

Pengenalan Algoritma dan Pemrograman

Rahmatullah Arrizal S.Kom, M.T

S-1 Teknik
Informatika



From West Java for Indonesia to the World through SDGs

www.unpad.ac.id



Tujuan Pembelajaran

Setelah mengikuti pokok bahasan ini mahasiswa dapat :

- Memahami definisi algoritma,
- Memahami tujuan dan kegunaan algoritma dan pemrograman



Pokok Bahasan

1. Algoritma
2. Pemrograman





Pengenalan Algoritma



Kaitan Komputer & Permasalahan

Problematika → komputer → penyelesaian (?)

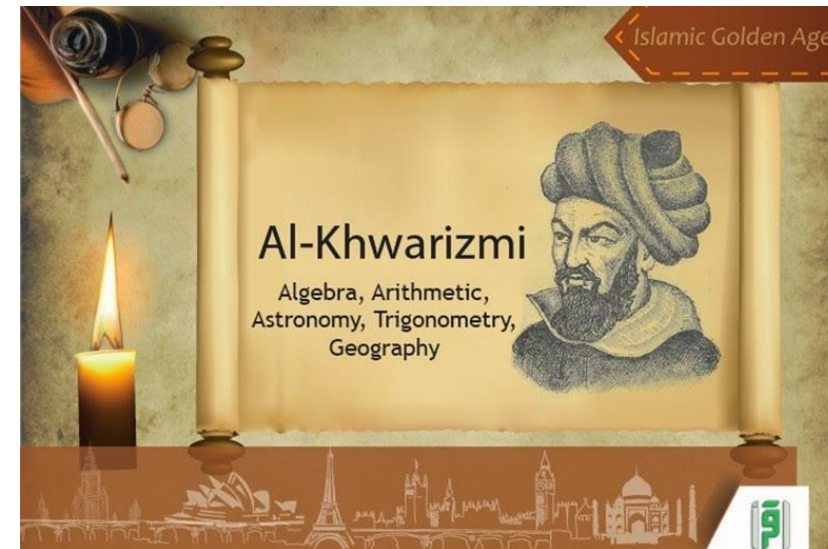
→ harus ditanamkan dalam bentuk program

- Perintah kepada komputer harus sistematis, rinci serta memenuhi kaidah logika tertentu.
- Perintah yang memiliki karakteristik seperti itu dinamakan algoritma.



Istilah Algoritma

- Algoritma berasal dari kata **al-khawarizmi (Abu Ja'far Muhammad Ibnu Musa Al-Khawarizmi)**,
- Merupakan ahli matematika dan astronomi dari Persia, dan juga merupakan penulis buku "Aljabar wal Muqobala".
- Beliau dianggap sebagai pencetus pertama algoritma karena di dalam buku tersebut Abu Ja'far menjelaskan langkah-langkah dalam menyelesaikan berbagai persoalan aritmatika (aljabar).





Pengertian Algoritma

- Algoritma biasanya didefinisikan sebagai rangkaian terurut langkah-langkah yang logis dan sistematis yang disusun untuk menyelesaikan suatu masalah.
- Algoritma dikatakan sebagai jantung ilmu komputer atau informatika
- Pemakaian: Tidak cuma dalam ilmu komputer tetapi juga dalam bidang lain (kehidupan sehari-hari)





Tujuan Algoritma

- Memberikan petunjuk tentang langkah-langkah logika penyelesaian masalah dalam bentuk yang mudah dipahami nalar manusia sebagai acuan yang membantu dalam mengembangkan program komputer.
- Pemahaman terhadap algoritma akan mencegah sejak dini kemungkinan terjadinya kesalahan logika pada program komputer yang dikembangkan.



