



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71231042</b>
<b>Nama Lengkap</b>	<b>Revaldo Fransisco Hohary</b>
<b>Minggu ke / Materi</b>	<b>15 / 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%)

### MATERI 1

#### A. Pengantar Regex

Pada Bab String kita membahas cara-cara untuk mengakses string, memanipulasi string, dan berbagai situasi pengolahan string lainnya, termasuk string yang ada di file. Fakta bahwa pengalaman tersebut menunjukkan bahwa kita menghadapi banyak tantangan untuk melakukan pengolahan string dengan metode konvensional atau standar; ada metode pengolahan string yang lebih mudah dan cepat yang menggunakan ekspresi reguler. Regex adalah ekspresi pola yang terdiri dari kumpulan karakter yang digunakan untuk menemukan pola (pattern) yang sama dengan pola regex di dalam string lain yang ingin dicari. Secara khusus, regex membantu dalam parsing string, yang biasanya dilakukan hanya dengan perintah `split()` dan `find()`.

Meskipun Regex sangat baik untuk mencari dan mengekstrak pola, polanya cukup kompleks. Python adalah salah satu bahasa pemrograman yang mendukung library regex dengan cara `import re`. Salah satu alat yang paling mudah digunakan dari `re` library adalah `search()`. Kita akan menggunakan file `mbox-short.txt` dan 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)
```

Bisa dilihat dari kode di atas bahwa `re.search` dapat diganti dengan perintah `find()` pada string biasa. Pola-pola ini belum menggunakan kemampuan regex sepenuhnya.

From: [stephen.marquard@uct.ac.za](mailto:stephen.marquard@uct.ac.za)  
From: [louis@media.berkeley.edu](mailto:louis@media.berkeley.edu)  
From: [zqian@umich.edu](mailto:zqian@umich.edu)  
From: [rjlowe@iupui.edu](mailto:rjlowe@iupui.edu)  
From: [zqian@umich.edu](mailto:zqian@umich.edu)  
From: [rjlowe@iupui.edu](mailto:rjlowe@iupui.edu)  
From: [cwen@iupui.edu](mailto:cwen@iupui.edu)  
From: [cwen@iupui.edu](mailto:cwen@iupui.edu)  
From: [gsilver@umich.edu](mailto:gsilver@umich.edu)  
From: [gsilver@umich.edu](mailto:gsilver@umich.edu)  
From: [zqian@umich.edu](mailto:zqian@umich.edu)  
From: [gsilver@umich.edu](mailto:gsilver@umich.edu)  
From: [wagnermr@iupui.edu](mailto:wagnermr@iupui.edu)  
From: [zqian@umich.edu](mailto:zqian@umich.edu)  
From: [antranig@caret.cam.ac.uk](mailto:antranig@caret.cam.ac.uk)  
From: [gopal.ramasammycook@gmail.com](mailto:gopal.ramasammycook@gmail.com)

From: [david.horwitz@uct.ac.za](mailto:david.horwitz@uct.ac.za)  
From: [david.horwitz@uct.ac.za](mailto:david.horwitz@uct.ac.za)  
From: [david.horwitz@uct.ac.za](mailto:david.horwitz@uct.ac.za)  
From: [david.horwitz@uct.ac.za](mailto:david.horwitz@uct.ac.za)  
From: [stephen.marquard@uct.ac.za](mailto:stephen.marquard@uct.ac.za)  
From: [louis@media.berkeley.edu](mailto:louis@media.berkeley.edu)  
From: [louis@media.berkeley.edu](mailto:louis@media.berkeley.edu)  
From: [ray@media.berkeley.edu](mailto:ray@media.berkeley.edu)  
From: [cwen@iupui.edu](mailto:cwen@iupui.edu)  
From: [cwen@iupui.edu](mailto:cwen@iupui.edu)  
From: [cwen@iupui.edu](mailto:cwen@iupui.edu)

Jumlah: 27

Kita harus mengubah parameter fungsi pencarian pada `re.search` menjadi `re.search("From")` jika kita ingin mencari baris yang dimulai dengan 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)
```

Kode di atas dapat digantikan juga dengan fungsi `string.startswith("From :")`.

