



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230987
Nama Lengkap	Prastha Pradipta Purusa
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.

Pengantar Regex

Dalam pembelajaran mengenai string, kita menemui berbagai kesulitan dalam pengolahan string dengan teknik standar. Namun, teknik regular expression menjadi solusi yang lebih mudah dan cepat. Regular expression, atau regex, merupakan kumpulan karakter yang digunakan untuk mencari pola yang sama dengan pola regex di dalam string lain. Dengan regex, kita dapat melakukan pencarian, penggantian, dan penghapusan string berdasarkan pola tertentu. Python, salah satu bahasa pemrograman yang mendukung regex melalui library re, memungkinkan penggunaan fungsi seperti `search()`. Meskipun regex sangat powerful dalam searching dan extracting pola, namun polanya cukup rumit dan tidak semua bahasa pemrograman mendukung library regex. Contoh program menggunakan `search`:

```
121 import re
122 file=open('textcoba.txt', 'r')
123 jumlah = 0
124 for line in file.readlines():
125     line=line.rstrip()
126     if re.search('Mencari', line):
127         jumlah += 1
128         print(line)
129 print("Count: ",jumlah)
```

```
Mencari:
Mencari is a
Mencari-compatible
Mencari-debugging
Mencari-error handling
adiain aja Mencari
Count: 6
```

Jika menggunakan regex dapat menggunakan symbol `^` untuk mencari awalan yang menggunakan 'Mencari' menjadi '^Mencari'. Contoh program:

```

1  import re
2  file=open('textcoba.txt', 'r')
3  jumlah = 0
4  for line in file.readlines():
5      line=line.rstrip()
6      if re.search('^Mencari', line):
7          jumlah += 1
8          print(line)
9  print("Count: ",jumlah)

```

```

Mencari:
Mencari is a
Mencari-compatible
Mencari-debugging
Mencari-error handling
Count: 5

```

Dari perbandingan kedua program tersebut dapat dilihat hasil outputnya dimana kode pertama memunculkan semua yang ada kata 'Mencari' sedangkan kode kedua memunculkan yang ada kata 'Mencari' saja.

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

Macam-macam special character pada Python:

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

Escaped Character pada regex:

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"

Himpunan karakter pada 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

Fungsi Regex pada Python:

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

Contoh penggunaan findall() :

```

131 import re
132 file = open('textcoba.txt', 'r')
133 baca = file.readline()
134 for line in file:
135     line = line.rstrip()
136     x = re.findall(r'\bd\w*|\w*d\b', line)
137     print(x)

```

```

[]
[]
['debugging']
[]
[]
[]
['dmfspfkdd']

```

Program tersebut mencari semua yang berawalan d ataupun yang berakhiran d dengan menggunakan r sebelum string memastikan bahwa backslash dianggap sebagai karakter literal dalam ekspresi reguler, bukan karakter escape membuat pola ekspresi reguler lebih jelas dan mengurangi kemungkinan kesalahan karena tidak perlu memperhatikan karakter escape. \b digunakan untuk mengetahui apakah karakter berada di awal atau di akhir. d\w* mencocokkan kata yang dimulai dengan 'd'. fungsi | atau OR digunakan untuk atau. \w*d\b mencocokkan kata yang diakhiri dengan 'd'.

Contoh penggunaan search():

```

139 import re
140 text = "Ke Padang beli nasi Padang 13 ribu"
141 x = re.search("\d", text)
142 print(x.start())

```

```

[Running] pyth
27

```

Program tersebut mengecek dimana digit pertama kali ada. Digit pertama kali ada di index ke 27 yaitu angka 1 dan output yang muncul adalah 27.

Contoh program menggunakan split():

```

144 import re
145 text = "Ke Padang beli nasi Padang 13 ribu"
146 x1 = re.split("\s", text)
147 x2 = re.split("\s", text, 2)
148 print(x1)
149 print(x2)

```

```
['Ke', 'Padang', 'beli', 'nasi', 'Padang', '13', 'ribu']  
['Ke', 'Padang', 'beli nasi Padang 13 ribu']
```

Program tersebut memisahkan setiap kata. X1 digunakan untuk memisahkan setiap kata dan setiap spasi. X2 sama seperti X1 hanya saja mulai index ke 2 sampai terakhir digabung sedangkan 0 dan 1 terpisah.

Contoh penggunaan sub():

```
151 import re  
152 text = 'beli 1 tambah 3 gratis 2 makan ayam'  
153 x = re.sub('\s',text, text)  
154 print(x)
```

```
[Running] python -u "c:\Users\HP\prasss020904\bab15\sc.py"  
belibeli 1 tambah 3 gratis 2 makan ayam1beli 1 tambah 3 gratis 2 makan ayamtambahbeli 1 tambah 3  
gratis 2 makan ayam3beli 1 tambah 3 gratis 2 makan ayamgratisbeli 1 tambah 3 gratis 2 makan  
ayam2beli 1 tambah 3 gratis 2 makan ayammakebeli 1 tambah 3 gratis 2 makan ayamayam
```

Program tersebut mengreplace spasi atau whitetab menggunakan sub dan \s dan text. Space kosong akan diisi dengan text dan menghasilkan output tersebut.

Contoh program lain yaitu mana saja yang ada a – d dan jumlahnya:

```
156 import re  
157  
158 text = "Ke Padang beli nasi Padang 13 ribu"  
159 x = re.findall("[a-d]", text)  
160 print(x)  
161 hitung = 0  
162 for i in x:  
163     hitung += 1  
164 print(hitung)
```

```
[Running] python -u "c:\Users\HP\prasss020904\bab15"  
['a', 'd', 'a', 'b', 'a', 'a', 'd', 'a', 'b']  
9
```

Program tersebut mengecek a-d dengan findall [a-d].

