

PEMROGRAMAN DASAR

NOTASI ALGORITMA

Kelas/Semester : X RPL / 1

Guru Mapel : SUPRIATI,S.Kom

Mata Pelajaran : Pemrograman Berorientasi Objek

Tujuan :

1. Secara mandiri siswa dapat menjelaskan konsep algoritma
2. Secara mandiri siswa dapat membuat notasi algoritma berupa Indonesia structure
3. Secara mandiri siswa dapat membuat notasi algoritma berupa flowchart
4. Secara mandiri siswa dapat membuat notasi algoritma berupa Pseudocode

Materi Pembelajaran :

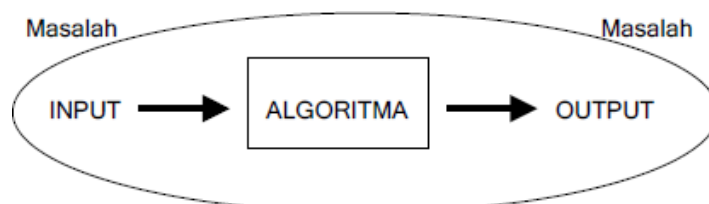
A. Pengenalan Algoritma



Gambar 1. Abu Ja'far Muhammad Ibnu Musa Al-Khawarizmi

Algoritma pertama kali diperkenalkan Oleh Ahli Matematika : Abu Ja'far Muhammad Ibnu Musa Al Khawarizmi. Seorang Ilmuwan Persia yang menulis kitab Al jabr Muqabala (Rules of restoration and Reduction) sekitar tahun 825 M.

Algoritma terdiri dari dua suku kata, yaitu: algoris dan ritmis (Bahasa Yunani). Algoris atau logis berarti sesuatu yang dapat bernilai benar atau salah. Ritmis berarti urutan yang sistematis. Berdasarkan asal katanya, Algoritma dapat diartikan sebagai urutan langkah-langkah untuk menyelesaikan masalah yang disusun atau ditulis secara logis dan sistematis.



Gambar 2. Algoritma dalam Menyelesaikan Masalah

B. Ciri-Ciri Algoritma

Berikut ini merupakan ciri-ciri Algoritma:

1. Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas
2. Setiap langkah harus didefinisikan dengan tepat dan tidak memiliki lebih dari dua arti (*ambiguous*)
3. Algoritma memiliki satu atau lebih masukan (*input*)
4. Algoritma memiliki nol atau lebih keluaran (*output*)
5. Algoritma harus sederhana

C. Keuntungan Menggunakan Algoritma

Berikut ini beberapa keuntungan menggunakan algoritma:

1. Pembuatan atau penulisan Algoritma tidak tergantung pada bahasa pemrograman manapun, artinya penulisan algoritma independen dari bahasa pemrograman dan komputer yang melaksanakannya
2. Notasi Algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman
3. Apapun bahasa pemrogramannya, *output* yang dihasilkan akan persis sama jika Algoritma yang digunakan sama.

D. Syarat Algoritma yang Baik

1. Tingkat kepercayaannya tinggi (*reability*) Hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.
2. Pemrosesan yang efisien (*cost rendah*)
Proses harus diselesaikan secepat mungkin dan frekuensi kalkulasi yang sependek mungkin.
3. Sifatnya general
Bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.
4. Bisa dikembangkan (*expandable*)
Haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.
5. Mudah dimengerti
Siapapun yang melihat, dia akan bisa memahami algoritma Anda. Susah dimengertinya suatu program akan membuat susah di-*maintenance* (kelola).
6. Portabilitas yang tinggi (*portability*)
Bisa dengan mudah diimplementasikan di berbagai *platform komputer*.
7. *Precise* (tepat, betul, teliti)
8. Efektif

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

9. Harus *terminate*

Jalannya algoritma harus ada kriteria berhenti.

10. *Output* yang dihasilkan tepat.

E. Notasi Algoritma

Ada beberapa notasi Algoritma:

1. Notasi I : menyatakan langkah-langkah algoritma dengan untaian kalimat deskriptif (*English/Indonesia structure*)
2. Notasi II : menggunakan diagram alir (*flowchart*)
3. Notasi III : menggunakan *pseudocode*

Keterangan:

Notasi I

Algoritma Menggunakan Kalimat Deskriptif (*English/Indonesia Structure*) atau terkadang disebut sebagai Bahasa Natural

Basisnya adalah bahasa Inggris, tetapi bisa juga menggunakan bahasa Indonesia. Algoritma ini digunakan untuk menggambarkan suatu algoritma yang akan dikomunikasikan kepada pemakai sistem.

