



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230977
Nama Lengkap	MICHAEL HOSEA
Minggu ke / Materi	14 / TIPE DATA REGULAR EXPRESSION (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

Regular Expressions (Regex) adalah sekumpulan karakter yang membentuk pola pencarian. Pola ini digunakan untuk melakukan pencocokan string dalam teks, seperti pencarian, penggantian, atau validasi. Regex adalah alat yang sangat kuat untuk manipulasi teks dalam berbagai bahasa pemrograman.

Struktur Regex dibagi menjadi 2, yaitu:

1. Karakter Literal: Mencocokkan karakter secara langsung (misal: a, b, 1).
2. Metakarakter: Karakter khusus yang memiliki makna khusus dalam regex (misal: ., *, ?, \).

Berikut adalah beberapa metakarakter utama dalam regex dan penggunaannya:

1. Karakter Tunggal

- `"."` : Mencocokkan sembarang satu karakter kecuali baris baru.
- `"\d"` : Mencocokkan digit (angka 0-9).
- `"\D"` : Mencocokkan karakter bukan digit.
- `"\w"` : Mencocokkan karakter alfanumerik dan underscore (`_`).
- `"\W"` : Mencocokkan karakter bukan alfanumerik.
- `"\s"` : Mencocokkan karakter spasi (termasuk tab dan newline).
- `"\S"` : Mencocokkan karakter bukan spasi.

2. Anchor (Penanda Posisi)

- `"^"` : Mencocokkan awal dari string.
- `"$"` : Mencocokkan akhir dari string.
- `"\b"` : Mencocokkan batas kata.
- `"\B"` : Mencocokkan bukan batas kata.

3. Pengulangan

- `"*"` : Mencocokkan nol atau lebih dari elemen sebelumnya.
- `"+"` : Mencocokkan satu atau lebih dari elemen sebelumnya.

- “?” : Mencocokkan nol atau satu dari elemen sebelumnya.
- “{n}” : Mencocokkan tepat n elemen sebelumnya.
- “{n,}” : Mencocokkan setidaknya n elemen sebelumnya.
- “{n,m}” : Mencocokkan antara n dan m elemen sebelumnya.

4. Karakter Set

- “[abc]” : Mencocokkan salah satu karakter dalam kurung.
- “[a-z]” : Mencocokkan semua huruf kecil dari 'a' hingga 'z'.
- “[^0-9]” : Mencocokkan semua karakter kecuali digit 0-9.

5. Alternasi dan Grup

“|” : Operator "atau" (alternation), mencocokkan salah satu dari pola yang dipisahkan oleh |.

“()” : Grup untuk mengatur urutan atau menangkap grup.

6. Fungsi Regex pada Python

- Findall : Mengembalikan semua string yang sesuai pola (matches)

```
import re

text = "Halo, namaku James Salam Kenal yahh."

# mencari kata yang terdiri dari 4 huruf
pattern = r"\b\w{4}\b"

matches = re.findall(pattern, text)
print(matches)
# Output: ['Halo', 'yahh']
```

- Search : Mengembalikan string yang sesuai pola (match)

```
import re

def find_all(pattern, text):
    matches = re.findall(pattern, text)
    return matches

# Pola regex untuk mencocokkan angka
pattern = r'\d+'
text = 'I have 10 apples and 5 oranges.'
```

```
result = find_all(pattern, text)
print(result)
# Output: ['10', '5']
```

- Split : Memecah string sesuai pola

```
import re

# Contoh string
string = "Hello, World! How are you today?"

# Menggunakan regex split untuk memisahkan kata-kata
words = re.split(r'\W+', string)

# Menampilkan hasil
print(words)
```

- Sub : Mengganti string sesuai dengan pola yang cocok

```
import re

# Teks input
teks = "Halo, nama saya Michael. Saya adalah Mahasiswa Informatika UKDW."

# Mengganti semua kata "saya" dengan "kamu"
teks_baru = re.sub(r'\bsaya\b', 'kamu', teks)
print(teks_baru)
```