designed by freepik



Contoh: Algoritma Login ke Facebook

- User : Masukkan username
- User : Masukkan Password
 - Sistem :Cek apakah username terdaftar?
 - Sistem :Jika **tidak terdaftar**, maka sistem akan mengeluarkan peringatan adanya kesalahan. Jika **terdaftar** maka, selanjutnya sistem akan mengecek apakah username dan password cocok (sesuai dengan database)?
 - Sistem :Jika Username dan password **cocok** maka user diperbolehkan masuk ke Facebook ybs. Jika **tidak cocok** maka sistem akan mengeluarkan peringatan adanya kesalahan
- User : Dapat mengakses halaman facebook ybs



Ciri Algoritma yang Baik

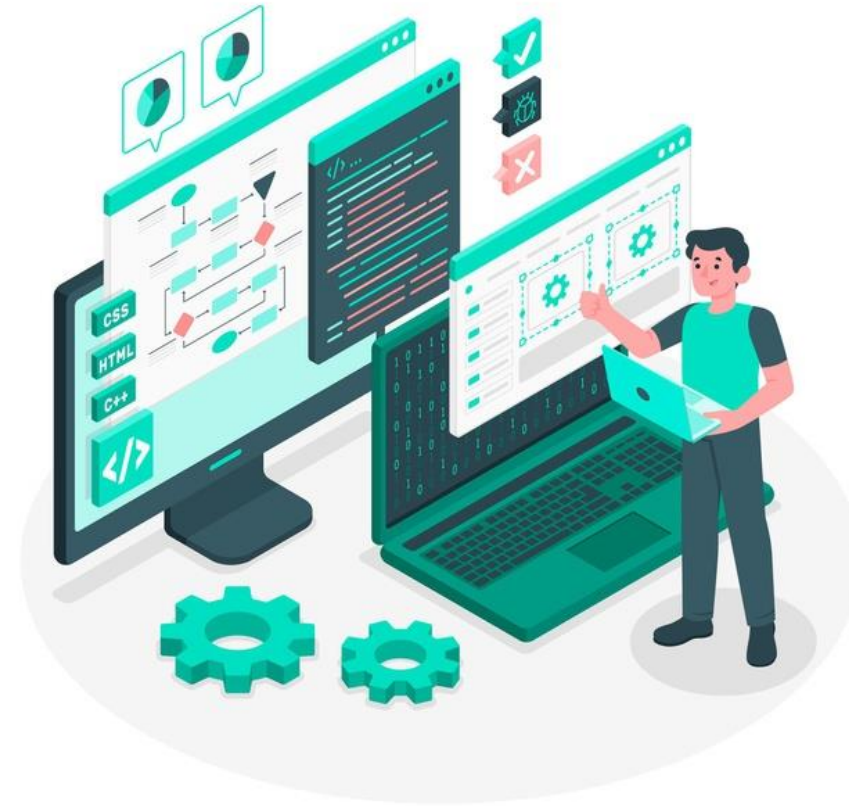
Menurut Donald E. Knuth, ada beberapa ciri algoritma

1. Algoritma mempunyai awal dan akhir, suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas. Dengan kata lain, suatu algoritma memiliki langkah yang terbatas.
2. Setiap langkah harus didefinisikan dengan tepat, sehingga tidak memiliki arti ganda, tidak membingungkan (*not ambiguous*)
3. Memiliki masukan (input) atau kondisi awal
4. Memiliki keluaran (output) atau kondisi akhir
5. Algoritma harus efektif, bila diikuti benar-benar maka akan menyelesaikan masalah.



Oleh sebab itu maka:

- Setiap instruksi dalam algoritma harus jelas dan bermakna unik/tunggal.
- Seluruh rangkaian instruksi harus selesai dalam waktu yang berhingga.
- Bersifat umum dan universal artinya dapat dipakai menyelesaikan berbagai masalah yang tipenya serupa tanpa perlu dimodifikasi sekaligus bisa diimplementasikan dengan berbagai macam bahasa pemrograman.





