

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
Minggu ke / Materi	14 / Regex

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 1

PENGANTAR REGEX

Teknik pengolahan *string* yang mudah dan cepat adalah menggunakan *regular expression*. *Regular expression* adalah kumpulan karakter yang digunakan untuk menemukan pola (*pattern*) yang sama dengan pola regex di dalam *string* lain yang ingin dicari (mencocokkan teks dengan karakter dalam alfabet ASCII atau latin, serta karakter Unicode).

Regex membantu dalam pencarian *string* dengan pola tertentu, mengganti string dengan pola tertentu, dan menghapus *string* dengan pola tertentu. Penggunaan regex membantu dalam parsing string, sehingga tidak hanya menggunakan *split()* dan *find()*.

Pemanfaatan Python Regex.

1. **Pencarian** : Python RegEx dapat melakukan pencarian terhadap suatu keberadaan suatu kata kunci (Keyword) atau pola dalam suatu string yang diberikan. Seperti contohnya yaitu Python RegEx dapat mencari semua email dalam sebuah dokumen string.
2. **Pencocokan dan Ekstraksi** : Python RegEx juga dapat melakukan pengecekan (mencocokkan) berdasarkan pola tertentu dan mengekstrak informasi yang sesuai dengan pola tersebut. Seperti contohnya, Python RegEx dapat mengekstrak nomor telepon dari sebuah string yang kompleks.
3. **Validasi Input** : Python RegEx biasanya juga dapat digunakan untuk melakukan pemeriksaan terhadap sebuah input pengguna yang diberikan untuk dilakukan pengecekan terhadap kesesuaian dengan format yang diinginkan, seperti contohnya yaitu Python RegEx dapat melakukan pengecekan alamat email atau nomor telepon yang bersifat valid.
4. **Penggantian** : Python RegEx tentunya bisa mengganti sebuah string yang sesuai dengan pola tertentu berdasarkan teks lain yang telah diberikan dan ditetapkan. Contohnya, Python RegEx dapat mengganti semua kata “merah” dengan kata “biru” dalam sebuah string.

source: <https://medium.com/@myskill.id/regular-expression-python-84a2468f5d88>

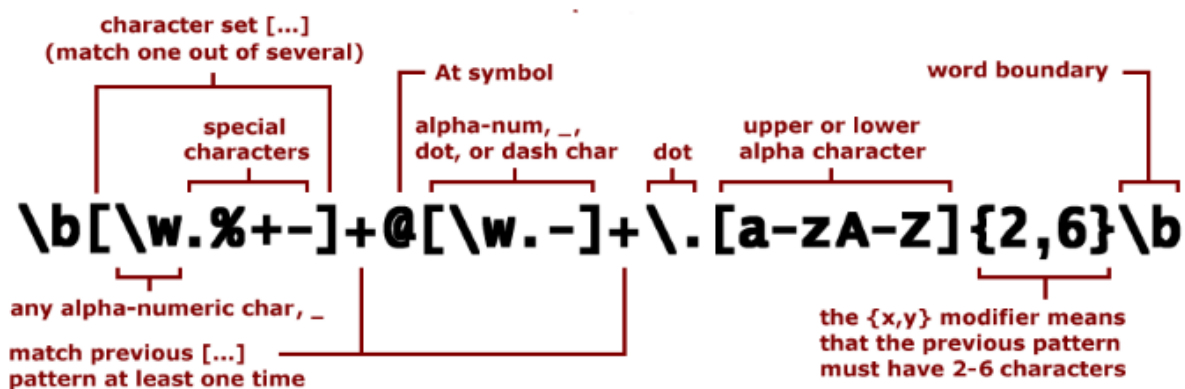
Tidak semua bahasa pemrograman mendukung *regular expression library*. Python merupakan salah satu bahasa yang mendukung *library* regex dengan cara *import re*. Salah satu fungsi yang paling mudah digunakan dari *library re* adalah *search()*.

Memanggil *library* Regex.

```
import re
```

Konsep pada Regex.

