



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71210695
Nama Lengkap	Cahaya Sampebua
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

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

Pada materi sebelumnya, kita sudah belajar mengenai pengolahan string. Terkadang kita merasa sulit untuk mengakses elemen pada string tertentu. Disitulah kita membutuhkan regular expression. Regex digunakan untuk mengakses sub string dari suatu string dengan menggunakan pola. Misal kita ingin mengambil tanggal dari sebuah string

```
>>> string = "Pada tanggal 17 Agustus 1945, hari kemerdekaan Indonesia"
```

Seperti yang kita tahu, pola dari tanggal adalah MM-DD-YYYY ataupun YYYY-MM-DD. Intinya mengandung angka dan tanda strip ("-"). Jadi, untuk mengakses tanggal pada string tersebut kita menggunakan regex dengan pola angka dan strip.

Ambil contoh yang lebih mudah. Misal kita ingin mengakses kata "tanggal" dalam string tersebut. kita dapat menggunakan fungsi **re.search()** untuk mencari letak pola yang kita kehendaki. **re.search()** akan mengembalikan nilai match dan span dengan tipe re.Match.

```
>>> string = "Pada tanggal 17 Agustus 1945, hari kemerdekaan Indonesia"
>>> import re
>>> re.search("tanggal", string)
<re.Match object; span=(5, 12), match='tanggal'>
>>> type(re.search("tanggal", string))
<class 're.Match'>
```

Dalam regex, ada beberapa symbol yang kita gunakan untuk menentukan pola-pola karakter tertentu. Ada metacharacters, special sequences, dan sets.

Metacharacters are characters with a special meaning:

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
\$	Ends with	"planet\$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
	Either or	"falls stays"
()	Capture and group	

Gambar 1 Sumber: https://www.w3schools.com/python/python_regex.asp

Special Sequences

A special sequence is a `\` followed by one of the characters in the list below, and has a special meaning:

Character	Description	Example
<code>\A</code>	Returns a match if the specified characters are at the beginning of the string	<code>"\AThe"</code>
<code>\b</code>	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	<code>r"\bain"</code> <code>r"ain\b"</code>
<code>\B</code>	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	<code>r"\Bain"</code> <code>r"ain\B"</code>
<code>\d</code>	Returns a match where the string contains digits (numbers from 0-9)	<code>"\d"</code>
<code>\D</code>	Returns a match where the string DOES NOT contain digits	<code>"\D"</code>
<code>\s</code>	Returns a match where the string contains a white space character	<code>"\s"</code>
<code>\S</code>	Returns a match where the string DOES NOT contain a white space	<code>"\S"</code>
<code>\w</code>	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore <code>_</code> character)	<code>"\w"</code>
<code>\W</code>	Returns a match where the string DOES NOT contain any word characters	<code>"\W"</code>
<code>\Z</code>	Returns a match if the specified characters are at the end of the string	<code>"Spain\Z"</code>

Gambar 2 Sumber: https://www.w3schools.com/python/python_regex.asp

Sets

A set is a set of characters inside a pair of square brackets `[]` with a special meaning:

Set	Description
<code>[arn]</code>	Returns a match where one of the specified characters (<code>a</code> , <code>r</code> , or <code>n</code>) are present
<code>[a-n]</code>	Returns a match for any lower case character, alphabetically between <code>a</code> and <code>n</code>
<code>[^arn]</code>	Returns a match for any character EXCEPT <code>a</code> , <code>r</code> , and <code>n</code>
<code>[0123]</code>	Returns a match where any of the specified digits (<code>0</code> , <code>1</code> , <code>2</code> , or <code>3</code>) are present
<code>[0-9]</code>	Returns a match for any digit between <code>0</code> and <code>9</code>
<code>[0-5][0-9]</code>	Returns a match for any two-digit numbers from <code>00</code> and <code>59</code>
<code>[a-zA-Z]</code>	Returns a match for any character alphabetically between <code>a</code> and <code>z</code> , lower case OR upper case
<code>[+]</code>	In sets, <code>+</code> , <code>*</code> , <code>.</code> , <code> </code> , <code>()</code> , <code>\$</code> , <code>{}</code> has no special meaning, so <code>[+]</code> means: return a match for any <code>+</code> character in the string

Gambar 3 Sumber: https://www.w3schools.com/python/python_regex.asp

Beberapa fungsi regex:

1. **re.findall()** berfungsi untuk mengembalikan list berisi pola yang dikehendaki.
2. **re.search()** berfungsi untuk mengembalikan string yang sesuai dengan pola.
3. **re.split()** berfungsi mengembalikan list dengan isi sub string yang telah terpecah oleh pola.
4. **re.sub** mengganti string dengan pola yang cocok.

