

ALGORITME :: Pengantar Pemrograman

TIM PENGAJAR PEMROGRAMAN
Departemen Ilmu Komputer IPB

Pertemuan 1

IDENTITAS MATA KULIAH

Nama Mata Kuliah	:	ALGORITME Pengantar Pemrograman
Kode Mata Kuliah	:	KOM101
Koordinator	:	Julio Adisantoso (JAS)
Semester	:	Genap 2015/2016
Pengajar	:	JAS, HRS, AAS, HKH
PJ Praktikum	:	AAS
Jadwal Kuliah	:	K1: Kamis (08.00-09.40) CCR 2.09 K2: Kamis (13:00-14:40) TL 1.01
Jadwal Praktikum	:	Ditentukan Sekretariat ILKOM

PENENTUAN NILAI AKHIR

- Praktikum, tugas, dan kuis : 10% (*jika sesuai standard*)
- UTS dan UAS Tertulis : 60-70%
- UTS dan UAS Praktikum : 30%
- Catatan: **Tidak ada ujian perbaikan**

PERANGKAT PERKULIAHAN

- Peserta: Mahasiswa Mayor ILKOM
- Situs latihan dan praktikum: <http://apps.cs.ipb.ac.id/lx>
- Site material elektronik (resources): <http://lms.ipb.ac.id> dan <http://julio.staff.ipb.ac.id>
- Bahasa Pemrograman : C (**Geany**—**CodeBlocks**)
- Referensi Utama:
C: How to Program. 3rd Ed. Deitel, H. M.and P. J. Deitel.
Pearson Education Prentice Hall. Prentice-Hall, New Jersey.

TATA TERTIB

- Kehadiran paling lambat 15 menit setelah dosen masuk kelas/lab. **TERLAMBAT, TIDAK DAPAT MASUK KELAS, DAN DIANGGAP TIDAK HADIR.**
- Berpakaian sesuai ketentuan TaTib IPB
- Minimum kehadiran 11 kali masing-masing untuk kuliah dan praktikum (syarat untuk UAS)
- Handphone harus dinonaktifkan (*silent*), no chatting/BBM/FB/etc saat kuliah maupun praktikum
- Tidak diperbolehkan mengoperasikan laptop saat kuliah, kecuali jika diperintahkan oleh dosen pengajar.

KEJUJURAN AKADEMIK

- Setiap **KECURANGAN** akan diberikan imbalan nilai 0 (**huruf mutu E**) pada mata kuliah ini
 - Menyontek ataupun bekerja sama pada saat ujian atau kuis
 - Menyalin tugas hasil pekerjaan pihak lain (sering terjadi pada tugas praktikum)
 - Titip tanda tangan kehadiran
- Imbalan (sanksi) akan diberikan untuk si pelaku maupun yang memberikan kesempatan

MENGAPA BELAJAR ALGORITME?

- Lulusan Mayor Ilmu Komputer harus mampu menulis program komputer (kode program) untuk memecahkan masalah
- Mengapa belajar algoritme? Kita ingin tahu apakah kode program yang ditulis ...
 - berdasarkan ide atau metode yang tepat?
 - menggunakan sumberdaya yang efisien?
 - merupakan solusi yang tepat bagi masalah yang ingin diselesaikan?
- *If you write code that does not solve the problem, or if it solves the problem but uses resources inefficiently, then your code doesn't really help. Computer programming is not just programming language syntax and using a development environment. At its core, computer programming is solving problems.*

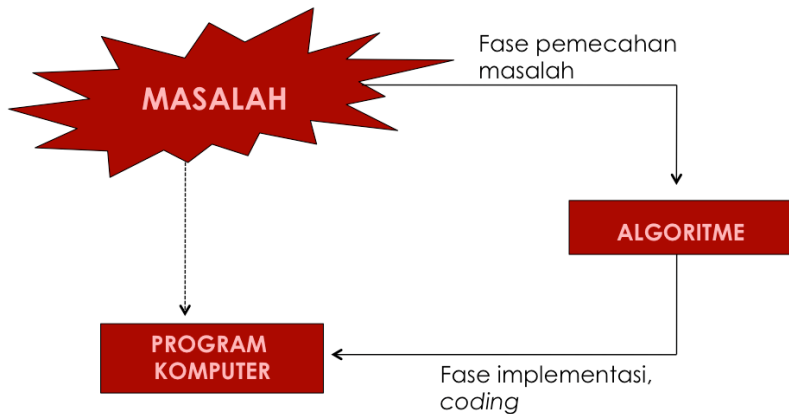
PROGRAM KOMPUTER

- Komputer hanyalah sebuah **mesin**.
- Komputer bekerja sesuai **perintah** atau **instruksi** yang diberikan.
- Perintah atau instruksi berupa "**kode/notasi**" yang diberikan kepada komputer untuk memecahkan suatu masalah disebut **program komputer**.
- Program komputer ini memandu kerja dari komputer yang secara berstruktur disusun oleh manusia.
- Bahasa yang digunakan untuk menyusun program komputer disebut **bahasa pemrograman komputer**.

