

IF5110 Teori Komputasi

3. Mesin Turing (Bagian 3)

Oleh: Rinaldi Munir

Program Studi Magister Informatika STEI-ITB

Credit Title

Semua bahan pada *power point* ini bersumberkan dari:

Hans Dulimarta, *Catatan Kuliah Matematika Informatika (Bagian Mesin Turing)*, Program Magister Informatika ITB, 2003.

Mesin Turing Universal

- *Emulator*: perangkat lunak yang menduplikasi (atau mengemulasi) fungsi suatu sistem komputer (*guest*) pada sistem komputer lain (*host*) yang berbeda sedemikian sehingga kelakuan yang diemulasi menyerupai kelakuan sistem sebenarnya (*guest*).

Contoh: *emulator* ponsel Nokia yang dijalankan pada komputer *desktop*.

- Dalam hal ini, emulator harus mengetahui karakteristik sistem komputer *guest*.

- Hal yang serupa dapat pula dilakukan oleh sebuah mesin Turing.
- Jika karakteristik mesin Turing T disajikan dalam suatu pengkodean tertentu, maka akan dapat dibuat mesin Turing lain (sebut saja U) yang dapat mensimulasikan perilaku T dengan membaca pengkodean tersebut.
- Mesin Turing U seperti ini dinamakan mesin Turing universal

- Pembentukan kode yang menggambarkan karakteristik suatu mesin Turing

$$T = (Q, \{0, 1\}, \{0, 1, B\}, \delta, q_1, B, \{q_2\})$$

dilakukan dengan cara:

(a) Simbol-simbol 0, 1, dan B dilambangkan berturut-turut sebagai simbol X_1 , X_2 , dan X_3 .

(b) Arah gerakan L dan R dilambangkan sebagai simbol D_1 dan D_2 .

(c) Setiap gerakan mesin Turing T , $\delta(q_i, X_j) = (q_k, X_l, D_m)$, dapat dituliskan sebagai 5-tuple (i, j, k, l, m) yang dikodekan sebagai string biner $C = 0^i 10^j 10^k 10^l 10^m$

(d) Jika mesin Turing T memiliki sebanyak r gerakan, maka seluruh pergerakan yang ada dapat dikodekan sebagai:

$$111C_1 11C_2 11 \dots 11C_r 111$$

(e) Rangkaian kode yang menyatakan perilaku mesin T ini dapat dijadikan input bagi mesin Turing U yang kemudian meniru perilaku T .

- **Contoh:** Misalkan terdapat mesin Turing yang memiliki gerakan seperti pada tabel berikut:

	0	1	B
q_1		$(q_2, 0, R)$	
q_2	$(q_3, 1, L)$	$(q_2, 1, R)$	$(q_3, 1, L)$
q_3	$(q_4, 0, R)$	$(q_3, 1, R)$	$(q_4, 0, R)$
q_4			

- Pengkodean setiap gerakan ditunjukkan pada tabel berikut:

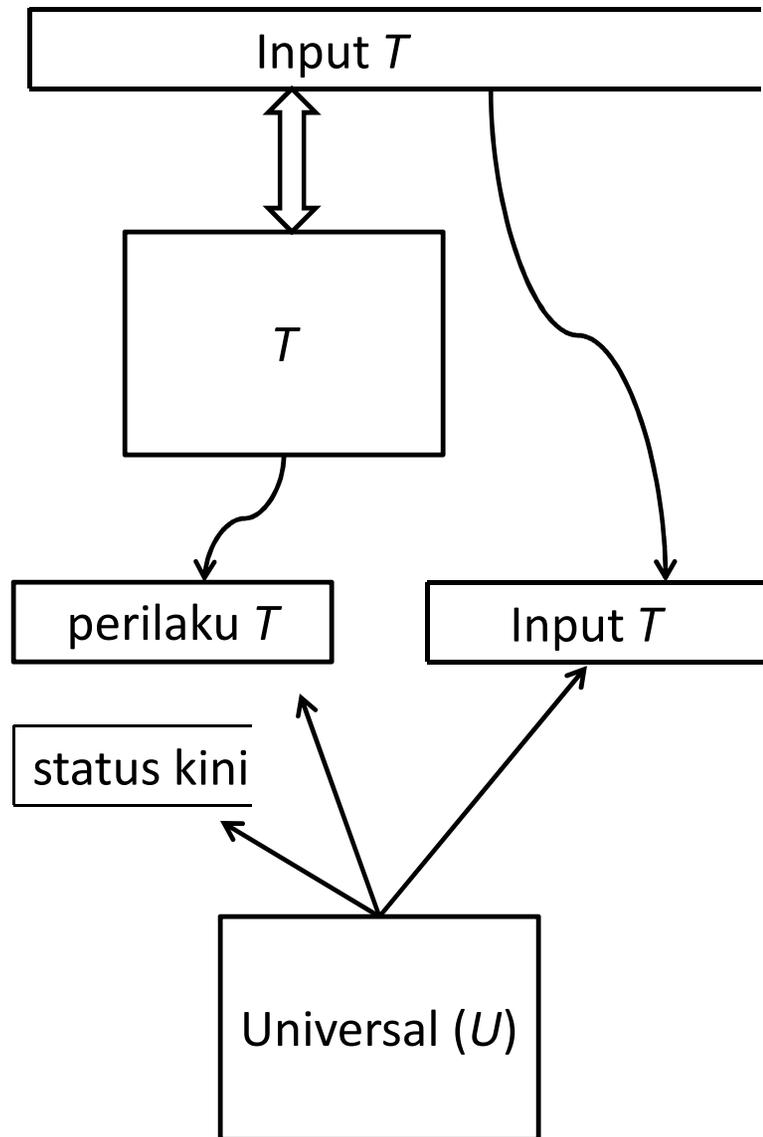
Gerakan	Kode
$\delta(q_1, 1) = (q_2, 0, R)$	0 1 00 1 00 1 0 1 00
$\delta(q_2, 0) = (q_3, 1, L)$	00 1 0 1 000 1 00 1 0
$\delta(q_2, 1) = (q_2, 1, R)$	00 1 00 1 00 1 00 1 00
$\delta(q_2, B) = (q_3, 1, L)$	00 1 000 1 000 1 00 1 0
$\delta(q_3, 0) = (q_4, 0, R)$	000 1 0 1 0000 1 0 1 00
$\delta(q_3, 1) = (q_3, 1, L)$	000 1 00 1 000 1 00 1 0
$\delta(q_3, B) = (q_4, 0, L)$	000 1 000 1 0000 1 0 1 00

