KOMPUTASI PEMROGRAMAN

Danang Wahyu Utomo
danang.wu@dsn.dinus.ac.id
+6285 740 955 623

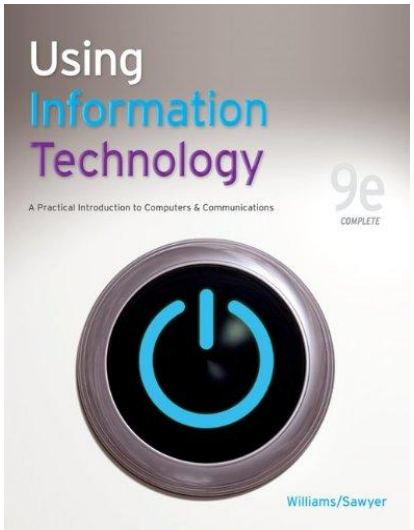
RENCANA KEGIATAN PERKULIAHAN SEMESTER

W	Pokok Bahasan
1	Pengenalan Teknologi Informasi
2	Konsep Sistem Komputer & Pengenalan Perangkat Keras
3	
4	Data Storage
5	Perangkat Lunak
6	
7	Data dan Informasi
8	**Ujian Tengah Semester**

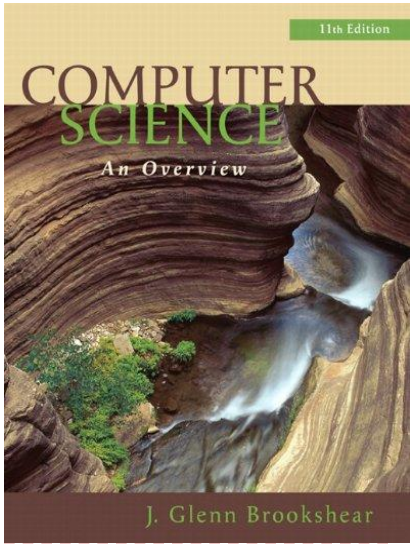
W	Pokok Bahasan
9	Komputasi Pemrograman
10	
11	Rekayasa Perangkat Lunak
12	Komunikasi data & Jaringan Komputer
13	
14	Etika dan dampak sosial teknologi informasi
15	Teknologi Terkini / Advance Topik
16	**Ujian Akhir Semester**

Reference

- Bruce K William, Stacey C. Sawyer – Using Information Technology : A Practical Introduction to Computers & Communications 9th Edition (2010)



- J. Glenn Brookshear – Computer Science : An Overview 11th Edition (2011)



Content

- Teori Komputasi
- Mesin Turing
- Complexity of Problem
- Paradigma Pemrograman

Fungsi dan Komputasi

- Kita perlu memahami apa yang dapat dan tidak dapat dilakukan oleh komputer
- Kita perlu memahami konsep dari fungsi komputasi

Fungsi dan Komputasi

- Fungsi

  dalam pengertian matematika adalah korespondensi antara sekumpulan nilai – nilai yang mungkin dan sekumpulan nilai output, sehingga setiap nilai input yang mungkin diberikan satu output.

  Brookshear, 2011

  contoh :

  fungsi konversi dari ‘yard’ ke meter

Fungsi dan Komputasi

- Computing the Function

  proses menentukan nilai output tertentu dimana sebuah fungsi diberikan pada sebuah nilai input

- Kemampuan untuk menghitung fungsi penting, karena dengan fungsi komputasi dapat digunakan untuk memecahkan masalah

Brookshear, 2011

Fungsi dan Komputasi

- Contoh

  sebuah sistem konversi suhu, dimana fungsi input dan output ditentukan dan disimpan dalam sebuah tabel sebagai berikut :

C	F
10	50
20	68
50	122
100	212
…	…

Fungsi dan Komputasi

- Pendekatan seperti ini tidak dapat merepresentasikan keseluruhan nilai, karena tidak ada batasan pasangan nilai input – output :

