

Pertemuan Ke-9

**STRUKTUR DASAR
LOGIKA ALGORITMA**



Oleh:

Imana Malia kondou, S.T.

Tips Sukses Menjadi Mahasiswa

- Pergunakan waktu dengan maksimal
- Buat buku tugas dan kerjakan tugas segera
- Jaga komunikasi dengan teman dan dosen
- Baca buku
- Searching di internet
- Diskusi
- Nikmati
- Selamat Berjuang !!!

Tujuan Pembelajaran

- mengajarkan konsep-konsep dan teknik-teknik dasar pemrograman imperatif dan struktur datanya.
- konsep dan logika berpikir komputer,
- cara perancangan dan analisis masalah,
- yang kemudian dipecahkan dengan menggunakan komputer menggunakan algoritma dan pemrograman terstruktur.
- Selain itu juga diperkenalkan dan diajarkan penggunaan bahasa pemrograman (Bahasa Pascal), dan flowchart
- mampu menggambarkan logika jalannya program secara tertulis dengan algoritma (pseudo code) dan dilengkapi dengan diagram alir (flow chart) menggunakan suatu bahasa pemrograman tertentu

PENGERTIAN DASAR LOGIKA dan ALGORITMA



Sejarah Algoritma

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

BAHASA PEMROGRAMAN

- Program harus ditulis dalam suatu bahasa yang dimengerti oleh komputer yaitu dalam Bahasa pemrogram dibedakan menjadi :
 - Bahasa tingkat rendah (low level language) :
bahasa yang berorientasi ke mesin.
 - Bahasa tingkat tinggi (high level language) :
bahasa yang berorientasi ke manusia (seperti bahasa inggris)
contoh bahasa Pascal, bahasa C dll.

BAHASA PEMROGRAMAN

- Program yang ditulis dalam bahasa pemrograman akan diterjemahkan ke dalam bahasa mesin (kenal dengan biner digit) dengan menggunakan penterjemah.

Penterjemah :

- **Interpreter** : menterjemahkan baris per baris instruksi. Contoh bahasa Basic.
- **Compiler** : menterjemahkan setelah seluruh instruksi ditulis. Contoh bahasa Pascal, C, C++, dll.

