

OTOMATA 01

PENGERTIAN OTOMATA DAN KONSEP BAHASA

Dosen : Arya Yudhi W, S.Kom, M.Kom.

Email : arya@if.its.ac.id

APA ITU OTOMATA?

Otomata adalah MODEL.
Model dari sistem yang akan kita komputasikan.

Semua bentuk sistem, diskrit, kontinu, hybrid (gabungan event diskrit dan kontinu dalam satu sistem) dapat dimodelkan oleh teori otomata.

Salah satu penggunaan otomata adalah untuk kompilasi. Kompilasi adalah ilmu yang mempelajari bagaimana kita dapat merancang & membangun bahasa pemrograman.

PENDAHULUAN

Komputasi menjadi isu penting karena mempelajari bagaimana kita dapat **merancang mesin** yang mampu melakukan proses-proses intelektual (yang mulanya hanya dapat dilakukan manusia)

Namun dalam merancang seringkali kita terkendala dengan berbagai macam batasan. Tetapi apakah benar jika batasan-batasan (yang dimiliki komputer) pada dasarnya disebabkan oleh kelemahan programmer (manusia), bukan batasan intrinsik yang dimiliki mesin/komputer.

Jika Ya, maka kita berharap agar batasan-batasan tersebut dapat tereduksi melalui pengembangan **teori komputasi**.

PENDAHULUAN

Sub bidang apapun dalam ilmu informatika pasti memiliki 2 komponen :

1. Ide/gagasan dirupakan ke dalam bentuk **MODEL KOMPUTASI**

Beberapa disiplin ilmu yang diadopsi :

■ Neuron Nets	➔	Finite Automata
■ Sistem Logika Formal	➔	Proof Methods
■ Sistem Tata Bahasa	➔	Psycho-Linguistic:

1. Apakah arti bahasa itu ?
2. Bagaimana manusia mengembangkan bahasa ?
3. Bagaimana manusia memahami bahasa ?
4. Bagaimana manusia mengajarkan bahasa ke anak-anaknya ?
5. Bagaimana cara menyatakan gagasan ?
6. Bagaimana manusia membangun kalimat dari gagasan yang ada dalam pikirannya ?

PENDAHULUAN



2. Teknik rekayasa untuk mengimplementasikan model ke dalam sebuah bentuk sistem yang terkomputasi (programming/coding)

PENDAHULUAN

Noam Chomsky, ▶ membuat **model matematis** untuk mendeskripsikan bahasa sekaligus menjawab pertanyaan ttg psycho-linguistic
▶ membuat perangkat formal untuk memodelkan properti bahasa (disebut **Grammar**)



McCulloch & Pitts, merancang **Finite Automata** untuk memodelkan neuron nets



Stephen Kleene, menemukan model representasi lain dari automata melalui Regular Expression



Alan Turing, menemukan model untuk mengidentifikasi apakah sebuah permasalahan dapat dikomputasi ➔ **Mesin Turing**

PENDAHULUAN

3 model mesin komputasi yang akan kita pelajari dalam otomata :

1. Finite Automata (FA)

(sejauh ini) telah dimanfaatkan untuk merancang lexical analyzer, aplikasi editor teks, pengenalan pola, fault tolerant system, dll

2. Pushdown Automata (PDA)

(sejauh ini) telah dimanfaatkan untuk mengenali bahasa yang berstruktur context- free grammar, kamus data, query, script, parsing, dll

3. Turing Machine (TM)

mesin turing dapat dimanfaatkan untuk mengidentifikasi ketidakmungkinan penulisan sebuah program komputer. Sejauh ini kita dapat meyakini bahwa jika suatu persoalan tidak dapat dimodelkan oleh mesin turing, maka persoalan tersebut tidak akan mungkin dapat diselesaikan secara komputatif oleh mesin komputasi apapun!

TERMINOLOGI BAHASA

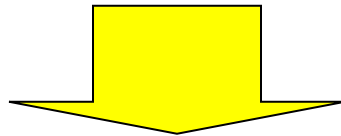
Manfaat bahasa adalah sebagai media komunikasi yang menggunakan sekumpulan simbol dan dikombinasikan menurut aturan sintaksis tertentu (grammar).

Sementara Semantik bahasa mendefinisikan bagaimana sebuah kalimat dapat diinterpretasikan/diartikan secara benar (sesuai dengan grammar-nya).

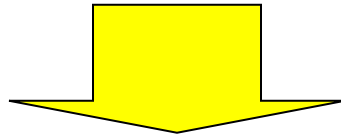
Terminologi penting di dalam memahami teori bahasa adalah pemahaman terhadap **alphabet** dan **string**.