- Berdasarkan kode gerakan yang disajikan pada tabel di atas, maka kode mesin Turing dapat dituliskan sebagai:

111010010010100**11** 0010100010010**11**00100100100100**11**
 001000100010010**11** 000101000010100**11**00010010001001
 0**11**00010001000010100**111**

Simulasi oleh Mesin Turing Universal

- Andaikan mesin Turing universal U akan mensimulasikan pengenalan string masukan w oleh mesin Turing T seperti ditunjukkan pada Gambar 1.
- Untuk membantu kerjanya, mesin Turing U dilengkapi oleh tiga pita.
- Pita pertama berisi deskripsi mesin Turing T yang akan disimulasikan, pita kedua berisi rangkaian simbol yang akan dikenali oleh T , dan pita ketiga berisi status kini dari mesin T .



Gambar 1. Simulasi T oleh mesin Turing Universal U

- Mesin Turing universal U bekerja dengan cara berikut:
 1. Pita 2 akan diinisialisasi dengan input T , dan pita 3 diisi dengan simbol 0 untuk menyatakan status awal T , yaitu q_1 .
 2. Jika pita 3 berisi simbol 00 maka pensimulasikan T oleh U dihentikan karena berarti mesin Turing sudah mencapai status akhirnya, q_2 . (lihat kembali deskripsi mesin Turing T yang diberikan pada halaman 4).
 3. Misalkan X_j adalah simbol yang sedang dibaca pada pita 2 dan pita 3 berisi simbol 0^i yang menyatakan status kini q_i dari T . Mesin Turing U harus memeriksa pita 1 untuk menemukan string yang dimulai dengan 110^i10^j1 (yang menandakan transisi $\delta(q_i, X_j)$). Ada dua kemungkinan kasus yang terjadi:

Kasus 1: Jika tidak ditemukan *string* tersebut, maka simulasi dihentikan dan berarti input w tidak diterima oleh T .

Kasus 2: Jika ditemukan, string $110^i10^j10^k10^l10^m$, maka

(a) simpan 0^k pada pita 3

(b) tuliskan simbol X_l pada sel yang sedang dibaca pada pita 2

(c) gerakkan *head* 2 ke arah D_m .

Halting Problem

- Masalah untuk mengetahui apakah sebuah mesin Turing akan berhenti jika mengolah suatu input dinamakan **masalah perhentian** (*halting problem*).
- Akan ditunjukkan bahwa tidak ada algoritma yang dapat memeriksa keadaan tersebut (tidak ada algoritma untuk menjawab masalah tersebut).
- Mesin Turing yang menerima input yang tidak dikenalnya akan mungkin mengalami *infinite loop*.

- **Contoh:** Tinjau mesin Turing untuk mengenal '111'. Gerakan mesin Turing diperlihatkan pada tabel berikut:

	1	B
q_1	$(q_2, 1, R)$	
q_2	$(q_3, 1, R)$	(q_1, B, L)
q_3	$(q_4, 1, R)$	(q_2, B, L)
q_4	$(q_5, 1, R)$	(q_3, B, L)
q_5	$(q_4, 1, L)$	(q_4, B, L)

- Rangkaian deskripsi sesaat mesin Turing dalam mengenali input '11' adalah:

$q_1 1 1 B \quad | \quad 1 q_2 1 B \quad | \quad 1 1 q_3 B \quad | \quad 1 q_2 1 B \quad | \quad \dots \quad | \quad 1 1 q_3 B$



