



# Turing Machines

Kuliah Teori Bahasa dan Automata  
Program Studi Ilmu Komputer  
Fasilkom UI

Prepared by:  
Suryana Setiawan

# Spesifikasi: 2 sifat penting

- Mampu mendeskripsikan segala komputasi.
  - Setingkat komputer tetapi tidak terbatas FSM/PDA
- Sederhana, agar penjelasan formal dapat dilakukan.
  - Sesederhana FSM/PDA, tapi tidak seperti sekompleks komputer

# Manfaat dan Masalah Stack

- Adanya stack meningkatkan secara signifikan kemampuan suatu FSM, dari hanya mengenali Bahasa Reguler ke bahasa Context Free.
  - Namun, mekanisme LIFO pada stack masih membatasi kemampuan komputasinya.
- Perlu struktur yang menggantikan stack sehingga
  - memungkinkan mengakses isi storage tersebut secara lebih fleksibel.
  - memungkinkan akses suatu data dalam storage tanpa mengganggu data pada posisi lainnya.

# Definisi Intuitif: Turing Machines

- Bayangkan suatu FSM yang dilengkapi storage berbentuk tape
  - **Tape** berbentuk **linear**, setiap posisi, atau **square**, terurut dari kiri ke kanan,
  - Setiap posisi dapat menyimpan **satu simbol tape** atau **kosong** ( $\square$ ),
  - **Kapasitas** (panjang tape) tak berhingga (tidak berujung baik di kiri maupun di kanan).
- **Read/write** dilakukan melalui sebuah **head**
  - Head bergerak secara sikluensial dari satu posisi ke posisi berikutnya (arah R)/sebelumnya (arah L).

# Mekanisme Komputasi

- Komputasi dilakukan menurut **current state**, **current data** tape (pada head), untuk bertransisi ke **next-state**, meng-**update isi tape** (pada head), **arah pemindahan head**.
- Dengan adanya tape, input string bisa diasumsikan sudah langsung ditaruh di dalam tape, sehingga mesin segera bekerja pada tape.
- Di konfigurasi awal dibuat **konvensi**: head berada di **posisi kosong tepat sebelum simbol terkiri string input** pada tape.

# Definisi Formal: Turing Machines

- Suatu Turing Machine  $M$  adalah 6-tuple  $(K, \Sigma, \Gamma, \delta, s, H)$ , dimana:
  - $K$ : himpunan terbatas **status**.
  - $\Sigma$ : **alfabet input** yang tidak berisi  $\square$
  - $\Gamma$ : **alfabet tape**, termasuk  $\Sigma$  dan  $\square$
  - $s$ : **status mulai**
  - $H$ : himpunan **status halting**,  $H \subseteq K$
  - $\delta$ : **fungsi transisi**, yang memetakan:  
$$(K-H) \times \Gamma \rightarrow K \times \Gamma \times \{\rightarrow, \leftarrow\}$$

non-halting tape symbol   state   tape symbol aksi head
- Note: definisi TM deterministik

# Fungsi Transisi $\delta$

- Suatu transisi  $((q_0, a), (q_1, b, A)) \in \delta$ 
  - menyatakan dari
    - **current state**  $q_0$ ,
    - saat head **membaca**  $a$  dari tape (pada posisi head),
  - Bertransisi menjadi
    - **Next state**  $q_1$ ,
    - sambil mengubah tape (pada posisi head) **menjadi**  $b$ , serta **bergerak ke arah**  $A$ , yaitu satu posisi ke kiri (“ $\leftarrow$ ”) atau ke kanan (“ $\rightarrow$ ”).
- $\delta$  adalah fungsi, berarti mesin bersifat **deterministik**, dan setiap kemungkinan  $(q, a)$  harus terdefinisi.
  - Karena sejumlah  $(q, a)$  tidak akan terjadi, untuk kejelasan pembacaan, seringkali itu disembunyikan.