ALGORITME

- Pemrograman komputer harus dilakukan secara **sistematis**, dimulai dengan merancang atau mendesain langkah-langkah yang tepat yang menjelaskan jawaban dari suatu masalah.
- Langkah-langkah yang disusun secara **berstruktur** dan **terurut** untuk menjawab suatu persoalan dengan menggunakan bahasa manusia inilah yang sering disebut dengan **algoritme**.
- Tingkatan algoritme:
 - **high-level**: diagram, math, idea
 - **low-level**: specific code (narrative, pseudocode, programming language implementation)

ALGORITME vs PROGRAM



CONTOH SOLUSI MASALAH

Contoh 1: Volume Kotak

Diketahui dimensi suatu kotak (panjang, lebar, tinggi). Hitung volume kotak.

- Ide: $\text{volume} = \text{panjang} \times \text{lebar} \times \text{tinggi}$.
- Dekomposisi:
 - Input: panjang, lebar, tinggi
 - Proses: kalikan $\text{panjang} \times \text{lebar} \times \text{tinggi}$
 - Output: volume
- Algoritme (**narrative**):
 - 1 dapatkan input dari user (panjang, lebar, tinggi)
 - 2 kalikan $\text{panjang} \times \text{lebar} \times \text{tinggi}$ dan simpan hasilnya
 - 3 tampilkan hasilnya (volume)

ALGORITME : Flowchart



NOTASI ALGORITME

Notasi Algoritme #1

```
read(panjang, lebar, tinggi)
volume ← panjang * lebar * tinggi
print(volume)
```

Notasi Algoritme #2: bentuk FUNGSI

```
Input: panjang, lebar, tinggi
Output: volume
volume ← panjang * lebar * tinggi
return volume
```

PSEUDOCODE

- **Pseudocode** adalah cara **informal** untuk menuliskan algoritme atau rancangan program komputer
- Bertujuan untuk mendapatkan ide secara cepat dan juga memudahkan membaca algoritme tanpa aturan **grammar** yang ketat.
- Ada banyak cara menuliskan pseudocode, **tidak ada aturan yang baku**. Namun untuk mengurangi **amiguity**, maka dibuat **konvensi**.
- **Baca:**
http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html
<http://www.cs.iit.edu/~cs100/ProblemSolving.pdf>.

PSEUDOCODE

Contoh Pseudocode

```
DECLARE panjang, lebar, tinggi, volume  
READ panjang, lebar, tinggi  
COMPUTE volume=panjang*lebar*tinggi  
PRINT volume
```

- **Variable:** DECLARE
- **Input:** READ, OBTAIN, GET
- **Output:** PRINT, DISPLAY, SHOW
- **Compute:** COMPUTE, CALCULATE, DETERMINE
- **Initialize:** SET, INIT
- **Add one:** INCREMENT, BUMP

CONTOH PROGRAM C

Hanya sebagai gambaran proses utuh:

MASALAH → IDE → PSEUDOCODE → PROGRAM

Program C

```
#include <stdio.h>
int main() {
    int panjang, lebar, tinggi, volume;
    scanf("%d %d %d", &panjang, &lebar, &tinggi);
    volume=panjang*lebar*tinggi;
    printf("%d\n", volume);
    return 0;
}
```

Akan dibahas nanti secara detil.

LATIHAN

Buat **PSEUDOCODE** untuk masalah-masalah berikut:

- 1 Menjumlahkan 3 bilangan.
- 2 Dapatkan nilai diskriminan (atau determinan) dari persamaan kuadrat $ax^2 + bx + c = 0$.
- 3 Menentukan luas segitiga yang memiliki alas dan tinggi tertentu.
- 4 Mendapatkan nilai suhu Celcius dari nilai suhu Fahrenheit.
- 5 Mendapatkan nilai Kilometer dari nilai jarak Mil.

HOMEWORK

Buat **PSEUDOCODE** untuk masalah-masalah berikut:

- 1 Mendapatkan nilai jawaban dari suatu persamaan kuadrat $ax^2 + bx + c = 0$.
- 2 Mendapatkan selisih dari dua bilangan.
- 3 Mendapatkan selisih (dalam detik) dari dua nilai waktu (jam, menit, detik). Khusus soal ini, lengkapi jawaban dengan Flowchart.
- 4 Menghitung jumlah dari $1 + 2 + \dots + n$

Jawaban **diketik** dan disimpan dalam format **PDF**, dikumpulkan melalui **LMS-IPB** paling lambat **Rabu, 17 Februari 2016** pukul **17:00**.