C	F
10	50
20	68
50	122
100	212
…	…

Fungsi dan Komputasi

- Pendekatan yang lebih baik untuk fungsi komputasi adalah dengan menggunakan rumus :

F = C * 1,8 + 32

Fungsi dan Komputasi

- Computable Function

  fungsi dimana nilai output dapat ditentukan secara algoritmik dari nilai input

- Non Computable Function

  fungsi dimana nilai output tidak dapat ditentukan secara jelas, tahap demi tahap dari nilai input

Fungsi dan Komputasi

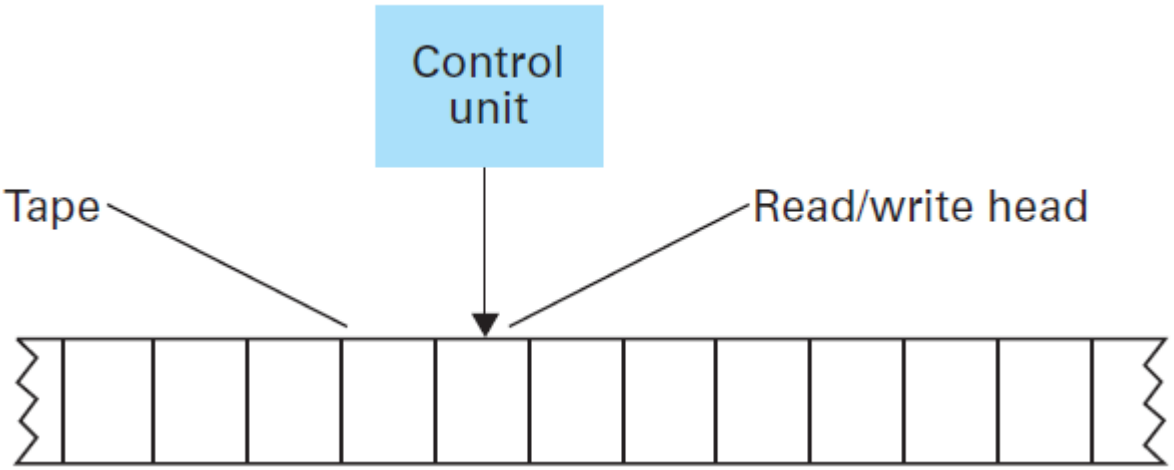
- Perbedaan antara computable dan non computable fungsi penting di bidang ilmu komputer
- Mesin (komputer) hanya dapat melakukan tugas yang dijelaskan melalui algoritma
- Dengan mengetahui kemampuan dari sebuah mesin untuk menghitung keseluruhan set dari computable function, maka dapat dibangun sebuah mesin yang memiliki kemampuan yang diinginkan
- Untuk memahami kemampuan dan batasan – batasan dari mesin, para peneliti mengusulkan dan mempelajari berbagai perangkat komputasi

Turing Machine

- Salah satunya adalah ‘Mesin Turing’ yang diusulkan oleh Alan M. Turing pada 1936 dan masih digunakan hingga saat ini sebagai alat untuk mempelajari kekuatan dari proses algoritmik