Infinite loop

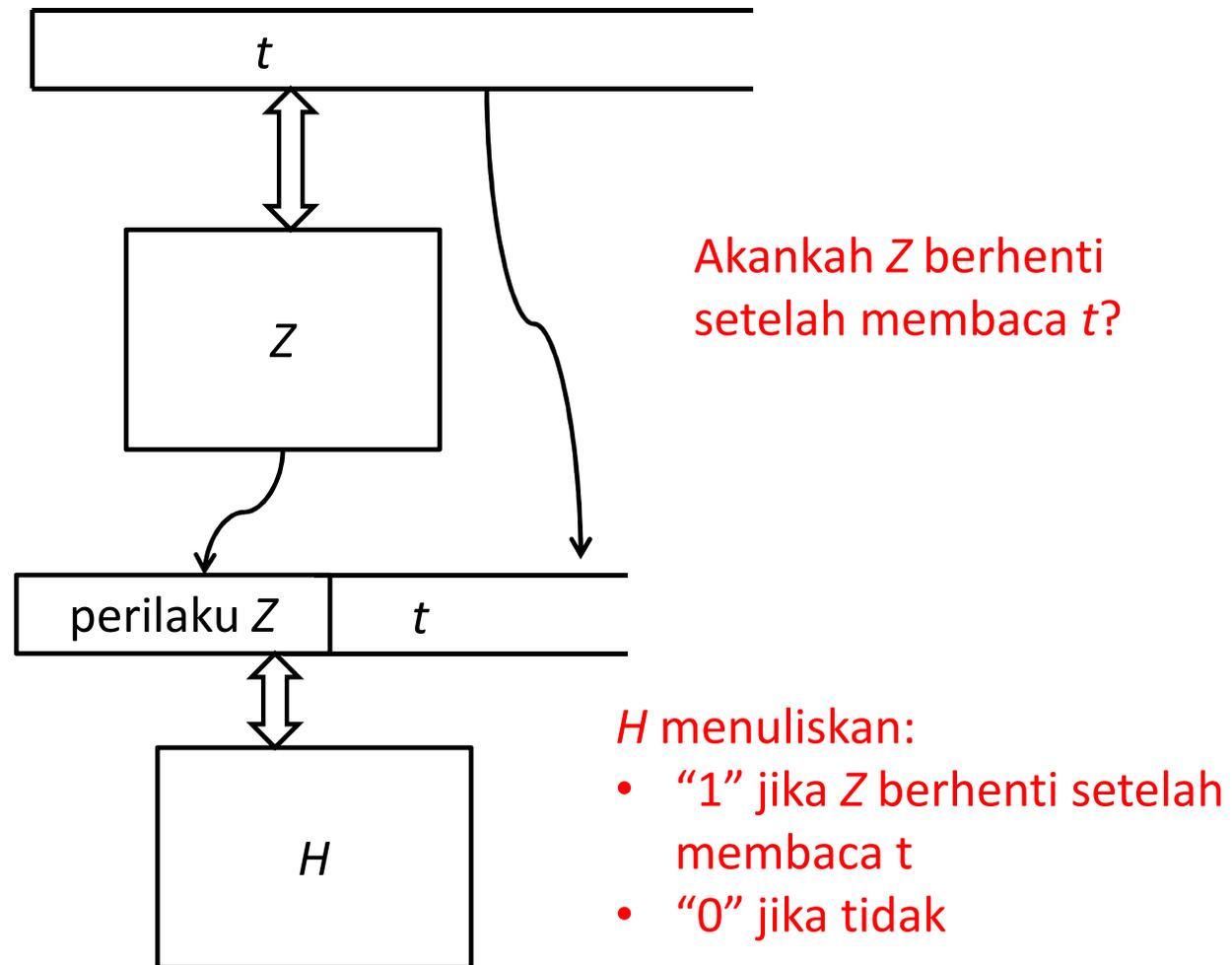
- “Bukti” bahwa algoritma untuk menjawab masalah perhentian memang tidak mungkin ada adalah sbb:

Misalkan ada sebuah mesin Turing Z yang mengolah input t dan ingin diketahui apakah Z suatu saat akan berhenti jika selesai membaca input t .

Seandainya algoritma untuk masalah perhentian ini ada dan dapat diterapkan pada mesin Turing universal H yang mempunyai perilaku demikian, maka H menuliskan:

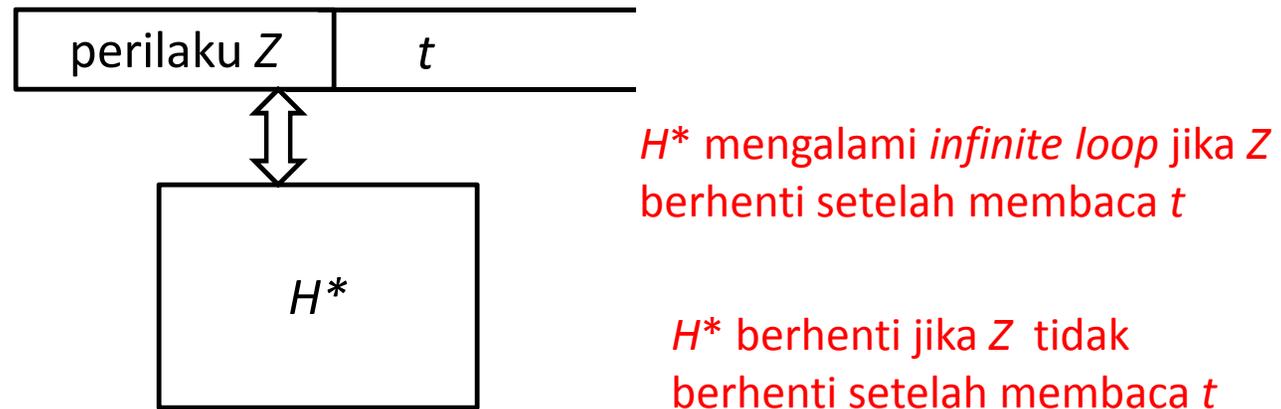
- “1” jika Z berhenti setelah membaca t
- “0” jika tidak

