



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230989>
Nama Lengkap	<Yohanes Nevan Adventus Wibawa>
Minggu ke / Materi	14 / ReGex

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.

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: ".

```
import re

# Membuka file mbox-short.txt
handle = open('mbox-short.txt')

count = 0 # Inisialisasi penghitung

# Iterasi setiap baris dalam file
for line in handle:
    line = line.rstrip() # Menghapus spasi atau newline di akhir baris
    if re.search('From:', line): # Mencari pola "From:" dalam baris
        count += 1 # Menambah penghitung jika ditemukan
        print(line) # Mencetak baris yang cocok

# Mencetak jumlah total baris yang cocok
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
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14>
```

Output =

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 # Mengimpor library regex

# Membuka file mbox-short.txt
handle = open('mbox-short.txt')

count = 0 # Inisialisasi penghitung

# Iterasi setiap baris dalam file
for line in handle:
    line = line.rstrip() # Menghapus spasi atau newline di akhir baris
    if re.search('^From:', line): # Mencari pola "From:" di awal baris
        count += 1 # Menambah penghitung jika ditemukan
        print(line) # Mencetak baris yang cocok

# Mencetak jumlah total baris yang cocok
print("Count: ", count)

```

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

Sebelum menggunakan fungsi regex, penting untuk memahami terlebih dahulu meta character atau special character dan kegunaannya dalam pola regex, seperti yang dijelaskan pada tabel 14.1.

Pada Python, terdapat beberapa special character (escaped characters) yang perlu diketahui, seperti yang dijelaskan pada tabel 14.2. Selain itu, dalam Python, kita juga dapat menggunakan himpunan karakter dengan simbol [] untuk mencocokkan berbagai karakter dalam pola regex.

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 14.2: 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"

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	(sayalkamu)	saya atau kamu sebagai satu kesatuan
	atau	\\{ }d \\{ }s	1 digit atau 1 spasi

Kegiatan Praktikum

Penggunaan Fidali

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

```
import re # Mengimpor library regex

# Mendefinisikan string yang akan dicari
txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"

# Mencari semua kemunculan kata "mata" dalam string
x = re.findall("mata", txt)

# Mencari semua kemunculan kata "saya" dalam string
y = re.findall("saya", txt)

# Mencetak semua kemunculan kata "mata"
for i in x:
    print(i)

# Mengecek apakah ada kemunculan kata "saya" dan mencetak pesan sesuai
if y:
    print("Ada yang cocok!")
else:
    print("Tidak ada yang cocok!")
```

```
mata
mata
mata
mata
mata
Tidak ada yang cocok!
```

Output =

Kode yang Anda berikan mencari semua kemunculan kata "mata" dan "saya" dalam string txt menggunakan fungsi re.findall()

Kode tersebut menggunakan library `re` untuk mencari pola dalam sebuah string. String `txt` didefinisikan dengan teks "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari". Fungsi `re.findall()` digunakan untuk mencari semua kemunculan kata "mata" dalam string dan hasilnya disimpan dalam variabel `x`. Dengan cara yang sama, pencarian kata "saya" dalam string dilakukan dan hasilnya disimpan dalam variabel `y`. Kemudian, kode mencetak setiap kemunculan kata "mata" yang ditemukan dalam string dengan mengiterasi daftar `x`.

Selanjutnya, kode memeriksa apakah ada kemunculan kata "saya" dengan mengecek apakah `y` berisi elemen. Jika `y` tidak kosong, mencetak "Ada yang cocok!"; jika tidak, mencetak "Tidak ada yang cocok!". Kode ini mendemonstrasikan penggunaan fungsi `re.findall()` untuk mencari dan mencetak pola dalam string serta memeriksa hasil pencarian tersebut.

Penggunaan search

```
import re # Mengimpor library regex

# Mendefinisikan string yang akan dicari
txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"

# Mencari keberadaan spasi pertama dalam string
x = re.search("\s", txt)

# Mencari keberadaan kata "saya" dalam string
y = re.search("saya", txt)

# Mengecek apakah spasi ditemukan dan mencetak posisinya
if x:
    print("Spasi ditemukan di:", x.start())
else:
    print("Spasi tidak ditemukan.")

