

# String Matching dengan Regular Expression

Masayu Leylia Khodra

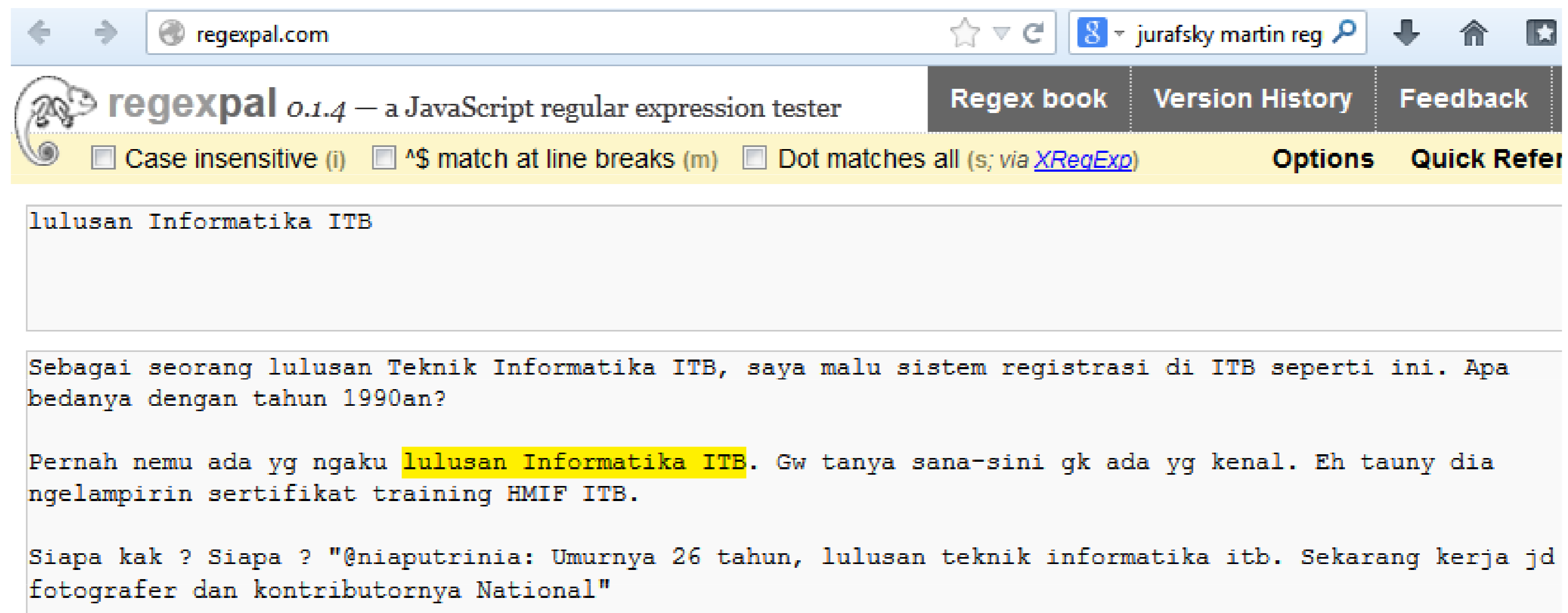
Referensi:

Chapter 2 of *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, by Daniel Jurafsky and James H. Martin  
15-211 *Fundamental Data Structures and Algorithms*, by Ananda Gunawardena

## String Matching: Definisi

- Diberikan:
  1.  $T$ : teks (*text*), yaitu (*long*) *string* yang panjangnya  $n$  karakter
  2.  $P$ : *pattern*, yaitu *string* dengan panjang  $m$  karakter (asumsi  $m \ll n$ ) yang akan dicari di dalam teks.Carilah (*find* atau *locate*) lokasi pertama di dalam teks yang bersesuaian dengan *pattern*.