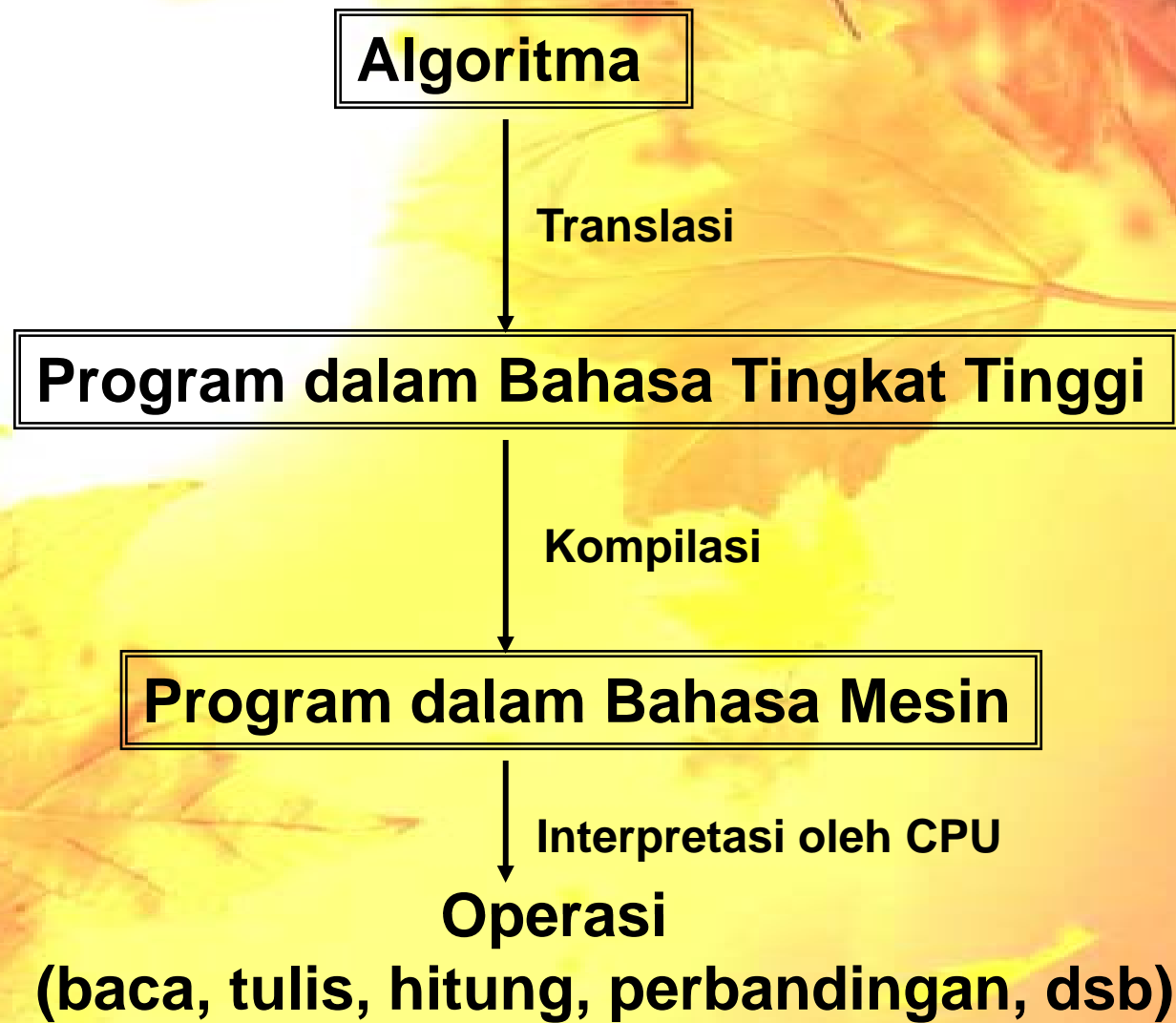
Langkah Pemrograman Komputer

- Mendefinisikan Masalah
Masalahnya apa, apa peran komputer, masukan dan keluaran
- Menentukan solusi
Bagaimana pemecahannya, membagi program atas beberapa modul
- Memilih Algoritma
Berpengaruh terhadap unjuk kerja
- Menulis program
Jenis permasalahan, bahasa yang dikuasai
- Menguji program
Kompilasi baik, keluaran sesuai dengan keinginan
- Menulis dokumentasi
Kegunaan dari suatu pernyataan
- Merawat program
Kemunculan bug baru, tambahan atas fasilitas baru

Definisi Algoritma

- Urutan langkah-langkah untuk memecahkan masalah
- Urutan logis pengambilan keputusan 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.

Tahapan Pemecahan Masalah



Kriteria Pemilihan Algoritma

- Algoritma harus berhenti
- Setiap langkah harus tepat dan tidak berarti dua (ambiguous)
- Algoritma memiliki nol atau lebih masukan (input)
- Algoritma memiliki nol atau lebih keluaran (output)
- Algoritma harus sangkil (efektif)

Suatu Algoritma yg terbaik (**The Best**): “*Suatu algoritma harus menghasilkan output yg tepat guna (efektif) dlm waktu yg relatif singkat & penggunaan memori yg relatif sedikit (efesien) dgn langkah yg berhingga & prosedurnya berakhir baik dlm keadaan diperoleh suatu solusi ataupun tdk ada solusinya.*”

Tahap Analisa Algoritma

- 1. Bagaimana merencanakan algoritma**
- 2. Bagaimana menyatakan suatu algoritma (menulis algoritma)**
- 3. Bagaimana validitas suatu algoritma.**
- 4. Bagaimana Menganalisa suatu Algoritma.**
- 5. Bagaimana Menguji Program dari suatu Algoritma**

Proses, Instruksi, dan Aksi

- Algoritma merupakan deskripsi urutan pelaksanaan suatu **proses**
- Algoritma tersusun oleh sederetan langkah **instruksi** yang logis
- Tiap langkah instruksi akan mengerjakan suatu **tindakan (Aksi)**
- Bila Aksi dilaksanakan, maka sejumlah **operasi** yang bersesuaian akan dikerjakan oleh CPU

Struktur Dasar Algoritma

- **RUNTUNAN (SEQUENCE)**

- aksi-aksi dalam algoritma yang dikerjakan secara Berurutan
- contoh :

A1. Aksi 1

A2. Aksi 2

A3. Aksi 3

CONTOH : Algoritma Tukar Isi Bejana

Deskripsi:

1. Tuang larutan dari bejana A ke bejana C
2. Tuang larutan dari bejana B ke bejana A
3. Tuang larutan dari bejana C ke bejana B

Struktur Dasar Algoritma(2)

- **Pemilihan (selection)**

- aksi dikerjakan jika kondisi tertentu terpenuhi
- contoh :

IF kondisi THEN
aksi

IF kondisi THEN
aksi1

ELSE
aksi2

IF kondisi1 THEN
Aksi1

ELSE
IF kondisi2 THEN
Aksi2

ELSE
Aksi3

(Aksi 3 dilakukan jika kondisi 1 dan 2 tidak terpenuhi)

