

# Pengantar Pemrograman

Pendahuluan

Baharuddin Aziz

Semester I 2019/2020

# Tujuan Perkuliahan

- Mengetahui program, pemrograman, dan bahasa pemrograman
- Memahami bagaimana program dieksekusi
- Mengetahui kompilator
- Mengetahui peran bahasa pemrograman pada dunia industri

Apa itu program?

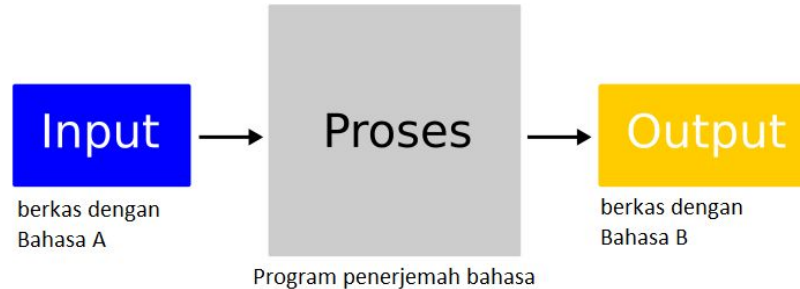
# Apa itu program? (1/2)

**Program** adalah serangkaian instruksi yang dieksekusi oleh mesin untuk mencapai suatu tujuan tertentu. Biasanya, program dapat menerima masukan, memprosesnya, dan menghasilkan suatu keluaran.



# Apa itu program? (2/2)

**Contoh:** program *penerjemah bahasa* menerima *berkas dalam suatu bahasa* sebagai masukan, kemudian *menerjemahkannya*, lalu menghasilkan keluaran berupa *hasil terjemahan*.



# Paradigma Pemrograman

**Paradigma:** sudut pandang penyelesaian persoalan dengan **[program]**

Paradigma yang ada:

- Fungsional
- **Prosedural**
- Object Oriented
- dll.

# Pemrograman Prosedural

# Pemrograman Prosedural (1/3)

Pada kuliah ini paradigma yang dibahas adalah **prosedural**, sehingga yang akan dipelajari adalah pemrograman prosedural.

## **Pemrograman prosedural:**

dihasilkan berdasarkan dekomposisi “aksional”, menjadi **Aksi** yang akan dijalankan secara berurutan (**sekuensial**).



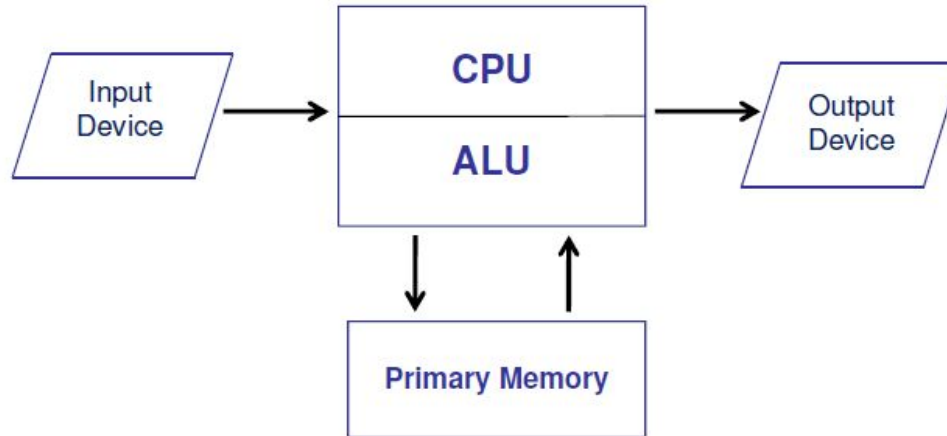
# Pemrograman Prosedural (2/3)

**Aksi** yang dimaksud:

- Jelas kondisi awal (*initial state*), kondisi akhir (*final state*), dan harus dalam waktu terbatas.
- Dapat didekomposisi menjadi **Sub Aksi**.

Aksi diterjemahkan menjadi sederetan instruksi (**aksi primitif**) yang dapat dijalankan oleh mesin. Berikut ini adalah model mesin yang menjadi dasar berpikir dalam membuat program. Oleh karena mesin sangat primitif, maka perlu ada abstraksi.

# Pemrograman Prosedural (3/3)



Model mesin ini menjadi dasar berpikir dalam membuat program.  
Oleh karena mesin sangat primitif, maka perlu ada abstraksi

# Contoh Pemrograman Prosedural (1/5)

Pemrograman prosedural dapat dianalogikan dengan kegiatan sehari-hari, misalnya **menyiapkan makanan dengan salah satu bahan.**



# Contoh Pemrograman Prosedural (2/5)

Berdasarkan gambar di atas, dapat dilihat bahwa

- kondisi **awal**: beli kentang
- kondisi **akhir**: kentang goreng / kentang rebus / sup (?)

Dari kondisi awal, diperlukan beberapa langkah untuk dapat mencapai kondisi akhir.

# Contoh Pemrograman Prosedural (3/5)

Jika diuraikan ke dalam beberapa urutan aksi, maka:

1. Ketika membeli kentang, apakah tersedia di penjual yang kita kunjungi?
2. Jika tidak, apakah perlu periksa penjual lain atau berhenti mencari?
3. Jika iya, lanjut mengupas kentang yang sudah dibeli.
4. Setelah mengupas beberapa, apakah cukup untuk membuat masakan?
5. Jika tidak, apakah perlu menambah kentang yang perlu dikupas atau dicukupkan saja?
6. Jika sudah cukup, lanjut mengolah kentang menjadi masakan, misal digoreng, direbus, atau sup.

