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

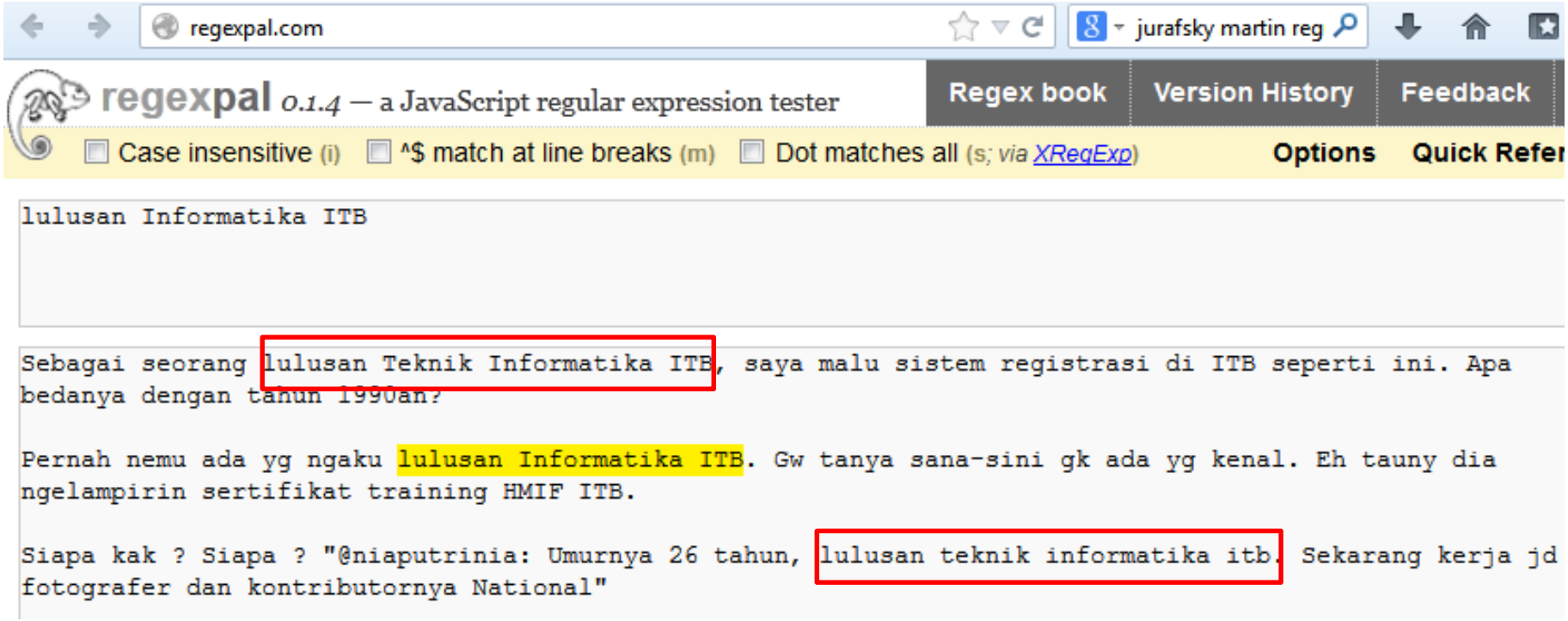
15-211 Fundamental Data Structures and Algorithms, by Ananda Gunawardena

