

# Minggu-1

## Pengantar Algoritma & Pemrograman



Disusun oleh : Petrus Sokibi, S.Kom., M.Kom.

# Pengertian Dasar Logika Dan Algoritma

## Sejarah Algoritma

Asal kata Algoritma berasal dari nama Abu Ja'far Mohammed Ibn Musa al-Khowarizmi, ilmuwan Persia yang menulis kitab al jabr w'al-muqabala (rules of restoration and reduction) sekitar tahun 825 M

## Definisi Algoritma

- Urutan langkah-langkah untuk memecahkan masalah
- Urutan logis pengambilan putusan untuk memecahkan masalah  
**urutan langkah logis**, berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat.
- Alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis.
  - Alur pikiran yang artinya algoritma seseorang dapat berbeda dari algoritma orang lain.
  - tertulis, yang artinya dapat berupa kalimat, gambar, atau tabel tertentu.

# Definisi algoritma (lanjutan..)

- Adalah kumpulan instruksi-instruksi tersendiri yang biasanya disebut source code yang dibuat oleh programmer (pembuat program)
- Program adalah kumpulan instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat untuk menyelesaikan suatu persoalan.

Dalam bidang komputer, algoritma sangat diperlukan dalam menyelesaikan berbagai masalah pemrograman, terutama dalam komputasi numeris. Tanpa algoritma yang dirancang baik maka proses pemrograman akan menjadi salah, rusak, atau lambat dan tidak efisien.

## Note:

Algoritma Di butuhkan untuk memerintah computer mengambil langkah-langkah tertentu untuk menyelesaikan masalah

Algoritma → Pemrograman → Program

Agar algoritma dapat memerintah (diproses) komputer, maka diubah menjadi bentuk program (melalui proses pemrograman).

# Sifat Algoritma

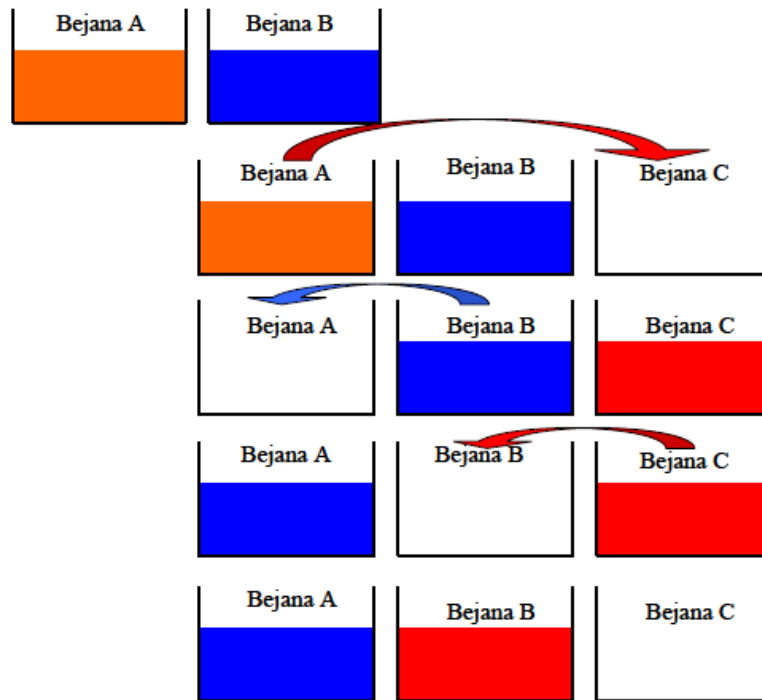
- Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman.
- Tidak tergantung pada suatu bahasa pemrograman.
- Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun.

## Notasi Penulisan Algoritma

- Menggunakan bahasa natural (Bahasa manusia: Indonesia, Inggris)  
Kelemahannya masih sering membingungkan (ambigu) / sulit dipahami.
- Menggunakan Flowchart  
Kelebihan : alur algoritma dapat dilihat secara visual  
Kekurangan : repot pembuatannya jika algoritma panjang
- Menggunakan Pseudocode  
Sudah dekat dengan bahasa pemrograman, tetapi sulit dimengerti oleh orang yang belum tahu pemrograman

# Menggunakan Bahasa Natural

Contoh Algoritma Tukar Isi Bejana, perhatikan ilustrasi berikut ini :