- Untuk memudahkan pembuktian, perilaku mesin Z dan masukan t ditulis pada pita yang sama (tidak dipisah seperti pensimulasian oleh mesin Turing universal yang dijelaskan sebelumnya), seperti gambar berikut:



Gambar 2. Pemeriksaan perhentian $\langle Z, t \rangle$ oleh H

- Misalkan kemudian mesin H diganti oleh mesin H^* seperti gambar berikut:

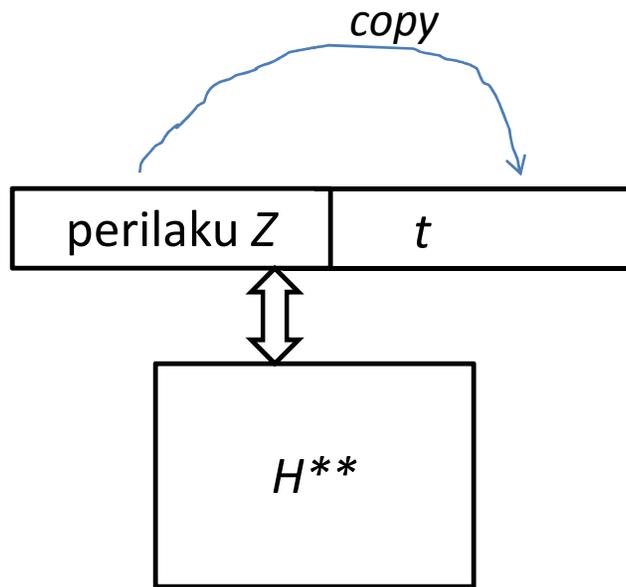


Gambar 3. Pemeriksaan perhentian $\langle Z, t \rangle$ oleh H^*

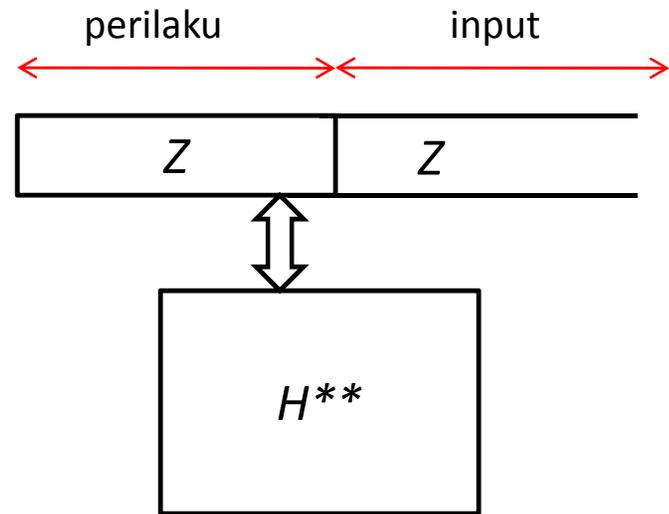
- Mesin H^* memiliki perilaku:
 - H^* mengalami *infinite loop* jika Z berhenti setelah membaca t
 - H^* berhenti jika Z tidak berhenti (mengalami *infinite loop*) setelah membaca t

- Andaikan perilaku mesin H^* dikembangkan lagi menjadi mesin Turing H^{**} . Mesin Turing H^{**} membaca hanya perilaku Z , kemudian menyalin perilaku tersebut ke bagian kanan, lalu menirukan perilaku H^* .
- Dalam keadaan demikian dapat dianggap bahwa efek yang terjadi adalah proses pemeriksaan perhentian $\langle Z, Z \rangle$ oleh H^* .
- Dengan demikian perilaku H^{**} dapat digambarkan sebagai berikut: Jika diberikan deskripsi perilaku mesin Z , maka:
 - H^{**} mengalami *infinite loop* jika Z akhirnya berhenti setelah membaca deskripsi perilakunya sendiri, dan
 - H^{**} akhirnya berhenti jika Z tidak berhenti (mengalami *infinite loop*) setelah membaca deskripsi perilakunya sendiri.

