



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230978</b>
<b>Nama Lengkap</b>	<b>Jonathan Satriani Gracio Andrianto</b>
<b>Minggu ke / Materi</b>	<b>14 / Regular Expression</b>

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%)

### Pengantar Regex

Dalam bab mengenai String, kita telah mempelajari beberapa teknik untuk mengakses, memanipulasi, dan menangani berbagai kasus pengolahan string lainnya, termasuk string yang terdapat dalam file. Pengalaman tersebut menunjukkan bahwa pengolahan string menggunakan teknik biasa atau standar bisa cukup sulit. Ada metode pengolahan string yang lebih mudah dan cepat, yaitu dengan menggunakan regular expression.

Regular expression adalah pola berbentuk kumpulan karakter yang digunakan untuk mencocokkan pola tertentu dalam string yang ingin ditemukan. Regex memudahkan kita dalam mencari string dengan pola tertentu, mengganti string berdasarkan pola tersebut, dan menghapus string dengan pola yang diinginkan. Pada dasarnya, regex membantu dalam parsing string yang biasanya hanya dilakukan dengan perintah `split()` dan `find()`.

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

```
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)
```

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlw@iupui.edu
From: zqian@umich.edu
From: rjlw@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: wagnerm@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", maka kita 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

Sebelum menggunakan fungsi regex perlu diketahui terlebih dahulu meta character / special character dan kegunaannya pada pola regex seperti pada tabel 1.1

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

### 1.1 Special Character pada Python

Pada Python terdapat beberapa special character (escaped characters) seperti pada tabel 1.2

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"

### 1.2 Escaped Character pada Regex

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

[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

1.3 Himpunan Karakter pada Regex

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

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

1.4 Fungsi Regex pada Python

## Kegiatan Praktikum

### Penggunaan findall

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

```
import re

txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
x = re.findall("mata", txt)
y = re.findall("saya", txt)
for i in x:
    print(i)

if (y):
    print("Ada yang cocok!")
else:
    print("Tidak ada yang cocok!")

#output
mata
mata
mata
mata
```

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

Contoh lain :

```
import re
handle=open('mbox-short.txt')
for line in handle:
    line=line.rstrip()
    x=re.findall('\S+@\S+', line)
    if len(x)>0:
        print(x)
#output
['step ['stephen.marquard@uct.ac.za']
['<postmaster@collab.sakaiproject.org>']
['<200801051412.m05ECIaH010327@nakamura.uits.iupui.edu>']
['<source@collab.sakaiproject.org>;']
['<source@collab.sakaiproject.org>;']
['<source@collab.sakaiproject.org>;']
['apache@localhost']
Dst..
```

### Penggunaan search

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)

print("Spasi ditemukan di:", x.start())
print(y)
#output
Spasi ditemukan di: 4
None
```

Contoh lain fungsi search: Pada mbox kita ingin menemukan kata-kata:

X-DSPAM-Confidence: 0.847

5X-DSPAM-Probability: 0.0000

X-DSPAM-Confidence: 0.6178

X-DSPAM-Probability: 0.0000

Untuk melakukannya dapat digunakan regex: X-.\*: [0-9.]+

```
import re
handle=open('mbox-short.txt')
for line in handle:
    line=line.rstrip()
    if(re.search('^X-.*: [0-9.]+', line)):
        print(line)
#output
X-DSPAM-Confidence: 0.8475
X-DSPAM-Probability: 0.0000
X-DSPAM-Confidence: 0.6178
Dst..
```

### Penggunaan split

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

```
import re
txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
y = re.split("\s", txt, 1) #split 1 kata pertama
print(x)
#output
['The', 'rain', 'in', 'Spain']
['The', 'rain', 'in', 'Spain']
```

### Penggunaan sub

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

```
import re
txt = "Sang mata-mata sedang memata-matai kasus kaca mata di toko Matahari"
x = re.sub("\s", "-", txt) #mengganti spasi dengan -
print(x)
y = re.sub("\s", "*", txt, 2) #mengganti spasi dengan * 2 saja
print(y)
#output
Sang-mata-mata-sedang-memata-matai-kasus-kaca-mata-di toko-Matahari
Sang*mata-mata*sedang memata-matai kasus kaca mata di toko Matahari
```

## BAGIAN 2: LATIHAN MANDIRI (60%)

### SOAL 1

#### Source Code

```
import re
from datetime import datetime

teks = """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)."""

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

hasil = []
for tanggal_ in tanggal:
    tanggal_bro = datetime.strptime(tanggal_, '%Y-%m-%d')
    selisih = sekarang - tanggal_bro
    tanggal_hasil = tanggal_bro.strftime('%d-%m-%Y')
    hasil.append(f"{tanggal_bro} selisih {selisih.days} hari")

for item in hasil:
    print(item)
```

#### Output

```
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
```

#### Penjelasan

Langkah pertama yaitu mengimpor library `re` untuk regex dan `datetime` untuk manipulasi tanggal. Kita cari tanggal yang ada pada 'teks' dengan regular expression. Dan kita gunakan `datetime.now()` untuk mencari tanggal sekarang. Kemudian kita buat list kosong untuk nantinya kita isi dengan hasil. Selanjutnya kita buat perulangan dari tanggal tanggal yang ada pada teks. Pada setiap iterasi, kita ubah tanggal yang masih string itu menjadi objek 'tanggal\_bro' agar nanti bisa dihitung selisih harinya. Kemudian kita hitung selisihnya, dan kita ubah format tanggal dari YY-MM-DD menjadi DD-MM-YY dengan `strftime()`. Dan kita masukkan semua hasil tersebut ke dalam list hasil dan kita cetak.

## Soal 2

### Source Code

```
import re
import random
import string

teks = """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"""

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

def password_acak(length = 8) :
    sandi = string.ascii_letters + string.digits
    hasil = ""
    for i in range(length) :
        hasil += random.choice(sandi)
    return hasil

for j in email :
    user = j.split('@')[0]
    password = password_acak()
    print(f"{j} username : {user}, password : {password}")
```

### Output

```
anton@mail.com username : anton, password : WmKSG0u8
budi@gmail.co.id username : budi, password : ii9yMD05
slamet@getnada.com username : slamet, password : ewVt75So
matahari@tokopedia.com username : matahari, password : OyrMglzc
```

### Penjelasan

pertama kita import library 're', 'random' untuk pengacakan, dan 'string' untuk akses string dan angka. Kemudian kita cari email yang ada di teks dengan regular expression. Selanjutnya kita buat fungsi dengan nama 'password\_acak' dengan parameter default yaitu length = 8 yaitu panjangnya 8. Dalam fungsi tersebut kita buat karakternya yaitu dengan gabungan angka, huruf kecil dan huruf besar dan disimpan di variable sandi, dan kita buat passwordnya dengan perulangan dengan range length yaitu 8 dan setiap iterasi variable hasil diisi dengan random karakter. Kemudian buat perulangan untuk mencetak output. Setiap iterasi kita harus split email yang di teks dengan fungsi 'split.(@)' dan usernya di index[0]. Dan kita panggil fungsi password\_acak untuk setiap email mendapat password acak.



Link Github

[https://github.com/JonathanAndrianto123/laporan\\_alpro\\_terakhir\\_71230978.git](https://github.com/JonathanAndrianto123/laporan_alpro_terakhir_71230978.git)