## String Matching : Contoh 1



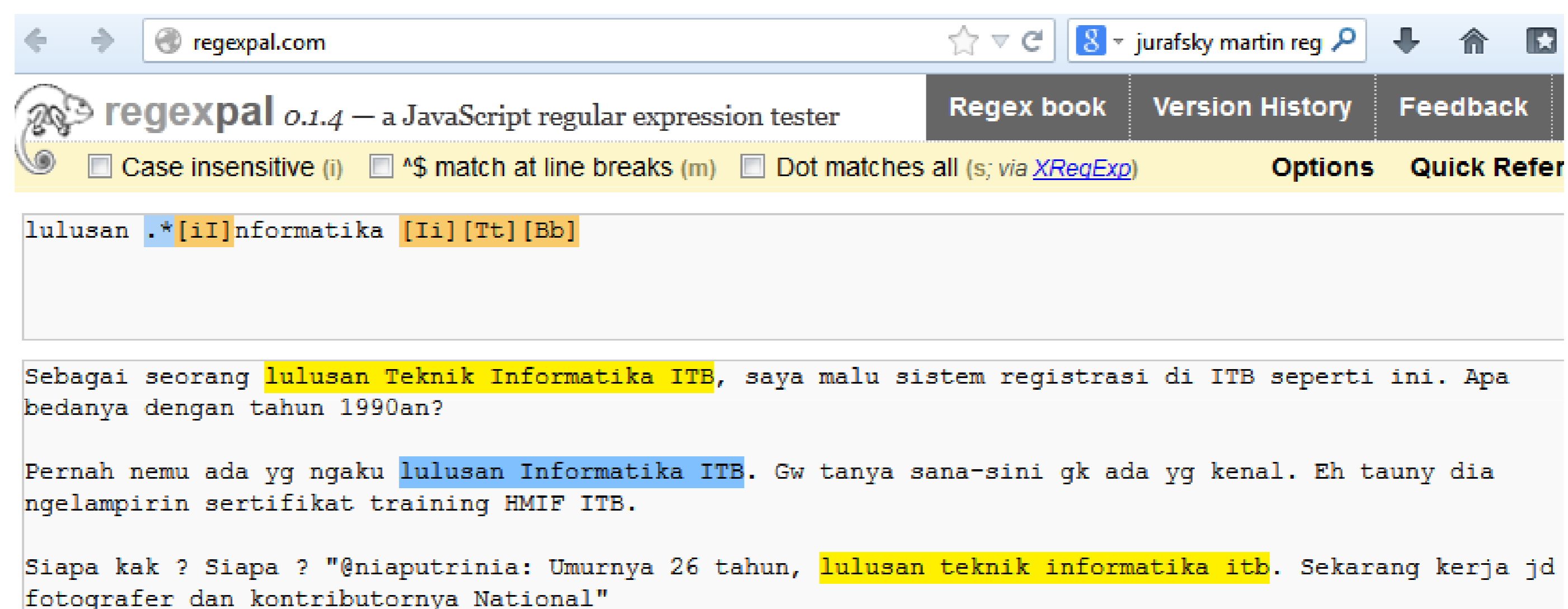
lulusan Informatika ITB

Sebagai seorang lulusan Teknik Informatika ITB, saya malu sistem registrasi di ITB seperti ini. Apa bedanya dengan tahun 1990an?

Pernah nemu ada yg ngaku **lulusan Informatika ITB**. Gw tanya sana-sini gk ada yg kenal. Eh tauny dia ngelampirin sertifikat training HMIF ITB.

Siapa kak ? Siapa ? "@niaputrinia: Umurnya 26 tahun, lulusan teknik informatika itb. Sekarang kerja jd fotografer dan kontributornya National"

## String Matching : Contoh 2



lulusan **.\*[iI]nformatika [Ii][Tt][Bb]**

Sebagai seorang **lulusan Teknik Informatika ITB**, saya malu sistem registrasi di ITB seperti ini. Apa bedanya dengan tahun 1990an?

