

IF5110 Teori Komputasi

2. Mesin Turing (Bagian 2)

Oleh: Rinaldi Munir

Program Studi Magister Informatika STEI-ITB

Peranan Mesin Turing

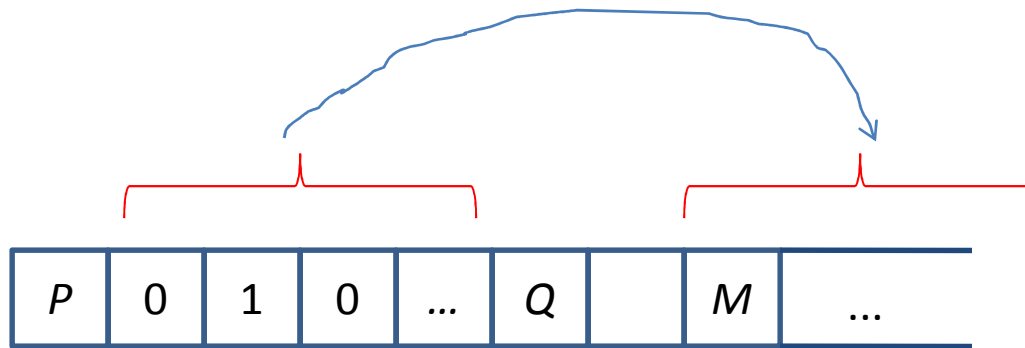
- Bahasa yang diterima oleh mesin Turing dinamakan *recursively enumerable* (r.e).
- Istilah *enumerable* digunakan untuk menunjukkan sifat bahwa himpunan jenis ini anggota-anggotanya dapat didaftarkan (di-enumerasi) oleh sebuah mesin Turing.
- Pengertian *recursive* sama dengan *recursion* pada *programming*.
- Kelas dari bahasa r.e sangat luas dan mencakup bahasa yang tidak dapat ditentukan keanggotaannya.

- Jika L adalah bahasa *recursive enumerable*, maka bila *string* di dalam L dikenali oleh M , maka M akan berhenti.
- Sebaliknya jika *string* bukan anggota L , maka mesin Turing gagal mengenali (mesin Turing tidak akan berhenti).
- Salah satu contoh kelas r.e adalah himpunan rekursif, yang dalam hal ini semua *string* yang menjadi anggotanya akan dikenali oleh sekurang-kurangnya satu mesin Turing.
- Kelas himpunan rekursif merupakan *proper* subkelas dari kelas r.e.

- Dengan adanya pita yang dapat dibaca maupun ditulis maka mesin Turing dapat menirukan perilaku sebuah *Finite State Automata* (FSA) maupun *Push Down Automata* (PDA).
- Dengan kemampuannya ini, maka mesin Turing mampu mengenali bahasa formal yang termasuk ke dalam kelas bahasa regular (bahasa yang dikenali oleh FSA), bahasa bebas konteks (bahasa yang dikenali oleh PDA). Maupun bahasa formal yang termasuk ke dalam kelas yang lebih tinggi.
- Selain sebagai pengenalan bahasa yang termasuk ke dalam kelas r.e, mesin Turing juga dapat digunakan untuk:
 1. Menyalin (*copy*) simbol
 2. Penghitung fungsi-fungsi *integer*
 3. Mengeksekusi prosedur (*subroutine*)

Mesin Turing Penyalin Simbol

- Pada contoh mesin Turing pengenalan 0^n1^n terlihat bahwa operasi CARI X (pencarian simbol tertentu di dalam pita) merupakan operasi yang mendasar.
- Operasi penanganan simbol yang lainnya adalah penyalinan simbol.
- Kita akan merancang mesin Turing yang dapat menyalin *string* biner yang terletak di antara simbol P dan Q ke suatu posisi sel baru yang ditandai dengan simbol M .



Mesin bekerja dari status awal q_1 dengan *head* pada posisi simbol P .

