



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	<71241079>
Nama Lengkap	<Natalie Neysa Jessica Soesanto>
Minggu ke / Materi	08 / Pengolahan String dan Regular Expression

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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 String

String merupakan rangkaian karakter yang digabungkan menjadi satu kesatuan dan digunakan dalam program komputer untuk menyimpan kalimat, baik panjang maupun pendek. Setiap karakter diwakili oleh kode ASCII. Tidak semua bahasa pemrograman memiliki tipe data string, misalnya bahasa C. String pada dasarnya adalah kumpulan karakter atau array/list of character, dan termasuk tipe data kompleks karena menyimpan lebih dari satu nilai.

MATERI 2

Pengaksesan String dan Manipulasi String

Source Code :

```
1  namesaya = "Antonius Rachmat C"
2  temansaya1 = "Yuan Lukito"
3  temansaya2 = 'Laurentius Kuncoro'
4  temansaya3 = "Matahari" + 'Bakti'
5
6  print(temansaya3)
7  print(namesaya[0])
8  print(namesaya[9])
9  print(temansaya1[1])
10
11 huruf = temansaya2[0]
12 print(huruf)
```

Output :

```
MatahariBakti
A
R
U
L
```

String pertama kali dibuat dengan mendeklarasikan variabel dan langsung mengisinya dengan data. String dapat diakses sebagai satu kesatuan lewat nama variabel, atau per karakter lewat indeks. Indeks string dimulai dari 0 dan harus berupa bilangan bulat. Pada memori komputer, string disimpan berurutan dalam bentuk list karakter, seperti contoh string "banana" yang setiap hurufnya memiliki indeks dari nol.

b	a	n	a	n	a
[0]	[1]	[2]	[3]	[4]	[5]

MATERI 3

Operator dan Metode String

Pada String, kita bisa mengecek apakah suatu kalimat merupakan substring dari suatu kalimat dengan menggunakan operator in. Hasil dari operator ini adalah True / False.

Source Code :

```
1 kalimat = "saya mau makan"
2 data = "saya"
3 print(data in kalimat)
4 print("mau" in kalimat)
5 print("dia" in kalimat)
```

Output :

```
True
True
False
```

Pada String juga dapat dilakukan perbandingan (comparison) yang juga menghasilkan True atau False.

Source Code :

```
1 if "saya" > "dia":
2     print("Ya")
3 else:
4     print("Tidak")
5
6 if "dua" == "dua":
7     print("Sama")
```

Output :

```
Ya
Sama
```

Fungsi Len

Kita dapat menggunakan fungsi len(<string>) untuk mengetahui jumlah karakter dalam sebuah string. Untuk mengambil huruf terakhir, gunakan indeks len(<string>) – 1 karena penomoran indeks dimulai dari 0.

Source Code:

```
1 kalimat = "universitas kristen duta wacana yogyakarta"
2 print(len(kalimat))
3
4 terakhir = kalimat[len(kalimat)-1]
5 print(terakhir)
6
7 terakhir_versi2 = kalimat[-1]
8 print(terakhir_versi2)
9 terakhir2 = kalimat[-2]
10 print(terakhir2)
```

Output :

```
42
a
a
o t
```

Traversing String

Untuk dapat menampilkan string dengan cara ditampilkan huruf demi huruf adalah dengan menggunakan loop yang dilakukan per huruf dengan 2 cara :

- Dilakukan dengan akses terhadap indeks

Source Code :

```
1 kalimat = "indonesia jaya"
2 i = 0
3 while i < len(kalimat):
4     print(kalimat[i],end='')
5     i += 1
```

Output :

```
indonesia jaya
```

- Dilakukan tanpa akses terhadap indeks secara otomatis

Source Code :

```
1 kalimat = "indonesia jaya"
2 for kal in kalimat:
3     print(kal,end='')
```

Output :

```
indonesia jaya
```

String Slice

String slice digunakan untuk menampilkan substring pada sebuah string menggunakan indeks dari awal tertentu sampai akhir-1 tertentu, dengan format <string>[awal:akhir]. Indeks awal dimulai dari 0, dan bagian awal atau akhir bisa dikosongkan.

