

# Teori Bahasa & Otomata

## Pertemuan 12



**Egi Safitri, S.Mat., M.Si**

# Deskripsi Otomata dan Teori Bahasa

- Otomata dan Teori Bahasa (OTB) matakuliah wajib
- Komponen utama dari Ilmu Komputer yaitu model dan gagasan mendasar mengenai komputasi.

suatu bentuk yang memiliki fungsi-fungsi dari **komputer digital**, menerima input, menghasilkan output, bisa memiliki penyimpanan sementara, dan mampu membuat keputusan dalam **menstransformasikan** dari input ke output

# Referensi

- Firrar Utdirartatmo, Teori Bahasa dan Otomata, JJ Learning Yogyakarta, 2001
- Bambang Hariyanto, Teori Bahasa, Otomata, dan Komputasi serta Terapannya, Informatika Bandung, 2004
- Dean Kelley, Otomata dan Bahasa-bahasa Formal, PT. Prenhallindo, Jakarta, 1999

# Contents

1

- **Definisi dan pengertian**

2

- **Kedudukan OTB**

3

- **Tata Bahasa**

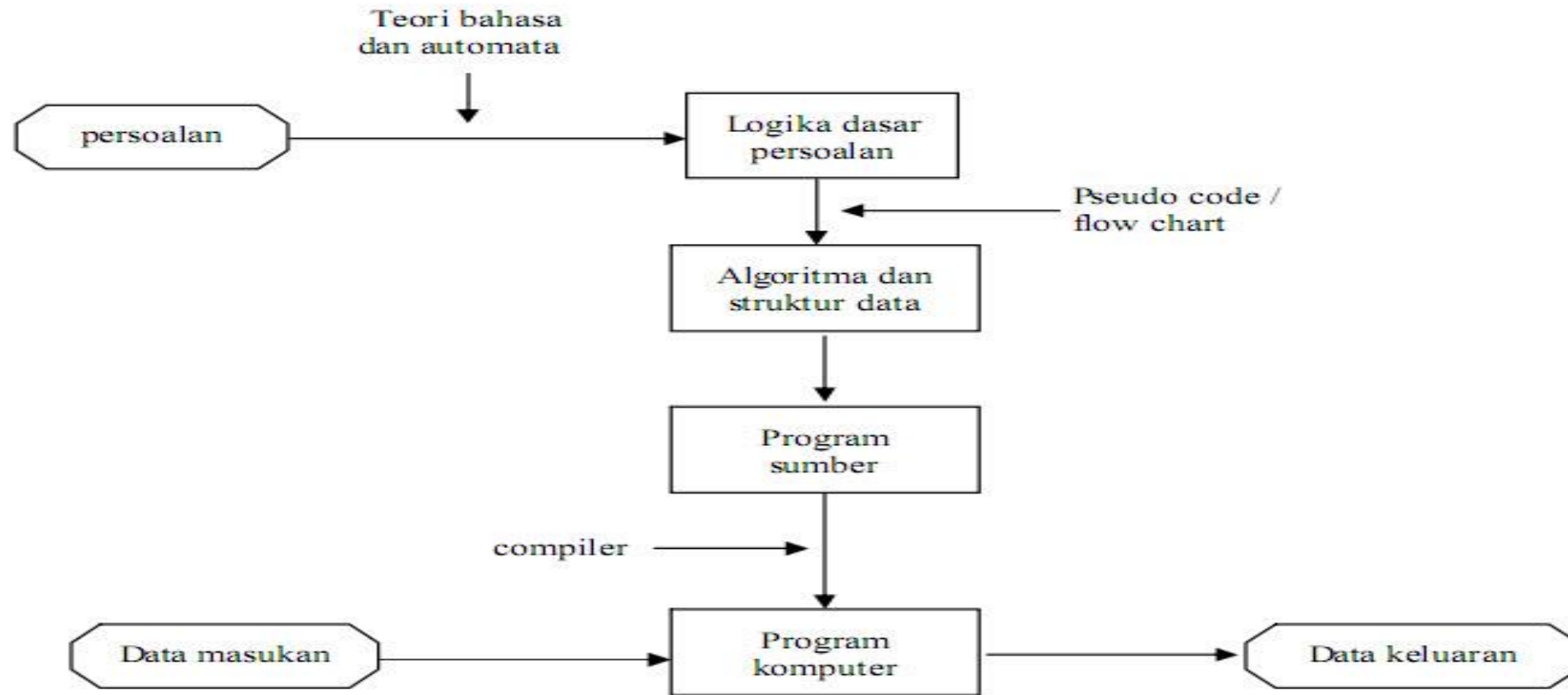
4

- **Klasifikasi Tata Bahasa**

# 1. Otomata & Teori Bahasa?

- Bagian dari teori komputasi pada ilmu komputer
- model matematika yang memiliki fungsi dari komputer digital yaitu menerima input, menghasilkan output, bisa memiliki penyimpanan sementara dan mampu membuat keputusan dalam transformasi input ke output
- Penekanannya adalah pada **pemecahan masalah**

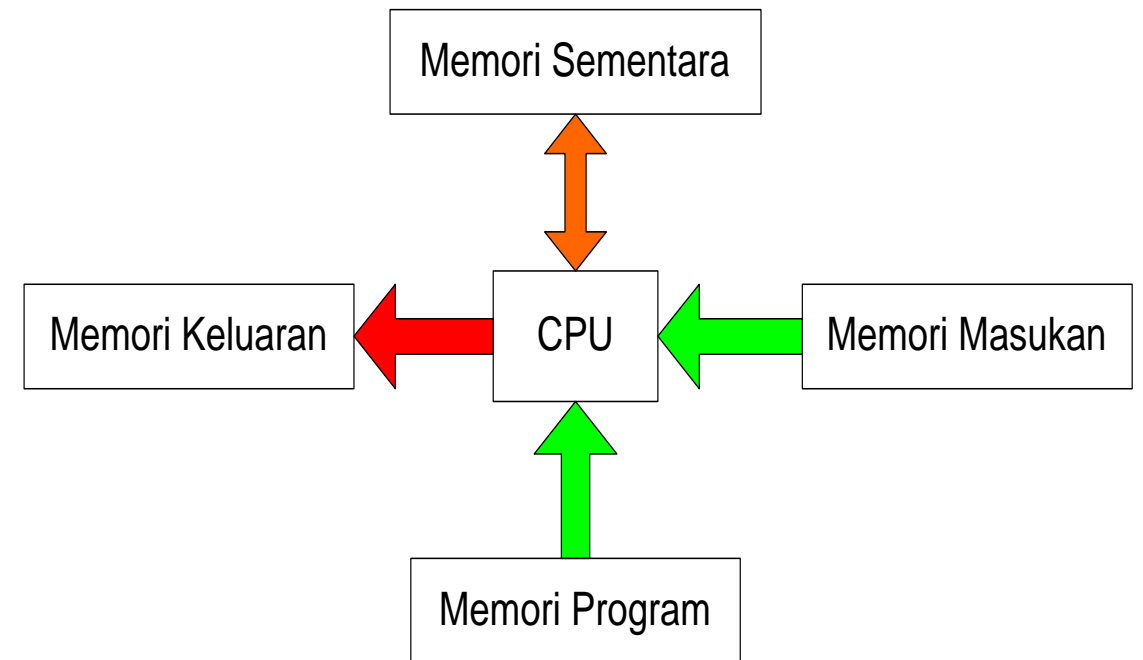
# 1. Otomata & Teori Bahasa?