String Matching: Definisi

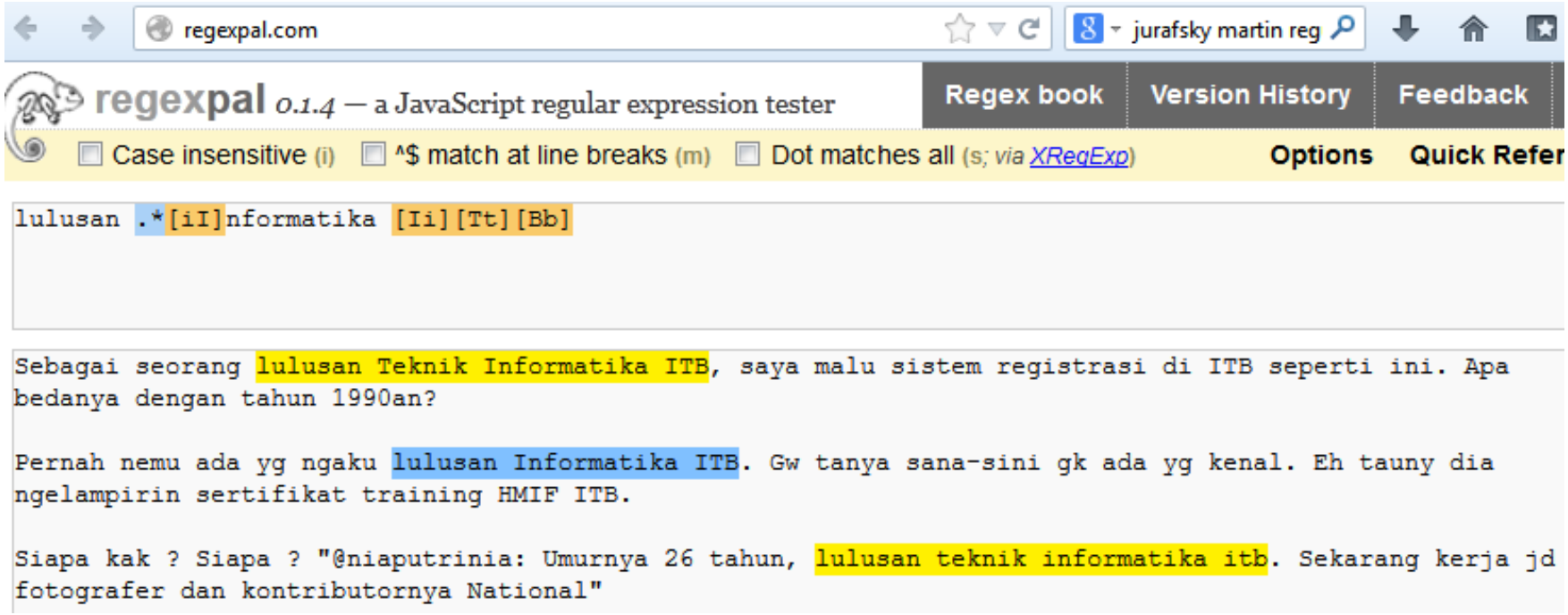
- Diberikan:
  1. *T*: teks (*text*), yaitu (*long*) *string* yang panjangnya *n* karakter
  2. *P: pattern*, yaitu *string* dengan panjang *m* karakter (asumsi *m* <<< *n*) yang akan dicari di dalam teks.

