

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71231006
Nama Lengkap	Fransiskus Tedyanto seran
Minggu ke / Materi	14/ Reguler 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
2023

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.

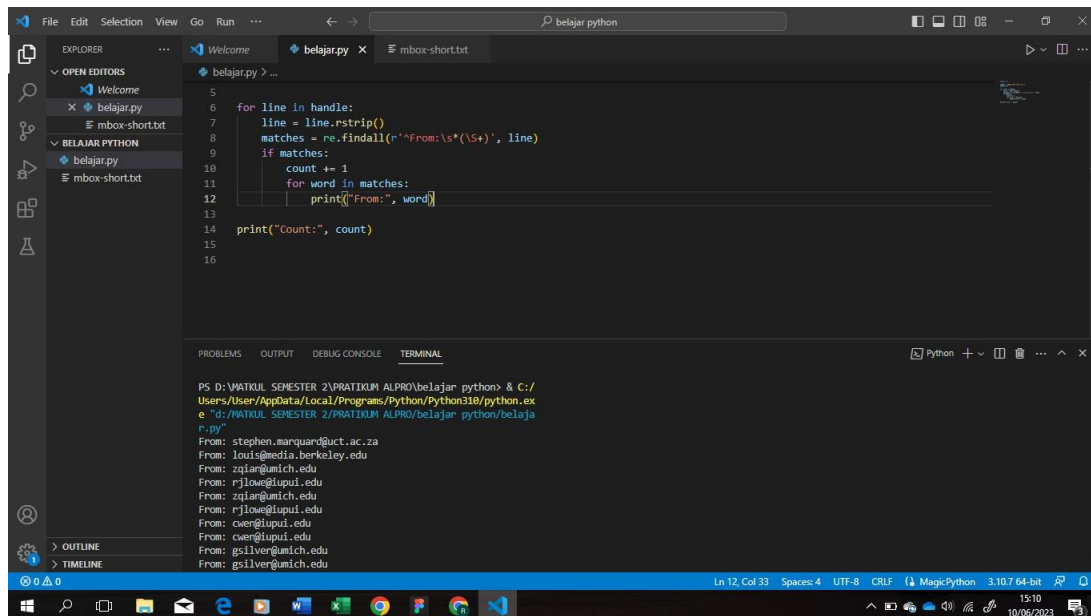
Paengantar Regex

Regex (Regular Expression) adalah sebuah metode yang digunakan untuk mencocokkan pola atau urutan tertentu dalam sebuah teks. Di dalam konteks python, modul re menyediakan implementasi regex. Dengan menggunakan regex, kita dapat melakukan pencarian, pengecekan, serta manipulasi teks dengan sangat efisien. Kita dapat mencari pola tertentu dalam sebuah string, memeriksa apakah sebuah string cocok dengan pola yang ditentukan, dan mengganti bagian-bagian tertentu dari sebuah string berdasarkan pola yang ditentukan.

Regex sangat powerful dalam searching dan extracing pola namun memiliki pola yang cukup rumit. Tidak semua bahasa pemrograman mendukung regular expression library. Python merupakan salah satu bahasa yang mendukung library search. Salah satu fungsi yang paling mudah digunakan dari library re adalah `search()`. Berikut merupakan contoh bagaimana cara kita menampilkan semua string pada file `mbox-short.txt` dengan menggunakan fungsi `re.search()`.

The screenshot shows a Windows 10 desktop with a Visual Studio Code editor window. The editor is open to a file named 'mbbox-short.txt' in a project named 'belajar.py'. The code in the editor is a simple Python script that reads a file and prints the count of lines. The terminal window at the bottom shows the output of the script, which is 'Count: 1'. The status bar at the bottom indicates the file is 'mbbox-short.txt' and the encoding is 'UTF-8'.