# Mengecek apakah kata "saya" ditemukan dan mencetak hasilnya
if y:
    print("Kata 'saya' ditemukan.")
else:
    print("Kata 'saya' tidak ditemukan.")
```

```
ce '\s'
x = re.search("\s", txt)
Spasi ditemukan di: 4
Kata 'saya' tidak ditemukan.
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14>
```

Output =

Kode tersebut mengimpor library regex (re) dan mendefinisikan string `txt` yang berisi teks yang akan diperiksa. Dengan menggunakan fungsi `re.search()`, kode mencari keberadaan spasi pertama dalam string menggunakan pola `\s`. Jika spasi ditemukan, posisi spasi pertama dicetak dengan menggunakan metode `x.start()`, yang memberikan indeks awal dari spasi yang ditemukan. Jika tidak ditemukan spasi, pesan "Spasi tidak ditemukan." akan dicetak.

Selanjutnya, kode mencari kata "saya" dalam string `txt`. Karena kata "saya" tidak ada dalam string, hasil pencarian adalah `None`. Kode kemudian memeriksa hasil pencarian; jika kata "saya" ditemukan, akan mencetak "Kata 'saya' ditemukan.", namun jika tidak ditemukan, akan mencetak "Kata 'saya' tidak ditemukan.". Kode ini berguna untuk mendemonstrasikan penggunaan fungsi `re.search()` dalam menemukan dan mencocokkan pola dalam string.

```

import re # Mengimpor library regex

# Membuka file mbox-short.txt
handle = open('mbox-short.txt')

# Iterasi setiap baris dalam file
for line in handle:
    line = line.rstrip() # Menghapus spasi atau newline di akhir baris
    # Mencari baris yang dimulai dengan "X-", diikuti oleh karakter apa saja (.),
    kemudian ": ", dan diikuti oleh angka atau titik
    if re.search('^X-.*: [0-9.]+' , line):
        print(line) # Mencetak baris yang cocok

```

```

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
X-DSPAM-Confidence: 0.7626
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.7556
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.7002
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.7615
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.7601

```

Output =

Kode ini menggunakan library re untuk mengolah regex dan membaca file mbox-short.txt baris per baris. Untuk setiap baris, kode menghapus karakter spasi atau newline di akhir baris menggunakan `rstrip()`. Kemudian, menggunakan fungsi `re.search()`, kode mencari baris yang sesuai dengan pola regex `'^X-.*: [0-9.]+'`. Pola ini mencocokkan baris yang dimulai dengan "X-", diikuti oleh karakter apapun (`.*`), kemudian ": ", dan diakhiri dengan satu atau lebih angka atau titik (`[0-9.]`). Jika sebuah baris cocok dengan pola ini, baris tersebut dicetak. Pola ini efektif untuk menemukan header dalam email yang dimulai dengan "X-" dan diikuti oleh angka, seperti "X-DSPAM-Confidence: 0.8475". Kode ini mendemonstrasikan penggunaan regex untuk mencari pola spesifik dalam teks yang dibaca dari sebuah file.

Penggunaan split

```
import re # Mengimpor library regex

# Mendefinisikan string yang akan dibagi
txt = "The rain in Spain"

# Membagi string berdasarkan spasi
x = re.split("\s", txt)
print(x) # Mencetak hasil pembagian

# Membagi string berdasarkan spasi, tetapi hanya pada pembagian pertama
y = re.split("\s", txt, 1)
print(y) # Mencetak hasil pembagian
```