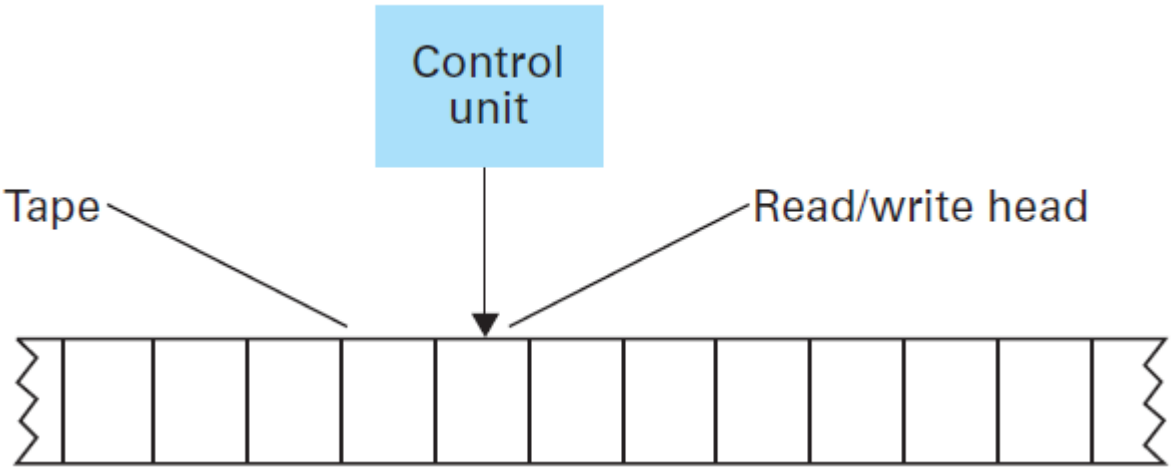
Turing Machine

- Mesin Turing terdiri dari sebuah control unit yang dapat membaca dan menuliskan simbol pada sebuah pita (tape) melalui sebuah *head*
- Tape tersebut dapat diperpanjang tanpa batas dan terbagi dalam cell - cell



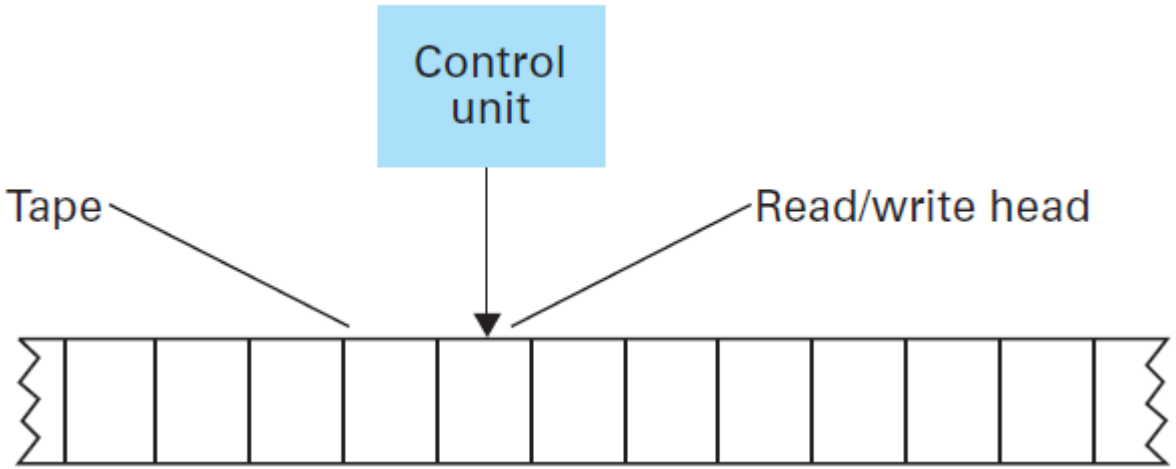
Turing Machine

- Setiap cell dapat memuat salah satu dari himpunan simbol terhingga
- Himpunan simbol tersebut disebut alphabet



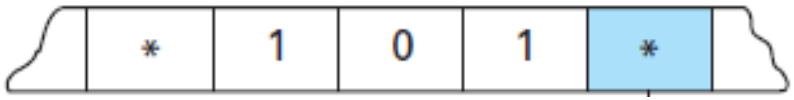
Turing Machine

- Saat melakukan komputasi, mesin turing harus berada pada salah satu kondisi tertentu yang disebut *state*
- Mesin turing memulai komputasi dari state yang disebut *start state* dan berakhir ketika mencapai *halt state*



Turing Machine

- Contoh :
  - Diketahui sebuah tape bernilai 5 (101_2)