1. **Pola (*Pattern*)** : Pola merupakan bentuk dari representasi terhadap pola pencarian yang ingin dilakukan pencarian terhadap suatu string. Contohnya, jika ingin mencari semua angka dalam string, kita dapat menggunakan pola `'\d+'`, yang mencocokkan satu atau lebih digit.
2. **Fungsi `'re.compile()'`** : Kita dapat mengompilasi pola RegEx menggunakan fungsi `'re.compile()'` untuk meningkatkan efisiensi jika kita ingin menggunakannya secara berulang-ulang.
3. **Mencocokkan (*Matching*)** : Operasi paling dasar dalam RegEx adalah mencocokkan untuk menentukan apakah pola cocok dengan string yang diberikan. Hal ini dilakukan dengan menggunakan metode `'match()'` atau `'search()'`.
4. **Subskrip dan Grup (*Capture Group*)** : Python RegEx dapat menggunakan subskrip seperti `'()'` untuk menangkap bagian-bagian tertentu dari string yang cocok dengan pola. Hal ini memungkinkan kita untuk mengambil informasi yang spesifik dari string yang cocok.
5. **Metakarakter (*Metacharacters*)** : RegEx memiliki beberapa metakarakter seperti `'.'` (cocokkan karakter apa pun) dan `'+'` (cocokkan satu atau lebih) yang berfungsi untuk menyusun pola yang lebih rumit.
6. **Karakter Escape (*Escape Character*)** : Karakter seperti `'.'` dapat dihindari menjadi metakarakter dengan menggunakannya secara bersama karakter escape `'\'`.
7. **Modifier** : Kita bisa menambahkan modifier seperti `'i'` (ignore case) dan `'s'` (dot matches all) kepada sebuah pola untuk mengubah cara pencocokan dilakukan.
8. **Fungsi `'re.sub()'`** : RegEx tentunya tidak hanya digunakan untuk pencocokan, tetapi juga bisa digunakan untuk penggantian. Dengan menggunakan `'re.sub()'`, kita dapat menggantikan string yang cocok dengan pola dengan string yang lain.



Parse: username@domain.TLD (top level domain)

Gambar 1. Regex Character

(source: <https://www.novixys.com/blog/python-regular-expression-tutorial-part-1/>)

Karakter Khusus:

1. `.` : Cocokkan dengan karakter apa pun kecuali karakter baris baru.
2. `*` : Cocokkan dengan nol atau lebih kemunculan pola sebelumnya.
3. `+` : Cocokkan dengan satu atau lebih kemunculan pola sebelumnya.
4. `?` : Cocokkan dengan nol atau satu kemunculan pola sebelumnya.
5. `[]` : Cocokkan dengan salah satu karakter dalam tanda kurung siku.
6. `^` : Cocokkan dengan awal string.
7. `$` : Cocokkan dengan akhir string.
8. `\` : Melarikan karakter khusus. Misalnya, `\\. Untuk mencocokkan titik (.) secara harfiah.`

MATERI 2

FUNGSI REGEX

RE.FINDALL()

Kecocokan dikembalikan dalam bentuk *list*.

```
pola = r"\d+"
teks = "Sekarang umurmu sudah 21, ya! Sampai jumpa di tahun 2025"

angka = re.findall(pola, teks)
print("Angka:", angka)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Angka: ['21', '2025']
```

RE.FINDITER()

```
pola = r"\d+"
teks = "Sekarang umurmu sudah 21, ya! Sampai jumpa di tahun 2025"

angka = re.finditer(pola, teks)
for i in angka:
    print("Angka:", i.group())
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Angka: 21
Angka: 2025
```

RE.SUB()

```
pola = r"\d+"
pengganti = "Sampai jumpa!"
teks = "Jangan lupa telepon, ya! 56486032"

hasil = re.sub(pola, pengganti, teks)
print("Kata-kata terakhir:", hasil)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Kata-kata terakhir: Jangan lupa telepon, ya! Sampai jumpa!
```

RE.FULLMATCH()

```
pola = r"[a-zA-Z]+"
teks = "Hello"

hasil = re.fullmatch(pola, teks)
if hasil:
    print("sesuai:", hasil.group())
else:
    print("Tidak ada yang sama")
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
sesuai: Hello
```

RE.SPLIT()

```
pola = r"\s+"
teks = "Aku melihat Surga terbuka dan Anak Allah duduk di
sebelah kanan Bapa."

hasil = re.split(pola, teks)
print("Kata terpisah:", hasil)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Kata terpisah: ['Aku', 'melihat', 'Surga', 'terbuka', 'dan', 'Anak', 'Allah', 'duduk', 'di', 'sebelah', 'kanan', 'Bapa.']
```

RE.COMPILE()

```
pola = r"\d+"
kompilasi_pola = re.compile(pola)

teks1 = "Pola yang sama dalam 2 teks"
teks2 = "Hello angka 3126, selamat pagi!"

sama1 = kompilasi_pola.findall(teks1)
sama2 = kompilasi_pola.findall(teks2)

print("Kesamaan pola teks1:", sama1)
print("Kesamaan pola teks2:", sama2)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Kesamaan pola teks1: ['2']
Kesamaan pola teks2: ['3126']
```

RE.ESCAPE()

```
teks = "Menambahkan simbol \ pada kata sebelum spasi dalam teks"
tambah_escaped = re.escape(teks)
print("Escaped teks:", tambah_escaped)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_14/Modul.py
Escaped teks: Menambahkan\ simbol\ \\ pada\ kata\ sesudah\ spasi\ dalam\ teks
```

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

Latihan 14.1 Anda 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.