## **B. 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 14.1 Pada Python terdapat beberapa special character (escaped characters) seperti pada tabel 14.2 Pada Python terdapat beberapa penggunaan himpunan character dengan menggunakan simbol [], pada tabel 14.3 Pada Python terdapat 4 buah fungsi yang bisa dipakai untuk menggunakan Regex seperti pada tabel 14.4

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

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" "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"

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 14.4: 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

### C. Kegiatan Praktikum

#### a. Penggunaan findall

Berikut contoh penggunaan findall:

```
import re
txt = "dia marah-marah waktu pulang kampus"
x = re.findall("marah", txt)
y = re.findall("dia", txt)
for i in x:
    print(i)
if (y):
    print("Ada yang cocok!")
else:
    print("Tidak ada yang cocok!")
```

outpunya :

```
marah
marah
Ada yang cocok!
```

Contoh lain findall:

```
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)
```

Outputnya:

```
['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']
['source@collab.sakaiproject.org;']
['stephen.marquard@uct.ac.za']
['source@collab.sakaiproject.org']
....dst
```

## b. Penggunaan Search

Berikut contoh penggunaan Search:

```
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)
```

Outputnya:

```
3
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
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  
dst...

**c. Penggunaan split**

Berikut adalah contoh penggunaan split:

```
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)
```

Hasil: ['The','rain','in','Spain'] ['The','rain in Spain']

**d. Penggunaan sub**

Berikut contoh penggunaan menggunakan sub:

```
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)
```

Ouputnya: Sang-mata-mata-sedang-memata-matai-kasus-kaca-mata-di toko-Matahari

Sang\*mata-mata\*sedang memata-matai kasus kaca mata di toko Matahari

**D. Link Github**

<https://github.com/FrealY0901/Tugas-PrakAIProLast.git>

## BAGIAN 2: LATIHAN MANDIRI (60%)

### SOAL 1

```
latihan1.py / ...
1  import re
2  from datetime import datetime
3
4  def tampilkan_tanggal(teks):
5      tanggal = re.findall(r'\d{4}-\d{2}-\d{2}', teks)
6      for tgl in tanggal:
7          tgl_obj = datetime.strptime(tgl, '%Y-%m-%d')
8          selisih = (datetime.now() - tgl_obj).days
9          print(f'{tgl} 00:00:00 selisih {selisih} hari')
10
11 teks = 'Pada tahun 2020-03-01 masa pembelajaran dilaakukan serentak s
12 tampilkan_tanggal(teks)
```

1. `import re` dan `from datetime import datetime`: Ini adalah perintah untuk mengimpor modul `re` dan `datetime` yang digunakan dalam program. Modul `re` digunakan untuk melakukan operasi regex (regular expression), sedangkan modul `datetime` digunakan untuk melakukan operasi waktu.
2. `def tampilkan_tanggal(teks):`: Ini adalah definisi fungsi `tampilkan_tanggal` yang menerima satu parameter, yaitu `teks`. Fungsi ini akan menampilkan tanggal-tanggal yang ada dalam teks, serta menghitung selisih hari antara tanggal tersebut dengan tanggal sekarang.
3. `tanggal = re.findall(r'\d{4}-\d{2}-\d{2}', teks)`: Ini adalah perintah untuk mencari semua kemunculan pola yang sesuai dengan regex yang diberikan, yaitu `\d{4}-\d{2}-\d{2}`. Pola ini akan mencari semua tanggal yang berformat YYYY-MM-DD. Kemudian, hasil pencarian disimpan dalam variabel `tanggal`.
4. `for tgl in tanggal:`: Ini adalah perulangan yang akan melakukan operasi untuk setiap tanggal yang ada dalam variabel `tanggal`.
5. `tgl_obj = datetime.strptime(tgl, '%Y-%m-%d')`: Ini adalah perintah untuk mengubah string tanggal menjadi objek `datetime`. Fungsi `datetime.strptime` digunakan untuk mengubah string menjadi objek `datetime` dengan format yang diberikan, yaitu `%Y-%m-%d`.
6. `selisih = (datetime.now() - tgl_obj).days`: Ini adalah perintah untuk menghitung selisih hari antara tanggal sekarang dengan tanggal yang diambil dari variabel `tgl_obj`. Fungsi `datetime.now()` digunakan untuk mendapatkan tanggal sekarang, kemudian dikurangi dengan tanggal yang diambil dari variabel `tgl_obj`. Hasilnya disimpan dalam variabel `selisih`.
7. `print(f'{tgl} 00:00:00 selisih {selisih} hari')`: Ini adalah perintah untuk menampilkan tanggal dan selisih hari yang dihasilkan.



- teks = 'Pada tahun 2020-03-01 masa pembelajaran dilaakukan serentak secara daring, seluruh masyarakat akan diliburkan dikarenakan penerapan lockdown selama dua minggu atau sampe pada 2020-04-05, tetapi lockdown terus berlanjut sampai dengan 2021-01-10.': Ini adalah teks yang akan digunakan dalam program.
- tampilkan\_tanggal(teks): Ini adalah perintah untuk memanggil fungsi tampilkan\_tanggal dengan parameter teks.

Outputnya:

```
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Regex> & C:\Users\Asus\Documents\UKDW\PrakAlPro\Regex\latihan1.py
2020-03-01 00:00:00 selisih 1556 hari
2020-04-05 00:00:00 selisih 1521 hari
2021-01-10 00:00:00 selisih 1241 hari
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Regex> █
```

## SOAL 2

```
import random
import re

def generate_password(email):
    username = re.findall(r'[a-zA-Z0-9]+(?:=|@)', email)[0]
    password = ''.join(chr(random.randint(33, 126)) for _ in range(8))
    return f'{email} username: {username} , password: {password}'

emails = ['doal@mail.com', 'ciss@gmail.co.id', 'jiii@jar.com', 'lah@pen.com']
for email in emails:
    print(generate_password(email))
```

- import random dan import re: Ini adalah perintah untuk mengimpor modul random dan re yang digunakan dalam program. Modul random digunakan untuk menghasilkan angka acak, sedangkan modul re digunakan untuk melakukan operasi regex (regular expression).
- def generate\_password(email):: Ini adalah definisi fungsi generate\_password yang menerima satu parameter, yaitu email. Fungsi ini akan mengembalikan sebuah string yang berisi email, username, dan password yang dihasilkan.
- username = re.findall(r'[a-zA-Z0-9]+(?:=|@)', email)[0]: Ini adalah perintah untuk mengambil username dari email. Fungsi re.findall digunakan untuk mencari semua kemunculan pola yang sesuai dengan regex yang diberikan, yaitu [a-zA-Z0-9]+(?:=|@). Pola ini akan mencari semua karakter alfanumerik yang diikuti oleh tanda @. Kemudian, [0] digunakan untuk mengambil hanya username pertama yang ditemukan.

4. `password = ''.join(chr(random.randint(33, 126)) for _ in range(8))`: Ini adalah perintah untuk menghasilkan password acak dengan panjang 8 karakter. Fungsi `random.randint(33, 126)` digunakan untuk menghasilkan angka acak antara 33 dan 126, yang merupakan ASCII code untuk karakter-karakter yang dapat digunakan sebagai password. Kemudian, fungsi `chr` digunakan untuk mengubah angka tersebut menjadi karakter. Selanjutnya, `join` digunakan untuk menggabungkan semua karakter tersebut menjadi sebuah string.
5. `return f'{email} username: {username} , password: {password}'`: Ini adalah perintah untuk mengembalikan sebuah string yang berisi email, username, dan password yang dihasilkan.
6. `emails = ['doal@mail.com', 'ciss@gmail.co.id', 'jiii@jar.com', 'lah@pen.com']`: Ini adalah daftar email yang akan digunakan dalam program.
7. `for email in emails::` Ini adalah perulangan yang akan melakukan operasi untuk setiap email yang ada dalam daftar emails.
8. `print(generate_password(email))`: Ini adalah perintah untuk menampilkan hasil dari fungsi `generate_password` untuk setiap email yang ada dalam daftar emails.

**Outputnya:**

```
doal@mail.com username: doal , password: 1?.X[u*J
ciss@gmail.co.id username: ciss , password: fjlo*SoU
jiii@jar.com username: jiii , password: O6X-7}vk
lah@pen.com username: lah , password: KK1SF)lI
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\Regex> 
```