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

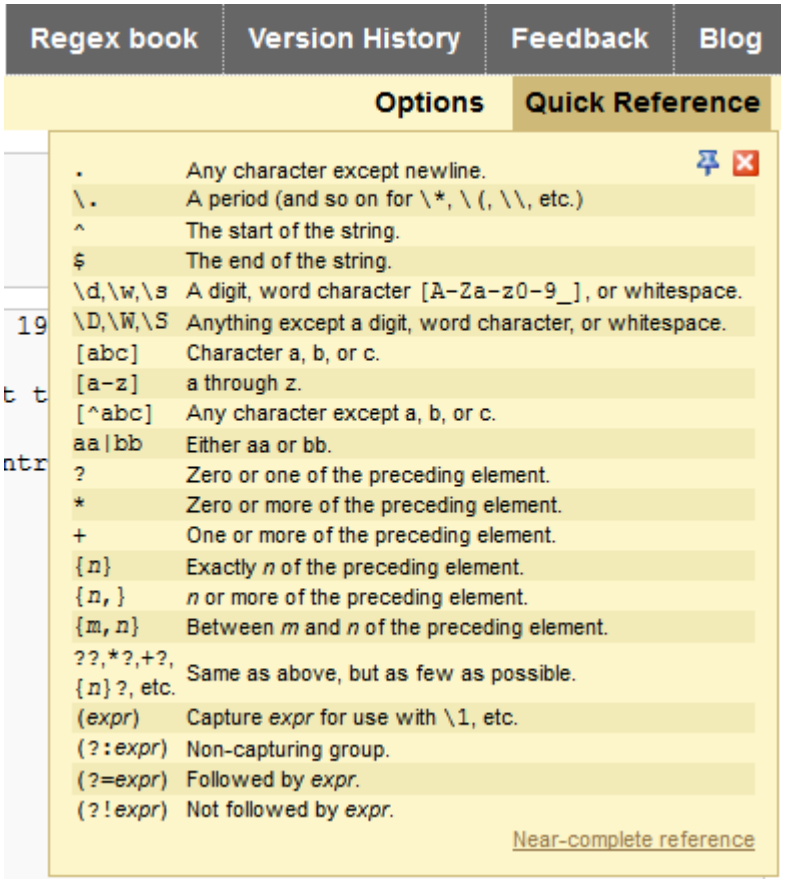
Contoh 1: Exact Matching



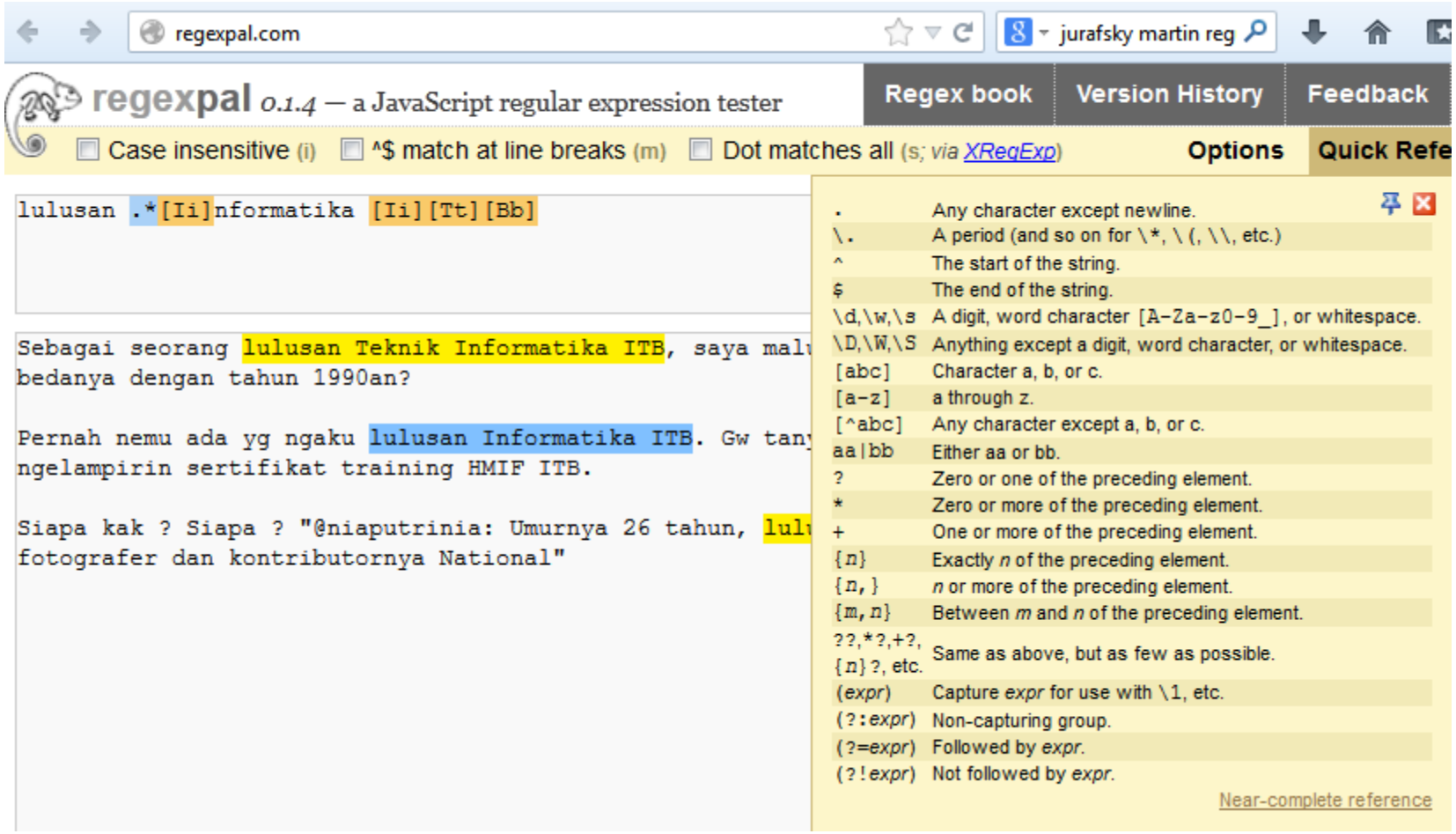
Contoh 2: Regex Matching



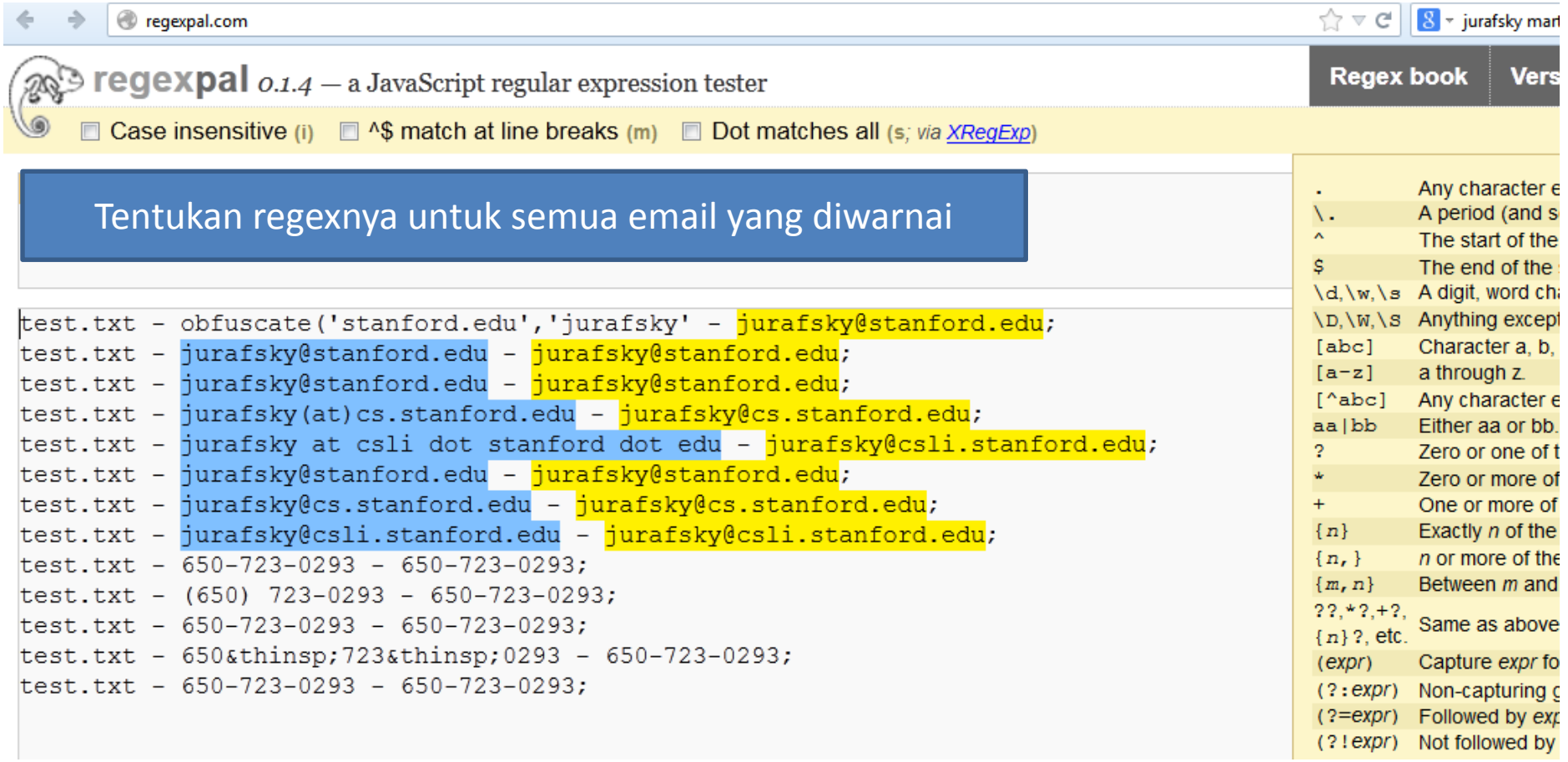
Notasi Umum Regex



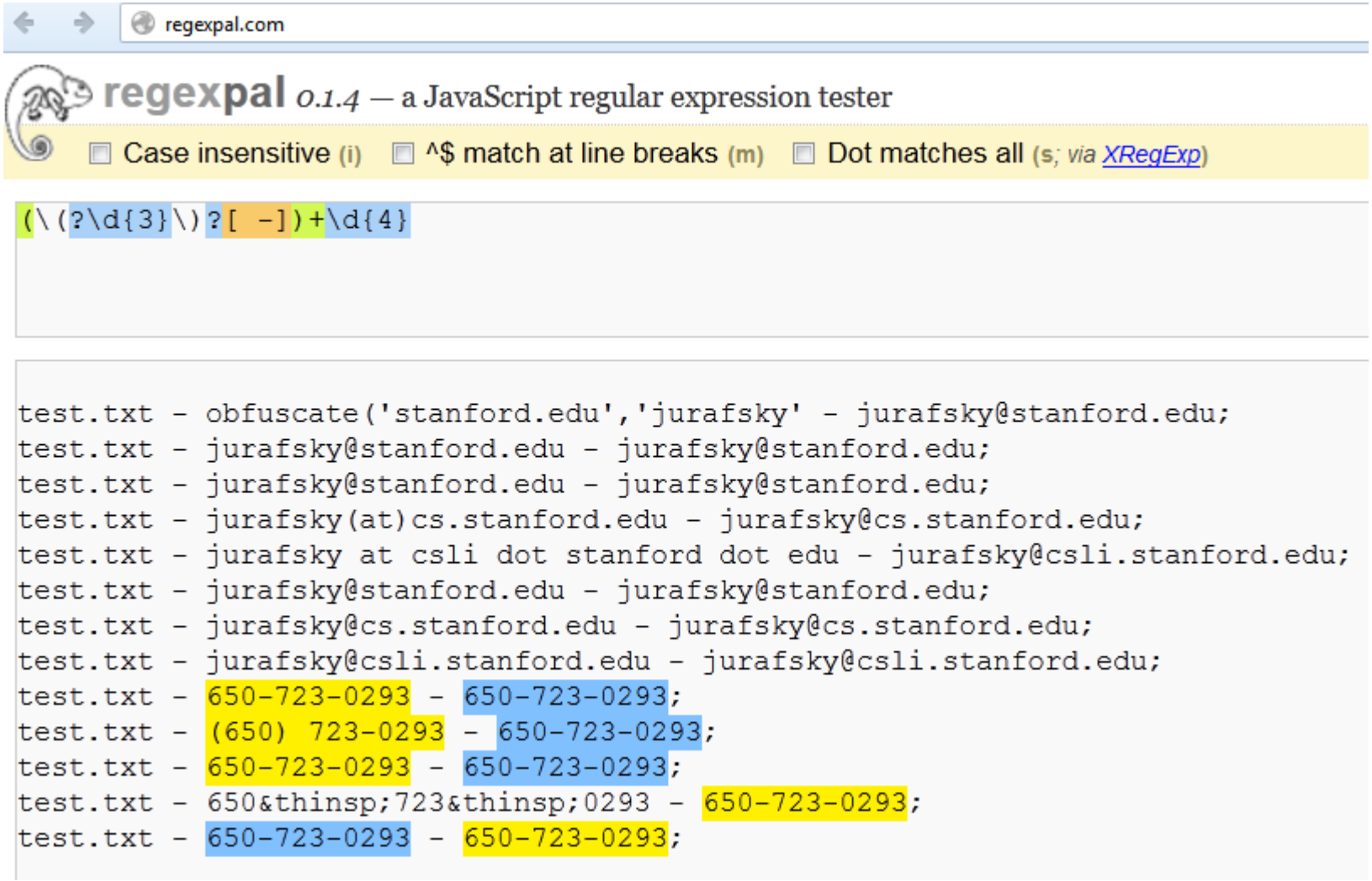
Notasi Regex



Contoh 3: Regex for Email



Contoh 4: Regex for Phone Number



Basic Regular Expression Patterns

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

RE	Match	Example Patterns
`/[wW]oodchuck/`	Woodchuck or woodchuck	"Woodchuck"
`/[abc]/`	'a', 'b', *or* 'c'	"In uomini, in soldati"
`/[1234567890]/`	any digit	"plenty of 7 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 'Drenched Blossoms'"
`/[a-z]/`	a lowercase letter	"my beans were impatient to be hoed!"
`/[0-9]/`	a single digit	"Chapter 1: 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'	"I have no exquisite reason for't"
[^\.]	not a period	"our resident Djinn"
[e^]	either 'e' or '^'	"look up ^ now"
a^b	the pattern 'a^b'	"look up a^ b now"