# 1. Otomata & Teori Bahasa?

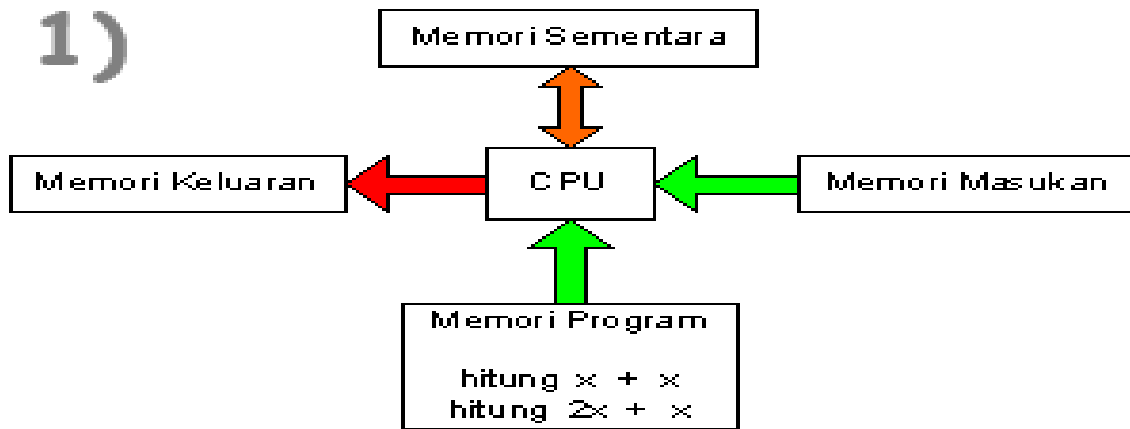
Bagaimana proses komputasi untuk :

$$f(x) = x + x + x$$

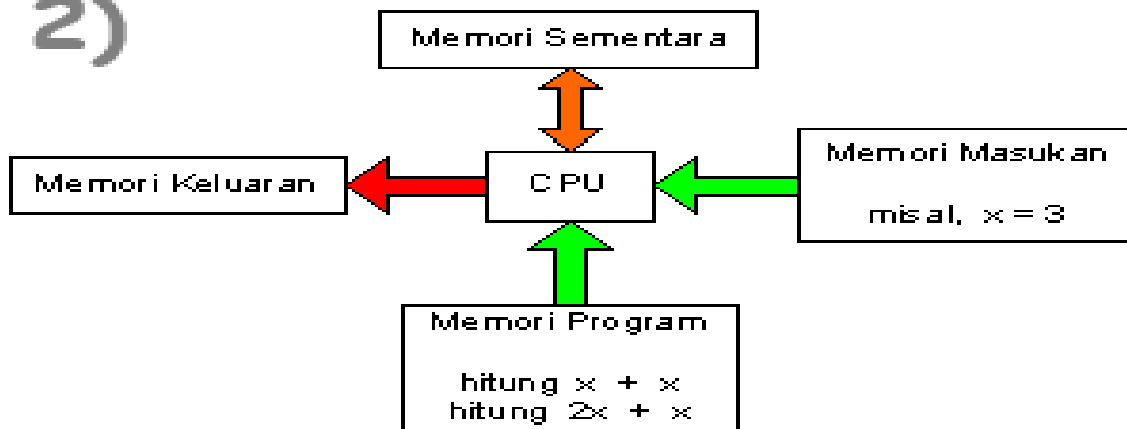


# 1. Otomata & Teori Bahasa?

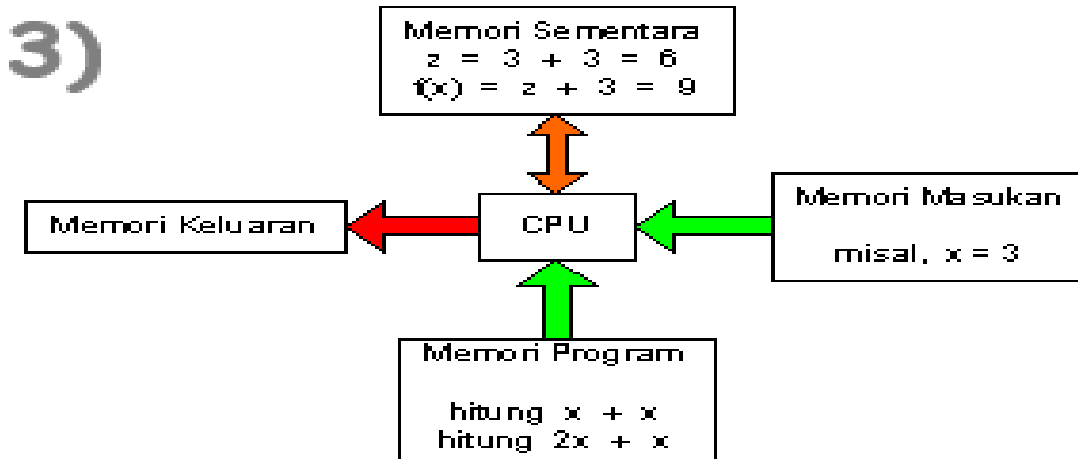
1)