Pernah nemu ada yg ngaku **lulusan Informatika ITB**. Gw tanya sana-sini gk ada yg kenal. Eh tauny dia ngelampirin sertifikat training HMIF ITB.

Siapa kak ? Siapa ? "@niaputrinia: Umurnya 26 tahun, **lulusan teknik informatika itb**. Sekarang kerja jd fotografer dan kontributornya National"

## Notasi Umum Regex

Regex book	Version History	Feedback	Blog
Options		Quick Reference	
.	Any character except newline.		
\.	A period (and so on for \*, \ (, \ \, etc.)		
^	The start of the string.		
\$	The end of the string.		
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.		
\D,\W,\S	Anything except a digit, word character, or whitespace.		
[abc]	Character a, b, or c.		
[a-z]	a through z.		
[^abc]	Any character except a, b, or c.		
aa bb	Either aa or bb.		
?	Zero or one of the preceding element.		
*	Zero or more of the preceding element.		
+	One or more of the preceding element.		
{n}	Exactly n of the preceding element.		
{n, }	n or more of the preceding element.		
{m, n}	Between m and n of the preceding element.		
??,*?,+?, {n}?, etc.	Same as above, but as few as possible.		
(expr)	Capture expr for use with \1, etc.		
(?:expr)	Non-capturing group.		
(?=expr)	Followed by expr.		
(?!expr)	Not followed by expr.		

[Near-complete reference](#)

## String Matching : Contoh 2

regexpal.com

jurafsky martin reg

regexpal 0.1.4 — a JavaScript regular expression tester

Case insensitive (i)  \$ match at line breaks (m)  Dot matches all (s; via XRegExp)

Options Quick Reference

lulusan .\*[Ii]nformatika [Ii][Tt][Bb]

Sebagai seorang lulusan Teknik Informatika ITB, saya mal bedanya dengan tahun 1990an?

Pernah nemu ada yg ngaku lulusan Informatika ITB. Gw tan ngelampirin sertifikat training HMIF ITB.

Siapa kak ? Siapa ? "@niaputrinia: Umurnya 26 tahun, lulu fotografer dan kontributornya National"

Any character except newline.  
A period (and so on for \\*, \ (, \ \, etc.)  
The start of the string.  
The end of the string.  
A digit, word character [A-Za-z0-9\_], or whitespace.  
Anything except a digit, word character, or whitespace.  
Character a, b, or c.  
a through z.  
Any character except a, b, or c.  
Either aa or bb.  
Zero or one of the preceding element.  
Zero or more of the preceding element.  
One or more of the preceding element.  
Exactly n of the preceding element.  
n or more of the preceding element.  
Between m and n of the preceding element.  
Same as above, but as few as possible.  
Capture expr for use with \1, etc.  
Non-capturing group.  
Followed by expr.  
Not followed by expr.

[Near-complete reference](#)

## String Matching : Contoh 3

regexpal.com

regexpal 0.1.4 — a JavaScript regular expression tester

Case insensitive (i)  ^\$ match at line breaks (m)  Dot matches all (s; via [XRegExp](#))

`[\\w-]+ ?(?:@|\\(?:at\\)? ?(?:[\\w-]+(\\.| ?dot ?)))+[a-zA-Z]{2,4}`

```
test.txt - obfuscate('stanford.edu','jurafsky' - jurafsky@stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky(at)cs.stanford.edu - jurafsky@cs.stanford.edu;
test.txt - jurafsky at csli dot stanford dot edu - jurafsky@csli.stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky@cs.stanford.edu - jurafsky@cs.stanford.edu;
test.txt - jurafsky@csli.stanford.edu - jurafsky@csli.stanford.edu;
test.txt - 650-723-0293 - 650-723-0293;
test.txt - (650) 723-0293 - 650-723-0293;
test.txt - 650-723-0293 - 650-723-0293;
test.txt - 650&thinsp;723&thinsp;0293 - 650-723-0293;
test.txt - 650-723-0293 - 650-723-0293;
```

