



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

|                           |                                       |
|---------------------------|---------------------------------------|
| <b>NIM</b>                | <b>71230976</b>                       |
| <b>Nama Lengkap</b>       | <b>Galih Pramana Chandra Prasetya</b> |
| <b>Minggu ke / Materi</b> | <b>14/ RegEx</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%)

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

Regex, kependekan dari Regular Expression, adalah sebuah pola yang dituliskan menggunakan kumpulan karakter untuk menemukan kecocokan dengan pola yang sama dalam string lain. Sederhananya, Regex membantu kita dalam mengolah string dengan lebih efektif dan fleksibel.

Manfaat utama Regex:

- **Pencarian String:** Menemukan string yang sesuai dengan pola tertentu, seperti kata kunci dalam dokumen atau memfilter data.
- **Penggantian String:** Mengubah string yang sesuai dengan pola tertentu, seperti mengganti nama variabel dalam dokumen.
- **Penghapusan String:** Menghapus string yang sesuai dengan pola tertentu.

Contoh Penggunaan Regex:

- **Validasi Input:** Memastikan format input pengguna sesuai dengan pola yang ditentukan, seperti email, nomor telepon, atau tanggal.
- **Ekstraksi Data:** Mengambil informasi spesifik dari teks, seperti alamat, nomor telepon, atau tanggal lahir dari dokumen.
- **Pemformatan Teks:** Mengubah format teks secara otomatis, seperti mengubah huruf besar menjadi kecil, atau menambahkan spasi antar kata.

Contoh penggunaan `search()` pada regex:

```
import re
handle=open('teks1.txt')
count = 0
for line in handle:
    line=line.rstrip()
    if re.search('halo', line):
        count += 1
print(line)
print("Count: ",count)
```

Output:

```
halo guys hai aku bla
Count: 1
```

Meta Character, Escaped Character, Set of Character, dan Fungsi Regex pada Library Python

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   |

*Diambil dari modul*

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" "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"        |

*Diambil dari modul*

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                                 |

*Diambil dari modul*

Tabel 14.4: Fungsi Regex pada Python

| 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        |

*Diambil dari modul*

**Penggunaan findall:**

```
import re

txt = "hello // planet"

x = re.findall("h.*o", txt)
print(x)
```

Output:

```
['hello']
```

**Penggunaan search:**

```
import re

txt = "saya berkuliah di universitas kristen duta wacana"

x = re.search("\s", txt)
y = re.search("saya", txt)

print("Spasi ditemukan di:", x.start())
print(y)
```

Output:

```
Spasi ditemukan di: 4
<re.Match object; span=(0, 4), match='saya'>
```

## 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.

### SOAL 1

Sc:

```
#Latihan 14.1
import re
from datetime import datetime

sekarang = datetime.now()

tahun = sekarang.year
bulan = sekarang.month
hari = sekarang.day
b = (f'{hari}-{bulan}-{tahun}')
grup = re.compile(r'\d{4}-\d{2}-\d{2}')
txt = open("n.txt", "r")
txt2 = txt.read()
a = []
findd = grup.findall(txt2)

for x in findd:
    v = x.split('-')
    c = (v[2]+"-"+v[1]+"-"+v[0])
    a.append(c)
date_format = "%d-%m-%Y"
b = datetime.strptime(b, date_format)
for y in a:
    c = datetime.strptime(y, date_format)
    d = b - c
    print(f'{y} 00:00:00 selisih {d.days} hari')
```

output:

```
17-08-1945 00:00:00 selisih 28781 hari
11-11-1785 00:00:00 selisih 87133 hari
08-06-1783 00:00:00 selisih 88020 hari
02-05-1889 00:00:00 selisih 49341 hari
```

Penjelasan:

- Mengimpor modul 're' untuk regex dan modul 'datetime' untuk tanggal dan waktu.
- Mengambil tanggal dan waktu saat ini menggunakan 'datetime.now()'.
- Mengambil tahun, bulan, dan hari dari tanggal sekarang dan menyimpannya dalam variabel 'tahun', 'bulan', dan 'hari'.
- Membuat objek regular expression dengan pola `\d{4}-\d{2}-\d{2}` yang cocok dengan tanggal dalam format YYYY-MM-DD dan menyimpannya dalam variabel grup.
- Membuka file teks "n.txt" dan membaca isinya.
- Menggunakan metode **findall()** dari objek regex untuk menemukan semua tanggal dalam teks.
- Mengubah format string b menjadi objek datetime menggunakan `datetime.strptime()`.
- Mengubah setiap tanggal dalam list a menjadi objek datetime.
- Menghitung selisih antara tanggal sekarang dan setiap tanggal dalam list a menggunakan operasi pengurangan.
- Mencetak tanggal, waktu, dan selisih hari untuk setiap tanggal dalam list a.

## SOAL 2

Sc:

```
#latman 14.2
import re
import random
import string

with open('x.txt', 'r') as file:
    teks = file.read()

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

for email in email_list:
    username = email.split('@')[0]
    password = ''.join(random.choices(string.ascii_letters +
string.digits, k=8))
    print(f"{email} username: {username} , password: {password}")
```

output:

```
anton@gmail.com username: anton , password: 3nJ6fI2i
budi@gmail.co.id username: budi , password: 4n76oavq
slamet@getnada.com username: slamet , password: zFGqoWQn
matahari@tokopedia.com username: matahari , password: wCEBZ3vm
```

Penjelasan:

- Mengimpor modul **re** untuk bekerja dengan regular expressions, **random** untuk menghasilkan kata sandi acak, dan **string** untuk mendapatkan kumpulan karakter ASCII.
- Menggunakan pernyataan **with open('x.txt', 'r') as file:** untuk membuka file teks "x.txt" untuk dibaca, dan kemudian membaca seluruh isinya dan menyimpannya dalam variabel **teks**.
- Menggunakan regular expression **email\_regex** untuk mencocokkan alamat email dalam teks. Regular expression ini mencocokkan alamat email yang sesuai dengan format standar.
- Menggunakan **re.findall()** untuk mencari semua alamat email yang cocok dalam teks dan menyimpannya dalam **list email\_list**.
- Melakukan iterasi melalui setiap alamat email dalam **list email\_list**.
- Memisahkan alamat email untuk mendapatkan nama pengguna (**username**) dengan membaginya berdasarkan tanda "@" menggunakan metode **split('@')**.



- Menghasilkan kata sandi acak sepanjang 8 karakter dengan menggunakan `random.choices()` untuk memilih karakter dari kumpulan karakter ASCII huruf besar, huruf kecil, dan angka. Kata sandi kemudian disatukan menggunakan `join()` dan disimpan dalam variabel `password`.
- Mencetak alamat email, nama pengguna, dan kata sandi yang dihasilkan.

Link github: <https://github.com/GalihPramana/Praktikum-Alpro-71230976.git>