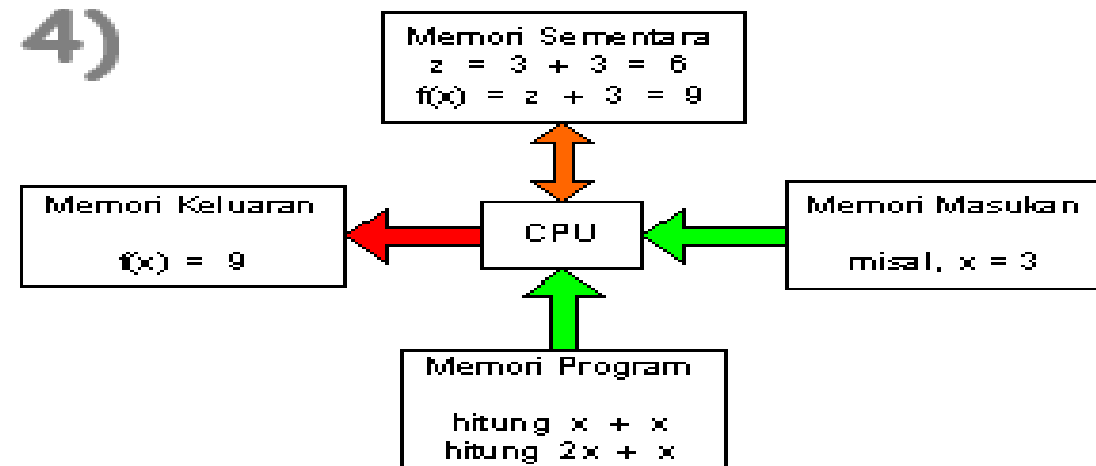
2)



3)



4)



## 2. Kedudukan OTB

**Sub bidang apapun dalam ilmu informatika pasti memiliki 2 komponen :**

1. Ide/gagasan di representasikan dalam bentuk **Model komputasi**

beberapa disiplin ilmu yang dapat di adaptasi:

- Neuro Nets → Finite Automata
- Sistem Logika Formal → Proof method
- Sistem Tata Bahasa → Psycho-Linguistic