Dari program di atas kita dapat melihat bahwa `re.search` bisa diganti dengan menggunakan perintah `re.findall()` pada string biasa. Pola pada contoh di atas kita akan menggantinya dengan kemampuan regex yang seutuhnya. Berikut merupakan contohnya:



```
5
6 for line in handle:
7     line = line.rstrip()
8     matches = re.findall(r'^From:.*', line)
9     if matches:
10        count += 1
11        for word in matches:
12            print("From:", word)
13
14 print("Count:", count)
15
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\WATKUL SEMESTER 2\PRATIUM ALPRO\belajar python> & C:/
Users/User/AppData/Local/Programs/Python/Python318/python.ex
e "d:\WATKUL SEMESTER 2\PRATIUM ALPRO\belajar python\belaja
r.py"
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjloue@iupui.edu
From: zqian@umich.edu
From: rjloue@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: gsilver@umich.edu
From: gsilver@umich.edu
```

Untuk memodifikasi program yang sebelumnya kita hanya menampilkan kata-kata awal yang cocok dengan pola `from`, kita dapat menggunakan fungsi `re.findall()` untuk mencari kata-kata awal pada setiap baris yang dimulai dengan `from`. Dalam pola `'r'^From:.*(s\s+)'` digunakan untuk mencocokkan kata-kata awal setelah kata `from` `'^'` mengindikasikan awal dari baris `s*` digunakan untuk mencocokkan spasi (jika ada) setelah kata `from`, dan `(\s+)` digunakan untuk mencocokkan satu atau lebih karakter yang bukan spasi setelahnya. Kemudian, kita menggunakan `re.findall()` untuk mencari semua kata yang cocok dengan pola tersebut pada setiap baris. Jika ada hasil pencarian (`matches`), maka kita menambahkan 1 ke variabel `count` dan mencetak kata-kata awal (`word`) yang cocok dengan pola. Bagian akhir program, kita mencetak jumlah kemunculan yang dihitung (`count`).

MATERI 2

Meta character, escaped character, set of character, dan fungsi regex pada library python

Sebelum menggunakan fungsi regex perlu diketahui meta character/ special character dan kegunaannya. Pada python terdapat beberapa special character. Pada python juga terdapat beberapa penggunaan himpunan character dengan menggunakan simbol `[]`. Berikut merupakan special character, dan escaped character yang bisa digunakan 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

Special character pada python

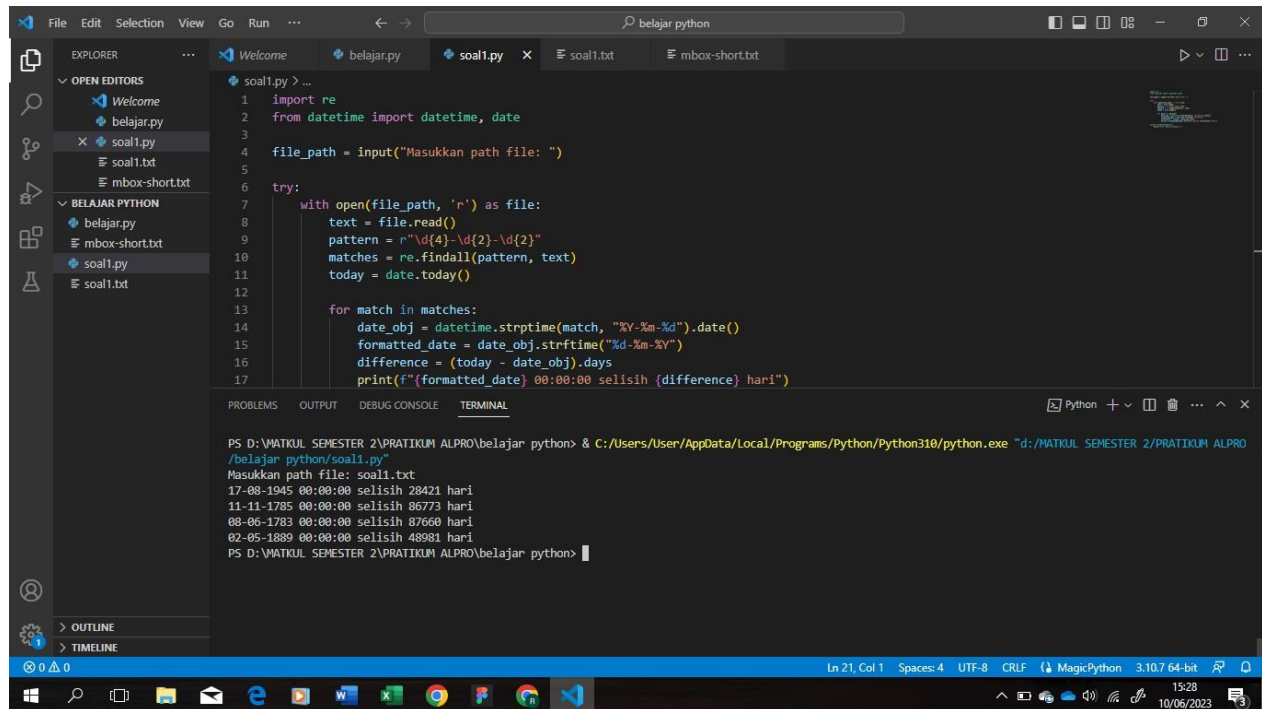
Special Characters	Kegunaan	Contoh
\b	Digunakan untuk mengetahui apakah suatu pola berada di awal kata atau akhir kata	"R\\bin" "Rain\\b"
\d	Digunakan untuk mengetahui apakah karakter adalah sebuah digit (0 s/d 9)	\d
\D	Digunakan untuk mengetahui apakah karakter yang bukan digit	\D
\s	Digunakan untuk 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"