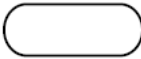

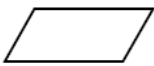
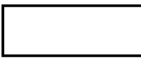
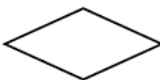
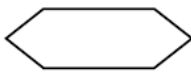
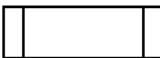




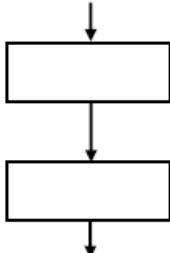
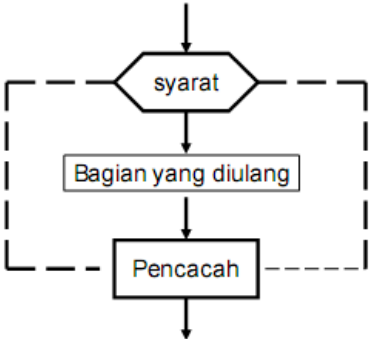
**Dengan menggunakan bahasa natural, langkah-langkah algoritma dapat dituliskan sebagai berikut :**

- 1) Isi Bejana C (Kosong) dari Bejana A (warna Merah)
- 2) Isi Bejana A (setelah Kosong) dari Bejana B (warna Biru)
- 3) Isi Bejana B (setelah Kosong) dari Bejana C (warna Merah)
- 4) Maka Isi Bejana A = Biru; Bejana B = Merah dan ; Bejana C = Kosong

# Menggunakan Flowchart

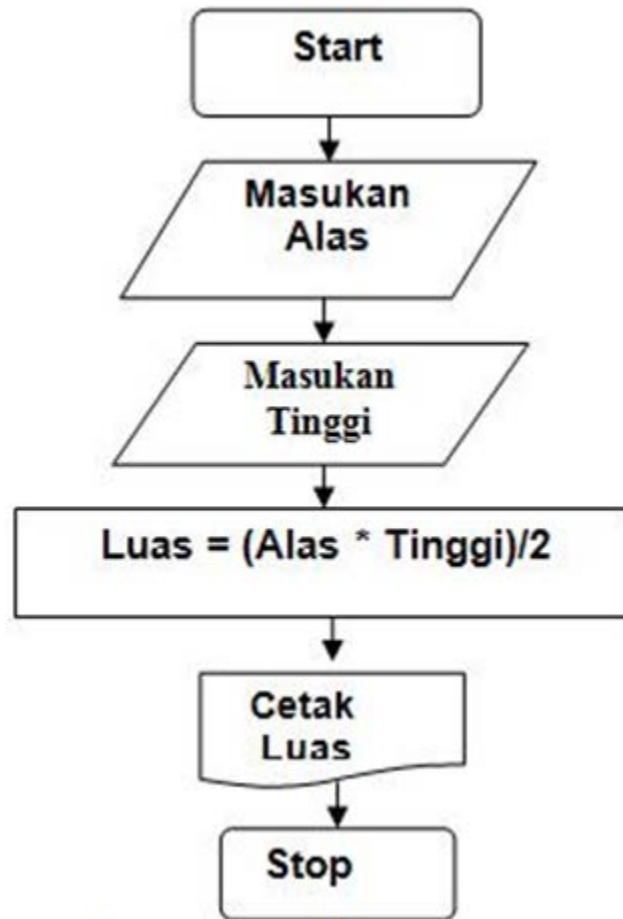
- Menggambarakan secara keseluruhan urutan proses / logika, dimana persoalan yang ada akan diselesaikan, dan menunjukkan tingkat dari detail penyelesaian persoalan.

KETERANGAN	LAMBANG
Mulai / Selesai (Terminator)	
Aliran Data	
Input / Output	
Proses	
Percabangan (Decision)	
Pemberian nilai awal suatu variabel (Preparation)	
Memanggil prosedur / fungsi (Call)	
Connector (di halaman yang sama)	

Connector (di halaman lain)	
Sequence process	
Perulangan	

# Contoh Flowchart

Berikut ini adalah contoh Flowchart menghitung Luas Segitiga :



# PSEUDOCODE

- Adalah cara untuk menuliskan algoritma yang digunakan seseorang dalam pemrograman dalam bahasa high-level lebih ringkas dari algoritma.
- Pseudocode dituliskan dengan kombinasi Bahasa universal yang mudah di pahami dan notasi matematika.
- **Pseudocode tidak terlalu detail** dibandingkan dengan program. Hal-hal mendetail dalam program yang sifatnya teknis tidak dibahas di dalam Pseudocode.