Contoh Lain:

Tabel 1.1. Contoh-Contoh Algoritma dalam Kehidupan Sehari-hari

No.	Proses	Algoritma	Contoh Langkah dalam Algoritma
1	Membuat kue	Resep kue	Masukkan telur ke dalam wajan, kocok sampai mengembang
2	Membuat pakaian	Pola pakaian	Gunting kain dari pinggir kiri bawah ke arah kanan sejauh 5 cm
3	Merakit mobil	Panduan merakit	Sambungkan komponen A dengan komponen B
4	Kegiatan sehari-hari	Jadwal harian	Pukul 06.00: mandi pagi, pukul 07.00: berangkat kuliah
5	Mengisi voucher HP	Panduan pengisian	Tekan 888, masukkan nomor voucher

- Berikan contoh yang lain?



Skema Dasar dalam Algoritma

- **Skema Runtunan (*sequence*)**
 - Prosedur proses dalam algoritma yang dilakukan secara terurut langkah demi langkah.
 - Sebuah urutan terdiri dari satu atau lebih instruksi. Tiap instruksi dilaksanakan secara berurutan sesuai dengan urutan pelaksanaan, artinya suatu instruksi akan dilaksanakan setelah instruksi sebelumnya telah selesai dilaksanakan.
- **Skema penyeleksian (*selection*)**
 - Suatu instruksi dikerjakan jika suatu kondisi tertentu dipenuhi. Dengan adanya proses ini maka ada kemungkinan beberapa jalur aksi yang berbeda berdasarkan kondisi yang ada.
- **Skema Pengulangan (*looping*)**
 - Suatu proses melakukan eksekusi suatu program secara berulang-ulang pada suatu blok instruksi tertentu yang terkendali.



Penulisan (Notasi) algoritma

- Uraian deskriptif
- Diagram-alir (*flowchart*)
- *Pseudocode* (kode semu)



Uraian Deskriptif

- Dengan notasi bergaya uraian, deskripsi setiap langkah dijelaskan dengan bahasa yang gamblang.
- Proses diawali dengan kata kerja seperti baca' atau 'membaca', 'hitung' atau 'menghitung', 'bagi' atau 'membagi', 'ganti' atau 'membagi', dan sebagainya,
- Pernyataan kondisional dinyatakan dengan 'jika ... maka ...'.
- Perulangan dengan 'Ulangi .. Selama ...'



Contoh: menyelesaikan permasalahan menghitung volume suatu bola

Algoritma Hitung_Volume_Bola

Deskripsi :

- Masukkan jari-jari lingkaran (r)
- Hitung volume bola dengan rumus $V = \frac{4}{3} \times \pi \times r^3$
- Tampilkan volume bola






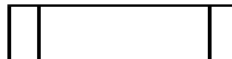





Diagram Alir / Flow Chart

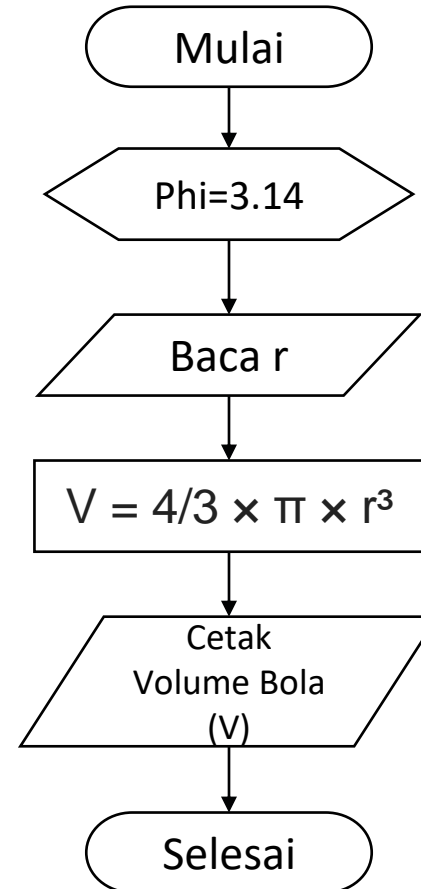
- Penulisan algoritma dilakukan dengan menggunakan diagram-diagram.
- Setiap diagram mewakili satu instruksi / perintah tertentu.
- Urutan perintah dalam suatu algoritma digambarkan dengan anak panah (dari suatu diagram ke diagram lain).
- Tidak cocok untuk penulisan algoritma yang panjang karena menimbulkan kerumitan.
- Dari segi struktur pemrograman tidak dianjurkan untuk dipakai karena bentuk penulisannya jauh berbeda dengan implementasinya pada bahasa pemrograman tertentu.
- Sejak tahun 1980-an penulisan dengan diagram alir mulai ditinggalkan, kecuali untuk menuliskan langkah-langkah global sebuah algoritma.