H^{**} menyalin perilaku Z

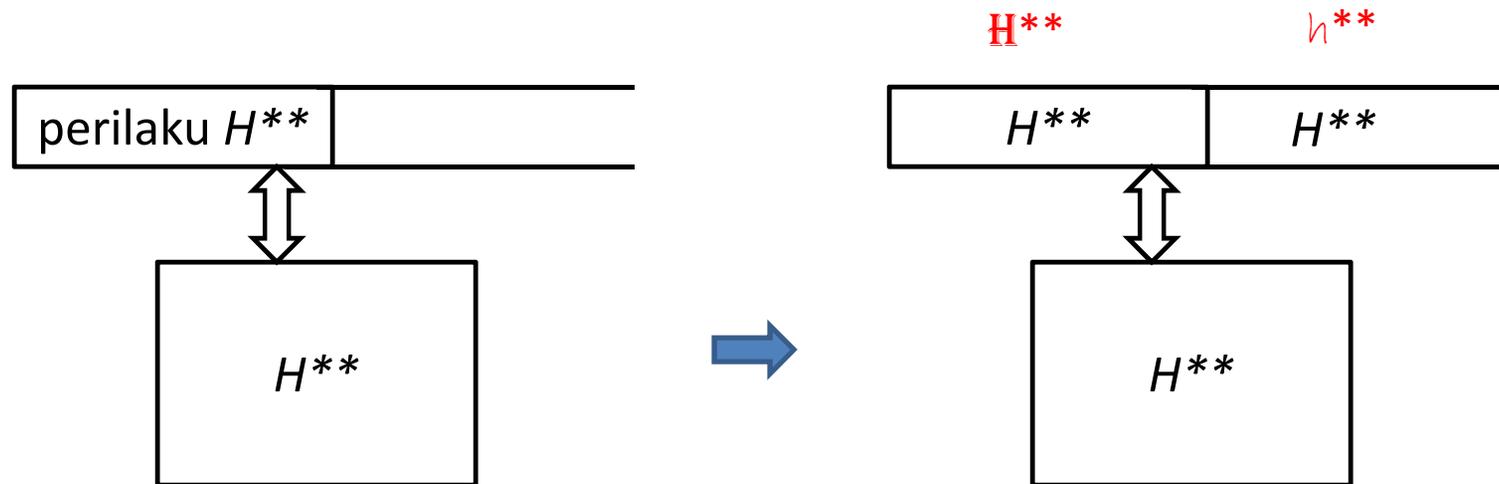


H^{**} menirukan perilaku H^*



Gambar 4. Pemeriksaan perhentian $\langle Z, Z \rangle$ oleh H^{}**

- Karena Z pada Gambar 4 dapat menggambarkan mesin Turing manapun, maka ia dapat diganti dengan H^{**} sendiri sehingga hal ini dapat digambarkan pada Gambar 5.



Gambar 5. Pemeriksaan perhentian $\langle H^{}, H^{**} \rangle$ oleh H^{**}**

Perilaku H^{**} :

- Mengalami *infinite loop* jika H^{**} berhenti setelah membaca h^{**} .
- Berhenti jika H^{**} mengalami *infinite loop* setelah membaca h^{**} .

Pada keadaan yang digambarkan oleh Gambar 5, dapat ditunjukkan sifat-sifat berikut:

- H^{**} mengalami *infinite loop* jika H^{**} akhirnya berhenti setelah membaca deskripsi perilakunya sendiri, dan
- H^{**} akhirnya berhenti jika H^{**} tidak berhenti (mengalami *infinite loop*) setelah membaca deskripsi perilakunya sendiri.

- Dari kedua sifat di atas terlihat adanya hal yang bertentangan:

*H*** mengalami *infinite loop* jika dan hanya jika *H*** berhenti.

- Hal ini tidak mungkin terjadi, sehingga akibatnya dapat disimpulkan:

H tidak ada \implies *H** tidak ada \implies *H* tidak ada**

- Dengan kata lain, algoritma untuk menjawab masalah perhentian tidak mungkin ada!

Variasi-Variasi Mesin Turing

- Terdapat beberapa variasi mesin Turing. Meskipun terdapat lebih dari satu variasi, namun tidak ada peningkatan kemampuan pengenalan bahasa dari masing-masing varian.
- Dengan kata lain, variasi-variasi mesin Turing tersebut merupakan mesin yang ekuivalen.
- Beberapa variasi mesin Turing:

1. *Two-way Infinite tape*

Mesin Turing ini memiliki pita simbol yang tidak terbatas pada kedua ujungnya. Pada mesin Turing biasa, *head* tidak dapat bergerak lebih kiri dari sisi pita.

2. *Multitrack*

Mesin Turing ini memiliki sebuah pita yang memiliki lebih dari satu jalur (*track*) penulisan/pembacaan simbol. Simbol-simbol yang berada pada “kolom” yang sama akan dibaca sekaligus oleh sebuah *head* tunggal.

3. *Multitape*

Mesin Turing ini memiliki beberapa pita yang masing-masing dapat dibaca oleh *head* yang saling bebas. Setiap pita memiliki *head* tersendiri. Aksi yang dilakukan salah satu *head* pada pitaanya tidak bergantung dari aksi *head* yang lain.