```
ce '\s'
x = re.split("\s", txt)
c:\Users\LENOVO\Tugas PrakAlPro\Tugas
nce '\s'
y = re.split("\s", txt, 1)
['The', 'rain', 'in', 'Spain']
['The', 'rain in Spain']
```

Output =

Kode tersebut menggunakan fungsi `re.split()` untuk membagi string `txt` berdasarkan pola yang ditentukan. Berikut adalah penjelasan rinci tentang cara kerja kode ini: ``python import re # Mengimpor library regex # Mendefinisikan string yang akan dibagi txt = "The rain in Spain" # Membagi string berdasarkan spasi x = re.split("\s", txt) print(x) # Mencetak hasil pembagian # Membagi string berdasarkan spasi, tetapi hanya pada pembagian pertama y = re.split("\s", txt, 1) print(y) # Mencetak hasil pembagian `` Penjelasan dalam paragraf:

Kode ini menggunakan library `re` untuk mengolah regex dan mendefinisikan string `txt` dengan nilai "The rain in Spain". Fungsi `re.split("\s", txt)` digunakan untuk membagi string `txt` berdasarkan karakter spasi (`\s`). Hasil pembagian ini disimpan dalam variabel `x` dan dicetak, menghasilkan daftar `['The', 'rain', 'in', 'Spain']`, di mana string dibagi menjadi empat elemen pada setiap spasi. Selanjutnya, kode menggunakan `re.split("\s", txt, 1)`, yang membagi string `txt` berdasarkan spasi tetapi hanya pada pembagian pertama. Hasil ini disimpan dalam variabel `y` dan dicetak, menghasilkan daftar `['The', 'rain in Spain']`, di mana string hanya dibagi menjadi dua elemen pada spasi pertama. Kode ini mendemonstrasikan penggunaan fungsi `re.split()` untuk membagi string berdasarkan pola regex dengan opsi pembatasan jumlah pembagian.

Penggunaan Sub

```
import re # Mengimpor library regex

# Mendefinisikan string yang akan diubah
txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"

# Mengganti semua spasi dalam string dengan tanda "-"
```

```

x = re.sub("\s", "-", txt)
print(x) # Mencetak hasil penggantian

# Mengganti dua spasi pertama dalam string dengan tanda "*"
y = re.sub("\s", "*", txt, 2)
print(y) # Mencetak hasil penggantian

C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14\kegiatan Praktikum 4.py:11
nce '\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
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14>

```

Output =

Kode tersebut menggunakan fungsi `re.sub()` dari library `re` untuk mengganti pola tertentu dalam string dengan karakter lain. Berikut adalah penjelasan rinci tentang cara kerja kode ini: ``python import re # Mengimpor library regex # Mendefinisikan string yang akan diubah txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari" # Mengganti semua spasi dalam string dengan tanda "-" x = re.sub("\s", "-", txt) print(x) # Mencetak hasil penggantian # Mengganti dua spasi pertama dalam string dengan tanda "*" y = re.sub("\s", "*", txt, 2) print(y) # Mencetak hasil penggantian ``

Penjelasan dalam paragraf: Kode ini menggunakan library `re` untuk mengolah regex dan mendefinisikan string `txt` dengan nilai "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari". Fungsi `re.sub("\s", "-", txt)` digunakan untuk mengganti semua karakter spasi (`\s`) dalam string `txt` dengan tanda "-" dan hasilnya disimpan dalam variabel `x`. Ketika `x` dicetak, hasilnya adalah "Sang-mata-mata-sedang-memata-matai-kasus-kaca-mata-di-toko-Matahari", di mana semua spasi dalam string asli telah diganti dengan tanda "-".

Selanjutnya, fungsi `re.sub("\s", "*", txt, 2)` digunakan untuk mengganti hanya dua spasi pertama dalam string `txt` dengan tanda "*". Hasil penggantian ini disimpan dalam variabel `y` dan ketika `y` dicetak, hasilnya adalah "Sang*mata-mata*sedang memata-matai kasus kaca mata di toko Matahari", di mana hanya dua spasi pertama dalam string asli yang diganti dengan tanda "*". Kode ini mendemonstrasikan penggunaan fungsi `re.sub()` untuk mengganti pola dalam string dengan karakter lain dan membatasi jumlah penggantian yang dilakukan.

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

```
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")

1945-08-17 00:00:00 selisih 28781 hari
1785-11-11 00:00:00 selisih 87133 hari
1783-06-08 00:00:00 selisih 88020 hari
1889-05-02 00:00:00 selisih 49341 hari
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14>
```

Output =

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

```
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}")

anton@mail.com username: anton, password: TZTe7OU1
budi@gmail.co.id username: budi, password: DoEw8foV
slamet@getnada.com username: slamet, password: 4DW9mUAE
matahari@tokopedia.com username: matahari, password: Do663XKy
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 14> █
```

Output =

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

Link = <https://github.com/YohanesNevan/Tugas-Laporan-Alpro-14.git>