Algoritma penyalinan simbol adalah sebagai berikut:

1. Simbol yang akan disalin ditandai dengan simbol khusus
2. Gerakkan *head* ke kanan hingga dijumpai posisi penyalinan yang tepat.
3. Tuliskan simbol yang disalin pada posisi tersebut.
4. Gerakkan *head* ke kiri sampai dijumpai simbol khusus di atas.
5. Ganti simbol khusus dengan simbol semula.

Fungsi pergerakan Mesin Turing:

$\delta(q_1, P) = (q_2, P, R)$	Longkap simbol P
$\delta(q_2, 0) = (q_3, \alpha, R)$	Tandai 0
$\delta(q_3, *) = (q_3, *, R)$	Longkap ke kanan simbol-simbol selain M
$\delta(q_3, M) = (q_4, 0, R)$	Ganti 'M' dengan simbol 0 yang sedang disalin
$\delta(q_4, *) = (q_5, M, L)$	Tulis 'M' pada posisi baru
$\delta(q_5, *) = (q_5, *, L)$	Longkap ke kiri simbol selain α
$\delta(q_5, \alpha) = (q_2, 0, R)$	kembalikan α menjadi 0
$\delta(q_2, 1) = (q_6, \beta, R)$	Tandai 1
$\delta(q_6, *) = (q_6, *, R)$	Longkap ke kanan simbol selain M
$\delta(q_6, M) = (q_7, 1, R)$	Ganti M dengan simbol 1 yang sedang disalin
$\delta(q_7, *) = (q_8, M, L)$	Tulis M pada posisi baru
$\delta(q_8, *) = (q_8, *, L)$	Longkap ke kiri simbol selain β
$\delta(q_8, \beta) = (q_2, 1, R)$	Kembalikan β menjadi 1
$\delta(q_2, Q) = (q_9, Q, R)$	Q menandakan akhir data

Mesin Turing Penghitung Fungsi Bilangan Bulat

- Mesin Turing dapat dipandang sebagai [enghitung fungsi bilangan bulat ke bilangan bulat.
- Untuk dapat melakukan penghitungan ini dibuat konvensi sebagai berikut:
 1. *Integer* bernilai k direpresentasikan sebagai 0^k (*string* yang terdiri dari “0” sebanyak k kali) pada pita masukan.
 2. Jika fungsi memiliki beberapa argumen sebagai input (i_1, i_2, \dots, i_m) , maka bilangan bulat ini dituliskan pada mesin Turing yang dibatasi oleh “1”, dituliskan $0^{i_1}10^{i_2}1\dots10^{i_m}$
 3. Hasil penghitungan fungsi (berupa *integer* p) akan dituliskan pada pita sebagai 0^p .

- Sebagai contoh, argumen dari fungsi $f(3, 7, 5)$ dapat dinyatakan sebagai:

000100000001000000

3 7 5

- Jika sebelum mesin Turin bekerja pada pita masukan terdapat rangkaian simbol yang merepresentasikan m argumen,

$$(i_1, i_2, \dots, i_m)$$

dan pada saat mesin Turing berhenti bekerja pada pita terdapat simbol 0^p , maka didefinisikan $f(i_1, i_2, \dots, i_m) = p$.

- Jika dalam mengolah input tersebut mesin Turing tidak berhenti bekerja, maka simbol yang tertera pada pita tidak dapat dipandang sebagai hasil perhitungan fungsi.
- Dalam hal ini dikatakan bahwa nilai fungsi untuk masukan tersebut tidak terdefinisi.
- Hal ini juga yang membedakan apakah fungsi yang dihitung oleh mesin Turing merupakan fungsi total atau fungsi parsial.
- Jika fungsi yang dihitung terdefinisi hasilnya untuk setiap nilai yang diberikan, maka fungsi tersebut dikatakan bersifat total, Jika tidak maka fungsi tersebut dikatakan bersifat parsial.