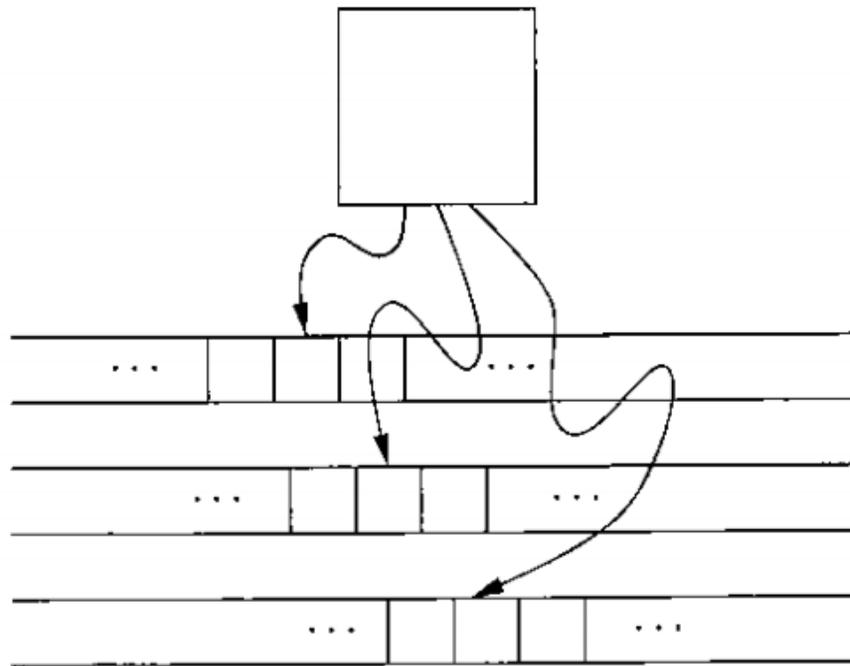


Figure 8.16: A multitape Turing machine

4. *Non-deterministic*

Mesin Turing ini memiliki satu pita yang terbatas pada salah satu ujungnya. Untuk setiap kombinasi status dan simbol pita yang sedang dibaca, mesin ini dapat memiliki sejumlah gerakan berikutnya.

5. *Multi-dimensional tape*

Mesin Turing ini memiliki pita yang multi dimensi. Mesin Turing biasa memiliki pita yang berdimensi satu. Untuk pita dua dimensi berarti *head* dapat berpindah dari satu sel ke sel lain yang terletak pada suatu bidang datar.

6. *Multihead*

Mesin Turing ini mirip dengan mesin Turing *multitape*, hanya bedanya mesin Turing *multihead* hanya satu pita. Setiap *head* pada pita tersebut dapat beraksi saling bebas satu sama lainnya.

Two-Way Infinite Tape

- Varian mesin Turing dengan ujung kiri tak terbatas sehingga head dapat bergerak ke kiri tanpa batas.
- Meskipun demikian, mesin Turing two-way infinite tape ekuivalen dengan mesin Turing biasa.
- Misalkan mesin Turing biasa adalah T_1 dan mesin Turing tanpa batas adalah T_2 :
$$T_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_1, B_1, F_1)$$
$$T_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_2, B_2, F_2)$$
- Akan ditunjukkan bahwa perilaku T_2 dapat disimulasikan oleh mesin Turing biasa T_1 begitu juga sebaliknya.

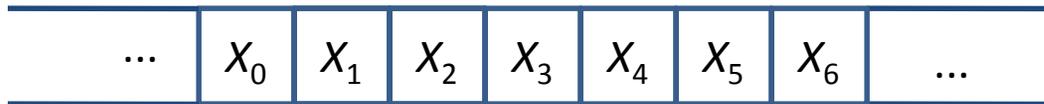
Simulasi T_2 oleh T_1

- Hal ini mudah dilakukan
- Mesin T_1 tidak pernah menggerakkan head-nya melewati sel terkiri pada pita.
- Sehingga, jika T_1 disimulasikan oleh T_2 maka sel-sel di sebelah kiri simbol masukan tidak akan pernah dimanfaatkan.

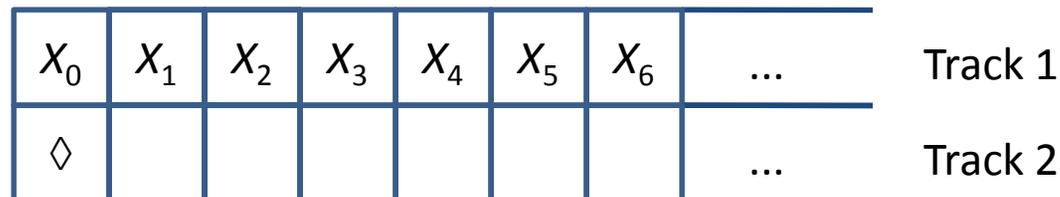
Simulasi T_1 oleh T_2

- Hal ini tidak mudah dilakukan karena ada kemungkinan T_2 pada bagian kiri sementara pita T_1 memiliki batas pada bagian kirinya.
- Sehingga, jika T_1 disimulasikan oleh T_2 maka sel-sel di sebelah kiri simbol masukan tidak akan pernah dimanfaatkan.

