



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231054
Nama Lengkap	Benedictus Karol Wojtyfa Putra Santoso
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.

MATERI 14.3.1

Regular expression adalah ekspresi pola yang berbentuk kumpulan karakter yang digunakan untuk menemukan pola (pattern) yang sama dengan pola regex di dalam string lain yang ingin dicari. Regex membantu kita mencari string dengan pola tertentu, mencari string dengan pola tertentu dan mengganti string dengan pola tertentu. Tidak semua bahasa pemrograman mendukung re.

Regex sudah banyak digunakan didalam pencocokan pola, misalnya :

- Username dan password
- E-mail, URL, dan HTML tag
- Alamat IP
- Nomor telepon

Sumber :

[https://anom.lecturer.pens.ac.id/Praktikum%20Bahasa%20Pemrograman%20dan%20Perangkat%20Penyuluhan%20Langsung/Praktikum%2003%20-%20Regular%20Expression%20\(regex\).pdf](https://anom.lecturer.pens.ac.id/Praktikum%20Bahasa%20Pemrograman%20dan%20Perangkat%20Penyuluhan%20Langsung/Praktikum%2003%20-%20Regular%20Expression%20(regex).pdf)

Manfaat regex dalam pemrograman :

- Untuk validasi data
- Untuk pencaharian
- Untuk find dan replace

Symbol :

() : untuk grouping

| : untuk logika or

[a-z] : untuk mencari karakter dari huruf a sampai z

[A-Z] : untuk mencari karakter dari huruf A sampai Z

[0-9] : untuk mencari angka dari 0 sampai 9

\w : untuk mencari kata

\d untuk mencari angka

Sumber :

https://www.petanikode.com/regex/#google_vignette

MATERI 14.3.2 (Meta character, escaped character, Set of Character, dan Fungsi Regex pada Library Python)

Meta Character :

. (titik) : satu karakter apapun kecuali baris baru. Sama dengan [^\n]

\d . \D : satu karakter digit/ non digit apapun

\w . \W satu karakter kata/non kata apapun, untuk ASCII

\s . \S : satu karakter spasi/bukan spasi untuk ASCII

Sumber :

<https://personal.ntu.edu.sg/ehchua/programming/howto/Regexe.html>

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\b in" "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

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/WOJTYFA/PrAlpro-Pert15-71231054>

SOAL 1

```
#latihan 15.1
import re
from datetime import datetime

def temukan_dan_konversikan(text):
    date_pattern = r'\d{4} - \d{2} - \d{2}'
    dates = re.findall(date_pattern, text)

    current_date = datetime.now()

    hasil = []
    for date_str in dates:
        date_obj = datetime.strptime(date_str, '%Y-%m-%d')

        format_date = date_obj.strftime('%d-%m-%Y')

        selisih = (current_date - date_obj).days

        hasil.append(f"{date_str} 00:00:00 selidih {selisih} hari")
    return hasil
```

Kita diminta untuk mencari seluruh teks yang berupa tanggal dengan format YYYY-MM-DD dan kemudian seluruh tanggal tersebut diambil dan ditampilkan kembali dalam format DD-MM-YYYY ditambah dengan perhitungan selisih dengan tanggal sekarang dalam hari.

```
✓ import re
  from datetime import datetime
```

Menggunakan re untuk mencari semua tanggal dengan format YYYY – MM – DD didalam teks.

```
✓ def temukan_dan_konversikan(text):
    date_pattern = r'\d{4} - \d{2} - \d{2}'
    dates = re.findall(date_pattern, text)
```

re untuk menemukan data tanggal dengan format YYYY – MM – DD didalam teks.

```
hasil = []
for date_str in dates:
    date_obj = datetime.strptime(date_str, '%Y-%m-%d')
```

Mengkonversi string tanggal menjadi objek 'datetime' agar memudahkan manipulasi tanggal.

```
format_date = date_obj.strftime('%d-%m-%Y')
```

Digunakan untuk mengkonverai menjadi format DD – MM – YYYY, menggunakan 'strftime'

```
selisih = (current_date - date_obj).days
```

Menghitung selisih tanggal dengan menggunakan seperti program diatas.

```
hasil.append(f"{date_str} 00:00:00 selidih {selisih} hari")
```

Menyimoan hasil dengan format yang diinginkan setelah itu menampilkannya.

SOAL 2

```
#latihan 15.2
import re
import random
import string

def generate_pass(length = 8):
    karakter = string.ascii_letters + string.digits

    password = ''.join(random.choice(karakter) for i in range(length))
    return password

def ekstrak_username_dan_generate_pass(text):
    email_pattern = r'\b[A-Za-z0-9. %+-] + @[A-Za-z0-9.-] + \.[A-Z|a-z]{2,}\b'
    emails = re.findall(email_pattern, text)

    hasil = []
    for email in emails:
        username = email.split('@')[0]
        password = generate_pass()
        hasil.append(f"{email} username: {username}, passeprd: {password}")
    return hasil
```

Kita diminta untuk mencari seluruh teks yang berupa email dan kemudian ambil semua username dari email tersebut untuk digenerate password random 8 karakter yang terdiri dari angka dan huruf.

```
def generate_pass(length = 8):
    karakter = string.ascii_letters + string.digits

    password = ''.join(random.choice(karakter) for i in range(length))
    return password
```

Membuat password dengan panjang 8 karakter, karakter dapat berupa angka / huruf. Password yang dibuat akan bersifar random.

```
def ekstrak_username_dan_generate_pass(text):
    email_pattern = r'\b[A-Za-z0-9. %+-] + @[A-Za-z0-9.-] + \.[A-Z|a-z]{2,}\b'
    emails = re.findall(email_pattern, text)
```

re digunakan untuk mencari semua teks email didalam sebuah teks. Mengambil email dengan memisahkan bagian sebelum @.

```
username = email.split('@')[0]
```

Mengeluarkan nama dari email.

```
password = generate_pass()
```

Membuat random password dengan panjang 8 karakter yang terdiri dari angka dan huruf.

```
hasil.append(f"{email} username: {username}, passeprd: {password}")
```

Menyimpan hasil dalam bentuk yang diinginkan lalu ditampilkan.