Contoh 1: Mesin Turing untuk menghitung $f(n) = n + 1$

- Pada keadaan awal di dalam pita terdapat rangkaian simbol dalam bentuk 0^n .
- Untuk menambahkan fungsi dengan 1, maka gerakan yang harus dilakukan adalah menambahkan satu simbol “0” pada sel paling kanan sehingga pada pita terdapat rangkaian simbol dalam bentuk 0^{n+1} .
- Jadi, algoritmanya adalah sebagai berikut:
 1. Geser *head* ke kanan sampai dijumpai “B”
 2. Ganti simbol “B” dengan “0”

- Fungsi pergerakan mesin Turing:

$\delta(q_0, 0) = (q_0, 0, R)$ Geser kanan

$\delta(q_0, B) = (q_1, 0, R)$ Ganti "B" dengan "0"

sehingga notasi formal mesin Turing penambah satu adalah:

$$M = (\{q_0, q_1\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \emptyset)$$

dengan definisi δ seperti di atas.

Contoh 2: Mesin Turing untuk menghitung $m - n$. Pengurangan $m - n$ didefinisikan sebagai $m - n$ jika $m \geq n$, dan sama dengan 0 jika $m < n$.

- Pada keadaan awal di dalam pita terdapat rangkaian simbol dalam bentuk $0^m 1 0^n$.
- Mesin berhenti bila di pita tercetak simbol 0^{m-n}
- Algoritmanya:
 1. M mengganti simbol 0 yang pertama dengan B.
 2. Kemudian dilakukan pencarian 1 ke kanan yang diikuti dengan 0, mengganti 0 dengan 1.
 3. Selanjutnya M kembali bergerak ke kiri sampai ketemu B
 4. Kembali ke 1

- Pengulangan berhenti jika:

Kasus 1. Ketika pencarian ke kanan simbol 0 mesin M tidak menemukan B .

Kasus 2. Mulai pencarian M tidak menemukan 0 untuk diganti dengan B .

- Fungsi transisinya adalah sebagai berikut:

1. $\delta(q_0, 0) = (q_1, B, R)$

Mulai pencarian, ganti simbol 0 pertama dengan B

2. $\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, 1) = (q_2, 1, R)$

Pencarian ke kanan mencari 1 yang pertama

3. $\delta(q_2, 1) = (q_2, 1, R)$

$\delta(q_2, 0) = (q_3, 1, L)$

Melakukan pencarian ke kanan melewati 1 hingga ketemu 0, lalu mengganti 0 dengan 1.

4. $\delta(q_3, 0) = (q_3, 0, L)$

$\delta(q_3, 1) = (q_3, 1, L)$

$\delta(q_3, B) = (q_0, B, R)$

Bergerak ke kiri simbol B , masuk ke status q_0 .

$$5. \delta(q_2, B) = (q_4, B, L)$$

$$\delta(q_4, 1) = (q_4, B, L)$$

$$\delta(q_4, 0) = (q_4, 0, R)$$

$$\delta(q_4, B) = (q_6, 0, R)$$

Jika dalam status q_2 diperoleh B sebelum 0 , kasusnya seperti nomor 1. Masuk ke status q_4 dan bergerak ke kiri mengganti simbol 1 dengan B sampai menemukan B . Simbol B ditukar dengan 0 , masuk ke status q_6 dan berhenti.

$$6. \delta(q_0, 1) = (q_5, B, R)$$

$$\delta(q_5, 0) = (q_6, B, R)$$

$$\delta(q_5, 1) = (q_6, B, R)$$

$$\delta(q_5, B) = (q_6, B, R)$$

Jika dalam status q_0 ditemukan 1 (bukan 0), kasusnya seperti nomor 1. M masuk ke status q_0 dan menghapus semua simbol.