- Untuk mengatasi hal ini, maka pita simbol yang digunakan di T_1 memiliki dua jalur (*track*) namun ujung kirinya tetap terbatas.
- Penempatan simbol-simbol pada T_1 dan T_2 adalah seperti pada gambar di bawah ini:



(a) Penempatan simbol pada pita T_2



(b) Penempatan simbol pada pita T_1

- Simbol \diamond pada jalur bawah pita T_1 digunakan untuk menandai sel terkiri. Simbol ini harus disisipkan oleh T_1 pada awal gerakannya.
- Pita jalur dua pada T_1 dapat dipandang sebagai pita T_2 yang dilipat pada posisi sel X_0 .
- Jika head pada pita T_2 bergerak di kanan X_0 maka *head* pada pita T_1 bergerak pada jalur atas dengan arah yang sama.
- Jika head pada pita T_2 bergerak di kirik X_0 maka *head* pada pita T_1 bergera pada jalur bawah dengan arah berlawanan.
- Jika simbol \diamond terbaca oleh T_1 maka berarti posisi lipatan pita terdeteksi dan T_1 harus beralih dari jalur atas ke jalur bawahnya, atau sebaliknya.

- .Dengan adanya pita dua jalur pada T_1 , maka *head* T_1 selalu membaca dua simbol sekaligus setiap saat. Pasangan simbol ini dapat ditulis sebagai $[X, Y]$, yang dalam hal ini X adalah simbol pada jalur atas dan Y simbol pada jalur bawah.
- Walaupun *head* membaca dua simbol sekaligus, namun T_1 tetap hanya mengubah satu simbol saja setiap saat.
- Untuk mengetahui *head* T_1 sedang mengolah simbol pada jalur atas atau jalur bawah, maka status-status T_1 dituliskan sebagai
 $[q, A]$ atau $[q, B]$

A menyatakan “atas” dan “ B ” menyatakan “bawah”

- Model matematis mesin Turing biasa:

$$T_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_1, B_1, F_1)$$

- Bagaimana pembentukan setiap komponen T_1 di atas? Caranya adalah sebagai berikut:

1. Himpunan status Q_1 berisi status-status dalam bentuk $[q, A]$ atau $[q, B]$. Suku pertama q berasal dari status di Q_2 . Selain status-status di atas, q_1 juga merupakan anggota Q_1 .
2. Himpunan simbol masukan Σ_1 terdiri dari simbol dalam bentuk $[a, B]$ dan suku pertama a adalah simbol masukan pada Σ_2 . Simbol B pada komponen kedua di sini melambangkan simbol *blank*, bukan “bawah” seperti status di atas.
3. Himpunan simbol pita Γ_1 akan berisi simbol-simbol dalam bentuk $[X, Y]$ yaitu menyatakan sepasang simbol yang diambil pada jalur atas dan jalur bawah pita. Simbol khusus \diamond yang digunakan untuk menandai tepi kiri pita hanya terdapat pada jalur bawah, sehingga komponen kedua (Y) dapat berupa simbol \diamond tersebut.

4. Status akhir F_1 berisi simbol-simbol dalam bentuk $[q, A]$ atau $[q, B]$ dan suku pertama q berasal dari F_2 .
5. Transisi δ_1 dibentuk dengan memperhatikan hal-hal berikut:
- (a) Jika T_2 melakukan gerakan awal (ke kiri atau ke kanan), T_1 harus menuliskan simbol \diamond pada jalur bawah pita. Hal ini dilakukan melalui dua gerakan berikut:

No.	Mesin T_2	Mesin T_1
(1)	$\delta_2(q_2, a) = (q, X, R)$	$\delta_1(q_1, [a, B]) = ([q, A], [X, \diamond], R)$
(2)	$\delta_2(q_2, a) = (q, X, L)$	$\delta_1(q_1, [a, B]) = ([q, B], [X, \diamond], R)$

Keterangan:

- Gerakan di atas dibentuk untuk setiap $a \in \Sigma \cup \{B\}$
- Pada gerakan (1), mesin T_2 mulai dengan bergerak ke kanan sehingga mesin T_1 akan mengolah simbol-simbol pada jalur atas.
- Pada gerakan (2), mesin T_2 mulai dengan bergerak ke kiri sehingga mesin T_1 akan mengolah simbol-simbol pada jalur bawah.

(b) Jika T_2 melakukan gerakan yang melintasi posisi X_0 (hal ini diketahui dari gerakan T_1 yang melintasi simbol \diamond) maka T_1 harus beralih lagi dari jalur atas ke jalur bawah, atau sebaliknya. Hal ini dilakukan melalui gerakan:

No.	Mesin T_2	Mesin T_1
(3)	$\delta_2(p, X) = (q, Y, R)$	$\delta_1([p, A], [X, \diamond]) = ([q, A], [Y, \diamond], R)$
(4)	$\delta_2(p, X) = (q, Y, R)$	$\delta_1([p, B], [X, \diamond]) = ([q, A], [Y, \diamond], R)$
(5)	$\delta_2(p, X) = (q, Y, L)$	$\delta_1([p, A], [X, \diamond]) = ([q, B], [Y, \diamond], R)$
(6)	$\delta_2(p, X) = (q, Y, L)$	$\delta_1([p, B], [X, \diamond]) = ([q, B], [Y, \diamond], R)$

Keterangan:

- Jika *head* berada pada posisi terkira pita, maka T_1 sedang membaca simbol dalam bentuk $[X, \diamond]$.
- Munculnya simbol \diamond pada komponen kedua (jalur bawah) menunjukkan bahwa kedudukan *head* pada jalur atas atau bawah tidak memegang peranan penting karena sesungguhnya simbol yang sedang diolah adalah simbol X .
- Hal ini menjelaskan terjadinya duplikasi gerakan T_1 pada pasangan (3) (4) dan (5)(6).
- Peralihan jalur (atas ke bawah atau bawah ke atas) terlihat jelas pada gerakan (4) dan (5).

