

String Matching dengan Regular Expression

Masayu Leylia Khodra

Referensi:

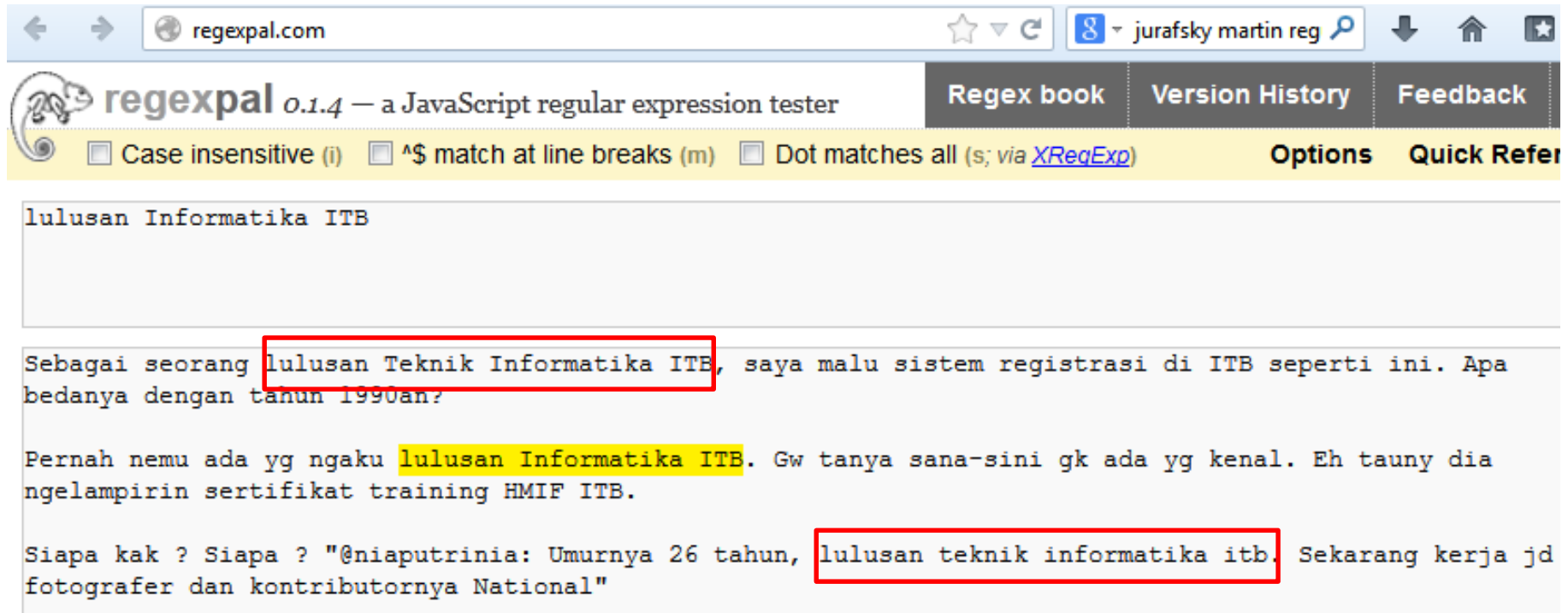
Chapter 2 of *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, by Daniel Jurafsky and James H. Martin
15-211 *Fundamental Data Structures and Algorithms*, by Ananda Gunawardena

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 $m \lll n$) yang akan dicari di dalam teks.

Carilah (*find* atau *locate*) di dalam teks yang bersesuaian dengan *pattern*.

Contoh 1: Exact Matching



The screenshot shows the regexpal.com website interface. The browser address bar displays 'regexpal.com'. The page title is 'regexpal 0.1.4 - a JavaScript regular expression tester'. The navigation menu includes 'Regex book', 'Version History', and 'Feedback'. The options bar shows three checked checkboxes: 'Case insensitive (i)', '^\$ match at line breaks (m)', and 'Dot matches all (s; via XRegExp)'. The search input field contains the text 'lulusan Informatika ITB'. The search results display three text snippets with exact matches highlighted in red boxes:

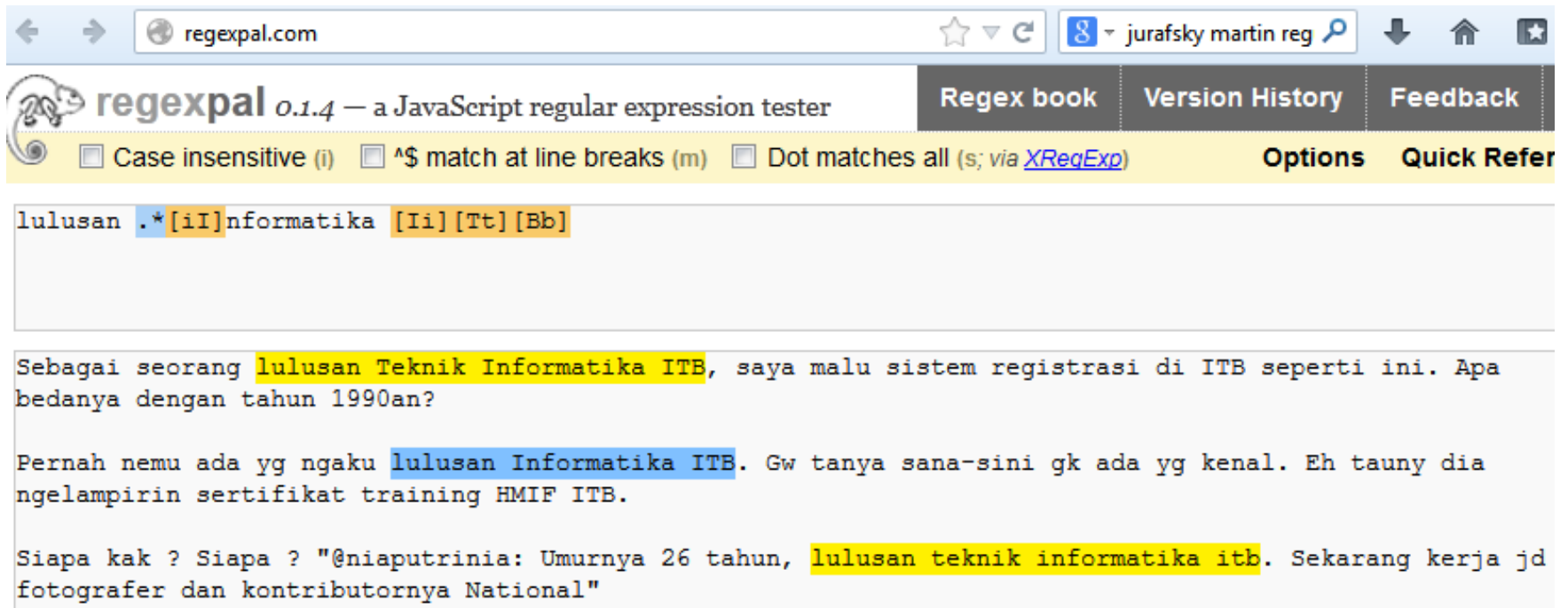
```
lulusan Informatika ITB
```

```
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"
```

Contoh 2: Regex Matching



regexpal 0.1.4 — a JavaScript regular expression tester

Regex book Version History Feedback

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



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?