## String Matching : Contoh 4

regexpal.com

regexpal 0.1.4 — a JavaScript regular expression tester

Case insensitive (i)  ^\$ match at line breaks (m)  Dot matches all (s; via [XRegExp](#))

`(\\(?:d{3}\\)?[ -])+\d{4}`

```
test.txt - obfuscate('stanford.edu','jurafsky' - jurafsky@stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky(at)cs.stanford.edu - jurafsky@cs.stanford.edu;
test.txt - jurafsky at csli dot stanford dot edu - jurafsky@csli.stanford.edu;
test.txt - jurafsky@stanford.edu - jurafsky@stanford.edu;
test.txt - jurafsky@cs.stanford.edu - jurafsky@cs.stanford.edu;
test.txt - jurafsky@csli.stanford.edu - jurafsky@csli.stanford.edu;
test.txt - 650-723-0293 - 650-723-0293;
test.txt - (650) 723-0293 - 650-723-0293;
test.txt - 650-723-0293 - 650-723-0293;
test.txt - 650&thinsp;723&thinsp;0293 - 650-723-0293;
test.txt - 650-723-0293 - 650-723-0293;
```

## Basic Regular Expression Patterns

- The use of the brackets [ ] to specify a disjunction of characters.

RE	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	" <u>W</u> oodchuck"
/[abc]/	'a', 'b', or 'c'	"In uomini, in soldati"
/[1234567890]/	any digit	"plenty of <u>7</u> to 5"

- The use of the brackets [ ] plus the dash – to specify a range.

RE	Match	Example Patterns Matched
/[A-Z]/	an uppercase letter	"we should call it ' <u>D</u> renched Blossoms'"
/[a-z]/	a lowercase letter	" <u>m</u> y beans were impatient to be hoed!"
/[0-9]/	a single digit	"Chapter <u>1</u> : Down the Rabbit Hole"

## Basic Regular Expression Patterns

- Uses of the caret ^ for negation or just to mean ^

RE	Match (single characters)	Example Patterns Matched
[^A-Z]	not an uppercase letter	"Oyfn pripetchik"
[^Ss]	neither 'S' nor 's'	" <u>I</u> have no exquisite reason for't"
[^\.]	not a period	" <u>o</u> ur resident Djinn"
[e^]	either 'e' or '^'	"look up <u>^</u> now"
a^b	the pattern 'a^b'	"look up <u>a^</u> b now"

- The question-mark ? marks optionality of the previous expression.

RE	Match	Example Patterns Matched
woodchucks?	woodchuck or woodchucks	" <u>w</u> oodchuck"
colou?r	color or colour	" <u>c</u> olour"

- The use of period . to specify any character

RE	Match	Example Patterns
/beg.n/	any character between beg and n	<u>b</u> egin, <u>b</u> eg'n, <u>b</u> egun

## Finite State Machines (FSM)

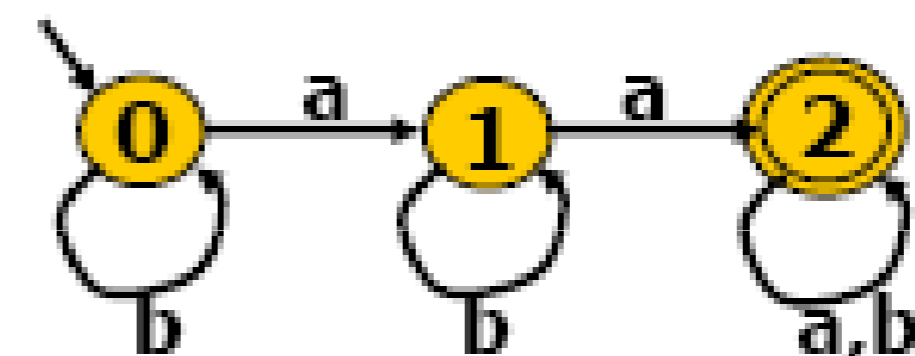
- FSM is a computing machine that takes
  - A string as an input
  - Outputs YES/NO answer
    - That is, the machine “accepts” or “rejects” the string