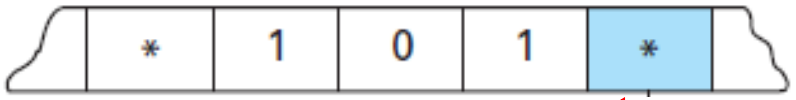


  - Dan sebuah tabel turing mesin untuk menambah nilai

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

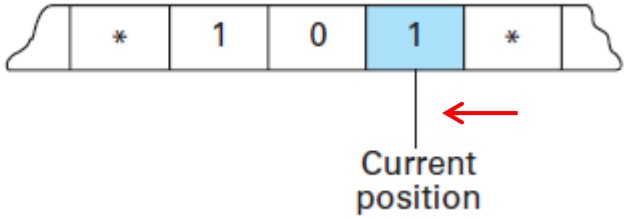
- Current State : START



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

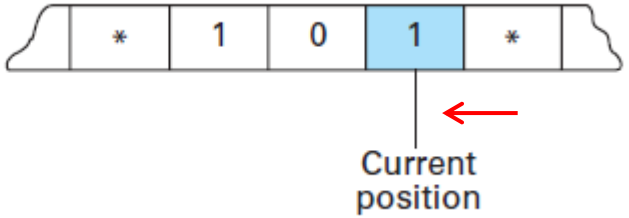
- Current State : START



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

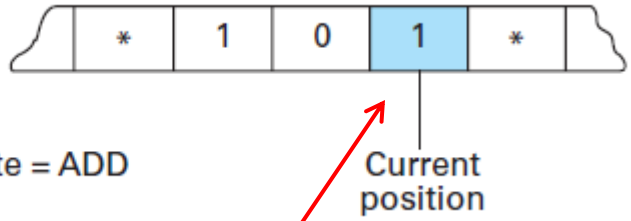
- Current State : START → New State : ADD



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

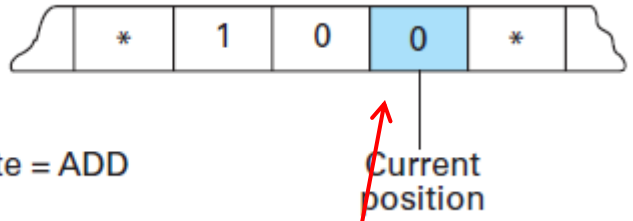
- Current State : ADD



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

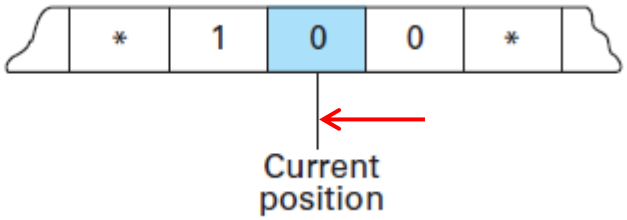
- Current State : ADD



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

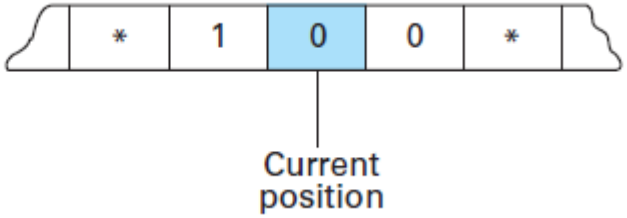
- Current State : ADD



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

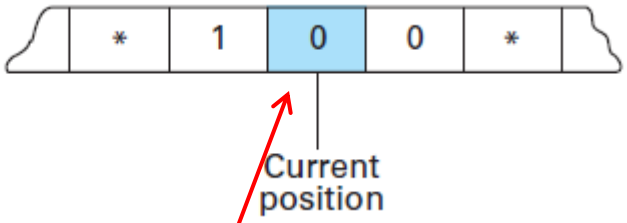
- Current State : ADD → New State : CARRY