Contoh : If hari hujan then pergi dengan becak
else pergi dengan motor

Struktur Dasar Algoritma (3)

- **PENGULANGAN (REPEATTION)**

- aksi-aksi yang dikerjakan berulang kali
- contoh :

- **FOR** pencacah pengulangan dari a sampai b **DO**

Aksi

(aksi dilakukan sebanyak hitungan pencacah pengulangan, yaitu dari a sampai b → yakni sebanyak $b-a+1$ kali)

- **REPEAT**

Aksi

UNTIL kondisi

(pengulangan aksi dilakukan sehingga kondisi/persyaratan berhenti terpenuhi)

- **WHILE** kondisi **DO**

Aksi

(selama kondisi/persyaratan pengulangan masih benar, maka aksi dikerjakan)

Contoh Repetition

- Algoritma Menulis_500_kalimat
- Menulis “Logika dan Algoritma”
- sebanyak 500 kali.

- DESKRIPSI :
- 1. Tulis “Logika dan Algoritma”
- 2. Tulis “Logika dan Algoritma”
- 3. Tulis “Logika dan Algoritma”
- 4. ...
- 5. ...
- 6. ...
- 7. ...
- .
- .
- 499. Tulis “Logika dan Algoritma”
- 500. Tulis “Logika dan Algoritma”

- Algoritma Menulis_500 kalimat
- Menulis “Logika dan Algoritma”
- sebanyak 500 kali.

- DESKRIPSI :
- Pencatat_juml_kalimat = 0
- Repeat
- tulis
- “Logika dan Algoritma”
- Naikkan pencatat_juml_kal dgn 1
- Until pencatat_juml_kalimat = 500

(Lebih Efisien)

Penulisan Algoritma

1. Menggunakan bahasa natural (Bahasa manusia: Indonesia, Inggris)

Kelemahannya masih sering membingungkan (ambigu) / sulit dipahami.

2. Menggunakan Flowchart

Baik karena alur algoritma dapat dilihat secara visual, tetapi repot pembuatannya jika algoritma panjang

3. Menggunakan Pseudocode

Sudah dekat dengan bahasa pemrograman, tetapi sulit dimengerti oleh orang yang belum tahu pemrograman

Penulisan Algoritma

- **TEKS ALGORITMA**
 - berisi langkah-langkah penyelesaian masalah yang ditulis dengan bahasa yang mudah dipahami
 - Struktur Teks algoritma tersusun dalam tiga bagian, yaitu:
 1. Bagian Kepala
 2. Bagian Deklarasi
 3. Bagian Deskripsi
- Setiap bagian disertai dengan penjelasan tentang maksud penulisan teks. Penjelasan ini ditulis dalam kurung seperti ini { }

Format dan contoh penulisan algoritma

- **Algoritma nama_algoritma** {penjelasan singkat uraian yang dilakukan oleh algoritma}
- **Deklarasi** {semua nama yang digunakan, meliputi nama-nama: tipe, konstanta, variable juga nama sub program dinyatakan dibagian ini}
- **Deskripsi** {semua langkah penyelesaian dituliskan disini}
- Algoritma Cetak_Ucapan {mencetak ucapan „Selamat Belajar „ dan diikuti dengan nama orang}
- Deklarasi nama : string;
- Deskripsi nama ← 'Ahmad'
write(„Selamat Belajar“, nama)

(Sumber : Rinaldi Munir, Algoritma dan Pemrograman, Informatika Bandung)

Flow Chart

- Flow chart suatu bagan/diagram yang menggambarkan aliran proses yang dikerjakan program dari awal sampai akhir.
- Flow chart adalah algoritma yang digambarkan dengan diagram
- Fungsi dari flow chart adalah mendeskripsikan urutan pelaksanaan suatu proses (sama dengan fungsi algoritma)

Ada 2 jenis Flowchart :

1. System Flowchart

Menggambarkan urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.

2. Program Flowchart




Menggambarkan urutan instruksi yang digambarkan dengan symbol tertentu untuk memecahkan masalah dalam suatu program.



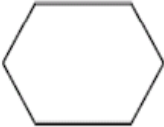
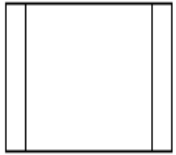


1: Pengertian Dasar dan Simbol-simbol Algoritma

Secara garis besar simbol *Flowchart* terdiri dari:

1. Simbol penghubung alur (*Flow Direction Symbols*)
2. Simbol Proses (*Processing Symbols*)
3. Simbol Input-Output (*Input-Output Symbols*)

Bentuk umum dari simbol-simbol Flow Chart antara lain:

SIMBOL	NAMA	KETERANGAN
	Terminator	Simbol Awal (Start) / Simbol Akhir (End)
	Flow Line	Simbol aliran / penghubung
	Proses	Perhitungan / pengolahan