Contoh rangkaian deskripsi sesaat:

(1) Pita berisi string 0010 (artinya 2 – 1)

$$\begin{array}{l} q_0 0010 \mid Bq_1 010 \mid B0q_1 0 \mid B01q_2 0 \mid \\ B0q_3 11 \mid Bq_3 011 \mid q_3 B011 \mid Bq_0 011 \mid \\ BBq_1 11 \mid BB1q_3 1 \mid BB11q_2 \mid BB1q_4 1 \mid \\ BBq_4 1 \mid Bq_4 \mid B0q_6 \end{array}$$

(2) Pita berisi string 010 (artinya 1 – 2)

$$\begin{array}{l} q_0 0100 \mid Bq_1 00 \mid B1q_2 00 \mid Bq_3 110 \mid \\ q_3 B110 \mid Bq_0 110 \mid BBq_5 10 \mid BBBq_5 0 \mid \\ BBBBq_5 \mid BBBBq_6 \end{array}$$

- Fungsi-fungsi lainnya yang dapat dihitung oleh Mesin Turing:

Fungsi	Operasi simbol
$\Sigma(x, y) = x + y$	Pergeseran 0^y ke kiri
$\Pi(x, y) = x \cdot y$	Salin 0^x sebanyak y kali
$E(x, y) = x^y$	Perkalian $\Pi(x, x)$ berulang kali
$F(x) = x!$	Perkalian $x(x - 1)(x - 2) \dots 2 \cdot 1$

- Untuk mewujudkan fungsi perkalian $\Pi(x, y)$ di atas, diperlukan penyalinan simbol dari suatu lokasi ke lokasi lain. Hal ini dapat dilakukan dengan memanfaatkan mesin Turing penyalin simbol.

- Pemanfaatan sebuah mesin Turing oleh mesin Turing lain mirip dengan pemanggilan prosedur atau *subroutine* oleh program lain (akan dijelaskan nanti).
- Selain pemanggilan *subroutine*, cara lain untuk membentuk mesin Turing yang “besar” dari sejumlah mesin Turing sederhana adalah melalui komposisi fungsi.

Komposisi Fungsi

- Bagaimana menghitung $f(x) = g(h(x))$?
- Misalkan: $h(x)$ dihitung oleh mesin Turing H
 $g(x)$ dihitung oleh mesin Turing G
- Algoritma menghitung $f(x)$:
 1. Dengan input 0^x pada pita, mesin Turing F mensimulasikan gerakan mesin H .
 2. Jika H selesai, output $h(x)$ akan tertera pada pita masukan dalam bentuk $0^{h(x)}$.
 3. Mesin Turing M kemudian mensimulasikan G dengan input $0^{h(x)}$ yang sudah tertera pada pita.
 4. Jika G selesai bekerja, maka pita akan tertera output $g(h(x))$.

- Dengan demikian, jika kita memiliki mesin Turing G dan H yang masing-masing menghitung fungsi $f(x)$ dan $g(x)$ maka komposisi $g(h(x))$ dapat diimplementasikan dengan membungkus kedua mesin Turing G dan H dengan sebuah mesin Turing baru F yang tugasnya mengaktifkan H dan G dalam urutan yang benar.
- Aksi pengaktifan ini dapat dipandang sebagai pemanggilan upa-program yang dilakukan oleh mesin Turing.

Pemanggilan *Subroutine*

- Pada contoh penghitungan fungsi di atas, terlihat adanya kebutuhan untuk mensimulasikan gerakan sebuah mesin Turing A oleh mesin Turing B .
- Aksi pensimulasikan ini dapat dipandang sebagai sebuah pemanggilan *subroutine* B oleh A .
- Hal ini dapat dilakukan jika himpunan status kedua mesin merupakan himpunan yang saling lepas.
- Untuk memanggil B , mesin Turing A harus memasuki status awal dari B . Transisi B juga otomatis harus merupakan transisi dari A .
- Dari status yang menyatakan B dalam keadaan berhenti maka mesin Turing A akan kembali memasuki status yang dimilikinya sendiri.

Contoh: Mesin Turing K akan menghitung perkalian *integer* m dan n , yaitu $m \times n$.