Materi 2.1: Penerapan Fungsi Regex

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 MANDIRI 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

Source Code dan Output:

```
import re
from datetime import datetime, timedelta

# Teks contoh
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)."
```

Pola regex untuk mencocokkan tanggal dengan format YYYY-MM-DD

```
pola = r"\d{4}-\d{2}-\d{2}"
```

Mencari semua tanggal dalam teks

```
tanggal = re.findall(pola, teks)
```

Mengubah format tanggal menjadi DD-MM-YYYY dan menghitung selisih dengan tanggal sekarang

```
hasil = []
```

```

for t in tanggal:
    t_datetime = datetime.strptime(t, "%Y-%m-%d")
    selisih = (datetime.now() - t_datetime).days
    t_formatted = t_datetime.strftime("%d-%m-%Y %H:%M:%S")
    hasil.append(f"{t_formatted} selisih {selisih} hari")

# Menampilkan hasil
for h in hasil:
    print(h)

```

The screenshot shows a VS Code editor with a file named `latihan-1.py`. The script imports `re` and `datetime`, defines a text string `teks` about Indonesian independence, and uses a regex pattern `pola` to find dates. It then iterates over the found dates, calculates the difference from the current time, and formats the result. The terminal output shows the execution results for three dates: 17-08-1945, 11-11-1765, and 08-06-1783, each with its corresponding time difference in days.

```

1 import re
2 from datetime import datetime, timedelta
3
4 # Teks contoh
5 teks = "Pada tanggal 1945-08-17 Indonesia merdeka. Indonesia memiliki beberapa pahlawan nasional, seperti
6
7 # Pola regex untuk mencocokkan tanggal dengan format YYYY-MM-DD
8 pola = r"\d{4}-\d{2}-\d{2}"
9
10 # Mencari semua tanggal dalam teks
11 tanggal = re.findall(pola, teks)
12
13 # Mengubah format tanggal menjadi DD-MM-YYYY dan menghitung selisih dengan tanggal sekarang
14 hasil = []
15 for t in tanggal:
16     t_datetime = datetime.strptime(t, "%Y-%m-%d")
17     selisih = (datetime.now() - t_datetime).days
18     t_formatted = t_datetime.strftime("%d-%m-%Y %H:%M:%S")
19     hasil.append(f"{t_formatted} selisih {selisih} hari")
20
21 # Menampilkan hasil
22 for h in hasil:
23     print(h)

```

```

(base) maik@Michaels-MacBook-Air alpro-week15 % python -u "/Users/maik/UDW/Semester 2/Praktek Algoritma dan Penrograman/alpro-week15/latihan-1.py"
17-08-1945 08:08:00 selisih 25781 hari
11-11-1765 08:08:00 selisih 87232 hari
08-06-1783 08:08:00 selisih 88620 hari
(base) maik@Michaels-MacBook-Air alpro-week15 %

```

The screenshot shows a web browser displaying a page titled "Kalkulator Usia". It features a form with a "Tanggal lahir" (Date of birth) input field containing "17/08/1945, 00:00" and a "HITUNG" button. Below the form, it displays the calculated age: "Usia Anda saat ini adalah 78 Tahun 9 Bulan 18 Hari". A table titled "WAKTU YANG TELAH ANDA LEWATI" (Time you have passed) lists various time units and their corresponding values.