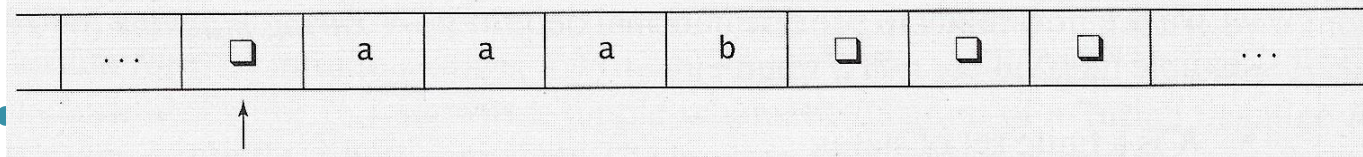
# Pengertian “simbol $\square$ ”

- Setiap posisi tape “berisi satu simbol tape”  $x \in \Gamma$  ( $x \neq \square$ ), atau “kosong”.
  - Jika disebut kosong, maka kenyataan yang sebenarnya, posisi tape tersebut berisi simbol “ $\square$ ” (blank).
  - Jika disebutkan tape hanya berisi string  $\alpha$ , maka kenyataan yang sebenarnya, yang lainnya berisi “ $\square$ ”.
- **Active tape:** isi tape dari non-blank ter kiri hingga non-blank terkanan, di tambah beberapa blank di luar itu yang terkait beroperasinya mesin tsb.
  - Selama komputasi, bagian tape yang kosong yang lain tidak perlu diperhatikan.

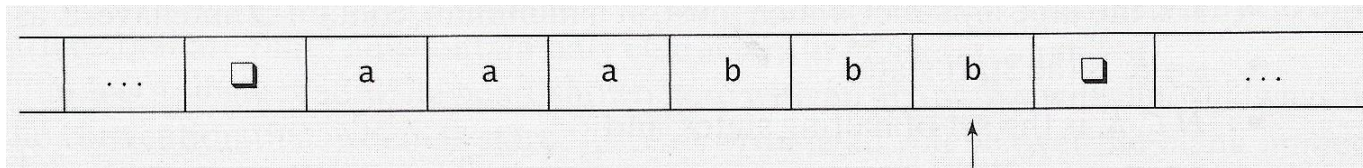


# Contoh (Deskripsi Masalah)

- TM  $M$  yang dapat memproses input string dari  $\{a^i b^j : 0 \leq j \leq i\}$  dengan menambahkan sejumlah  $b$  di belakang string agar menjadi  $a^i b^i$ .
- Saat mulai, isi tape sebagai berikut:



- Saat halt isi tape menjadi:



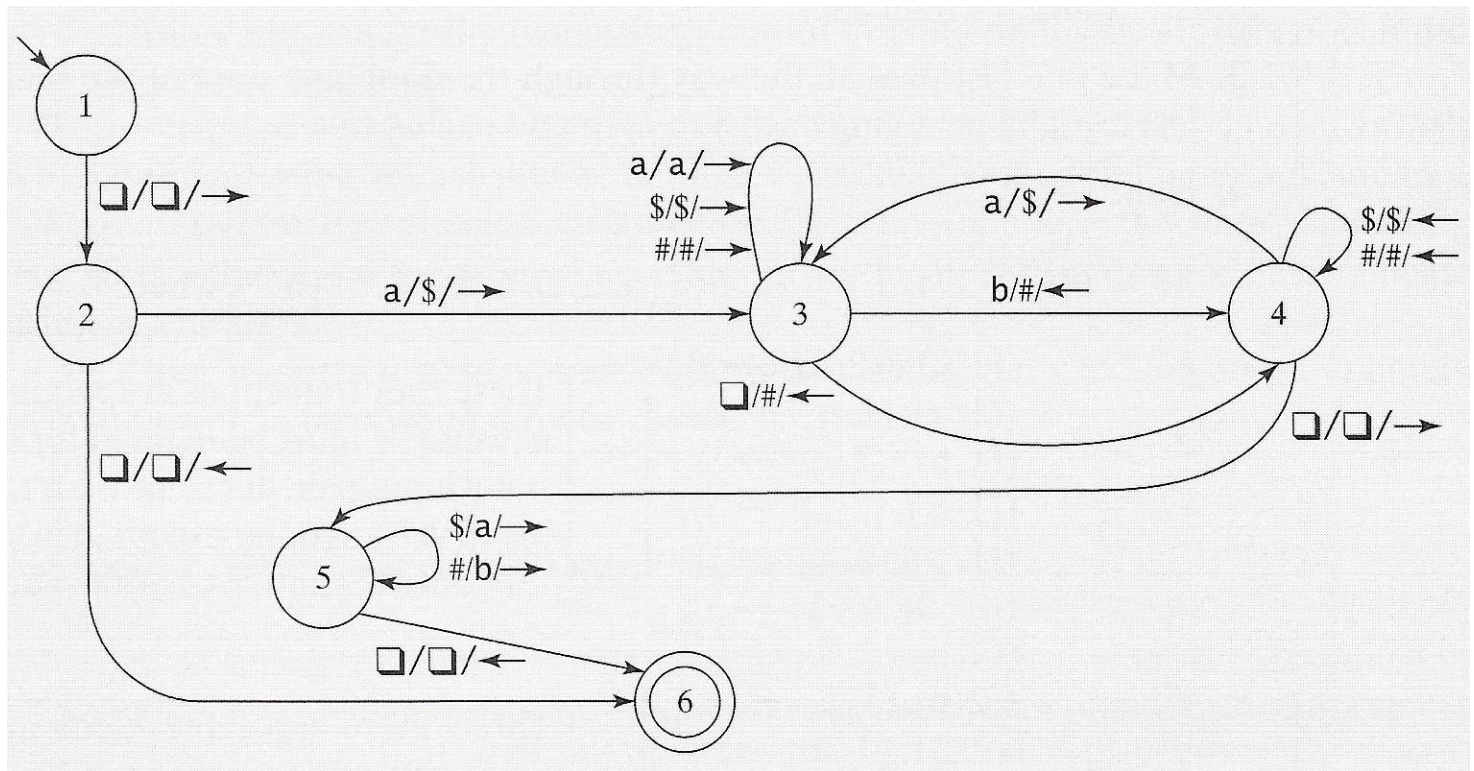
- Tanda panah menunjukkan posisi head
- Untuk contoh ini string input diasumsikan selalu benar  $\{a^i b^j : 0 \leq j \leq i\}$