Referensi: Gunawardena, 2006

## FSM Model

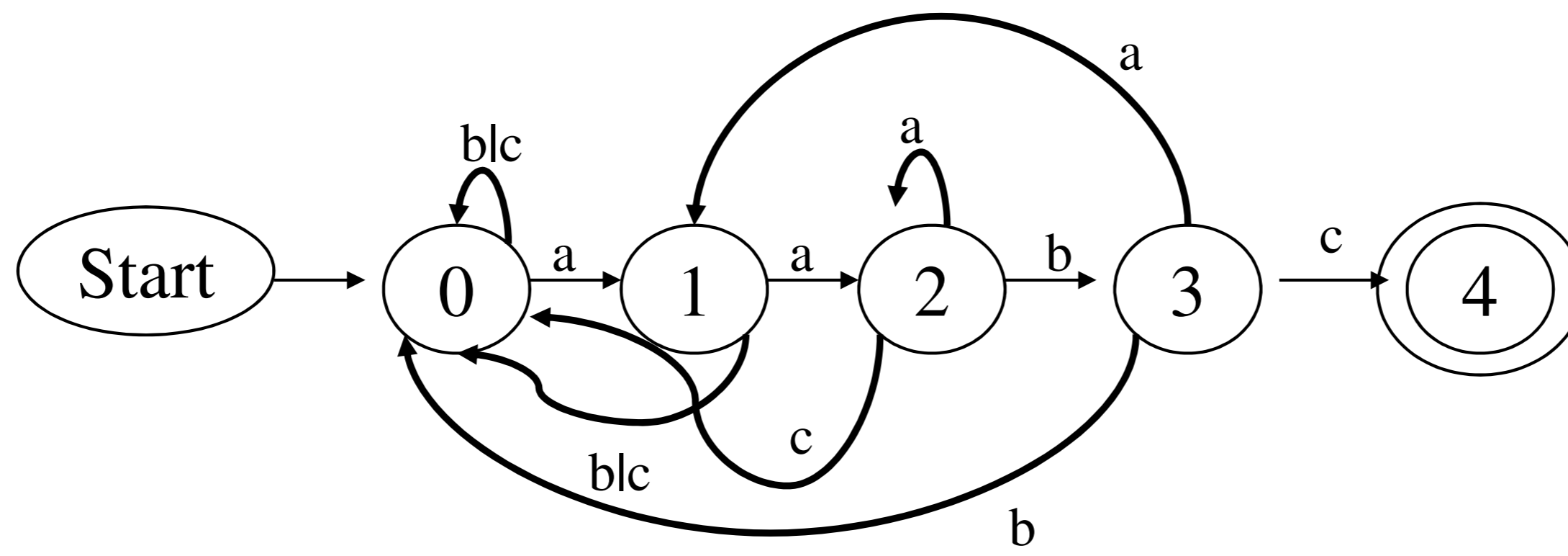
- **Input to a FSM**
  - Strings built from a fixed alphabet  $\{a,b,c\}$
  - Possible inputs: aa, aabbcc, a etc..
- **The Machine**
  - A directed graph
    - Nodes = States of the machine
    - Edges = Transition from one state to another
- **Special States**
  - Start ( $q_0$ ) and Final (or Accepting) ( $q_2$ )
- **Assume the alphabet is  $\{a,b\}$** 
  - Which strings are accepted by this FSM?



Referensi: Gunawardena, 2006

## FSM untuk String Matching

- Alphabet {a,b,c}
- Pattern "abc"
- String: aaaaaaaaaabcccccccccccccccc



Referensi: Gunawardena, 2006