Notasi Diagram Alir / Flow Chart



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

Contoh: Menghitung Volume Bola





Kode Semu / Pseudo Code

- Notasi yang menyerupai notasi Bahasa pemrograman tingkat tinggi, misalnya Bahasa Pascal dan C.
- Umumnya mempunyai notasi yang hampir mirip untuk beberapa instruksi dalam beberapa bahasa pemrograman, seperti notasi *if-then-else*, *while-do*, *repeat-until*, *read*, *write*, dan sebagainya



Struktur Pseudocode

- **Bagian kepala (header)** : memuat nama algoritma serta informasi atau keterangan tentang algoritma yang ditulis
- **Bagian Deklarasi (definisi variable)** : memuat definisi nama variable, nama tetapan, nama prosedur, nama fungsi, tipe data yang akan digunakan dalam algoritma.
- **Bagian Deskripsi (rincian langkah)** : memuat langkah-langkah penyelesaian masalah, termasuk beberapa perintah seperti baca data, tampilkan, ulangi, yang mengubah data input menjadi output.



Contoh: menghitung luas sebuah luas persegi panjang

Algoritma Luas_persegi_panjang

{Menghitung sebuah luas persegipanjang apabila panjang dan lebar persegipanjang tersebut diberikan}

Deklarasi

{Definisi nama peubah/variabel}

float panjang, lebar, luas

Deskripsi

read(panjang,lebar) // input / baca

luas = panjang * lebar

write(Luas) // output / tulis



Keuntungan Menggunakan Notasi *Pseudocode*

- Kemudahan mengkonversi atau mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap *pseudocode* dengan notasi bahasa pemrograman.
- Korespondensi ini dapat diwujudkan dengan tabel translasi dari notasi algoritma ke notasi bahasa pemrograman apapun.