Contoh:

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).

Hasil:

1945-08-17 00:00:00 selisih 27209 hari
1785-11-11 00:00:00 selisih 85561 hari
1783-06-08 00:00:00 selisih 86448 hari
1889-05-02 00:00:00 selisih 47769 hari

Pengerjaan soal dilakukan dengan menggunakan *library* regex dan *library* datetime, kemudian membuat pola tanggal dalam regex yang diinginkan berupa 4 digit (angka) untuk tahun, 2 digit untuk bulan, dan 2 digit untuk tanggal. Menggunakan fungsi regex *findall* untuk mencari pola tanggal itu di dalam teks yang ada. Lalu mengambil tanggal saat ini dengan fungsi *datetime.now()*. Dibuat *list* kosong untuk menyimpan tanggal dan selisih yang akan ditampilkan. Untuk setiap pola tanggal pada teks diambil dengan *datetime.strptime()*, dilakukan perhitungan selisih hari dari tanggal sekarang dengan tanggal pada teks. Format YYYY-MM-DD diubah ke DD-MM-YYYY dengan *.strftime()*.

SOURCE CODE:

```
import re
from datetime import datetime

def selisih_tanggal(teks):
    tanggal = r"\d{4}-\d{2}-\d{2}"
    cari_tanggal = re.findall(tanggal, teks)
    tanggal_sekarang = datetime.now()

    hasil = []
    for i in cari_tanggal:
        ambil_tanggal = datetime.strptime(i, "%Y-%m-%d")
        selisih_hari = (tanggal_sekarang - ambil_tanggal).days

        ubah_bentuk = ambil_tanggal.strftime("%d-%m-%Y")
        # print(ubah_bentuk)
        hasil.append(f"{i} selisih {selisih_hari} hari")
    return hasil
```

```

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).
"""

hasil_tanggal = selisih_tanggal(teks)
for i in hasil_tanggal:
    print(i)

```

OUTPUT:

```

PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe "d:/71200645_PrakAlpro_14/Latihan
14.1.py"
1945-08-17 selisih 28780 hari
1785-11-11 selisih 87132 hari
1783-06-08 selisih 88019 hari
1889-05-02 selisih 49340 hari
PS D:\71200645_PrakAlpro_14>

```

SOAL 2

Latihan 14.2 Anda 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.

Contoh:

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

Hasil:

anton@mail.com username: anton , password: 8u78A2UD
 budi@gmail.co.id username: budi , password: bdP066Ld
 slamet@getnada.com username: slamet , password: Ab1FiHXb
 matahari@tokopedia.com username: matahari , password: 5KYyaP6

Pengerjaan soal dilakukan dengan memanggil *library* `regex`, `random`, dan `string`. Dengan fungsi `buat_password` diberi parameter panjang password yang akan dibuat secara *random* adalah 8. Variabel karakter adalah kombinasi dari semua huruf (huruf besar + huruf kecil + angka). Sebanyak *length* dilakukan *looping* untuk memilih acak satu karakter dan menambahkannya ke variabel password. Pada fungsi `ambil_user_dan_beri_password` diberi parameter teks yang berisi daftar alamat *email* beserta nama penggunanya. Untuk mengambil alamat *email* dibuat pola *email* dengan `regex`. Kemudian menggunakan fungsi `findall()` untuk mencari teks yang sesuai dengan pola tersebut. Dibuat *list* kosong untuk menyimpan bagian *email* yang ingin disimpan, mengambil nama pengguna sebelum simbol “@”, lalu memanggil fungsi `buat_password`, kemudian *username* dan *password* disimpan dalam *list*.

SOURCE CODE:

```
import re
import random
import string

def buat_password(length = 8):
    karakter = string.ascii_letters + string.digits
    password = ""
    for i in range(length):
        password += random.choice(karakter)
    return password

def ambil_user_dan_beri_password(teks):
    pola_email = r"[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}"
    email = re.findall(pola_email, teks)

    hasil = []
    for i in email:
        username = i.split("@")[0]
        password = buat_password()
        hasil.append(f"{i} username: {username}, password: {password}")
    return hasil

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
"""

hasil = ambil_user_dan_beri_password(teks)
for i in hasil:
    print(i)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_14> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe "d:/71200645_PrakAlpro_14/Latihan
14.2.py"
anton@mail.com username: anton, password: T04ryTM9
budi@gmail.co.id username: budi, password: L3K4CYpB
slamet@getnada.com username: slamet, password: fb9tx3C9
matahari@tokopedia.com username: matahari, password: FUlebj6U
PS D:\71200645_PrakAlpro_14>
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_14