- Mesin Turing K menerima masukan dalam bentuk $0^m 1 0^n$.
- Setelah mesin Turing berhenti, pada pita akan tertera luaran dalam bentuk 0^{mn} .
- Algoritma perkalian m dan n adalah sebagai berikut:
 1. Tuliskan simbol '1' setelah $0^m 1 0^n$.
 2. Salin 0^n ke sebelah kanan simbol '1' sebanyak m kali.
 3. Setiap kali melakukan penyalinan 0^n hapus satu simbol '0' dari 0^m .
 4. Proses penyalinan dihentikan jika tidak ada lagi simbol '0' pada 0^m , sehingga pada akhir siklus kerja mesin Turing akan tertera 0^{mn} pada pita yang menyatakan hasil perkalian.

- Misalkan proses penyalinan 0^n ke lokasi simbol B di sebelah kanan dilakukan oleh mesin Turing C .
- Mesin C mulai dengan deskripsi sesaat $0^m 1 a 0^n 1 0^i$ dan berakhir pada deskripsi sesaat $0^m 1 e 0^n 1 0^{i+n}$. Dalam hal ini, a adalah status awal dan e adalah status akhir.
- Perilaku mesin Turing C diperlihatkan oleh Tabel 1 berikut:

Tabel 1

	0	1	2	B	Keterangan
a	$(b, 2, R)$	$(d, 1, L)$			Status awal
b	$(b, 0, R)$	$(b, 1, R)$		$(c, 0, L)$	Cari B , ganti dengan 0
c	$(c, 0, L)$	$(c, 1, L)$	$(a, 2, R)$		Cari 2
d		$(e, 1, R)$	$(d, 0, L)$		Cari 2 dengan 0, sampai ketemu 1

- Jika mesin Turing K mulai dengan status s dan *head* ditempatkan pada posisi paling kiri, maka sebelum mesin C diaktifkan, posisi *head* harus ditempatkan pada posisi awal rangkaian 0^n yang akan disalin.
- Pengaktifan mesin C dilakukan dengan cara memasuki status a , status awal C .
- Dengan demikian, sebelum C dapat dipanggil, mesin K harus melakukan beberapa aksi tambahan seperti yang dinyatakan pada Tabel 2 berikut. Aksi-aksi ini mengubah simbol '0' pertama lalu bergerak ke kanan sampai dijumpai '1' untuk kemudian memasuki status a , status awal mesin C .

Tabel 2. Persiapan pemanggilan C oleh K

	0	1	2	B	Keterangan
<i>S</i>	(f, B, R)				'0' pertama diganti 'B'
<i>f</i>	$(f, 0, R)$	$(a, 1, R)$			Cari 1 ke kanan, lalu panggil mesin C

- Dengan menggunakan tabel di atas, mesin Turing akan mengalami perubahan deskripsi sesaat dari awal $s0^m10^n$ yaitu sebelum proses penyalinan oleh mesin C dilakukan.
- Proses penyalinan dilakukan oleh mesin C dan akan berhenti pada status e dan diperoleh deskripsi sesaat $B^i 0^{m-i} 1e0^n 10^{ni}$

- Untuk melanjutkan mengulangi proses penyalinan melalui Tabel 1, deskripsi sesaat terakhir harus diubah menjadi $B^{j-1}0^{m-l-1} 1a0^n10^{ni}$. Hal ini dilakukan oleh Tabel 3 berikut:

Tabel 3. Penyiapan pemanggilan ulang C

	0	1	2	B	Keterangan
e	$(g, 0, L)$				Skip 0 tunggal di kiri
g		$(h, 1, L)$			Skip 1 tunggal di kiri
h	$(i, 0, L)$			(j, B, R)	Skip rangkaian 0^{m-l} jika ada
i	$(i, 0, L)$			(s, B, R)	Skip rangkaian 0^{m-l} sampai ketemu B
J		(k, B, R)			Status j dan k : bersihkan 10^n1
k	(k, B, R)				

Sumber:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, *Introduction To Automata Theory , Languages, and Computation 3rd Edition*, Addison Wesley, 2007.
2. Hans Dulimarta, *Catatan Kuliah Matematika Informatika (Bagian Mesin Turing)*, Program Magister Informatika ITB, 2003.
3. Judhi Santoso, *Catatan Kuliah Teori Komputasi*, Program Magister Informatika ITB