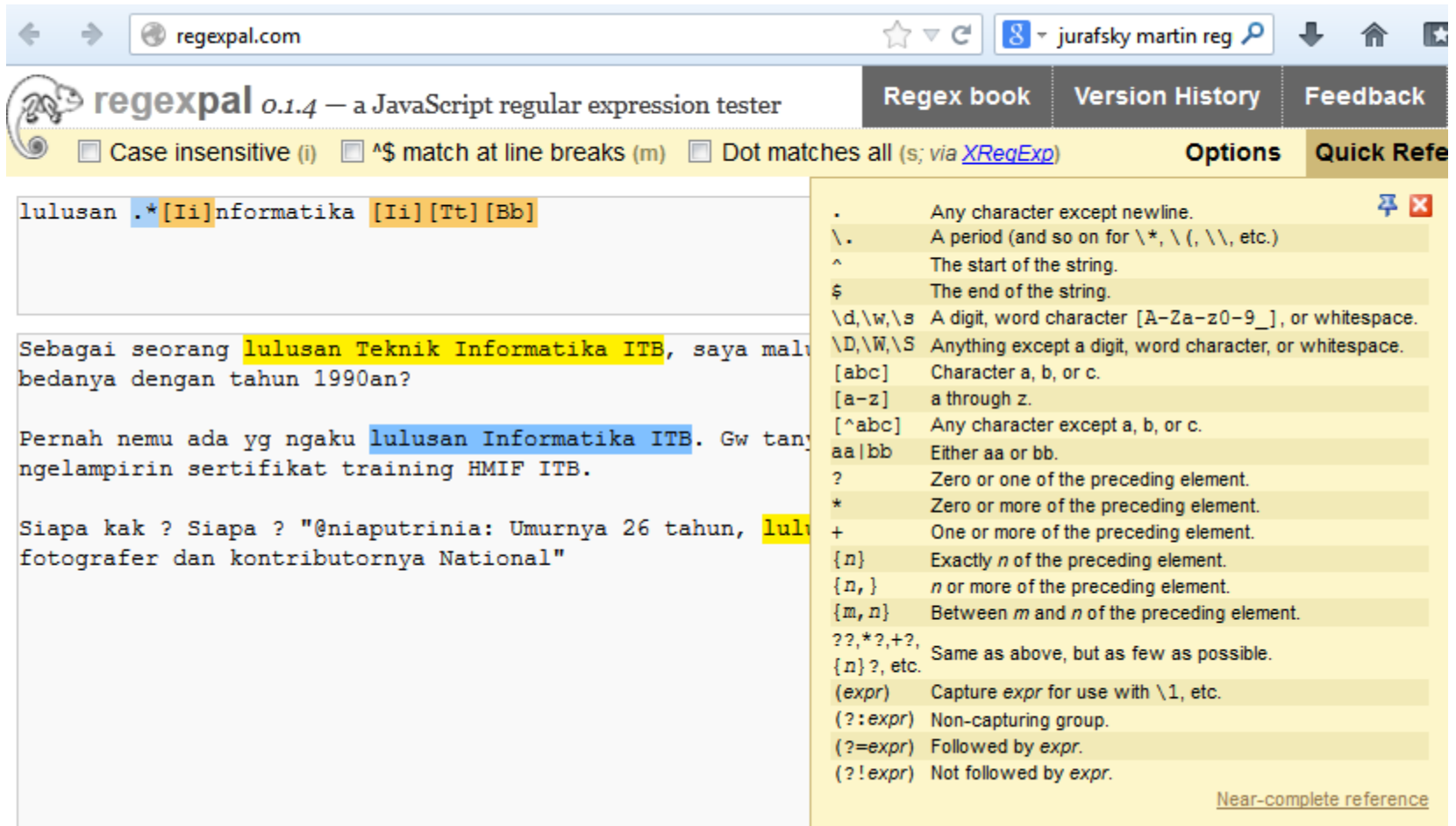
Pernah nemu ada yg ngaku lulusan Informatika ITB. Gw tanya sana-sini gk ada yg kenal. Eh tauny dia ngelampirin sertifikat training HMIF ITB.

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 <i>n</i> of the preceding element.		
{n,}	<i>n</i> or more of the preceding element.		
{m,n}	Between <i>m</i> and <i>n</i> of the preceding element.		
??,*?,+?, {n}?, etc.	Same as above, but as few as possible.		
(<i>expr</i>)	Capture <i>expr</i> for use with \1, etc.		
(?: <i>expr</i>)	Non-capturing group.		
(?= <i>expr</i>)	Followed by <i>expr</i> .		
(?! <i>expr</i>)	Not followed by <i>expr</i> .		
Near-complete reference			

Notasi Regex



The screenshot shows the website `regexpal.com` with the title "regexpal 0.1.4 — a JavaScript regular expression tester". The browser address bar shows "regexpal.com" and a search bar contains "jurafsky martin reg". The page has navigation links for "Regex book", "Version History", and "Feedback". Below the navigation, there are checkboxes for "Case insensitive (i)", "\$ match at line breaks (m)", and "Dot matches all (s; via XRegExp)".

The main content area shows a text input field with the regex `.*[Ii]nformatika [Ii][Tt][Bb]` applied to a sample text. The text is: "Sebagai seorang **lulusan Teknik Informatika ITB**, saya malu bedaanya dengan tahun 1990an? Pernah nemu ada yg ngaku **lulusan Informatika ITB**. Gw tanya ngelampirin sertifikat training HMIF ITB. Siapa kak ? Siapa ? "@niaputrinia: Umurnya 26 tahun, **lulu** fotografer dan kontributornya National"

On the right side, there is a "Quick Reference" table listing various regex symbols and their meanings:

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

[Near-complete reference](#)

Contoh 3: Regex for Email

regexpal.com

regexpal 0.1.4 — a JavaScript regular expression tester

Regex book Vers

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


Tentukan regexnya untuk semua email yang diwarnai

```
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;
```