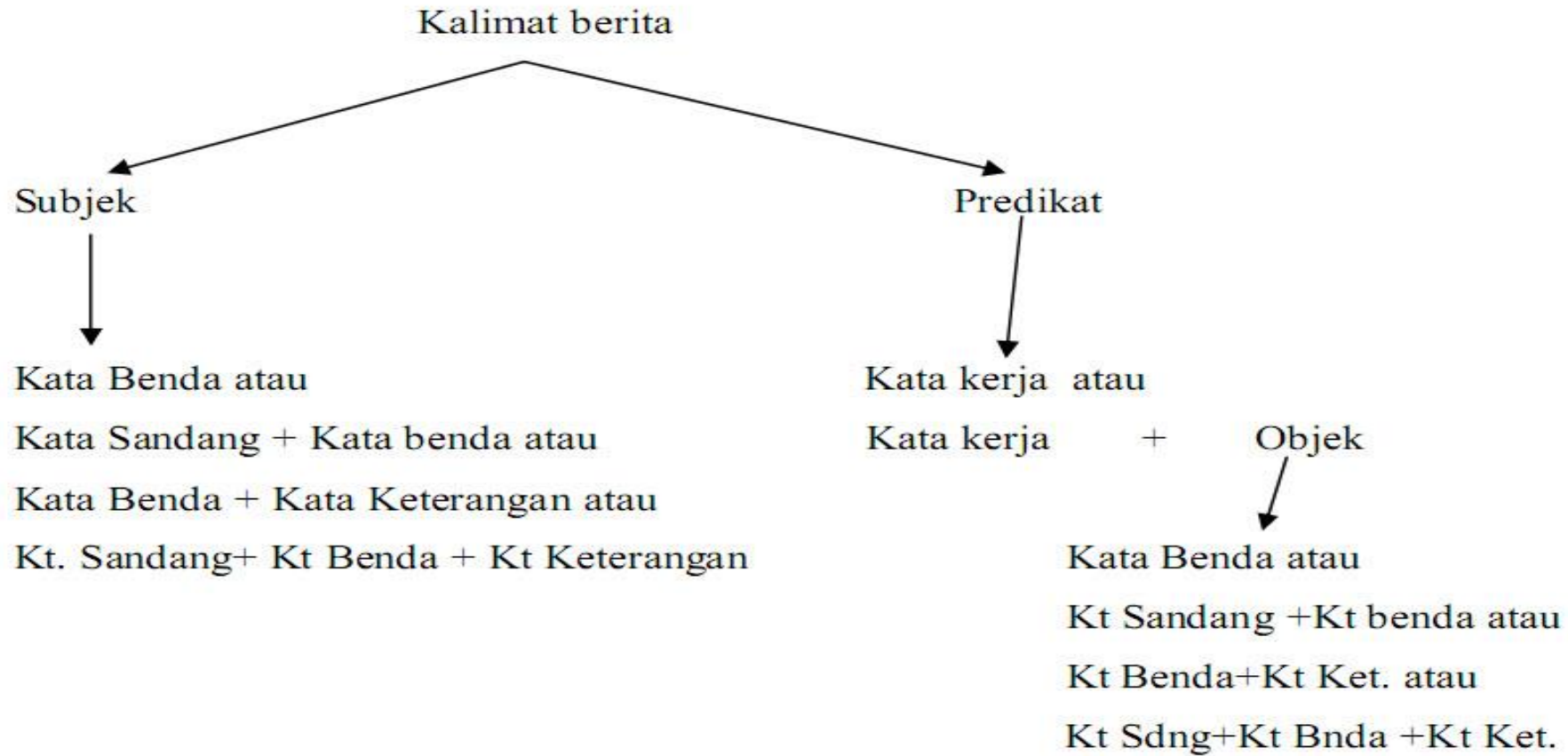
2. Teknik rekayasa untuk perancangan sistem komputasi, meliputi perangkat keras, perangkat lunak, khususnya penerapan rancangan dari teori .

**Otomata dan Teori Bahasa merupakan bagian pertama**

# 3. Tata Bahasa

- Penulisan suatu kalimat dalam sebuah bahasa, akan mengikuti suatu aturan tertentu yang berlaku pada bahasa tersebut.
- Aturan tersebut dikenal sebagai Tata Bahasa (Grammar).

# 3. Tata Bahasa



# 3. Tata Bahasa

1. Adik menulis
2. Kucing menggigit tikus
3. Si kambing cantik memakan sayuran segar

Periksalah kalimat berikut apakah memenuhi aturan tersebut :

1. Si tikus jorok mengejar kucing galak
2. Si bola besar menendang kambing jelek

# 3. Tata Bahasa

- **Tata bahasa (grammar):** kumpulan dari himpunan-himpunan variabel, simbol-simbol terminal, simbol awal yang dibatasi oleh aturan-aturan produksi.
- Pada tahun 1959 seorang ahli bernama **Noam Chomsky** membuat **model matematis** untuk mendeskripsikan Bahasa sekaligus menjawab pertanyaan ttg psycho-linguistic dan membuat perangkat formal untuk memodelkan Bahasa (disebut **Grammar**)

# 3. Tata Bahasa

Grammar  $G$  didefinisikan sebagai pasangan 4 tuple :  $V_t$  ,  $V_n$  ,  $S$ , dan  $P$ , dan dituliskan sebagai  $G(V_t, V_n, S, P)$ , dimana :

- $V_t$  : himpunan simbol-simbol terminal (alfabet) = kamus
- $V_n$  : himpunan simbol-simbol non terminal
- $S \in V$  : simbol awal (atau simbol start)
- $P$  : himpunan produksi

