14. Regular Expression

14.1 Tujuan Praktikum

Setelah mempelajari Bab ini, mahasiswa diharapkan dapat:

1. Dapat menjelaskan tentang regular expression.
2. Dapat menggunakan simbol dan fungsi regex secara umum.
3. Dapat menggunakan library regex pada Python.
4. Dapat menyelesaikan kasus-kasus regex pada Python.

14.2 Alat dan Bahan

Praktikum ini membutuhkan perangkat komputer yang memiliki spesifikasi minimum sebagai berikut:

1. Terkoneksi ke Internet dan dapat mengunduh package-package Python.
2. Mampu menjalankan sistem operasi Windows 10 atau Ubuntu Linux.

Perangkat lunak yang diperlukan untuk mendukung praktikum ini adalah sebagai berikut:

1. Python 3.7 atau 3.8 yang terinstall menggunakan Anaconda atau Installer Python lainnya.
2. Web Browser (Mozilla Firefox, Microsoft Edge atau Google Chrome).
3. Command Prompt (jika menggunakan Windows).
4. Terminal (jika menggunakan Linux).
5. Editor Python (Visual Studio Code, PyCharm, Spyder atau editor-editor lainnya yang mendukung Python).
6. File mbox-short.txt (di e-class)

14.3 Materi

14.3.1 Pengantar Regex

Pada bab String, kita sudah sedikit memperlajari mengenai teknik-teknik pengaksesan string, manipulasi string, dan berbagai kasus-kasus pengolahan string lainnya, termasuk string yang terdapat pada file. Dari pengalaman tersebut dapat dilihat bahwa kita cukup kesulitan untuk

melakukan pengolahan string dengan teknik biasa / standar. Terdapat teknik pengolahan string yang lebih mudah dan cepat dengan menggunakan bantuan regular expression.

Regular expression adalah ekspresi pola yang berbentuk kumpulan karakter yang digunakan untuk menemukan pola (pattern) yang sama dengan pola regex di dalam string lain yang ingin dicari. Regex membantu kita dalam pencarian string dengan pola tertentu, mengganti string dengan pola tertentu, dan menghapus string dengan pola tertentu. Intinya regex membantu dalam parsing string yang selama ini biasanya hanya menggunakan perintah split() dan find() saja.

Regex sangat powerful dalam searching dan extracting pola namun memiliki pola yang cukup rumit. Tidak semua bahasa pemrograman mendukung regular expression library. Python merupakan salah satu bahasa yang mendukung library regex dengan cara import re. Salah satu fungsi yang paling mudah digunakan dari library re adalah search().

Dengan menggunakn file mbox-short.txt, kita akan mencoba menampilkan semua string pada file tersebut yang **mengandung** pola "From: ".

```
import re
handle=open('mbox-short.txt')
count = 0
for line in handle:
    line=line.rstrip()
    if re.search('From:', line):
        count += 1
        print(line)
    print("Count: ",count)
```

Dari kode di atas kita dapat melihat bahwa re.search bisa saja diganti dengan menggunakan perintah **find()** pada string biasa. Pola pada contoh di atas belum menggunakan kemampuan regex yang seutuhnya.

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: gsilver@umich.edu
From: gsilver@umich.edu
From: zqian@umich.edu
From: gsilver@umich.edu
From: wagnermr@iupui.edu
From: zqian@umich.edu
From: antranig@caret.cam.ac.uk
From: gopal.ramasammycook@gmail.com
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: stephen.marquard@uct.ac.za
```

```
From: louis@media.berkeley.edu
From: louis@media.berkeley.edu
From: ray@media.berkeley.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
Jumlah:  27
```

Jika diinginkan mencari baris yang **diawali** dengan pola "From", maka kita harus mengubah parameter fungsi search pada re.search menjadi **re.search("^From")**.