	Input/Output Data	Pembacaan data (read) / penulisan (write)
	Decision	Simbol pernyataan pilihan
	Preparation	Inisialisasi/pemberian nilai awal
	Predefined Process (subprogram)	Proses menjalankan subprogram
	On Page Connector	Penghubung Flow chart pada satu halaman
	Off Page Connector	Penghubung Flow chart pada halaman berbeda

Contoh Deskripsi algoritma:

Sebuah prosedur ketika akan mengirimkan surat kepada teman:

1. Tulis surat pada secarik kertas surat
2. Ambil sampul surat atau amplop
3. Masukkan surat ke dalam amplop
4. Tutup amplop surat dengan lem perekat
5. Tulis alamat surat yg dituju, jika tdk ingat, lebih dahulu ambil buku alamat & cari alamat yg dituju, lalu tulis alamat tsb pd amplop surat.
6. Tempelkan perangko pada amplop surat
7. Bawa surat ke kantor pos utk diserahkan pd pegawai pos atau menuju ke bis surat untuk memasukkan surat ke dlm kotak/bis surat.

CONTOH KASUS DAN PENYELESAIAN

Ad.2. Bagaimana menyatakan suatu algoritma (menulis algoritma)

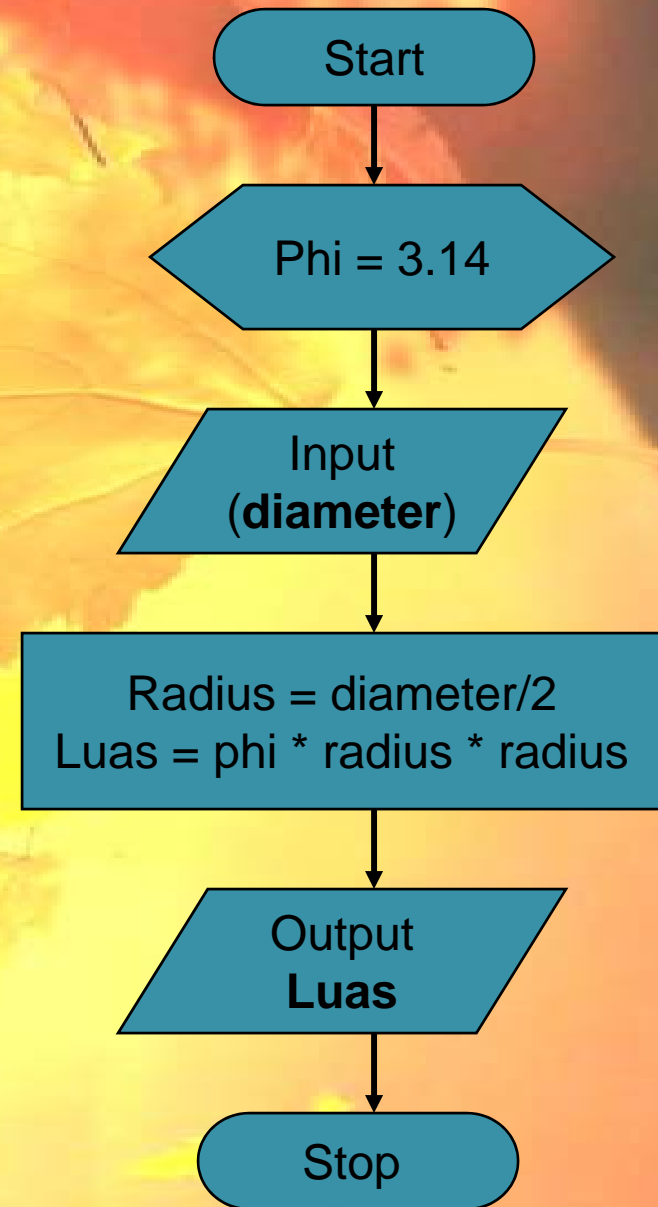
- **Dengan pseudocode**

Suatu cara penulisan algoritma agar ide dan logika dari algoritma dapat disampaikan/diekspresikan menggunakan gaya bahasa pemrograman tertentu.

```
Phi ← 3.14  
Input (diameter)  
Radius ← diameter / 2  
Luar ← phi * radius * radius  
Output (Luas)  
End
```

- Dengan flowchart / diagram alir

Program Flowchart, yaitu bagan yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah.



Sumber

- Algoritma dan Pemrograman dalam Bahasa Pascal dan C (Buku 1), Rinaldi Munir, Informatika Bandung
- Bahan Matakuliah Algoritma dan Pemrograman Universitas Atma Jaya Yogyakarta
- ilmukomputer.com