# 3. Tata Bahasa

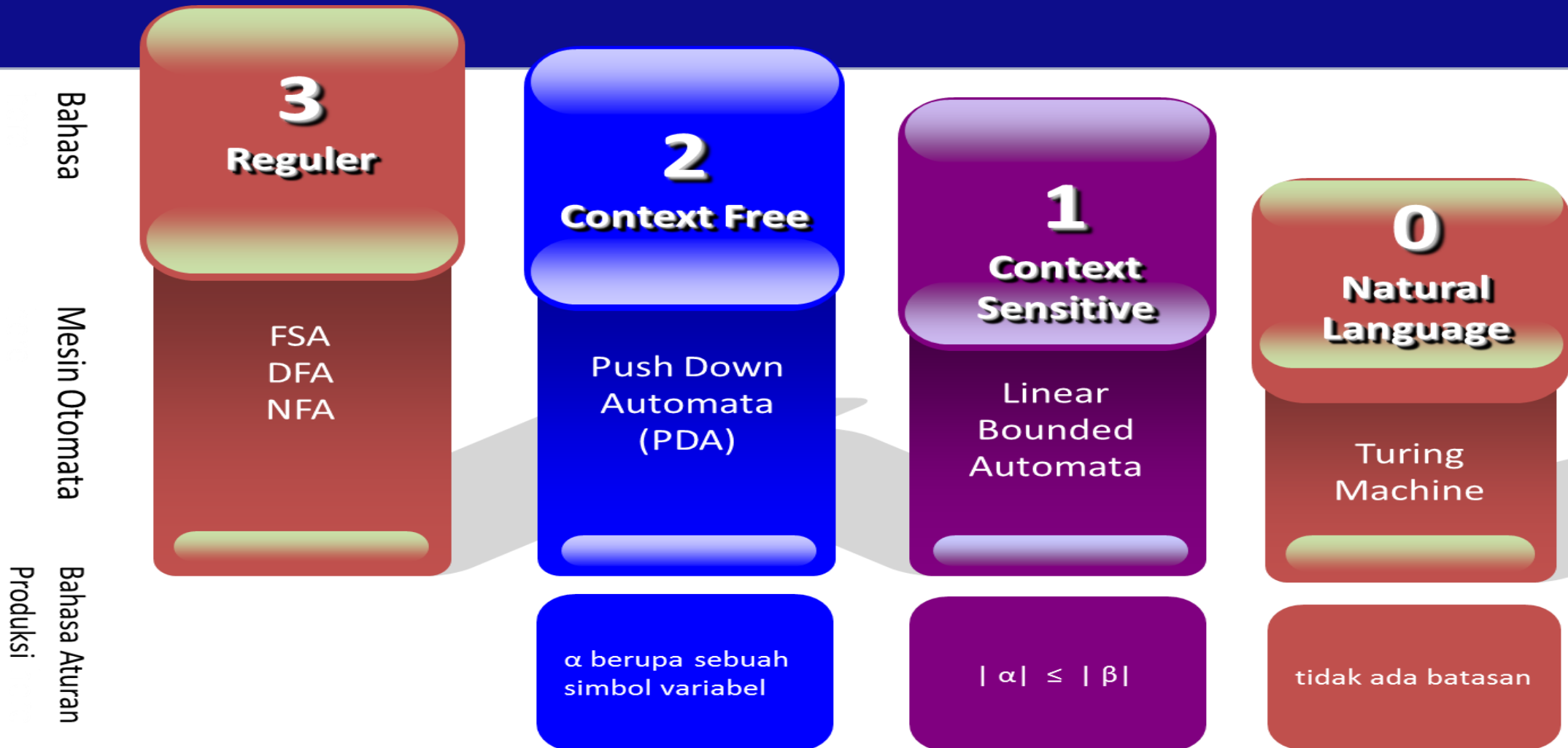
- $G1 : VT = \{I, \text{want}, \text{need}, \text{You}\}, V = \{S, A, B, C\}, P = \{S \rightarrow ABC, A \rightarrow I, B \rightarrow \text{want} \mid \text{need}, C \rightarrow \text{You}\}$