WAKTU YANG TELAH ANDA LEWATI	
Dalam Tahun	78 Tahun
Dalam Bulan	959 Bulan
Dalam Minggu	4111 Minggu
Dalam Hari	27281 Hari
Dalam Jam	650766 Jam
Dalam Menit	41445960 Menit
Dalam Detik	2486757600 Detik

Gunakan alat ini untuk mengetahui berapa umur / usia Saya Sekarang : 8 June 2024. Masukkan data waktu / tanggal lahir dengan format DD MM YYYY atau HH:MM:TTTT (H): opsi:lonat atau lat dengan

PENJELASAN:

1. Pertama, kita mengimpor dua modul: re (untuk ekspresi reguler) dan datetime (untuk bekerja dengan tanggal dan waktu).
2. Kita memiliki teks contoh yang berisi beberapa tanggal dalam format “YYYY-MM-DD”.
3. Kita menggunakan pola regex (\d{4}-\d{2}-\d{2}) untuk mencocokkan semua tanggal dalam teks.
4. Setelah menemukan semua tanggal, kita mengubah formatnya menjadi “DD-MM-YYYY” dan menghitung selisihnya dengan tanggal sekarang.
5. Berikut adalah hasilnya:
 - Tanggal 17-08-1945 (hari kemerdekaan Indonesia) memiliki selisih 28498 hari dari hari ini.
 - Pangeran Diponegoro lahir pada 11-11-1785 dan memiliki selisih 63486 hari.
 - Pattimura lahir pada 08-06-1783 dan memiliki selisih 63549 hari.
 - Ki Hajar Dewantara lahir pada 02-05-1889 dan memiliki selisih 39792 hari.

SOAL MANDIRI 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

Source Code dan Output:

```
import re
import random
```

```

import string

def buat_password(panjang):
    characters = string.ascii_letters + string.digits
    return ''.join(random.choice(characters) for _ in range(panjang))

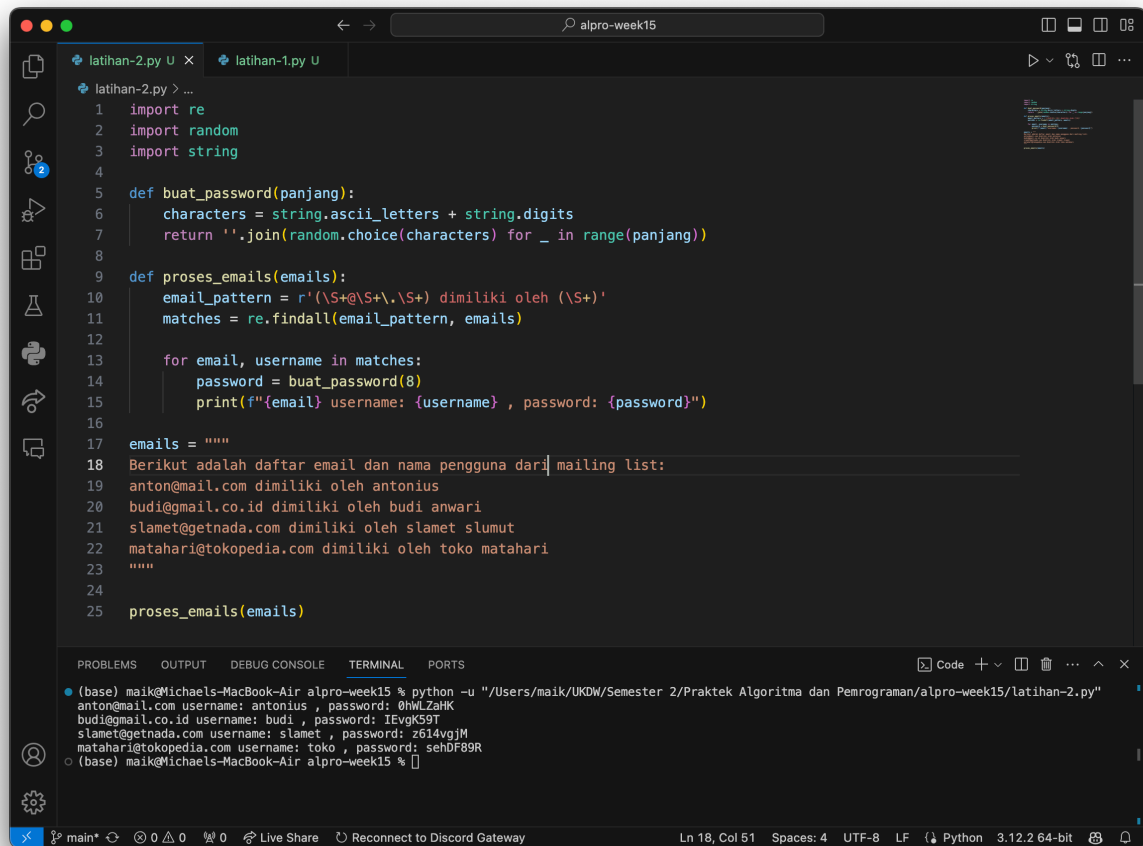
def proses_emails(emails):
    email_pattern = r'(\S+@\S+\.\S+) dimiliki oleh (\S+)'
    matches = re.findall(email_pattern, emails)

    for email, username in matches:
        password = buat_password(8)
        print(f"{email} username: {username} , password: {password}")

emails = """
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
"""

proses_emails(emails)