TERMINOLOGI BAHASA

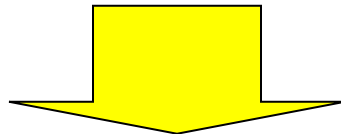
Bahasa = { Kalimat }



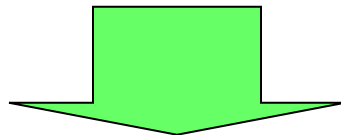
Kalimat = { Kata atau String }



Kata atau String = { Karakter }



Karakter = { Alphabet \cup Digit \cup Symbol }



Bahasa = { { { Alphabet \cup Digit \cup Symbol } } }

TERMINOLOGI BAHASA

Misal terdapat sebuah himpunan alfabet $\Sigma = \{x\}$
dan misalkan akan didefinisikan sebuah bahasa L_1

$$L_1 = \{ x, xx, xxx, xxxx, \dots \}$$

Maka L_1 dapat dinyatakan secara formal sebagai

$$L_1 = \{ x^n, \text{ untuk } n = 1, 2, 3, \dots \}$$

Atau, didefinisikan sebuah bahasa L_2

$$L_2 = \{ x, xxx, xxxxx, \dots \}$$

secara formal, L_2 dapat dinyatakan sebagai

$$L_2 = \{ x^n, \text{ untuk } n = 1, 3, 5, \dots \}$$

OPERASI PADA BAHASA

Misalkan terdapat 2 himpunan karakter sebarang L dan M. Maka operasi² yang dapat dilakukan terhadap kedua himpunan tersebut antara lain adalah :

No	Nama Operasi	Simbol	Keterangan
1	UNION	$L \cup M$	$\{ s \mid s \text{ ada di } L \text{ atau } M \}$
2	CONCATENATION	LM	$\{ st \mid s \text{ ada di } L \text{ dan } t \text{ ada di } M \}$
3	KLEENE CLOSURE	L^*	$L^* = \bigcup_{i=0}^{\infty} L^i$ (penggabungan nol atau lebih L)
4	POSITIVE CLOSURE	L^+	$L^+ = \bigcup_{i=1}^{\infty} L^i$ (penggabungan satu atau lebih L)
5	REVERSE of STRING	$\text{Rev}(x)$	Sebuah string x yang ditulis dalam urutan terbalik
6	LENGTH of STRING	$\text{Length}(x)$	Menghitung jumlah karakter pada sebuah string x
7	PALINDROME	$x = \text{Rev}(x)$	Rangkaian karakter dalam sebuah string x yang ditulis dalam urutan terbalik tetap menghasilkan string x

OPERASI PADA BAHASA

Contoh :

Misal terdapat himpunan string

$S = \{ a, aa, aaa \}$ dan $T = \{ bb, bbb \}$

Maka,

$$S \cup T = \{ a, aa, aaa, bb, bbb \}$$

$$S T = \{ abb, abbb, aabb, aabbb, aaabb, aaabbb \}$$

Contoh :

Misal terdapat sebuah himpunan alphabet $\Sigma = \{ 0, 1 \}$

Maka $\Sigma^* = \{ \lambda, 0, 1, 00, 01, 10, 11, 000, \dots \}$

Contoh :

Misal terdapat sebuah himpunan alphabet $\Sigma = \{ x \}$

Maka $\Sigma^+ = \{ x, xx, xxx, \dots \}$

OPERASI PADA BAHASA

Definisi Rekursif

Secara umum, definisi rekursif dapat diartikan sebagai upaya untuk mendefinisikan sebuah himpunan melalui penerapan (sebuah/sekelompok) aturan secara berulang-ulang.

Mendefinisikan himpunan melalui **Definisi Rekursif** akan melibatkan *three-steps process*, yaitu :

1. Menentukan **obyek dasar** dari himpunan yang akan dibentuk
2. Menentukan **aturan²** untuk membangkitkan obyek² lain dalam himpunan tersebut
3. Mendeklarasikan bahwa tidak ada obyek² lain yang dapat dibentuk, kecuali menggunakan aturan di atas.

OPERASI PADA BAHASA

Contoh : Misalkan kita akan mendefinisikan himpunan bilangan genap. Melalui definisi rekursif, masalah di atas dapat didekati dengan tahapan penyelesaian seperti berikut :

1. 2 adalah anggota himpunan bilangan genap (*2 adalah sebuah obyek dasar*)
2. Jika x adalah anggota dalam himpunan tersebut, maka $x + 2$ juga merupakan anggota himpunan bilangan genap (*$x + 2$ adalah aturan yang dapat memproduksi anggota himpunan bilangan genap yang lain seperti 4, 6, 8, dst*)
3. Jika y adalah anggota himpunan bilangan genap, maka y pasti dihasilkan dari aturan di atas (*deklarasi bahwa $x + 2$ adalah satu²nya aturan untuk memproduksi elemen himpunan bilangan genap*)

