



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230981
Nama Lengkap	Deo Dewanto
Minggu ke / Materi	15 / Regular Expression

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Pengantar ReGex

Regular expression adalah pola ekspresi yang terdiri dari kumpulan karakter yang digunakan untuk menemukan pola yang sesuai dalam string lain. Regex membantu kita dalam pencarian, penggantian, dan penghapusan string berdasarkan pola tertentu. Dengan kata lain, regex mempermudah parsing string yang biasanya dilakukan dengan perintah `split()` dan `find()`.

Regex sangat membantu dalam mencari dan mengekstrak pola, meskipun polanya cukup rumit. Tidak semua bahasa pemrograman mendukung library regular expression, namun Python adalah salah satu yang mendukungnya melalui `import re`. Salah satu fungsi yang paling mudah digunakan dari library `re` adalah `search()`.

Dengan menggunakan file `mbox-short.txt`, kita akan mencoba menampilkan semua string dalam file tersebut yang mengandung pola "From: ".

```
1  import re
2  handle = open('mbox-short.txt')
3  count = 0
4  for line in handle:
5      line = line.rstrip()
6      if re.search('From:', line):
7          count += 1
8          print(line)
9  print("Count: ",count)
```

```

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
Count: 27

```

Jika diinginkan mencari baris yang diawali dengan pola “From” ,makakita harus mengubah parameter Fungsi search pada re.search menjadi re.search(“From”)

Meta Character, Escaped Character, Set of Character, dan Fungsi Regex pada Library Python

Python

Sebelum menggunakan fungsi regex perlu diketahui terlebih dahulu meta character / special character dan kegunaannya pada pola regex seperti pada tabel 14.1

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?	Bolch 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	(sayalkamu)	saya atau kamu sebagai satu kesatuan
	atau	\()d\)s	1 digit atau 1 spasi

Pada Python terdapat beberapa special character (escaped characters) seperti pada tabel 14.2

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" "Ra-in\\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"

Pada Python terdapat beberapa penggunaan himpunan character dengan menggunakan simbol [], pada tabel 14.3

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

Pada Python terdapat 4 buah fungsi yang bisa dipakai untuk menggunakan Regex seperti pada table 14.4

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

Kegiatan Praktikum

PENGUNAAN FINDALL

SOURCE CODE AND OUTPUT

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

```
1 import re
2
3 txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
4 x = re.findall("mata", txt)
5 y = re.findall("saya", txt)
6 for i in x:
7     print(i)
8
9 if y:
10     print("Ada yang cocok!")
11 else:
12     print("Tidak ada yang cocok!")
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```
macbook@MacBook-Pro-2 Regular Expression % /usr/local/bin/python3 "/Users/macbook/Regul
ar Expression/kegiatan1.py"
mata
mata
mata
mata
Tidak ada yang cocok!
```

PENGUNAAN SEARCH

SOURCE CODE AND OUTPUT

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)

if x:
    print("Spasi ditemukan di:", x.start())
else:
    print("Spasi tidak ditemukan.")

if y:
    print("Kata 'saya' ditemukan.")
else:
    print("Kata 'saya' tidak ditemukan.")

import re
handle = open('mbox-short.txt')

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

```

/usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan2.py"
macbook@MacBook-Pro-2 Regular Expression % /usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan2.py"
/Users/macbook/Regular Expression/kegiatan2.py:4: SyntaxWarning: invalid escape sequence '\s'
  x = re.search("\s", txt)
Spasi ditemukan di: 4
Kata 'saya' tidak ditemukan.
X-DSPAM-Confidence: 0.8472
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6178
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6961
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7552
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7626
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7556
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7802
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7615
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7681
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7685
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6959
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7686
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7559
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7685
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6932
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.7558
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6526
X-DSPAM-Probability: 0.8000
X-DSPAM-Confidence: 0.6948

```

PENGUNAAN SPLIT

SOURCE CODE AND OUTPUT

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

```

1 import re
2
3 txt = "The rain in Spain"
4 x = re.split("\s", txt)
5 print(x)
6 y = re.split("\s", txt, 1)
7 print(y)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```

/usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan3.py"
macbook@MacBook-Pro-2 Regular Expression % /usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan3.py"
/Users/macbook/Regular Expression/kegiatan3.py:4: SyntaxWarning: invalid escape sequence '\s'
  x = re.split("\s", txt)
/Users/macbook/Regular Expression/kegiatan3.py:6: SyntaxWarning: invalid escape sequence '\s'
  y = re.split("\s", txt, 1)
['The', 'rain', 'in', 'Spain']
['The', 'rain in Spain']
macbook@MacBook-Pro-2 Regular Expression %