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

Jupyter Prastha Pradipta Purusa(71230987) Laprak 15 Last Checkpoint: 2 jam ya

```
File Edit View Insert Cell Kernel Widgets Help
[+] [-] [x] [y] [z] [a] [b] [c] [d] [e] [f] [g] [h] [i] [j] [k] [l] [m] [n] [o] [p] [q] [r] [s] [t] [u] [v] [w] [x] [y] [z] [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [Run] [Stop] [Restart] [Code] [v] [m]

In [ ]: import re
        from datetime import datetime

        with open("txt.txt", "r") as file:
            isi = file.read()

        sekarang = datetime.now()
        tanggal = re.findall(r'\d{4}-\d{2}-\d{2}', isi)

        for i in tanggal:
            format_tanggal = datetime.strptime(i, "%Y-%m-%d")
            selisih = (sekarang - format_tanggal).days
            tanggal_terformat = format_tanggal.strftime("%d-%m-%Y %H:%M:%S")
            print(f"{tanggal_terformat} selisih {selisih} hari")
```

```
[Running] python -u "c:\Users\HP\prasss02090
17-08-1945 00:00:00 selisih 28779 hari
11-11-1785 00:00:00 selisih 87131 hari
08-06-1783 00:00:00 selisih 88018 hari
02-05-1889 00:00:00 selisih 49339 hari
```

Line1: import regex

Line2: Mengimpor datetime dari modul datetime untuk tanggal dan waktu.

Line3: Membuka file txt.txt dengan read ("r") lalu menutupnya.

Line4: membaca seluruh isi txt dan menyimpan di variabel isi

Line5: Mendapatkan tanggal dan waktu saat ini dan menyimpannya dalam variabel sekarang

Line6: Menggunakan re.findall untuk mencari semua tanggal dalam format YYYY-MM-DD dalam isi dan menyimpan hasil pencarian dalam list tanggal.

Line7: Melakukan perulangan melalui setiap tanggal yang ditemukan.

Line8: Mengonversi string tanggal ke objek datetime menggunakan metode strptime dengan format "%Y-%m-%d"

Line9: Menghitung selisih antara tanggal saat ini (sekarang) dan tanggal yang ditemukan (format_tanggal) dalam satuan hari dengan .days()

Line10: Memformat ulang objek datetime (format_tanggal) menjadi string dengan format "%d-%m-%Y %H:%M:%S" dan disimpan di variabel tanggal_terformat

Line11: Memunculkan output hasil format tanggal dan selisih hari.

SOAL 2

```
In [ ]: import re
import random
import string

with open('txt2.txt', 'r') as file:
    isi = file.read()
line = isi.strip().split('\n')
hasil = []

for i in line:
    pola = re.compile(r'(.+?) dimiliki oleh (.+)')
    cocok = pola.match(i)
    if cocok:
        email = cocok.group(1).strip()
        nama = cocok.group(2).strip().split()[0]
        username = email.split('@')[0]
        panjang_password = 8
        acak = string.ascii_letters + string.digits
        password = ''.join(random.choice(acak) for i in range(panjang_password))
        hasil.append(f"{email} username: {username}, password: {password}")
for j in hasil:
    print(j)
```

```
[Running] python -u "c:\Users\HP\prasss020904\bab15\tempCodeRunnerR
anton@mail.com username: anton, password: IUt3mDhX
budi@gmail.co.id username: budi, password: 2LLcLxER
slamet@getnada.com username: slamet, password: 1KQG2TvF
matahari@tokopedia.com username: matahari, password: P985IClk
```

Line1: import regex

Line2: import random untuk menghasilkan bilangan acak.

Line3: import string berisi kumpulan konstanta yang mewakili alfabet ASCII, digit, tanda baca, serta fungsi-fungsi penting untuk manipulasi dan format string.

Line4: membaca file txt2.txt sebagai file lalu menutupnya.

Line5: membaca seluruh isi file txt dan disimpan di variabel isi

Line6: isi.strip() menghapus whitespace di teks, sementara .split('\n') memisahkan teks menjadi terpisah.

Line7: membuat list kosong untuk hasil

Line8: Perulangan setiap baris dalam line.

Line9: menghasilkan pola regex untuk mencocokkan string yang berisi email diikuti oleh "dimiliki oleh" dan nama.

Line10: mencocokkan pola dengan baris saat ini.

Line11: Memeriksa apakah pola cocok dengan baris

Line12: Mengambil bagian email dari hasil cocok dan menghapus spasi putih di awal dan akhir.

Line13: Mengambil bagian nama dari hasil cocok, menghapus whitespace di awal dan akhir, kemudian membagi berdasarkan spasi untuk mendapatkan kata pertama

Line14: Membagi email pada simbol '@' dan mengambil bagian pertama sebagai username.

Line15: Membuat panjang password 8 karakter

Line16: menggabungkan semua huruf ASCII (baik besar maupun kecil) dan angka menjadi satu string.

Line17: Membuat password acak dengan panjang yang ditentukan yaitu 8 dan menggunakan karakter dari acak.

Line18: Menambahkan string yang diformat berisi email, username, dan password ke dalam daftar hasil.

Line19: Perulangan untuk list hasil.

Line20: Membuat output setiap item yang berisi informasi email, username, dan password.

Link github:

<https://github.com/prasss020904/laprak15-71230987.git>