Source Code :

```
1 kalimat = "cerita rakyat"
2 awal = 0
3 akhir = 6
4 print(kalimat[awal:akhir])
5 print(kalimat[7:len(kalimat)])
6 print(kalimat[:5])
7 print(kalimat[5:])
8 print(kalimat[:])
```

Output :

```
• cerita
o rakyat
cerit
a rakyat
cerita rakyat
```

String merupakan data yang bersifat immutable! Immutable adalah bahwa data tersebut tidak bisa diubah saat program berjalan, hanya bisa diinisialisasi saja. Contoh :

Source Code :

```
1  kalimat = "satu"
2  kalimat[0] = "batu"
```

Output :

```
Traceback (most recent call last):
  File "c:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\S
  Regex\contoh8.py", line 2, in <module>
    kalimat[0] = "batu"
    ~~~~~^~~~~
TypeError: 'str' object does not support item assignment
```

Agar bisa diubah, maka harus disimpan dalam variabel yang berbeda.

Source Code :

```
1  kalimat = "satu"
2  kalimat_baru = kalimat[0] + "alah"
3  print(kalimat_baru)
```

Output :

salah

Berikut adalah beberapa method String yang sering digunakan :

Nama Method	Kegunaan	Penggunaan
capitalize()	untuk mengubah string menjadi huruf besar	string.capitalize()
count()	menghitung jumlah substring yang muncul dari sebuah string	string.count()
endwith()	mengetahui apakah suatu string diakhiri dengan string yang diinputkan	string.endswith()
startswith()	mengetahui apakah suatu string diawali dengan string yang diinputkan	string.startswith()
find()	mengembalikan indeks pertama string jika ditemukan string yang dicari	string.find()
islower() dan isupper()	mengembalikan True jika string adalah huruf kecil / huruf besar	string.islower() dan string.isupper()
isdigit()	mengembalikan True jika string adalah digit (angka)	string.isdigit()
strip()	menghapus semua whitespace yang ada di depan dan di akhir string	string.strip()
split()	memecah string menjadi token-token berdasarkan pemisah, misalnya berdasarkan spasi	string.split()

Semua fungsi/method di atas yang mengembalikan string, mengembalikan string baru, tidak mengubah yang aslinya, karena string bersifat immutable.

Operator * dan + pada String

Di Python, operator + bisa digunakan untuk menggabungkan dua string, sedangkan operator * bisa memperbanyak string sesuai jumlah perkalian.

Source Code :

```
1  kata1 = "saya"
2  kata2 = "makan"
3  kata3 = kata1 + " " + kata2
4  print(kata3)
5  kata4 = "ulang"
6  print(kata4 * 4)
7  kata4 = "ulang "
8  print(kata4 * 2)
```

Output :

```
• saya makan  
ulangulangulang  
ulang ulang
```

MATERI 4

Parsing String

Parsing string adalah cara menelusuri string bagian demi bagian untuk mendapatkan/menemukan/mengubah bagian string yang diinginkan. Contohnya, dari kalimat “Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka”, kita ingin mengambil tanggal, bulan, dan tahun, lalu menyusunnya menjadi 08/17/1945.