```

PENGUNAAN SUB

SOURCE CODE AND OUTPUT

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

```

1 import re
2
3 txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
4 x = re.sub("\s", "-", txt)
5 print(x)
6 y = re.sub("\s", "*", txt, 2)
7 print(y)

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

```

/usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan4.py"
macbook@MacBook-Pro-2 Regular Expression % /usr/local/bin/python3 "/Users/macbook/Regular Expression/kegiatan4.py"
/Users/macbook/Regular Expression/kegiatan4.py:4: SyntaxWarning: invalid escape sequence '\s'
  x = re.sub("\s", "-", txt)
/Users/macbook/Regular Expression/kegiatan4.py:6: SyntaxWarning: invalid escape sequence '\s'
  y = re.sub("\s", "*", txt, 2)
Sang-mata-mata-sedang-memata-matai-kasus-kaca-mata-di-toko-Matahari
Sang-mata-mata-sedang memata-matai kasus kaca mata di toko Matahari
macbook@MacBook-Pro-2 Regular Expression %

```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

SOURCE CODE AND OUTPUT

```
#Latihan 1
import re
from datetime import datetime

txt = """
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).
"""

dates = re.findall(r'\b\d{4}-\d{2}-\d{2}\b', txt)
now = datetime.now()

for date_str in dates:
    date_obj = datetime.strptime(date_str, '%Y-%m-%d')
    delta = now - date_obj
    formatted_date = date_obj.strftime('%d-%m-%Y')
    print(f"{date_obj} selisih {delta.days} hari")
```

✓ 0.0s

```
1945-08-17 00:00:00 selisih 28782 hari
1785-11-11 00:00:00 selisih 87134 hari
1783-06-08 00:00:00 selisih 88021 hari
1889-05-02 00:00:00 selisih 49342 hari
```

PENJELASAN

Kode tersebut dimulai dengan mengimpor library 're' untuk regex dan 'datetime' untuk manipulasi tanggal. Teks yang akan dianalisis didefinisikan dalam variabel 'txt', yang berisi beberapa tanggal dalam format 'YYYY-MM-DD'. Kode kemudian menggunakan fungsi 're.findall()' untuk mencari semua tanggal yang sesuai dengan format tersebut dan menyimpannya dalam daftar dates. Tanggal saat ini diperoleh menggunakan 'datetime.now()', Selanjutnya, kode mengiterasi setiap tanggal yang ditemukan dalam daftar dates, mengubah string tanggal menjadi objek datetime menggunakan 'datetime.strptime()', dan menghitung selisih hari antara tanggal tersebut dan tanggal saat ini. Tanggal dalam format "YYYY-MM-DD" diubah menjadi format 'DD-MM-YYYY' menggunakan metode strftime(). Akhirnya, kode mencetak tanggal asli dalam format objek 'datetime' dan selisih hari antara tanggal tersebut dan tanggal saat ini. Hasilnya adalah daftar tanggal dengan format baru serta jumlah hari yang telah berlalu sejak tanggal tersebut hingga saat ini.

SOAL 2

SOURCE CODE AND OUTPUT

```

#Latihan 2
import re
import random
import string

txt = """
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
"""

emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b', txt)

def generate_password(length=8):
    characters = string.ascii_letters + string.digits
    return ''.join(random.choice(characters) for _ in range(length))

for email in emails:
    username = email.split('@')[0]
    password = generate_password()
    print(f"{email} username: {username}, password: {password}")

```

✓ 0.0s

```

anton@mail.com username: anton, password: FAzR3vqK
budi@gmail.co.id username: budi, password: wI4MEE2q
slamet@getnada.com username: slamet, password: qCia4npu
matahari@tokopedia.com username: matahari, password: y9GwYYNr

```

PENJELASAN

Kode ini mengimpor library re untuk regex, random untuk pengacakan, dan string untuk

mengakses huruf dan angka. Teks yang akan dianalisis didefinisikan dalam variabel txt, yang berisi daftar email dan nama pengguna. Kode menggunakan fungsi re.findall() untuk mencari semua email dalam teks dan menyimpannya dalam daftar emails. Fungsi generate_password() didefinisikan untuk menghasilkan password acak sepanjang 8 karakter, terdiri dari gabungan huruf besar, huruf kecil, dan angka. Untuk setiap email yang ditemukan, username diekstrak sebagai bagian dari email sebelum simbol @. Password acak dihasilkan dengan memanggil fungsi generate_password(). Hasilnya ditampilkan dengan format yang menampilkan email, username, dan password acak yang dihasilkan. Kode ini mendemonstrasikan penggunaan regex untuk pencarian pola dalam teks, manipulasi string untuk ekstraksi data, dan pengacakan untuk pembuatan password.

LINK GITHUB : <https://github.com/Deodewanto07/LAPRAK-15-71230981-.git>