$S \rightarrow ABC$

→ IwantYou

→  $L(G1) = \{IwantYou, IneedYou\}$

# 4. Klasifikasi Tata Bahasa

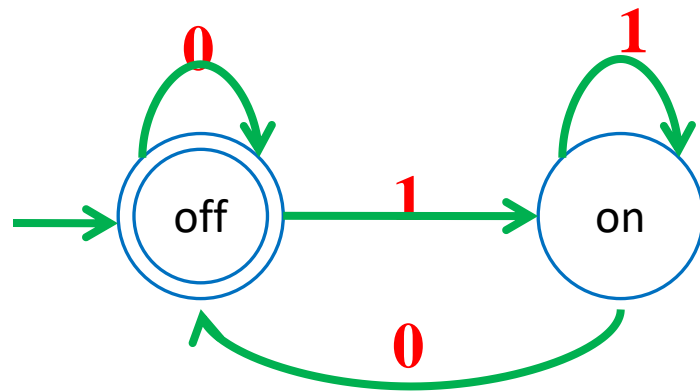


# Otomata Sederhana

---

Contoh mesin otomata sederhana :

- Otomata pada saklar listrik



- Bila mesin mendapat string input :
- 0 → diterima
- 010 → diterima
- ....???

- Sebuah string input diterima bila mencapai state akhir ( final state ) yang digambarkan dengan lingkaran ganda

# 5. Contoh Komputasi Mesin EDC (Electronic Data capture)

1. Misalnya customer belanja habis Rp.75000, state awal mesin diam.
2. Kemudian Kartu Debit digesekkan pada slot yang disediakan, lalu mesin siap menerima input nominal.
3. Setelah diinputkan nominalnya, (dalam contoh ini Rp.75000) muncul pilihan Yes atau No
4. Jika ditekan tombol No akan kembali ke state awal yaitu diam.
5. Ketika ditekan tombol Yes mesin siap menerima input Pin atau kata sandi.
6. Jika Pin yang diinputkan salah atau ditekan tombol No akan kembali ke State awal yaitu diam.
7. Jika Pin yang dimasukkan benar dan ditekan tombol Yes, mesin akan mencetak struk atau bukti transaksi sebanyak 2 lembar (1 untuk customer, 1 untuk mmini market).
8. Status mesin kembali diam.

# 5. Contoh Komputasi Mesin EDC (Electronic Data capture)

Misalkan :

total belanja Rp.75000

Pin atau kata sandi = 628

$\Sigma = \{\text{gesek}, 75000, Y, N, 6, 2, 8\}$

$Q = \{A, B, C, D, E, F, G, H\}$

$S = \{A\}$

$F = \{A, H\}$

Fungsi Transisi :

$\Delta = \{((A, \text{gesek}), B), ((B, 75000), C), ((C, N), A), ((C, Y), D),$   
 $((D, N), A), ((D, Y), A), ((D, 6), E), ((E, N), A), ((E, Y), A), ((E, 2), F), ((F, N), A),$   
 $((F, Y), A), ((F, 8), G), ((G, N), A), ((G, Y), H)\}$



# Konsep Bahasa

- **Abjad**
- **String**
- **Bahasa (Language)**
- **Bahasa Kosong**
- **Bahasa Universal dari  $\Sigma$**

# Konsep Bahasa

## ➤ Abjad

- Sebuah himpunan berhingga tak kosong dari simbol – simbol.
- Notasi :  $\Sigma$
- $\Sigma = \{a, b, c, \dots, x, y, z\}$  (Roman alphabet)
- $\Sigma = \{0, 1, \dots, 9\}$
- $\Sigma = \{0, 1\}$  (binary alphabet)

# Konsep Bahasa

## ➤ Bahasa (Language)

- Kumpulan dari string. Notasi : L

- $\Sigma = \{1, 2, 3, 4, 5\}$ ,

- Kumpulan  $\{1, 12, 123, 1234, 12345, 112\}$  adalah bahasa dari abjad tersebut

- $\Sigma = \{1\}$ ,

- Kumpulan  $\{1, 11, 111, 1111, \dots\}$  adalah sebuah Bahasa.