Escaped character pada regex

BAGIAN 2: LATIHAN MANDIRI

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



The screenshot shows a Visual Studio Code editor window with a Python file named `soal1.py` open. The code in the editor is as follows:

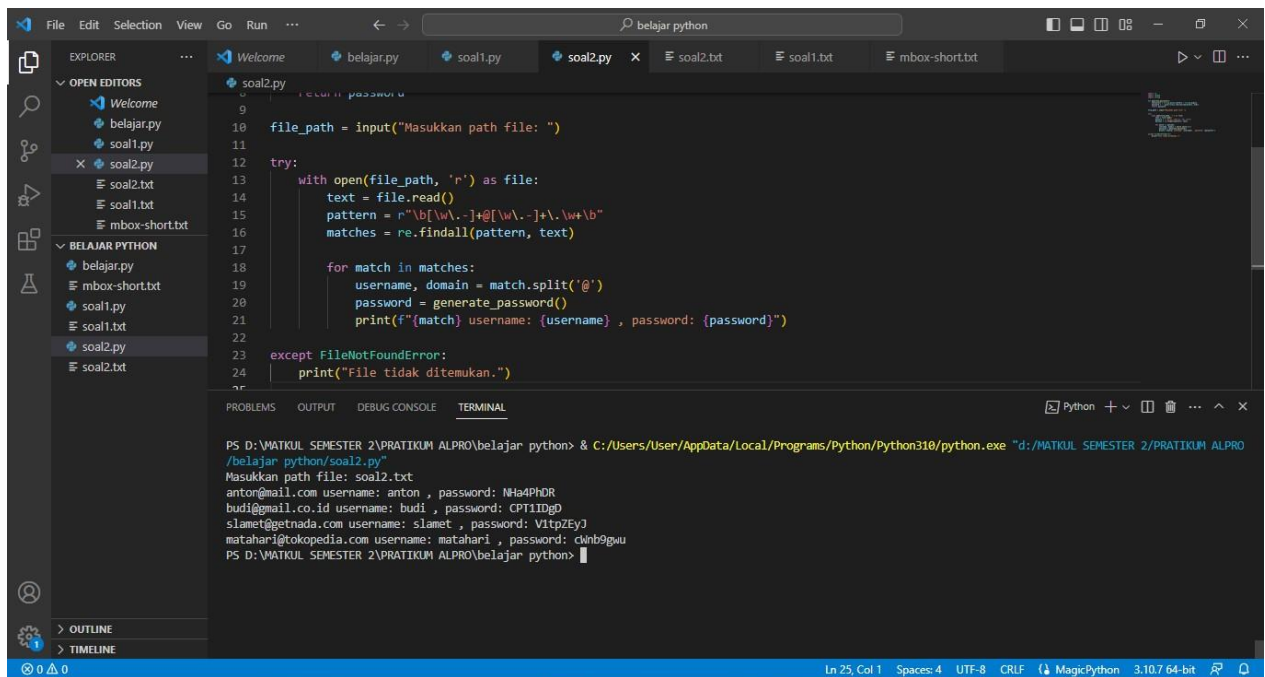
```
1 import re
2 from datetime import datetime, date
3
4 file_path = input("Masukkan path file: ")
5
6 try:
7     with open(file_path, 'r') as file:
8         text = file.read()
9         pattern = r"\d{4}-\d{2}-\d{2}"
10        matches = re.findall(pattern, text)
11        today = date.today()
12
13    for match in matches:
14        date_obj = datetime.strptime(match, "%Y-%m-%d").date()
15        formatted_date = date_obj.strftime("%d-%m-%Y")
16        difference = (today - date_obj).days
17        print(f"{formatted_date} 00:00:00 selisih {difference} hari")
```