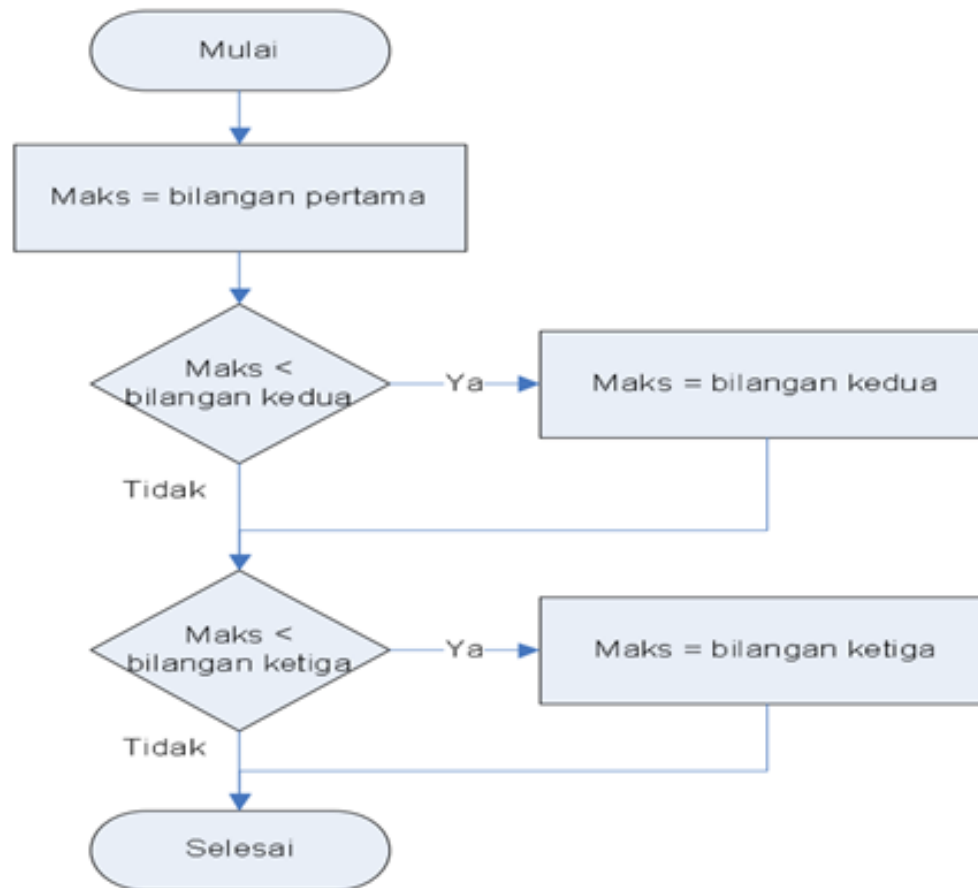
## Contoh Studi Kasus Sederhana

Buatlah Algoritma untuk memilih bilangan terbesar dari 3 buah bilangan

### Dengan Bahasa Natural

1. Ambil bilangan pertama dan set maks sama dengan bilangan pertama
2. Ambil bilangan kedua dan bandingkan dengan maks
3. Apa bila bilangan kedua lebih besar dari maks, set maks sama dengan bilangan kedua
4. Ambil blangan ketiga dan bandingan dengan maks
5. Apabila bilangan ketiga lebih besar dari maks, set maks sama dengan bilangan ketiga
6. Variabel maks berisi bilangan terbesar. Tampilkan hasilnya

# Dengan Flowchart



# Dengan Pseudocode

Maks  $\leftarrow$  bilangan pertama

if (maks < bilangan kedua)

mak  $\leftarrow$  bilangan kedua

if (maks < bilangan ketiga)

maks  $\leftarrow$  bilangan ketiga

# Latihan Mandiri

Buatlah Algoritma (gunakan bahasa natural dan flowchart)

- Memasak mie instan.
- Membuat teh manis.
- Penerimaan Mahasiswa Baru di UCIC.

Buatlah Pseudocode untuk :

- Menghitung Luas Persegi Panjang
- Menghitung Luas Lingkaran