OPERASI PADA BAHASA

Contoh :

Misal terdapat bahasa $L_1 = \{ \lambda, x, xx, xxx, \dots \}$. Melalui Definisi Rekursif, bahasa L_1 dapat dinyatakan melalui cara berikut :

1. λ adalah anggota L_1
2. Jika Q adalah sebarang string di L_1 , maka xQ juga merupakan string dalam L_1

Contoh :

Misal terdapat bahasa $L_2 = \{ x, xxx, xxxxx, \dots \}$. Melalui Definisi Rekursif, bahasa L_2 dapat dinyatakan melalui cara berikut :

1. x adalah anggota L_2
2. Jika Q adalah sebarang string di L_2 , maka xxQ juga merupakan string dalam L_2

METODE PENDEFINISIAN BAHASA

Jumlah alphabet dan digit memang berhingga (*finite*).
Tetapi, jumlah string/kata dan kalimat yang dapat dibentuk dari kombinasi alphabet dan digit tersebut bisa tak berhingga (*infinite*) !!!

Oleh karenanya, untuk mendefinisikan sebuah bahasa, umumnya tidak dilakukan dengan cara *me-listing* semua string dan kalimat yang dimiliki oleh bahasa tersebut, melainkan dengan mengemukakan syarat² yang dimiliki oleh bahasa yang bersangkutan.

Dengan kata lain, karena bahasa adalah suatu bentuk himpunan, maka cara mengekspresikan himpunan yang paling praktis adalah melalui **set theory notation**.

METODE PENDEFINISIAN BAHASA

Metode untuk mendefinisikan bahasa secara berhingga (untuk bahasa yang tidak berhingga) adalah melalui :

- **Grammar**

kita dapat membangun sebuah kalimat dengan sintaksis yang benar sesuai dengan kaidah yang telah ditetapkan pada serangkaian aturan yang disebut **production(s)**

- **Recognizer** atau **Finite Automata**

diberikan sebuah input string, maka recognizer akan melakukan penelusuran karakter per karakter untuk mengetahui apakah input string tersebut merupakan anggota suatu bahasa tertentu atau tidak

METODE PENDEFINISIAN BAHASA

Namun dalam kenyataannya, kedua metode tersebut sebenarnya teraplikasi untuk tujuan yang berbeda :

- **Grammar**
lebih berfungsi sebagai pembangkit string dan sentence.
- **Recognizer** atau **Finite Automata**
sesuai namanya, recognizer lebih berfungsi sebagai pengenal string atau sentence.

TUGAS MINGGUAN

1. Misal terdapat bahasa S^* dengan $S = \{a, b\}$
terdapat berapa banyak kata dalam bahasa di atas yang memiliki :
 - a. $\text{Length}(2)$
 - b. $\text{Length}(3)$
 - c. $\text{Length}(n)$
2. Misal terdapat bahasa S^* dengan $S = \{a, ab, ba\}$
 - a. apakah string (abbba) adalah anggota bahasa di atas?
 - b. Adakah cara lain untuk menjelaskan bahasa di atas (yang pasti bukan bahasa dengan string yang tidak mengandung bbb)?
3. Misalkan terdapat sebuah himpunan string $S = \{a, bb, bab, abaab\}$
 - a. Apakah abbabaabab dan abaabbabbaabb terdapat dalam S^* ?
 - b. Adakah string pada S^* yang memiliki karakter b berjumlah ganjil ?
4.
 - a. Misal $S = \{ab, bb\}$ dan $T = \{ab, bb, bbbb\}$. Tunjukkan bahwa $S^* = T^*$
 - b. Misal $S = \{ab, bb\}$ dan $T = \{ab, bb, bbb\}$. Tunjukkan bahwa $S^* \neq T^*$, tetapi $S^* \subset T^*$

TUGAS MINGGUAN

- 5.a. Buktikan jika x ada dalam palindrome, maka x^n juga terdapat dalam palindrome tersebut.
- b. Buktikan bahwa pada sebarang himpunan palindrome, jumlah string dengan $\text{length}(3)$ akan sama dengan jumlah string dengan $\text{length}(4)$
- 6. Buktikan bahwa untuk semua himpunan S : $(S^+)^* = (S^*)^+$
- 7. Buatlah Definisi Rekursif untuk :
 - a. Himpunan bilangan ganjil $\{ 1, 3, 5, 7, \dots \}$
 - b. Bahasa S^* dengan $S = \{ aa, b \}$
 - c. Himpunan The Powers_of_Two = $\{ 1, 2, 4, 8, 16, \dots \}$