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

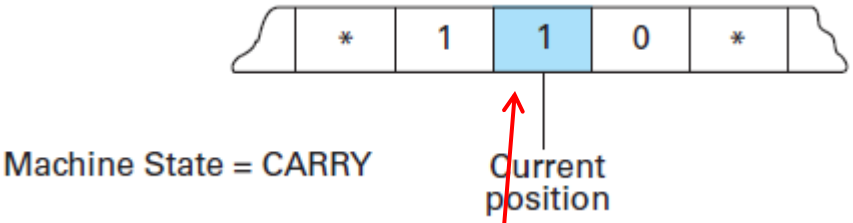
- Current State : CARRY



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

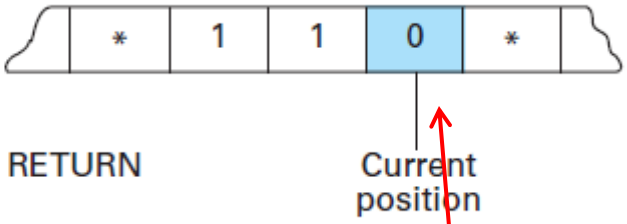
- Current State : CARRY



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

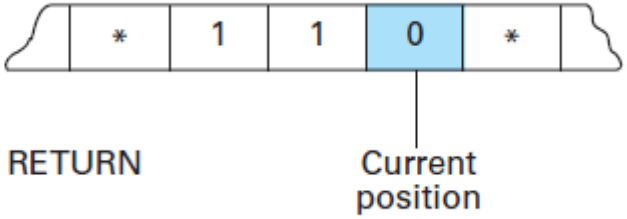
- Current State : CARRY



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

- Current State : CARRY → New State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

- Current State : RETURN

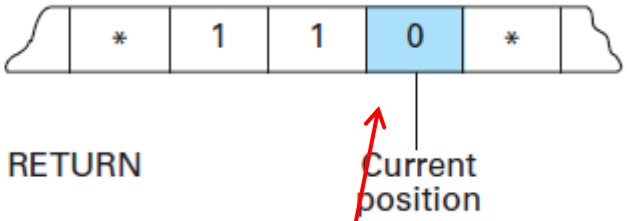
Machine State = RETURN

Current position

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

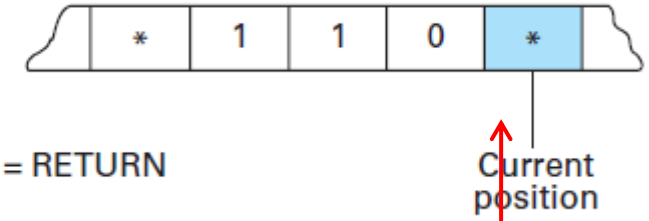
- Current State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

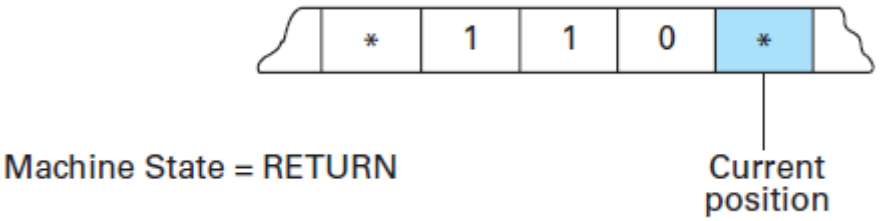
- Current State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

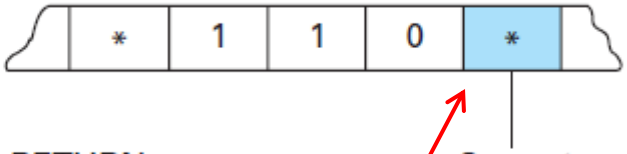
- Current State : RETURN → New State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