.	Any character e
\.	A period (and s
^	The start of the
\$	The end of the
\d,\w,\s	A digit, word ch
\D,\W,\S	Anything except
[abc]	Character a, b,
[a-z]	a through z.
[^abc]	Any character e
aa bb	Either aa or bb.
?	Zero or one of t
*	Zero or more of
+	One or more of
{n}	Exactly n of the
{n,}	n or more of the
{m,n}	Between m and
??,*?,+?,	Same as above
{n}?, etc.	
(expr)	Capture expr fo
(?:expr)	Non-capturing c
(?=expr)	Followed by exp
(?!expr)	Not followed by

Contoh 4: Regex for Phone Number

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 uo <u>m</u> ini, in soldat <u>i</u> ”
/ [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	“O <u>y</u> fn 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>woodchuck</u> ”
colou?r	color or colour	“ <u>colour</u> ”

- The use of period . to specify any character

RE	Match	Example Patterns
/beg.n/	any character between <i>beg</i> and <i>n</i>	<u>begin</u> , <u>beg’n</u> , <u>begun</u>

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



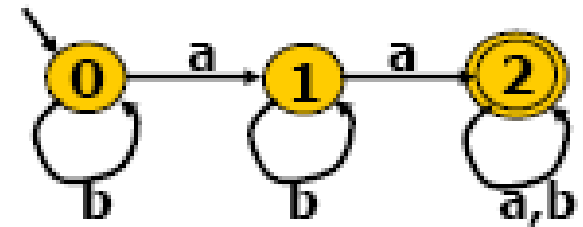
FSM Model

- **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**

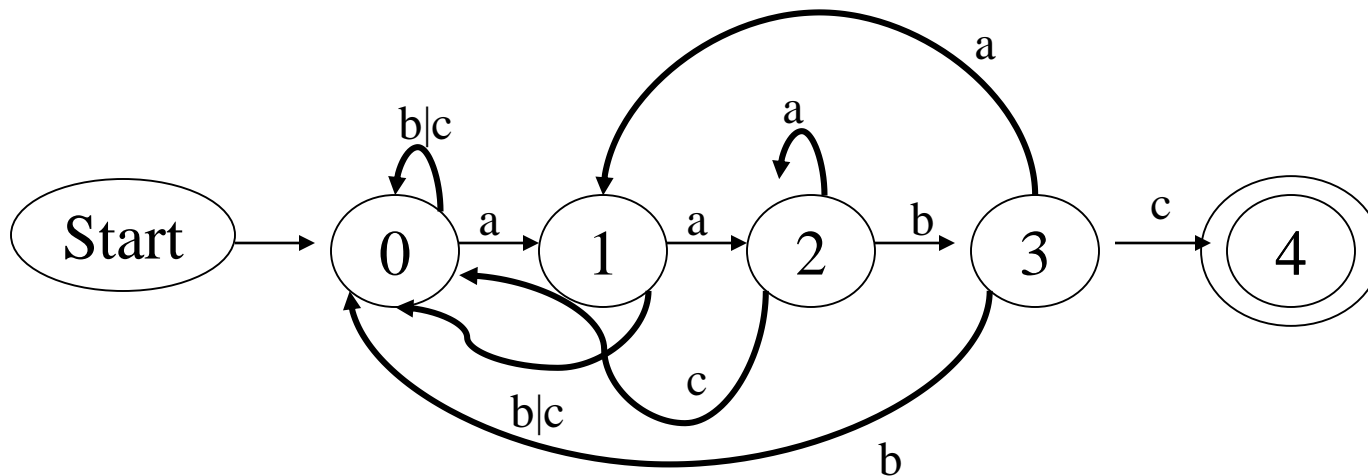
- Start (q0) and Final (or Accepting) (q2)

- **Assume the alphabet is {a,b}**

- Which strings are accepted by this FSM?

FSM untuk String Matching

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



Regex di Java

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

`\d{2}\.\d{2}`

```
#lalinBDG 09.16 : yg mau ke jln sudirman dsk, :  
@infobdg: #lalinBDG 08.23 : Macet (lagi) rancas  
#suaraBDG via @dionmudjenan: Hati-hati jembatan  
RT @quinsymegamira: leuwipanjang banyak anak j...
```

```
public static void extraction(Pattern myPattern, String str) {  
    String extract;  
  
    Matcher m;  
    m = myPattern.matcher(str);  
    while(m.find()) {  
        extract = m.group();  
        System.out.println(extract);  
    }  
}
```

```
public static void main(String[] args) {  
    resources.extraction(Pattern.compile("\\d{2}\\.\d{2}"), "@infobdg: #
```

it %

Debugger Console x veritransTools (run) x

run:

08.23

BUILD SUCCESSFUL (total time: 0 seconds)