Caranya :

- Pisahkan string berdasarkan spasi menjadi beberapa bagian (token)
- Cari token yang diawali angka, lalu pecah lagi berdasarkan tanda ‘-’
- Susun kembali hasilnya sesuai format yang diinginkan

Source Code :

```
1 kalimat = "Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka"  
2  
3 hasil = kalimat.split(" ")  
4  
5 for kal in hasil:  
6     if kal[0].isdigit():  
7         hasil2 = kal.split("-")  
8         print(hasil2[1]+"/"+hasil2[0]+"/"+hasil2[2])
```

Output :

```
08/17/1945
```

MATERI 5

Pengantar Regex

Pada bab String, kita telah mempelajari teknik dasar pengolahan string, namun metode standar sering terasa sulit. Untuk memudahkan, kita dapat menggunakan regular expression (regex), yaitu pola karakter yang digunakan untuk mencari, mengganti, atau menghapus bagian string yang sesuai pola tertentu. Regex sangat efektif untuk pencarian dan ekstraksi data, meskipun polanya cukup kompleks. Tidak semua bahasa mendukung regex tetapi Python mendukungnya melalui library re, dengan fungsi sederhana seperti search().

Dengan menggunakan file mbox-short.txt, kita akan mencoba menampilkan semua string pada file tersebut yang mengandung pola “From: ” menggunakan regex.

Contoh Source Code :

```

1 import re
2 handle = open('mbox-short.txt')
3 count = 0
4 for line in handle:
5     line = line.rstrip()
6     if re.search('From:', line):
7         count += 1
8         print(line)
9 print("Count:", count)

```

Disini, `re.search('From:', line)` sebenarnya dapat digantikan dengan metode `find()` biasa, karena belum menggunakan regex penuh.

Jika ingin mencari baris yang diawali dengan "From: ", gunakan symbol `^` di pola regex :

```

1 if re.search('^From:', line):

```

atau cukup dengan `startswith('From:')` untuk hasil yang sama.

MATERI 6

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

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	(sayaikamu)	saya atau kamu sebagai satu kesatuan
	atau	"\{d s"	1 digit atau 1 spasi

Escaped Character pada Regex :

Special Characters	Kegunaan	Contoh
\b	Digunakan untuk mengetahui apakah suatu pola berada di awal kata atau akhir kata	"R\b" "Rainb"
\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	"EndZ"

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

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.

SOAL 1

Source Code :

```
1 def anagram(kata1, kata2):
2
3     kata1 = kata1.lower().replace(" ", "")
4     kata2 = kata2.lower().replace(" ", "")
5
6
7     if len(kata1) != len(kata2):
8         return False
9
10
11     return sorted(kata1) == sorted(kata2)
12
13
14 print(anagram("mata", "atma"))
15 print(anagram("mata", "maat"))
16 print(anagram("mata", "taam"))
17 print(anagram("mata", "tama"))
```

Output :

```
True
True
True
True
```

Penjelasan : Program ini memeriksa apakah dua kata adalah anagram. Fungsi “anagram” yang berguna untuk membandingkan urutan huruf. Jika urutan sama, keduanya adalah anagram. Jika tidak, bukan anagram.

SOAL 2

Source Code :

```
1 import string
2 def menghitung_frekuensi(kalimat, kata):
3
4     kalimat = kalimat.lower()
5     kate = kalimat.split()
6     kate = [k.strip(string.punctuation) for k in kate]
7     f = kate.count(kata.lower())
8
9     return f
10
11 kalimat = "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan"
12 kata = "makan"
13 f = menghitung_frekuensi(kalimat, kata)
14
15 print(f"{kata} ada {f} buah")
```

Output :

```
makan ada 3 buah
```


Penjelasan : Program ini menghitung frekuensi kemunculan suatu kata dalam sebuah kalimat. Fungsi “hitung_frekuensi” menghitung berapa kali sebuah kata muncul dalam kalimat.

SOAL 3

Source Code :

```
1 def menghapus_spasi(x):
2     kate = x.split()
3     teks_baru = ' '.join(kate)
4     return teks_baru
5
6 x = "saya tidak suka memancing ikan "
7
8 print(menghapus_spasi(x))
```

Output :

```
saya tidak suka memancing ikan
```

Penjelasan : Fungsi “menghapus_spasi” berfungsi untuk memisahkan teks menjadi kata-kata dan menghapus semua spasi berlebih. Kemudian, kata-kata tersebut digabungkan kembali menjadi satu string dengan satu spasi antar kata. Hasilnya adalah string tanpa spasi berlebih.