## ➤ Bahasa Kosong

- Suatu bahasa yang tidak terdiri dari string. Notasi :  $\{\}$  atau  $\emptyset$

# Konsep Bahasa

- Bahasa universal dari  $\Sigma$ 
  - Bahasa yang terdiri dari semua string berdasarkan suatu abjad  $\Sigma$
  - Notasi  $\Sigma^*$
  - $\Sigma = \{1\}$ , maka  $\Sigma^* = \{\epsilon, 1, 11, 111, 1111, \dots\}$

# Aturan Produksi

Aturan produksi  $\alpha \rightarrow \beta$  yang diterapkan pada suatu string  $w = a\alpha c$  mengganti kemunculan  $\alpha$  menjadi  $\beta$ , sehingga string tersebut berubah menjadi  $w = a\beta c$ , sehingga dapat dituliskan

$$aac \Rightarrow a\beta c \text{ (} aac \text{ memproduksi } a\beta c \text{).}$$

➤ Produksi tersebut dapat diterapkan berkali-kali

$$w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \dots \Rightarrow w_n$$

➤ atau dapat dituliskan

$$w_1 \Rightarrow^* w_n$$

➤ jika minimal harus ada 1 aturan produksi yang diterapkan :

$$w_1 \Rightarrow^+ w_n$$

# Aturan Produksi

Contoh 1

Tatabahasa  $G = \{ \{S\}, \{a,b\}, S, P \}$  dengan aturan produksi  $P$  adalah

$$S \rightarrow aSb$$

$$S \rightarrow \varepsilon$$

maka dapat dihasilkan suatu string

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

sehingga dapat dituliskan

$$S \Rightarrow^* aabb$$

Bahasa yang dihasilkan dari tatabahasa tersebut adalah

$$L(G) = \{ \varepsilon, ab, aabb, aaabbb, aaaabbbb, \dots \}$$

atau dapat pula dituliskan

$$L(G) = \{ a^n b^n \mid n \geq 0 \}$$

# Aturan Produksi

Tata bahasa  $G = \{\{S,A\}, \{a,b\}, S, P\}$  dengan aturan produksi  $P$  adalah

$S \rightarrow Ab$

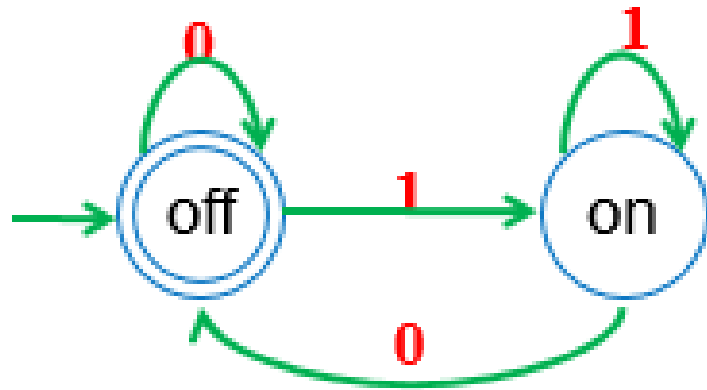
$A \rightarrow aAb$

$A \rightarrow \varepsilon$

tuliskan string apa yang di hasilkan pada aturan produksi di atas!

# Otomata Sederhana

- Contoh mesin otomata sederhana :
  - Otomata pada saklar listrik

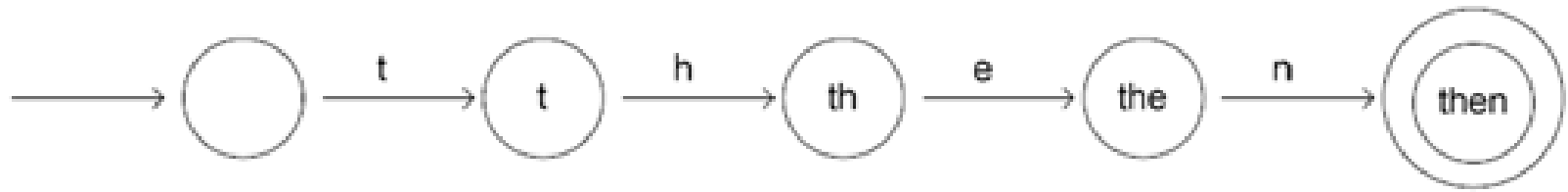


- Bila mesin mendapat string input :
- 0 → diterima
- 010 → diterima
- .....???

- Sebuah string input diterima bila mencapai state akhir ( final state ) yang digambarkan dengan lingkaran ganda

# Otomata Sederhana

- Finite automaton berikut dapat dinyatakan sebagai bagian dari lexical analyzer.



- Vending Machine