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

RE	Match	Example Patterns Matched
woodchucks?	woodchuck or woodchucks	"woodchuck"
colou?r	color or colour	"colour"

- The use of period . to specify any character

RE	Match	Example Patterns
/beg.n/	any character between *beg* and *n*	begin, beg'n, begun

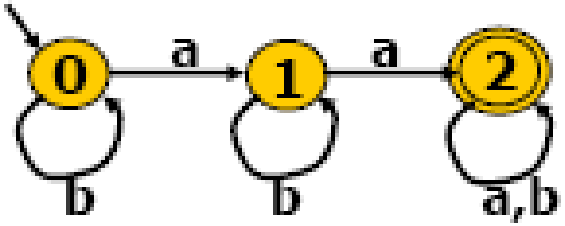
Finite State Machines (FSM)

- 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

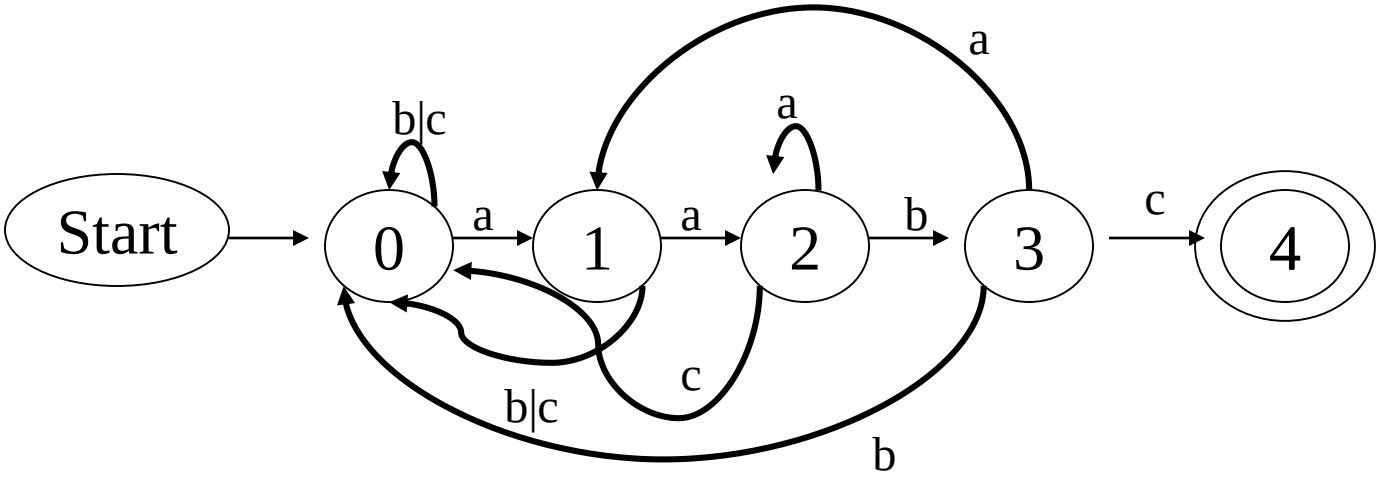
- Input to a FSM
  - Strings built from a fixed alphabet {a,b,c}
  - Possible inputs: aa, aabbcc, a etc..
- The Machine
  - A directed graph
    - Nodes = States of the machine
    - Edges = Transition from one state to another
- Special States
  - Start (q0) and Final (or Accepting) (q2)
- Assume the alphabet is {a,b}
  - Which strings are accepted by this FSM?



Referensi: Gunawardena, 2006

FSM untuk String Matching

- Alphabet {a,b,c}
- Pattern “aabc”
- String: aaaaaaaaaaabcddddddddddddddd



Referensi: Gunawardena, 2006

Regex di Java

regexpal 0.1.4 — a JavaScript regular expression tester

Regex book

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) rancae

#suaraBDG via @dionmudjenan: Hatihati jembatan

RT @quinsymegamira: leuwipanjang banyak anak ja
```

```
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: #
```

Debugger Console × veritransTools (run) ×

```
run:
08.23
BUILD SUCCESSFUL (total time: 0 seconds)
```