SOAL 4

Source Code :

```
1 def cari_terpanjang_terpendek(kalimat):
2     kate = kalimat.split()
3     kata_terpendek = kata_terpanjang = kate[0]
4     for kata in kate:
5         if len(kata) < len(kata_terpendek):
6             kata_terpendek = kata
7         elif len(kata) > len(kata_terpanjang):
8             kata_terpanjang = kata
9     return kata_terpendek, kata_terpanjang
10
11 kalimat = "red snakes and a black frog in the pool"
12 kata_terpendek, kata_terpanjang = cari_terpanjang_terpendek(kalimat)
13 print(f"terpendek: {kata_terpendek}, terpanjang: {kata_terpanjang}")
```

Output :

```
terpendek: a, terpanjang: snakes
```

Penjelasan : Fungsi “cari_terpanjang_terpendek” berfungsi untuk mencari kata terpanjang dan terpendek dalam suatu kalimat.

SOAL 5

Source Code :

```
1 import re
2 from datetime import datetime
3
4 def hitung_selisih_tanggal(teks):
5     tanggal_regex = r'\d{4}-\d{2}-\d{2}'
6     tanggal_list = re.findall(tanggal_regex, teks)
7     sekarang = datetime.now()
8
9     for tgl in tanggal_list:
10         tgl_obj = datetime.strptime(tgl, '%Y-%m-%d')
11         selisih = (sekarang - tgl_obj).days
12         tgl_ubah = tgl_obj.strftime('%Y-%m-%d 00:00:00')
13         print(f"{tgl_ubah} selisih {selisih} hari")
14
15 teks = """
16 Pada tanggal 1945-08-17 Indonesia merdeka. Indonesia memiliki beberapa pahlawan
17 nasional, seperti Pangeran Diponegoro (TL: 1785-11-11), Pattimura (TL: 1783-06-08) dan Ki
18 Hajar Dewantara (1889-05-02).
19 """
20
21 hitung_selisih_tanggal(teks)
```

Output :

```
1945-08-17 00:00:00 selisih 29108 hari
1785-11-11 00:00:00 selisih 87460 hari
1783-06-08 00:00:00 selisih 88347 hari
1889-05-02 00:00:00 selisih 49668 hari
```

Penjelasan : Fungsi “hitung_selisih_tanggal” berfungsi untuk mencari selisih antara tanggal pada teks dengan tanggal saat ini dihitung dan ditampilkan dengan format tersebut.

SOAL 6

Source Code :

```
1 import re
2 import random
3 import string
4
5 def buat_password():
6     return ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(8))
7
8 def cari_email(teks):
9     emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,7}\b', teks)
10     for email in emails:
11         username = email.split('@')[0]
12         print(f"{email} username: {username} , password: {buat_password()}")
13
14 teks = """
15 anton@mail.com dimiliki oleh antonius
16 budi@gmail.co.id dimiliki oleh budi anwari
17 slamet@getnada.com dimiliki oleh slamet slumut
18 matahari@tokopedia.com dimiliki oleh toko matahari
19 """
20
21 cari_email(teks)
```

Output :

```
anton@mail.com username: anton , password: ihU0AEyE
budi@gmail.co.id username: budi , password: u07f9FYx
slamet@getnada.com username: slamet , password: edIED16N
matahari@tokopedia.com username: matahari , password: k2wccIco
```

Penjelasan : Fungsi “buat_password” digunakan untuk membuat password random 8 karakter yang terdiri dari angka dan huruf. Fungsi “cari_email(teks)” untuk mencari semua email dalam teks lalu mengambil username dan menghasilkan password acak dengan memanggil fungsi “buat_password”.

Github : <https://github.com/babydoll-05/Laprak-8.git>