# Contoh Pemrograman Prosedural (4/5)

## AKSI MENGUPAS KENTANG

**Initial State (I.S.):**  $T_0$ , kentang di kantong, ada di rak dapur.

**Final State (F.S.):**  $T_1$ , kentang terkupas di panci, siap dimasak, kantong kembali ke rak dapur.

### **Sub-aksi:**

1. Ambil kantong kentang dari rak
2. Ambil panci dari lemari
3. Kupas kentang
4. Kembalikan kantong kentang ke rak

Aksi 1 dan 2 bebas urutan, bisa digabungkan.

# Contoh Pemrograman Prosedural (5/5)

- Harus dikerjakan dengan urutan yang diinginkan  
→ aksi **sekuensial**
- Saat tertentu (jika baju warna muda) perlu celemek  
→ aksi **kondisional/analisis kasus**
- Mengupas kentang dilakukan hingga pada kriteria tertentu (jumlah)  
→ **perulangan**
- *Mengupas kentang* bagian dari aksi *menyiapkan makan malam*  
→ **subprogram**

# Pemrograman



# Pemrograman

*Programming tidak sekedar coding.*

- **Pemrograman** adalah aktivitas menulis program. Program ditulis dengan bahasa pemrograman, sehingga mesin atau komputer dapat mengerti apa yang yang diinstruksikan.  
**Pemrograman: membuat spesifikasi, implementasi, *debugging*, *testing***
- Belajar pemrograman tidak sama dengan belajar bahasa pemrograman.
- **Analogi:** belajar [*tata bahasa, kosa kata*] Indonesia dengan belajar menulis makalah ilmiah dalam bahasa Indonesia.

***Programming: art or science?***

# Kegiatan dalam Pemrograman

- utak atik, simulasi mencari solusi
- analisis, pembuatan spesifikasi
- program *writing (coding)*
- program *execution (observation, debugging, testing)*
- program *reading*
- program *correctness & complexity analysis*

# Skala dan Kompleksitas Program

## Skala program: relatif

- Program kecil
- Program sedang
- Program besar

Selain skala, perlu diperhatikan **kompleksitas** program:

- algoritma kompleks
- teknik pemecahan persoalan lanjut: *dynamic programming, branch and bound, advanced searching, advanced data structure, dst.*

# Macam-macam *Programmer*

- *End user programmer, coder, component user*
- *Designer*
- *Framework and component provider*
- *CASE tools programmer*
- *Pemrogram sistem (system programmer)*
- *Tester*

# Bagaimana Komputer Menjalankan Program?

- Masa lalu, komputer diprogram dengan bahasa Assembly
- Bahasa Assembly mudah dimengerti oleh mesin.
- Membaca dan mengerti alur program Assembly cukup sulit bagi manusia.
- Pada tahun 1960-an, mulai diciptakan bahasa pemrograman tingkat tinggi, tetapi tidak bisa dimengerti secara langsung oleh mesin.
- Perlu penerjemahan bahasa pemrograman tingkat tinggi ke tingkat rendah, sehingga mesin dapat mengerti instruksi yang diberikan.
- Penerjemahan ini biasa dilakukan oleh program yang berperan sebagai kompilator, interpreter, atau keduanya.
- Dalam hal ini kita hanya akan membahas tentang **kompilator**.

# Kompilator

- **Kompilator** adalah program komputer yang dapat menerjemahkan bahasa pemrograman tingkat tinggi ke bahasa mesin. Hasil terjemahan ini dapat dimengerti oleh mesin, sehingga dapat dieksekusi oleh komputer dengan mudah. Aktivitas menerjemahkan ini disebut dengan kompilasi.
- Siklus kerja jika kita menggunakan kompilator adalah:  
**tulis program → kompilasi → eksekusi**  
**Kompilator:** mengasilkan *object code*, yang kemudian di *link* oleh *linker* menjadi *executable code*
- **Interpreter:** menerjemahkan dan melaksanakan instruksi demi instruksi.
- Masing-masing punya kelebihan dan kekurangan.

# Bahasa Pemrograman

# Bahasa-bahasa Pemrograman (1/2)

Bahasa pemrograman dibagi menjadi **tiga kelas utama**:

1. terkait dengan platform **bisnis**,
2. didedikasikan untuk pemrograman **sistem**, dan
3. terkait dengan dunia **web**.



# Bahasa-bahasa Pemrograman (2/2)

Berdasarkan indeks **TIOBE** bahasa pemrograman yang populer, yaitu:

Sep 2019	Sep 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.661%	-0.78%
2	2		C	15.205%	-0.24%
3	3		Python	9.874%	+2.22%
4	4		C++	5.635%	-1.76%
5	6	▲	C#	3.399%	+0.10%
6	5	▼	Visual Basic .NET	3.291%	-2.02%
7	8	▲	JavaScript	2.128%	-0.00%
8	9	▲	SQL	1.944%	-0.12%
9	7	▼	PHP	1.863%	-0.91%
10	10		Objective-C	1.840%	+0.33%

# Bahasa Pemrograman Berdasarkan Kebutuhan

## Keamanan

- Bahasa pemrograman yang dibutuhkan untuk keamanan aplikasi yang lebih baik, misal **Go** dari Google dan **Rust** dari Mozilla.

## Kesederhanaan

- Penting untuk memiliki solusi akses yang sederhana, misal VB dan VB .NET.

## Big Data

- Beberapa bahasa sangat cocok untuk analisis data dan data besar, misal **R**.

Pertanyaan?