Pengenalan Pemrograman



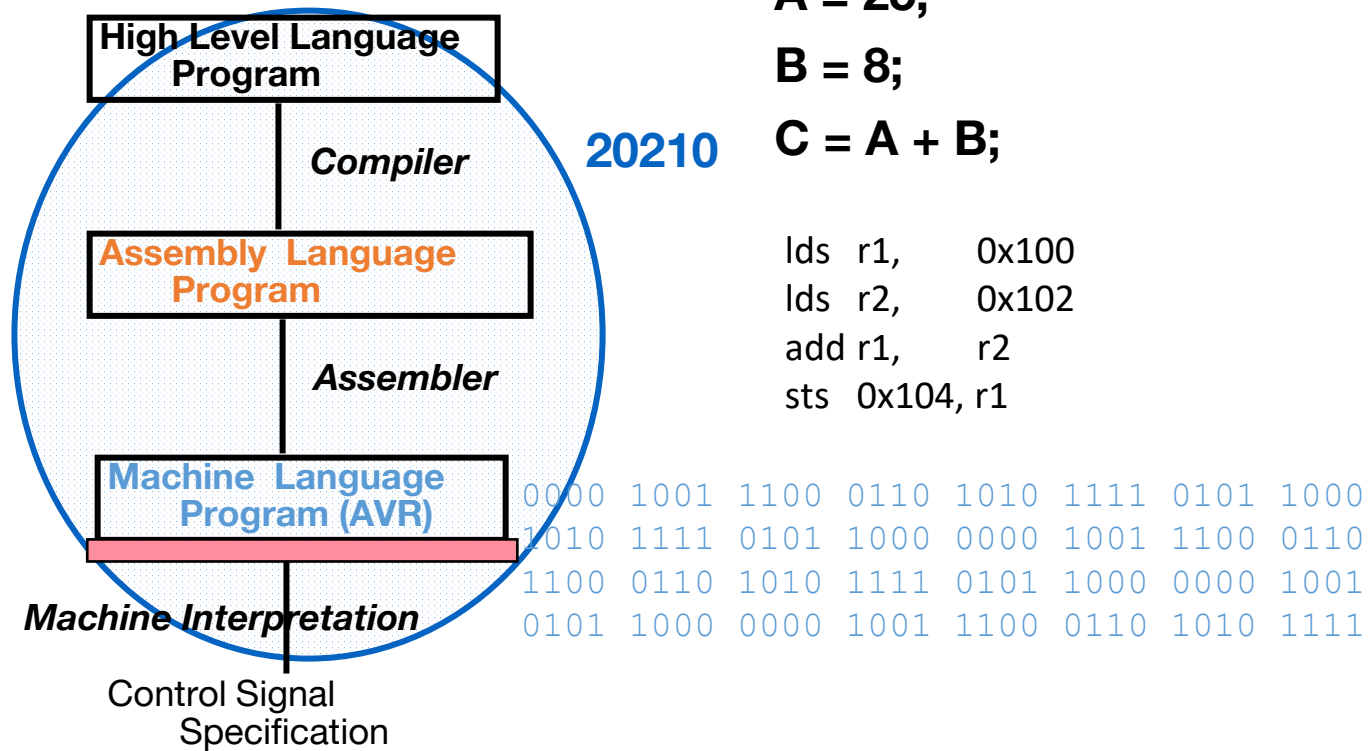
Pemrograman

- Program secara umum didefinisikan sebagai kumpulan instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang logis untuk menyelesaikan suatu persoalan yang dimengerti oleh komputer.
- Pemrograman adalah aktivitas yang berhubungan dengan pembuatan program dengan mengikuti kaidah bahasa pemrograman tertentu.
- Dalam konteks pemrograman terdapat sejumlah bahasa pemrograman seperti Pascal, Delphi, C, C++, C#, C++, dll.

Tingkatan Bahasa Pemrograman



Tingkat Bahasa Pemrograman (Tinggi hingga Rendah)



A = 25;

B = 8;