# Contoh (Operasi pada $M$ )

- Pindahkan head ke kanan satu posisi, jika simbol di bawah head adalah  $\square$ , maka halt
- Dalam loop:
  - Tandai (sebenarnya ganti) setiap a dengan \$.
  - Scan ke kanan menemukan b atau  $\square$ 
    - Jika b, tandai (sebenarnya ganti) dengan #, kemudian siap balik ke kiri.
    - Jika  $\square$ , berarti b habis, tetapi masih ada a tersisa, maka tuliskan # dan siap balik ke kiri.
    - Bali ke kiri untuk menemukan a atau  $\square$ ,
      - jika a kembali ke awal loop,
      - jika  $\square$ , semua a sudah ditangani, maka halt.
- Lakukan pass terakhir untuk mengganti \$ ke a, dan # ke b.

# Contoh (Mesin $M$ )

- $M = (\{1,2,3,4,5,6\}, \{a,b\}, \{a,b, \square, \$, \#\}, \delta, 1, \{6\})$ , dengan  $\delta = \{$   
((1,  $\square$ ), (2,  $\square$ ,  $\rightarrow$ )), ((1, a), (2, q,  $\rightarrow$ )), ((1, b), (2, q,  $\rightarrow$ )), ((1,  $\square$ ), (2,  $\square$ ,  $\rightarrow$ )), ((1, #), (2, #,  $\rightarrow$ )),  
((2,  $\square$ ), (6,  $\square$ ,  $\rightarrow$ )), ((2, a), (3,  $\square$ ,  $\rightarrow$ )), ((2, b), (3,  $\square$ ,  $\rightarrow$ )), ((2,  $\square$ ), (3,  $\square$ ,  $\rightarrow$ )), ((2, #), (3,  $\square$ ,  $\rightarrow$ )),  
((3,  $\square$ ), (4, #,  $\leftarrow$ )), ((3, a), (3, a,  $\rightarrow$ )), ((3, b), (4, #,  $\leftarrow$ )), ((3,  $\square$ ), (3,  $\square$ ,  $\rightarrow$ )), ((3, #), (3, #,  $\rightarrow$ )),  
((4,  $\square$ ), (5,  $\square$ ,  $\rightarrow$ )), ((4, a), (3,  $\square$ ,  $\rightarrow$ )), ((4,  $\square$ ), (4,  $\square$ ,  $\leftarrow$ )), ((4, #), (4, #,  $\rightarrow$ )),  
((5,  $\square$ ), (6,  $\square$ ,  $\rightarrow$ )), ((4, a), (3,  $\square$ ,  $\rightarrow$ )), ((5,  $\square$ ), (5, a,  $\leftarrow$ )), ((5, #), (5, b,  $\rightarrow$ ))}
- Note: 6 merupakan halting state maka transisi untuk 6 tidak perlu didefinisikan.