Gaya penulisannya :

- *Common Style* (menggunakan huruf besar di awal dan selanjutnya huruf kecil semua)
- *Capitalized Common Style* (menggunakan huruf besar semua)
- *Outline Common Style* (menggunakan nomor urut)
- *Narative Style* (berbentuk uraian)
- Gaya lain (tiap kata kunci ditulis dengan huruf besar semua)

Contoh : Menghitung rata-rata tiga buah bilangan

Algoritma dengan struktur bahasa Indonesia :

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

Notasi II

Flowchart

Metode untuk menggambarkan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah dimengerti, mudah digunakan dan standar. Tujuannya untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol standar. Terdapat 2 jenis flowchart, yaitu:

1. *System Flowchart*

Menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut.

Pita Magnetik	Kartu Flong/Keyboard	Punched Paper Tape	On Line Storage/VDU
Input/Output	Magnetic Drum	Process	Magnetic Disc
Off Line Storage	Proses Sortir	Proses Merge	Arus

Gambar 2. Sistem Flowchart

2. *Program Flowchart*

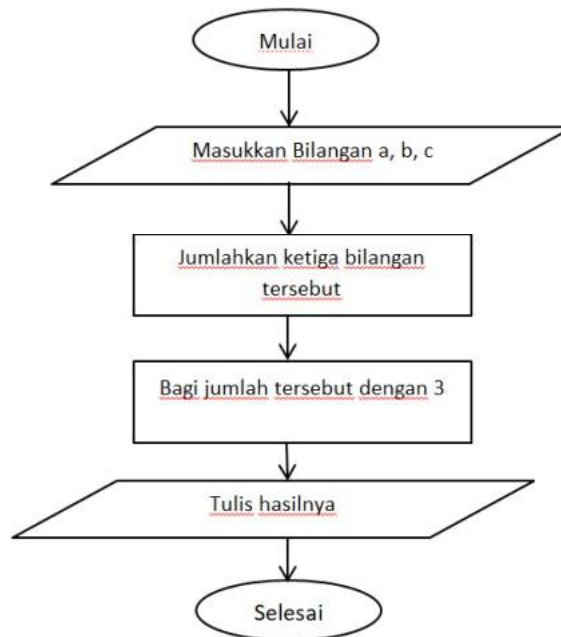
Menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Ada dua metode yaitu: *Conceptual Flowchart* dan *Detail Flowchart*.

Proses	Input Output	Keterangan
Pengujian	Pemberian Nilai Awal	Awal/Akhir Proses
Konektor pada satu halaman	Konektor pada halaman lain	Arah

Gambar 3. Program Flowchart

Contoh : Menghitung rata-rata tiga buah data

Algoritma dengan flowchart :



Mengapa Flowchart???

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 logikalogika 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.

Notasi III

Pseudocode

Pseudocode menggambarkan algoritma dengan menggunakan kode yang mirip dengan kode pemrograman yang sebenarnya. Pseudocode ini berbasis bahasa pemrograman seperti BASIC, PASCAL atau C, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada programmer. Pseudocode lebih rinci dari English/Indonesia *Structure*.

Aturan penulisan pseudocode :

1. Satu pseudocode satu baris
2. Pisahkan modul-modul atau kelompok pseudocode dengan memberikan spasi beberapa baris untuk mempermudah pembacaan
3. Pseudocode ditulis dengan huruf kapital, sedangkan komentar atau variabel dalam huruf kecil
4. Berikanlah tabulasi yang berbeda untuk penulisan pseudocode-pseudocode yang berada dalam kalang (*loop*) atau struktur kondisional
5. Lakukan pembatasan jumlah baris pseudocode setiap modulnya, misalnya 50–75 baris pseudocode per modul, sehingga tidak terlalu panjang.

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)*

Tugas

1. Buat algoritma untuk menghitung luas persegi panjang dengan
 - A. Struktur pseudocode
 - B. Flowchart
2. Buat algoritma makan dengan
 - A. Flowchart
 - B. Dengan struktur bahasa Indonesia