

# Introduction

# Teori Bahasa Dan Otomata

- Teori bahasa membicarakan bahasa formal (*formal language*), terutama untuk kepentingan perancangan kompilator (*compiler*) dan pemroses naskah (*text processor*).
- Bahasa formal adalah kumpulan *kalimat*. Semua kalimat dalam sebuah bahasa dibangkitkan oleh sebuah tata bahasa (*grammar*) yang sama.
- Sebuah bahasa formal bisa dibangkitkan oleh dua atau lebih tata bahasa berbeda.
- Dikatakan bahasa formal karena grammar diciptakan mendahului pembangkitan setiap kalimatnya.
- Bahasa Natural/manusia bersifat sebaliknya; grammar diciptakan untuk meresmikan kata-kata yang hidup di masyarakat. Dalam pembicaraan selanjutnya 'bahasa formal' akan disebut 'bahasa' saja.
- Otomata adalah mesin abstrak yang dapat mengenali (*recognize*), menerima (*accept*), atau membangkitkan (*generate*) sebuah kalimat dalam bahasa tertentu.

# Definisi

**Teori Otomata** adalah teori mengenai mesin-mesin abstrak, dan berkaitan erat dengan teori [bahasa formal](#). ada beberapa hal yang berkaitan dengan Otomata, yaitu Grammar. Grammar adalah bentuk abstrak yang dapat diterima (accept) untuk membangkitkan suatu kalimat otomata berdasarkan suatu aturan tertentu.

# Teori Automata Meliputi / mencakup

- Teori Bahasa Formal
- Regular Expression
- Finite Automata
- Non Deterministic Finite Automata
- Finite Automata dengan Output
- Context-free Grammar
- Pushdown Automata

# Pendahuluan

## Apa itu Komputer

- Alat untuk mengetik ?
- Komputer sangat kompleks - menerapkan teori matematika secara langsung (*computational model*)
  - ❑ Model dasar perhitungan
  - ❑ Rekayasa merancang sistem komputer H/W + S/W

# Pengertian Program Sebagai Obyek Matematika

- Diberlakukannya alasan (reason), logika (logic) dan aturan (rule) untuk kombinasi
- Apakah program mengoperasikan bilangan, teks atau apapun

# Hardware/Software

- Kekompleksan hardware dan software ditangani oleh processor untuk mengkonversi program kedalam bentuk execute (eksekusi)
- Execute file dapat dibentuk melalui :
  - ☐ Compile
  - ☐ Assembly
  - ☐ Dan sebagainya (interpreter)

# Beberapa Pengertian Dasar

- Simbol adalah sebuah entitas abstrak (seperti halnya pengertian *titik* dalam geometri). Sebuah huruf atau sebuah angka adalah contoh simbol.
- String adalah deretan terbatas (*finite*) simbol-simbol. Sebagai contoh, jika  $a$ ,  $b$ , dan  $c$  adalah tiga buah simbol maka  $abcb$  adalah sebuah string yang dibangun dari ketiga simbol tersebut.
- Jika  $w$  adalah sebuah string maka panjang string dinyatakan sebagai  $|w|$  dan didefinisikan sebagai cacahan (banyaknya) simbol yang menyusun string tersebut. Sebagai contoh, jika  $w = abcb$  maka  $|w| = 4$ .
- String hampa adalah sebuah string dengan nol buah simbol. String hampa dinyatakan dengan simbol  $\varepsilon$  (atau  $\wedge$ ) sehingga  $|\varepsilon| = 0$ . String hampa dapat dipandang sebagai simbol hampa karena keduanya tersusun dari nol buah simbol.
- Alfabet adalah himpunan hingga (*finite set*) simbol-simbol



# Operasi Dasar String (1)

- Prefik string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *no*/ atau lebih simbol-simbol paling belakang dari string  $w$  tersebut.

Contoh :  $abc$ ,  $ab$ ,  $a$ , dan  $\varepsilon$  adalah semua Prefix( $x$ )

- ProperPrefix string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *satu* atau lebih simbol-simbol paling belakang dari string  $w$  tersebut.

Contoh :  $ab$ ,  $a$ , dan  $\varepsilon$  adalah semua ProperPrefix( $x$ )

- Postfix (atau Sufix) string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *no*/ atau lebih simbol-simbol paling depan dari string  $w$  tersebut.

Contoh :  $abc$ ,  $bc$ ,  $c$ , dan  $\varepsilon$  adalah semua Postfix( $x$ )

# Operasi Dasar String (2)

- ProperPostfix (atau PoperSufix) string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *satu* atau lebih simbol-simbol paling depan dari string  $w$  tersebut.  
Contoh :  $bc$ ,  $c$ , dan  $\varepsilon$  adalah semua ProperPostfix( $x$ )
- Head string  $w$  adalah simbol paling depan dari string  $w$ .  
Contoh :  $a$  adalah Head( $x$ )
- Tail string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan simbol paling depan dari string  $w$  tersebut.  
Contoh :  $bc$  adalah Tail( $x$ )

# Operasi Dasar String (3)

- Substring string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *no*/ atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string  $w$  tersebut.

Contoh :  $abc, ab, bc, a, b, c$ , dan  $\epsilon$  adalah semua Substring( $x$ )

- ProperSubstring string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *satu* atau lebih simbol-simbol paling depan dan/atau simbol-simbol paling belakang dari string  $w$  tersebut.

Contoh :  $ab, bc, a, b, c$ , dan  $\epsilon$  adalah semua Substring( $x$ )

- Subsequence string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *no*/ atau lebih simbol-simbol dari string  $w$  tersebut.

Contoh :  $abc, ab, bc, ac, a, b, c$ , dan  $\epsilon$  adalah semua Subsequence( $x$ )

# Operasi Dasar String (4)

- Proper Subsequence string  $w$  adalah string yang dihasilkan dari string  $w$  dengan menghilangkan *satu* atau lebih simbol-simbol dari string  $w$  tersebut.

Contoh :  $ab, bc, ac, a, b, c$ , dan  $\varepsilon$  adalah semua Subsequence( $x$ )

- Concatenation adalah penyambungan dua buah string. Operator concatenation adalah *concat* atau tanpa lambang apapun.

Contoh :  $\text{concat}(xy) = xy = abc123$

- Alternation adalah pilihan satu di antara dua buah string. Operator alternation adalah *alternate* atau  $|$ .

Contoh :  $\text{alternate}(xy) = x | y = abc \text{ atau } 123$

- Kleene Closure :  $x^* = \varepsilon | x | xx | xxx | \dots = \varepsilon | x | x | x | \dots$
- Positive Closure :  $x^+ = x | xx | xxx | \dots = x | x | x | \dots$