan, maka hasil yang diharapkan tidak akan dapat diperoleh.

- Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan syarat yang sama. Tingkat kerumitan dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Umumnya, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki tingkat kerumitan yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan tingkat kerumitan yang tinggi.

- Perhatikan algoritma sederhana berikut :
- Luas Segitiga ??
- Contoh Algoritma\_Menghitung\_Luas\_Segitiga :

1. Mulai
2. Baca data alas dan tinggi
3. Luas adalah alas kali tinggi dibagi dua
4. Tampilkan Luas
5. Selesai.

- Algoritma di atas adalah algoritma yang sangat sederhana, hanya ada lima langkah. Pada algoritma ini tidak dijumpai perulangan ataupun pemilihan. Semua langkah dilakukan hanya satu kali.
- Sekilas algoritma di atas benar, namun apabila dicermati maka algoritma ini mengandung kesalahan yang mendasar, yaitu tidak ada pembatasan pada nilai data untuk alas dan tinggi.
- Bagaimana jika nilai data alas atau tinggi adalah bilangan 0 atau bilangan negatif ? Tentunya hasil yang keluar menjadi tidak sesuai dengan yang diharapkan.

- Dalam kasus seperti ini kita perlu menambahkan langkah untuk memastikan nilai alas dan tinggi memenuhi syarat, misalnya dengan melakukan pengecekan pada input yang masuk. Apabila input nilai alas dan tinggi kurang dari 0 maka program tidak akan dijalankan. Sehingga algoritma di atas dapat dirubah menjadi seperti contoh berikut.

- Algoritma\_Menghitung\_Luas\_Segitiga :

1. Mulai
2. Baca data alas dan tinggi
3. Periksa data alas dan tinggi, jika nilai data alas dan tinggi lebih besar dari nol maka lanjutkan ke langkah ke 4 jika tidak maka stop
4. Luas adalah alas kali tinggi dibagi dua
5. Tampilkan Luas
6. Selesai.



DARI PENJELASAN DI ATAS DAPAT DIAMBIL KESIMPULAN  
POKOK TENTANG ALGORITMA

1. Algoritma harus benar.

2. Algoritma harus berhenti.

3. Algoritma harus memberikan hasil yang  
benar.



# CARA PENULISAN ALGORITMA

- Dalam bahasa natural (Bahasa Indonesia, Bahasa Inggris, dan bahasa manusia lainnya)
  - Tapi sering membingungkan (*ambiguous*)
- Menggunakan *flow chart* (diagram alir)
  - Bagus secara visual akan tetapi repot kalau algoritmanya panjang
- Menggunakan pseudo-code
  - Sudah lebih dekat ke bahasa pemrograman, namun sulit dimengerti oleh orang yang tidak mengerti pemrograman

# PSEUDOCODE

- **Pseudo** berarti imitasi atau mirip atau menyerupai atau **tiruan**.
- **code** menunjuk pada kode program,
- berarti **pseudocode** adalah kode yang mirip dengan instruksi kode program yang sebenarnya. **Pseudocode** berbasis pada **bahasa pemrograman yang sesungguhnya**

# CONTOH PSEUDOCODE

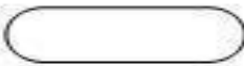








- Algoritma\_Luas\_Segitiga :

1. Start
2. Read alas, tinggi
3.  $\text{Luas} := \text{alas} * \text{tinggi} / 2$
4. Print Luas
5. End.

# FLOWCHART

- Flowchart atau bagan alir adalah skema/bagan (chart) yang menunjukkan aliran (flow) di dalam suatu program secara logika.
- Flowchart merupakan alat yang banyak digunakan untuk menggambarkan algoritma dalam bentuk notasi-notasi tertentu.
- Pada flowchart ada beberapa simbol penting yang digunakan untuk membuat algoritma

# SIMBOL FLOWCHART DAN FUNGSINYA

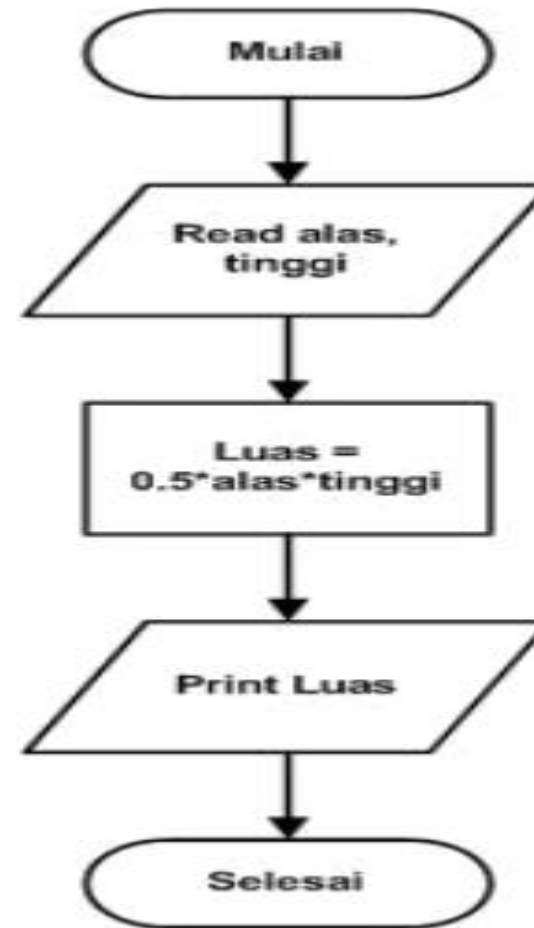
SIMBOL	NAMA	FUNGSI
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/pemberian harga awal
	<b>PROCESS</b>	Proses perhitungan/proses pengolahan data
	<b>INPUT/ OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

- Program Flowchart dapat terdiri dari dua macam, yaitu bagan alir logika program (program logic flowchart) dan bagan alir program komputer terinci (detailed computer program flowchart).
- Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika dan biasanya dipersiapkan oleh seorang analis system.
- bagan alir program komputer terinci digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci dan biasanya dipersiapkan oleh seorang programmer.





Bagan Alir Logika Program



Bagan Alir Program Komputer Terinci



# LATIHAN SOAL :

1. Buatlah algoritma dalam Bahasa natural, Pseudocode dan juga flowchart untuk menghitung :

- Volume Balok
- Luas Lingkaran

- Volume balok = panjang \* lebar \* tinggi
- Luas lingkaran =  $\pi * r * r$