```
import re
handle=open('mbox-short.txt')
count = 0
for line in handle:
    line=line.rstrip()
    if re.search('^From:', line):
        count += 1
        print(line)
print("Count: ",count)
```

Kode di atas dapat digantikan juga dengan fungsi string.startswith("From :").

14.3.2 Meta Character, Escaped Character, Set of Character, dan Fungsi Regex pada Libary Python

Sebelum menggunakan fungsi regex perlu diketahui terlebih dahulu meta character / special character dan kegunaannya pada pola regex seperti pada tabel 14.1
Pada Python terdapat beberapa special character (espcaped characters) seperti pada tabel 14.2
Pada Python terdapat beberapa penggunaan himpunan character dengan menggunakan simbol [], pada tabel 14.3
Pada Python terdapat 4 buah fungsi yang bisa dipakai untuk menggunakan Regex seperti pada tabel 14.4

14.4 Kegiatan Praktikum

14.4.1 Penggunaan findall

Kita akan mencoba penggunaan fungsi **findall** untuk mencari semua pola sebagai berikut:

```
import re

txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
x = re.findall("mata", txt)
y = re.findall("saya", txt)
for i in x:
    print(i)

if (y):
    print("Ada yang cocok!")
else:
    print("Tidak ada yang cocok!")
```

Tabel 14.1: Special Character pada Python

Karakter	Kegunaan	Contoh	Arti Contoh
[]	Kumpulan karakter	"[a-zA-Z]"	1 karakter antara a-z kecil atau A-Z besar
\{}	Karakter dengan arti khusus dan escaped character	\{}d	Angka / digit
.	Karakter apapun kecuali newline	say.n.	Tidak bisa diganti dengan karakter apapun, misal "sayang" akan valid
^	Diawali dengan	^From	Diawali dengan From
$	Dakhiri dengan	this$	Diakhiri dengan kata this
*	0 s/d tak terhingga karakter	\{}d*	ada digit minimal 0 maksimal tak terhingga
?	ada atau tidak (opsional)	\{}d?	Boleh ada atau tidak ada digit sebanyak
+	1 s/d tak terhingga karakter	\{}d+	Minimal 1 s/d tak terhingga karakter
{}	Tepat sebanyak yang ada para {}	\{}d{2}	Ada tepat 2 digit
()	Pengelompokan karakter / pola	(saya|kamu)	saya atau kamu sebagai satu kesatuan
|	atau	\{}d|\{}s	1 digit atau 1 spasi

Tabel 14.2: Escaped Character pada Regex

Special Characters	Kegunaan	Contoh
\b	Digunakan untuk mengetahui apakah suatu pola berada di awal kata atau akhir kata	"R\bin" "Rain\b"
\d	Digunakan untuk mengetahui apakah karakter adalah sebuah digit (0 s/d 9)	\d
\D	Digunakan untuk mengetahui apakah karakter yang bukan digit	\D
\s	Digunakan untuk mengetahui apakah karakter adalah whitespace (spasi, tab, enter)	\s
\S	Digunakan untuk mengetahui apakah karakter adalah BUKAN whitespace (spasi, tab, enter)	\S
\w	Digunakan untuk mengetahui apakah karakter adalah word (a-z, A-Z, 0-9, dan _)	\w
\W	Digunakan untuk mengetahui apakah karakter adalah BUKAN word (a-z, A-Z, 0-9, dan _)	\W
\A	Digunakan untuk mengetahui apakah karakter adalah berada di bagian depan dari kalimat	"\AThe"
\Z	Digunakan untuk mengetahui apakah karakter adalah berada di bagian akhir dari kalimat	"End\Z"

Hasil:

```
mata #pada mata-mata
mata #pada mata-mata
```

Tabel 14.3: Himpunan Karakter pada Regex

[abc]	Mencari pola 1 huruf a, atau b, atau c
[a-c]	Mencari pola 1 huruf a s/d c
[^bmx]	Mencari pola 1 huruf yang bukan b,m, atau x
[012]	Mencari pola 1 huruf 0, atau 1, atau 2
[0-3]	Mencari pola 1 huruf 0 s/d 3
[0-2][1-3]	Mencari pola 2 huruf: 01, 02, 03, 11, 12, 13, 21, 22, 23
[a-zA-Z]	Mencari pola 1 huruf a-Z

Tabel 14.4: Fungsi Regex pada Python

Nama Fungsi	**Kegunaan**
findall	mengembalikan semua string yang sesuai pola (matches)
search	mengembalikan string yang sesuai pola (match)
split	memecah string sesuai pola
sub	mengganti string sesuai dengan pola yang cocok

```
mata #pada memata-matai
mata #pada memata-matai
mata #kaca mata
Tidak ada yang cocok #karena tidak ada 'saya'
```

Perhatikan bagian Matahari tidak muncul karena Matahari menggunakan huruf besar

Contoh lain fungsi **findall**:

```
import re
handle=open('mbox-short.txt')
for line in handle:
    line=line.rstrip()
    x=re.findall('\S+@\S+', line)
    if len(x)>0:
        print(x)
```

Hasil:

```
['stephen.marquard@uct.ac.za']
['<postmaster@collab.sakaiproject.org>']
['<200801051412.m05ECIaH010327@nakamura.uits.iupui.edu>']
['<source@collab.sakaiproject.org>;']
['<source@collab.sakaiproject.org>;']
['<source@collab.sakaiproject.org>;']
['apache@localhost)']
['source@collab.sakaiproject.org;']
['stephen.marquard@uct.ac.za']
['source@collab.sakaiproject.org']
....dst
```

Ada beberapa format email yang tidak sesuai format, seperti mengandung karakter <, sehingga kita perlu mengganti format regex nya menjadi: [a-zA-Z0-9]§*@§*[a-zA-Z. Silahkan ubah dibagian baris ke-5.

14.4.2 Penggunaan search

Kita akan mencoba penggunaan fungsi **search** untuk mencari pola sebagai berikut:

```
import re

txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
x = re.search("\s", txt)
y = re.search("saya", txt)

print("Spasi ditemukan di:", x.start())
print(y)
```

Hasil:

```
4
None
```

Contoh lain fungsi **search**: Pada mbox kita ingin menemukan kata-kata:

```
X-DSPAM-Confidence: 0.847
5X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.6178
X-DSPAM-Probability: 0.0000
```

Untuk melakukannya dapat digunakan regex: **X̂-.*: [0-9.]+**

```
import re
handle=open('mbox-short.txt')
for line in handle:
    line=line.rstrip()
    if(re.search('^X-.*: [0-9.]+', line)):
        print(line)
```

Hasil:

```
X-DSPAM-Confidence: 0.8475
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.6178
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.6961
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.7565
X-DSPAM-Probability: 0.0000
dst...
```

14.4.3 Penggunaan split

Kita akan mencoba penggunaan fungsi **split** untuk memecah string sebagai berikut:

```
import re

txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
y = re.split("\s", txt, 1)  #split 1 kata pertama
print(x)
```

Hasil:

```
['The','rain','in','Spain']
['The','rain in Spain']
```

14.4.4 Penggunaan sub

Kita akan mencoba penggunaan fungsi **sub** untuk replace pola sebagai berikut:

```
import re

txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
x = re.sub("\s", "-", txt)  #mengganti spasi dengan -
print(x)
y = re.sub("\s", "*", txt, 2)   #mengganti spasi dengan * 2 saja
print(y)
```

Hasil:

```
Sang-mata-mata-sedang-memata-matai-kasus-kaca-mata-di toko-Matahari
Sang*mata-mata*sedang memata-matai kasus kaca mata di toko Matahari
```

14.5 Latihan Mandiri

Latihan 14.1 Anda diminta untuk mencari seluruh teks yang berupa tanggal dengan format YYYY-MM-DD dan kemudian seluruh tanggal tersebut diambil dan ditampilkan kembali dalam format DD-MM-YYYY ditambah dengan perhitungan selisih dengan tanggal sekarang dalam hari.

Contoh:

Pada tanggal 1945-08-17 Indonesia merdeka. Indonesia memiliki beberapa pahlawan nasional, seperti Pangeran Diponegoro (TL: 1785-11-11), Pattimura (TL: 1783-06-08) dan Ki Hajar Dewantara (1889-05-02).

Hasil:

1945-08-17 00:00:00 selisih 27209 hari
1785-11-11 00:00:00 selisih 85561 hari
1783-06-08 00:00:00 selisih 86448 hari
1889-05-02 00:00:00 selisih 47769 hari ■

Latihan 14.2 Anda diminta untuk mencari seluruh teks yang berupa email dan kemudian ambil semua username dari email tersebut untuk digenerate password random 8 karakter yang terdiri dari angka dan huruf.

Contoh:

Berikut adalah daftar email dan nama pengguna dari mailing list:
anton@mail.com dimiliki oleh antonius
budi@gmail.co.id dimiliki oleh budi anwari
slamet@getnada.com dimiliki oleh slamet slumut
matahari@tokopedia.com dimiliki oleh toko matahari

Hasil:

anton@mail.com username: anton , password: 8u78A2UD
budi@gmail.co.id username: budi , password: bdP066Ld
slamet@getnada.com username: slamet , password: Ab1FiHXb
matahari@tokopedia.com username: matahari , password: 5KYyaP6 ■