# Halting, Crash atau Forever Loop

- Mesin **berhenti** jika:
  - Mencapai **status halt** (status khusus untuk menyatakan mesin berhenti).
  - Terpaksa berhenti (**crash**) karena transisi untuk konfigurasi saat ini (status dan isi tape pada head) tidak terdefinisi.
- Mesin bisa **tidak pernah berhenti** (karena *forever-loop*)
  - Tidak dapat mencapai halt maupun crash.
  - Karena **kesalahan logika** dalam pendefinisian transisi atau karena *nature of its related problem* sendiri (tidak ada mesin ekivalen yang bisa halt/crash).



# Konfigurasi

- Saat komputasi **konfigurasi** mesin dinyatakan sebagai  $(q, \alpha, x, \beta)$  atau  $(q, \alpha \underline{x} \beta)$  dengan
  - $q$ : current state
  - $\alpha$ : string isi tape di kiri posisi head
  - $x$ : simbol tape pada posisi head
  - $\beta$ : string isi tape di kanan posisi head
  - $\alpha \underline{x} \beta$  isi active tape
- **Konfigurasi awal:**  $(s, \sqcup \gamma)$ , start state  $s$ , dan head berada pada square kosong tepat disamping kiri string.
- **Halting configuration:** jika  $q$  merupakan status halting.

# Yields dan Komputasi

- **Yield in one step:** untuk mesin TM  $M$ , relasi antara konfigurasi  $C_1$  dan  $C_2$  dimana  $C_2$  dicapai dari  $C_1$  setelah satu kali transisi, ditulis  $C_1 \vdash_M C_2$ .
- **Yields:** untuk mesin TM  $M$ , reflexive, transitive closure dari  $\vdash_M$ , ditulis  $\vdash_M^*$
- **Path:** dari mesin TM  $M$ , adalah suatu sikuens  $C_0, C_1, C_2, \dots$  dengan  $C_0$  adalah **konfigurasi awal**, apabila terjadi  $C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \dots$
- **Komputasi:** dari mesin TM  $M$ , adalah suatu path  $C_0, C_1, \dots, C_n$ , untuk  $n \geq 0$ , dengan  $C_n$  adalah **halting configuration**.
  - Disebut komputasi halt dalam  $n$  langkah,  $C_1 \vdash_M^n C_n$

# Contoh

- Untuk contoh menambahkan b sebelumnya, jika input string aaab , TM ybs menghasilkan komputasi:
- $(1, \underline{a}aaab\Box\Box) \vdash (2, \Box\underline{a}aab\Box\Box) \vdash (3, \Box\$a\underline{a}ab\Box\Box)$   
 $\vdash (3, \Box\$a\underline{a}ab\Box\Box) \vdash (3, \Box\$a\underline{a}ab\Box\Box) \vdash (4, \Box\$a\underline{a}\#\Box\Box)$   
 $\vdash (3, \Box\$a\underline{\$}\#\Box\Box) \vdash (3, \Box\$a\underline{\$}\#\Box\Box) \vdash (4, \Box\$a\underline{\$}\#\#\Box)$   
 $\vdash (4, \Box\$a\underline{\$}\#\#\Box) \vdash (4, \Box\$a\underline{\$}\#\#\Box) \vdash (3, \Box\$\$\$\underline{\$}\#\#\Box)$   
 $\vdash (3, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (3, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (3, \Box\$\$\$\underline{\$}\#\#\Box)$   
 $\vdash (4, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (4, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (4, \Box\$\$\$\underline{\$}\#\#\Box)$   
 $\vdash (4, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (4, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (4, \Box\$\$\$\underline{\$}\#\#\Box)$   
 $\vdash (4, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (5, \Box\$\$\$\underline{\$}\#\#\Box) \vdash (5, \Box\underline{a}\$\$\$\#\#\Box)$   
 $\vdash (5, \Box\underline{a}a\underline{\$}\#\#\Box) \vdash (5, \Box\underline{a}a\underline{\$}\#\#\Box) \vdash (5, \Box\underline{a}a\underline{\$}\#\#\Box)$   
 $\vdash (5, \Box\underline{a}a\underline{\$}\#\#\Box) \vdash (6, \Box\underline{a}a\underline{b}b\underline{b}\Box)$