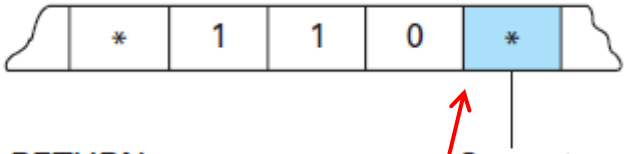
- Current State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

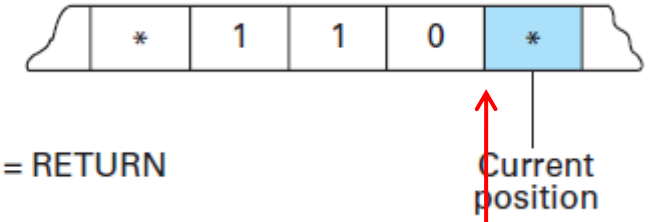
- Current State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

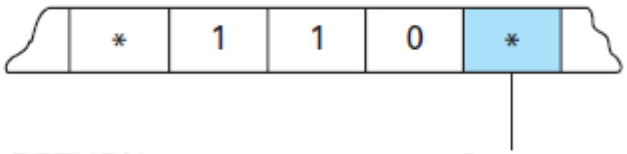
- Current State : RETURN



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

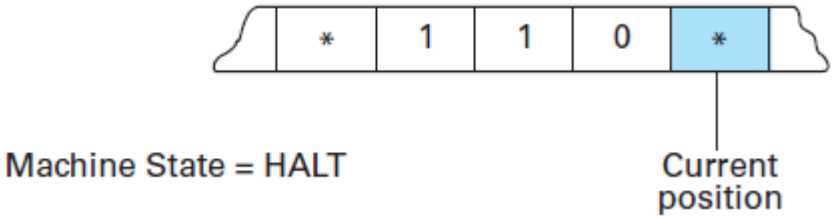
- Current State : RETURN → New State : HALT



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

- Current State : HALT

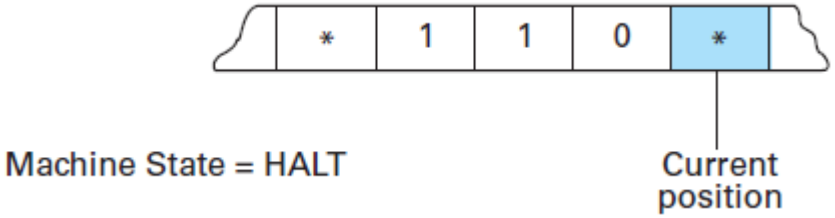


Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Turing Machine

- Current State : HALT

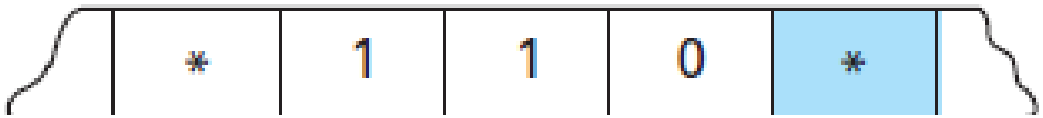
Nilai telah ditambahkan → Tape bernilai 6 (110_2)



Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Latihan

- Contoh :
  - Diketahui sebuah tape bernilai 6 (110_2)



  - Dan sebuah tabel turing mesin untuk menambah nilai

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	(Ignored)	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