Below the editor, the terminal window shows the execution of the script. The user enters the path `D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajar python\soal1.txt`, and the program outputs the following results:

```
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajar python> C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajar python\soal1.py"
Masukkan path file: soal1.txt
17-08-1945 00:00:00 selisih 28421 hari
11-11-1785 00:00:00 selisih 86773 hari
08-06-1783 00:00:00 selisih 87660 hari
02-05-1889 00:00:00 selisih 48981 hari
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajar python>
```

Pada program di atas fungsi `format_date()` dan `calculate_date_difference()` telah dihilangkan karena pemanggilan dan logika yang terkait dapat langsung dimasukkan ke dalam loop `for`. Hal ini membuat program lebih sederhana dengan mengurangi fungsi-fungsi tambahan yang sebenarnya hanya dipanggil sekali. Selain itu, variabel `today` yang mewakili tanggal saat ini dideklarasikan di luar loop `for` agar tidak perlu dihitung ulang setiap kali loop dieksekusi. Seluruh kode lainnya tetap sama dengan program sebelumnya, termasuk membaca file, mencocokkan tanggal dengan regex, dan mencetak hasil sesuai format yang diminta.

SOAL 2



```
9 def generate_password():
10     return ''.join(random.choices(string.ascii_letters + string.digits, k=8))
11
12 def main():
13     file_path = input("Masukkan path file: ")
14
15     try:
16         with open(file_path, 'r') as file:
17             text = file.read()
18             pattern = r"b[\w\.-]+\@([\w\.-]+\.\w+\b)"
19             matches = re.findall(pattern, text)
20
21             for match in matches:
22                 username, domain = match.split('@')
23                 password = generate_password()
24                 print(f"{match} username: {username} , password: {password}")
25
26     except FileNotFoundError:
27         print("File tidak ditemukan.")
28
29 if __name__ == '__main__':
30     main()
```

PS D:\VATKUL SEMESTER 2\PRATIUM ALPRO\belajar python> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/belajar python/soal2.py"

Masukkan path file: soal2.txt

anton@mail.com username: anton , password: Nts4PhdR

budi@gmail.co.id username: budi , password: CPTIIDgD

slamet@getnada.com username: slamet , password: VltzZEyJ

matahari@tokopedia.com username: matahari , password: chnb9gwu

PS D:\VATKUL SEMESTER 2\PRATIUM ALPRO\belajar python>

Program di atas adalah penggunaan `random.choices()` dengan parameter `k=8` untuk langsung menghasilkan password acak sepanjang 8 karakter, yang terdiri dari huruf dan angka. Selain itu, variabel `domain` tidak digunakan dalam output, sehingga pemisahan email menjadi username dilakukan menggunakan `split('@')` langsung ke variabel `username`. Seluruh kode lainnya tetap sama dengan program sebelumnya, termasuk membaca file, mencocokkan email dengan regex, dan mencetak hasil sesuai format yang diminta.