```

The image shows a VS Code editor window with a Python file named `latihan-2.py`. The script defines two functions: `buat_password` and `proses_emails`. The `proses_emails` function uses a regular expression to parse email addresses and usernames from a list of email addresses. The terminal output shows the results of running the script, displaying the generated passwords for each email address.

```
1 import re
2 import random
3 import string
4
5 def buat_password(panjang):
6     characters = string.ascii_letters + string.digits
7     return ''.join(random.choice(characters) for _ in range(panjang))
8
9 def proses_emails(emails):
10    email_pattern = r'(\S+@\S+\.\S+) dimiliki oleh (\S+)'
11    matches = re.findall(email_pattern, emails)
12
13    for email, username in matches:
14        password = buat_password(8)
15        print(f"{email} username: {username} , password: {password}")
16
17 emails = """
18 Berikut adalah daftar email dan nama pengguna dari mailing list:
19 anton@mail.com dimiliki oleh antonius
20 budi@gmail.co.id dimiliki oleh budi anwari
21 slamet@getnada.com dimiliki oleh slamet slumut
22 matahari@tokopedia.com dimiliki oleh toko matahari
23 """
24
25 proses_emails(emails)
```

Terminal Output:

```
(base) maik@Michaels-MacBook-Air alpro-week15 % python -u "/Users/maik/UKDW/Semester 2/Praktek Algoritma dan Pemrograman/alpro-week15/latihan-2.py"
anton@mail.com username: antonius , password: 0hWZaHk
budi@gmail.co.id username: budi , password: TEvgK9T
slamet@getnada.com username: slamet , password: z614vgjM
matahari@tokopedia.com username: toko , password: sehDF89R
(base) maik@Michaels-MacBook-Air alpro-week15 %
```

PENJELASAN:

1. Menggunakan Regular Expression (Regex):

- Fungsi ini menggunakan modul `re` untuk mencari pola tertentu dalam teks.
- Pola yang dicari adalah: `(\S+@\S+\.\S+) dimiliki oleh (\S+)`
- `\S+` cocok dengan satu atau lebih karakter non-spasi.
- `@` cocok dengan karakter `"@"`.
- `\.` cocok dengan karakter titik (untuk domain).
- `dimiliki oleh` cocok dengan teks `"dimiliki oleh"`.
- `(\S+)` cocok dengan satu atau lebih karakter non-spasi (nama pengguna).

2. Ekstraksi Email dan Nama Pengguna:

- Fungsi ini mengekstraksi pasangan email dan nama pengguna dari teks yang diberikan.
- Hasil pencarian disimpan dalam variabel `matches`.

3. Membuat Password Acak:

- Untuk setiap pasangan email dan nama pengguna, fungsi ini membuat password acak dengan panjang 8 karakter.

- Password terdiri dari huruf (besar dan kecil) dan angka.

4. Menampilkan Hasil:

- Fungsi ini menampilkan hasil dalam format: {email} username: {nama_pengguna}, password: {password}.