Complexity of Problem

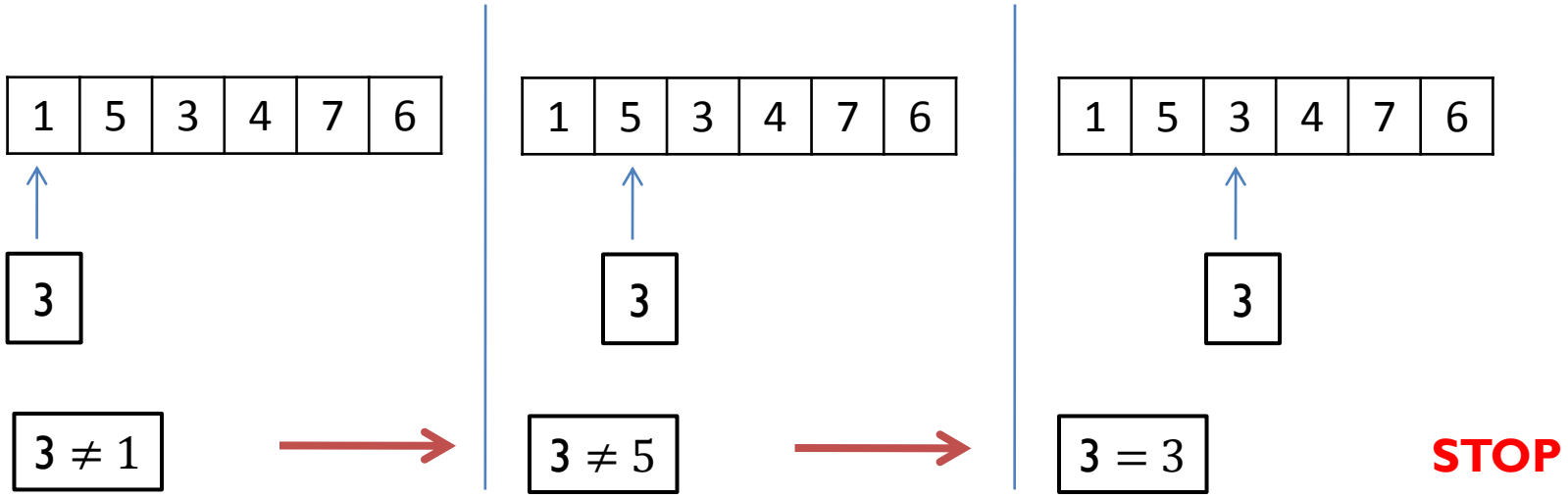
- Mesin memiliki kemampuan untuk mengeksekusi jutaan instruksi dalam tiap second
- Efficiency merupakan problem dalam suatu algoritma
- Contoh :
  - Student record $\rightarrow$ updating, searching, retrieving
  - Untuk menemukan data siswa menggunakan **pencarian** dalam daftar siswa

  Misal : sequential search dan binary search

Complexity of Problem

- Sequential search mulai pencarian dari list awal dan membandingkan semua elemen

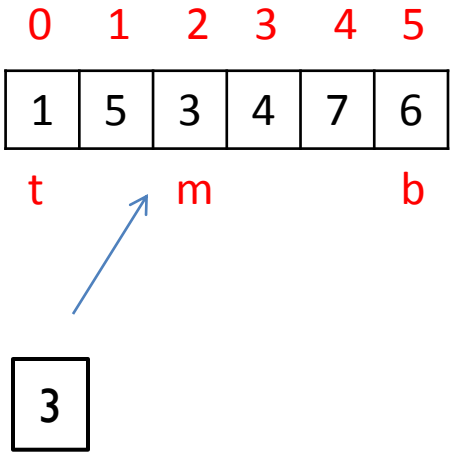
  misal : 1 5 3 4 7 6 x = 3



Complexity of Problem

- Binary search mulai pencarian dengan membandingkan nilai kunci dan nilai tengah dari semua elemen

  misal : 1 5 3 4 7 8 x = 3



Middle = (top + bottom) / 2

3 = 3, stop how about x=4 ?

Complexity of Problem

- efficiency dari suatu algoritma penting dalam *time* atau *storage space*
- Beberapa problem memiliki tingkat kompleksitas yang tinggi
- Tingkat kompleksitas diukur menggunakan notasi Θ (*big-theta*) untuk mengelompokkan algoritma berdasarkan waktu eksekusi yang diperlukan.
- Contoh :

Algoritma Sequential Search	→	Θ (N)
Algoritma Binary Search	→	Θ (log N)

TERIMA KASIH