



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
Minggu ke / Materi	14 / 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

Link Github: <https://github.com/Tohrubo/tugaspralpro14.git>

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Regex

Regex atau Regular Expression, adalah sebuah fungsi python dan Java yang dapat diimport untuk membuka berbagai fungsi lain. Kerja utama regex adalah untuk menemukan sebuah pola yang ada dalam sebuah string atau text panjang. Contoh singkat seperti di bawah ini:

```
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)
```

Gambar contoh penggunaan regex

Di import re untuk fungsi regex, kemudian dengan re, menulis sintaks re.search(a, b) dengan a sebagai pola yang ingin di cari dan b sebagai text yang ingin dianalisa polanya. Output dari kode di atas akan seperti berikut

```
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
```

Kode di atas akan mencari semua baris yang memiliki 'From: ' dan mengeprint baris itu.

MATERI 2: Regex Formats

Regex memiliki beberapa format / methods yang dapat digunakan untuk menemukan pola tertentu. format-format tersebut dapat saja dibuat seperti di bawah ini:

Tabel 1.11: Special Character pada Regex

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

Format-format ini penting digunakan untuk menemukan pola tertentu dengan tepat. Ada juga format umum untuk kategori tertentu seperti angka, huruf, alphanumeric, etc. Format itu sebagai berikut:

Special Characters	Kegunaan	Contoh
\b	Digunakan untuk mengetahui apakah suatu pola berada di awal kata atau akhir kata	"R\b" "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"

Ada juga format manual seperti ascii format atau letter format seperti di bawah ini:

[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

Dan yang terakhir ada juga beberapa macam fungsi-fungsi di regex, seperti:

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

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Welcome | Question1.py U x | Question2.py U | user.txt U
Question1.py > ...
1 import re
2 import datetime as dati
3
4 sentence = "Pada tanggal 1945-08-17 Indonesia merdeka. Indonesia r
5
6 dt = re.findall("\d{4}-\d{2}-\d{2}", sentence)
7
8 dtnow = dati.datetime.now()
9
10 for dates in dt:
11     ymd = dates.split("-")
12     dtthen = dati.datetime(int(ymd[0]), int(ymd[1]), int(ymd[2]))
13     print(f"{dtthen} selisih {(dtnow-dtthen).days} hari")
```

B. Output Result

```
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
```

C. Explanation

Kode ini menggunakan regex untuk menemukan pola tanggal dan datetime untuk mengubah tanggal menjadi variabel date supaya bisa dihitung selisih hari antara date tertentu. `re.findall()` akan menemukan pola tanggal, `datetime` akan memungkinkan untuk menggunakan variabel date untuk menghitung selisih hari.

SOAL 2

A. Source Code

```
Question2.py > ...
1 import re
2 import random
3 import string
4
5 handle = open("user.txt", "r")
6
7 text = handle.read()
8 users = re.findall("(\\w.+)", text)
9
10 emailst = []
11 para = text.split()
12 for word in para:
13     if "@" in word:
14         emailst.append(word)
15
16 passwords = []
17 for user in emailst:
18     psw = "".join(random.choice(string.ascii_letters + string.digits) for i in range(8))
19     passwords.append(psw)
20
21 for i in range(len(users)):
22     print(f"{emailst[i]} username: {users[i]}, password: {passwords[i]}")
23
24 handle.close()
```

```
user.txt
1 Berikut adalah daftar email dan nama pengguna dari mailing list:
2 anton@mail.com dimiliki oleh antonius
3 budi@gmail.co.id dimiliki oleh budi anwari
4 slamet@getnada.com dimiliki oleh slamet slumut
5 matahari@tokopedia.com dimiliki oleh toko matahari
```

B. Output Result

```
anton@mail.com username: anton, password: LXvMwCZq
budi@gmail.co.id username: budi, password: oHGk0kw8
slamet@getnada.com username: slamet, password: M6T7o3P6
matahari@tokopedia.com username: matahari, password: Tbw9oyeZ

anton@mail.com username: anton, password: jbynCFbA
budi@gmail.co.id username: budi, password: j6UbJMg2
slamet@getnada.com username: slamet, password: 7kNa5RRw
matahari@tokopedia.com username: matahari, password: b7FB92Lf
```

C. Explanation

Kode menggunakan `open()` untuk membuka `file.txt` yang berisi email user yang kemudian menggunakan regex untuk menemukan username. Kemudian untuk generate password alphanumeric random menggunakan `import random` dan `string` supaya bisa menggunakan `random.choice()` untuk memilih salah satu antara `string.ascii_letters` atau `string.digits` yang menggunakan `".join()` untuk menggabungkan tiap huruf/digit yang digenerate sebagai sebuah password berukuran 8.