C = A + B;

```
lds r1, 0x100
```

```
lds r2, 0x102
```

```
add r1, r2
```

```
sts 0x104, r1
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



Top 10 Programming Language (IEEE Spectrum Ver)

Rank	Language	Type	Score
1	Python▼	🌐 🖥️ ⚙️	100.0
2	Java▼	🌐 📱 🖥️	95.3
3	C▼	📱 🖥️ ⚙️	94.6
4	C++▼	📱 🖥️ ⚙️	87.0
5	JavaScript▼	🌐	79.5
6	R▼	🖥️	78.6
7	Arduino▼	⚙️	73.2
8	Go▼	🌐 🖥️	73.1
9	Swift▼	📱 🖥️	70.5
10	Matlab▼	🖥️	68.4

<https://spectrum.ieee.org/computing/software/the-top-programming-languages-2020>



Paradigma Pemrograman

Digunakan beberapa pendekatan dalam bidang pemrograman (paradigma) yaitu : sudut pandang tertentu yang diprioritaskan terhadap kelompok problema, realitas, keadaan dll.

Antara Lain:

- Pemrograman **Prosedural / Terstruktur**
- Pemrograman **Fungsional**
- Pemrograman **Berorientasi Objek**
- Pemrograman **Berorientasi Fungsi**
- Pemrograman **Deklaratif**



Paradigma Pemrograman

Pemrograman **Prosedural / Terstruktur**

- Berdasarkan urutan-urutan, sekuensial
- Program adalah suatu rangkaian prosedur untuk memanipulasi data. Prosedur merupakan kumpulan instruksi yang dikerjakan secara berurutan.
- Harus mengingat prosedur mana yang sudah dipanggil dan apa yang sudah diubah.
- Program dapat dibagi-bagi menjadi prosedur dan fungsi.
- Contoh: PASCAL dan C



Paradigma Pemrograman

- Pemrograman **Fungsional**
 - Berdasarkan teori fungsi matematika
 - Fungsi merupakan dasar utama program.
 - Contoh : Lisp, Matlab, R, dll
- Pemrograman **Berorientasi Obyek**
 - Pemrograman berdasarkan prinsip obyek, dimana obyek memiliki data/variabel/property dan method/event/prosedur yang dapat dimanipulasi
 - Contoh: C++, Object Pascal, dan C++.



Paradigma Pemrograman

- Pemrograman **Berorientasi Fungsi**
 - Pemrograman ini berfokus pada suatu fungsi tertentu saja. Sangat tergantung pada tujuan pembuatan bahasa pemrograman ini.
 - Contoh: SQL (Structured Query Language), HTML, XML dan lain-lain.
- Pemrograman **Deklaratif**
 - Pemrograman ini mendeskripsikan suatu masalah dengan pernyataan daripada memecahkan masalah dengan implementasi algoritma. Contoh: PROLOG



Langkah-langkah Penyelesaian Masalah dengan Pemrograman

- **Menganalisis masalah**

- Yaitu tindakan untuk mengidentifikasi informasi yang menjadi keluaran pemecahan masalah dan data-data yang menjadi masukan.
- Dengan kerangka pemecahan masalah :

Masukan → Proses → Keluaran

- **Membuat algoritma**

- Yaitu menuangkan ide dari pengidentifikasian masalah ke dalam bentuk algoritma baik dengan menggunakan Flowchart atau PseudoCode

- **Menuangkan algoritma ke dalam bentuk program.**

- Yaitu proses membuat kode dengan menggunakan sebuah bahasa pemrograman untuk mendapatkan hasil sesuai dengan permasalahan

- **Mengeksekusi dan menguji program (implementasi).**

- Yaitu proses pengujian terhadap suatu program yang digunakan, apakah nantinya berhasil sesuai yang diharapkan atau masih terjadi kesalahan-kesalahan.



Belajar memprogram TIDAK SAMA dengan belajar bahasa pemrograman

Belajar memprogram : belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah yang kemudian dituangkan dalam suatu notasi yang disepakati bersama (bahasa program)

==> lebih bersifat pemahaman persoalan, analisis, sintesis

==> designer

Belajar bahasa pemrograman : belajar memakai suatu bahasa, aturan sintaks (tata bahasa) setiap instruksi yang ada dan tata cara pengoperasian kompilator bahasa yang bersangkutan pada mesin tertentu

==> lebih bersifat keterampilan

==> juru kode (coder)

Ilmu pemrograman berkembang menggantikan “Seni” memprogram atau memprogram secara coba-coba (“trial and error”).



Referensi

- Akmal & Mira Suryani. 2016. Modul Algoritma dan Pemrograman. Program Studi Teknik Informatika, Universitas Padjadjaran
- Rinaldi Munir. 2011. Algoritma dan Pemrograman dalam Bahasa Pascal dan C (Edisi revisi). Penerbit: Informatika Bandung.



**ANY
QUESTIONS?**



Tugas

Buatlah algoritma dari permasalahan-permasalahan berikut dengan 3 bentuk yaitu:

- Uraian deskriptif
- Diagram-alir (*flowchart*)
- *Pseudocode* (kode semu)

1. Algoritma menghitung faktorial dari sebuah bilangan
2. Algoritma mengkonversi detik ke jam, menit, detik
3. Algoritma mencari nilai terbesar dari beberapa kumpulan bilangan
4. Algoritma mencari nilai pangkat dari suatu bilangan input n dan m (pangkat)
5. Algoritma mencari bilangan prima dari 1 – 50

- Kumpulkan dengan format PDF
- flowcart boleh menggunakan aplikasi opensource <https://app.diagrams.net/> atau aplikasi lainnya



Sesi Berakhir
TERIMA KASIH