Format penulisan regex 1-3 adalah `re.findall("pattern",string)` missal untuk findall.

Sedangkan `re.sub` ditulis dengan cara `re.sub("pattern","pengganti",string)`.

Contoh kasus dengan menggunakan regex.

1. Mencari kode pos

Kode pos memiliki pola yaitu 5 angka.

```
kalimat = "Saya tinggal di Yogyakarta. Saya tidak tinggal di Aceh. DKI  
Jakarta 12521, 11567, 14321.\n \n  
No. telpon 081221088021. NAD 23521. Jawa Tengah telepon +6285874556321."
```

Dapat menggunakan pola regex

```
re_kalimat = re.findall("\W(\d{5})\W",kalimat)
```

karena setiap angka kode pos diawali dan diakhiri dengan karakter non-alphanumeric.

Output:

```
['12521', '11567', '14321', '23521']
```

2. Mengganti karakter whitespace dengan symbol +

```
teks = "And the wonder of it all is that you just don't realize how  
much I love you."
```

Gunakan pola

```
5 re_teks = re.sub("\s", "+", teks)  
6 print(re_teks)
```

Output

```
And+the+wonder+of+it+all+is+that+you+just+don't+realize+how+much+I+love+you.
```

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.

Link GitHub → <https://github.com/cahaya-sampebua/Latihan-Mandiri-PrakALPro-14.git>

SOAL 1

```
#Soal 1

from datetime import date
import re

def selisihHari(data):
    re_data = re.findall("\d{4}-\d{2}-\d{2}",data)
    today = date.today()
    for i in re_data:
        re_i = re.split("-",i)
        year = int(re_i[0])
        month = int(re_i[1])
        day = int(re_i[2])

        past = date(year,month,day)

        hasil = str(today - past)
        hasil = hasil.split(" ")

        print(i,"00:00:00 selisih",hasil[0],"hari")

data = "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).\"

selisihHari(data)
```

Membuat list berisi tanggal dengan pola YYYY-MM-DD. Variabel today adalah tanggal hari ini. Untuk setiap indeks dalam list tadi, pisahkan setiap angka dengan strip. Lalu, gunakan untuk variabel past dimana isinya adalah tanggal yang ingin dicari selisihnya. Kurangi today dengan past untuk mendapatkanselisih hari. Print dengan format seperti dalam source code.

SOAL 2

```
#Soal 3

import re
import string
import random

def randomPassword(n):
    list_choice = list(string.ascii_letters + string.digits)
    password = ""
    for i in range(n):
        password += str(random.choice(list_choice))

    return password

def bikinPassword(data):
    list_email = list()
    list_uname = list()
    for line in data:
        re_email = re.findall("\w+@\w+\.\w+[\.\w+?]+", line)
        re_uname = re.findall("(\\w+)@", line)
        if re_email:
            list_email.append(re_email[0])
        if re_uname:
            list_uname.append(re_uname[0])

    for i in range(len(list_email)):
        print(list_email[i], "username:", list_uname[i] + ", password:", randomPassword(8))

data = open("email.txt", "r")

bikinPassword(data)
```

```
anton@mail.com username: anton, password: A955k1ue
budi@gmail.co.id username: budi, password: cpXFhb28
slamet@getnada.com username: slamet, password: gkK4TzjA
matahari@tokopedia.com username: matahari, password: Q0oquN8k
```

Import library yang diperlukan. Buat fungsi random password dengan parameter n untuk membuat n digit karakter random dengan menggunakan library string. Setelah itu, buat fungsi untuk membuat output seperti dalam soal. Buat list email dan list username dan isi dengan cara menggunakan regularexpression. Cari sesuai format email, yaitu yang ada @ nya, dan username adalah kata sebelum tanda@. Untuk i dalam range jumlah dari list email, print dengan format seperti dalam source code.