(c) Selain kedua gerakan khusus di atas, gerakan lain merupakan pensimulasian gerakan T_2 oleh T_1 yang bergerak pada jalur yang sama. Gerakan tersebut adalah:

No.	Mesin T_2	Mesin T_1
(7)	$\delta_2(p, X) = (q, Z, R)$	$\delta_1([p, A], [X, Y]) = ([q, A], [Z, Y], R)$
(8)	$\delta_2(p, X) = (q, Z, L)$	$\delta_1([p, A], [X, Y]) = ([q, A], [Z, Y], L)$
(9)	$\delta_2(p, X) = (q, Z, R)$	$\delta_1([p, B], [X, Y]) = ([q, A], [X, Z], L)$
(10)	$\delta_2(p, X) = (q, Z, L)$	$\delta_1([p, B], [X, Y]) = ([q, A], [X, Z], R)$

Keterangan:

- Pada gerakan (7) dan (8) simbol pita yang berubah terletak pada jalur atas dan arah gerakan T_1 sama dengan arah gerakan T_2 .
- Pada gerakan (9) dan (10) simbol pita yang berubah terletak pada jalur bawah dan arah gerakan T_1 terbalik dari arah T_2 .

Mesin Turing dengan Pita Berjalur Banyak

- Misalkan M adalah mesin Turing yang memiliki sebuah pita dengan k jalur.
- Karena semua simbol pada kolom sel yang sama dibaca sekaligus oleh mesin Turing, maka fungsi δ dari mesin Turing ini berbentuk:

$$\delta(p, [a_1, a_2, \dots, a_k]) = (q, [b_1, b_2, \dots, b_k], D)$$

- Dengan mengganti kombinasi simbol $[a_1, a_2, \dots, a_k]$ dan $[b_1, b_2, \dots, b_k]$ dengan simbol tunggal baru a dan b sehingga

$$\delta(p, a) = (q, b, D)$$

maka transisi di atas akan tampak seperti transisi pada mesin Turing biasa.

- Sebaliknya mesin Turing biasa dengan pita jalur tunggal dapat dengan mudah disimulasikan oleh mesin Turing dengan pita berjalur banyak. Dalam hal ini hanya salah satu jalur saja yang digunakan oleh mesin Turing dengan pita berjalur banyak.

Mesin Turing Non-Deterministik

- Semua mesin Turing yang dibahas sebelum ini adalah mesin Turing deterministik.
- Pada mesin Turing deterministik, fungsi transisi δ memiliki nilai tunggal, sedangkan pada mesin Turing non-deterministik untuk pasangan status dan simbol tertentu mungkin dijumpai lebih dari satu aksi.
- Mesin Turing non-deterministik M_n dapat disimulasikan oleh sebuah mesin Turing deterministik M_d . Dasar pemikirannya: untuk setiap status dan simbol pita, M_n mungkin memiliki lebih dari satu aksi yang banyaknya berhingga. Misalkan pilihan aksi ini diberi nomor 1, 2, 3, ..., k . Dengan penomoran ini, rangkaian aksi mesin Turing dapat dinyatakan sebagai rangkaian bilangan dari 1 sampai k .

- Untuk mensimulasikan M_n , mesin Turing M_d menggunakan tiga pita. Pita pertama menyimpan input yang diolah M_n , pita kedua digunakan untuk menyimpan rangkaian bilangan 1 sampai dengan k yang dibangkitkan secara teratur.
- Rangkaian ini akan dihasilkan dengan urutan mulai dari rangkaian terpendek yang terdiri dari satu simbol, dua simbol, dst.
- Untuk setiap urutan bilangan yang terdapat pada pita kedua, mesin M_d akan mensimulasikan perilaku M_n secara deterministik, yaitu menggunakan nomor aksi yang sudah dibangkitkan pada pita kedua.

- Dengan demikian jika M_n menerima rangkaian simbol masukan, haruslah ada sebuah rangkaian aksi yang menerimanya.
- Karena rangkaian aksi pada pita kedua dibangkitkan secara teratur, maka M_d dapat juga suatu saat menerima rangkaian simbol masukan tersebut.
- Sebaliknya, jika M_n tidak menerima, maka tidak ada rangkaian aksi yang dapat digunakan oleh M_d dalam menerima rangkaian simbol masukan tersebut.

Pekerjaan Rumah 2

- Buatlah laporan yang berisi segala hal tentang tiga varian mesin Turing yang lain, yaitu mesin Turing berpita banyak, mesin Turing multi dimensi, dan mesin Turing *multihead*. Laporan berisi:
 - deskripsi/spesifikasi mesin Turing tersebut
 - ekivalennya dengan mesin Turing biasa (mesin Turing biasa mensimulasikan mesin Turing varian, mesin Turing varian mensimulasikan mesin Turing biasa).