

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
Minggu ke / Materi	07 / Pengolahan String

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  
2023

## 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

#### STRING

Untaian karakter-karakter yang menjadi satu kesatuan dan digunakan dalam program komputer untuk menyimpan kalimat, baik panjang ataupun pendek.

Suatu jenis data yang mampu menyimpan huruf / karakter dalam kode ASCII. Tidak semua bahasa pemrograman memiliki tipe data String, seperti bahasa C. Tipe data ini merupakan jenis tipe data yang bukan tipe data dasar, karena tipe data ini pada dasarnya menyimpan lebih dari satu nilai tunggal sebagai satu kesatuan. Beberapa bahasa pemrograman ada yang menyebutkan bahwa String pada dasarnya adalah kumpulan tipe data karakter / *array of character* / *list of character*.

#### Pengaksesan String dan Manipulasi String

Indeks pada string dimulai dari 0 seperti pada *list*. Indeks string berupa bilangan bulat. Pada *memory* komputer, string disimpan secara urut menggunakan *list* yang berisi huruf-huruf dengan indeks yang dimulai dari nol.

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

### MATERI 2

#### OPERATOR DAN METODE STRING

##### Operator In

memeriksa apakah suatu kalimat merupakan substring dari suatu kalimat lain dengan menggunakan operator in. Hasil dari operator ini adalah True / False.

##### SOURCE CODE:

```
kalimat = "Dia pengikut Kristus"
data = "pengikut"
print(data in kalimat)      #True
print("Kristus" in kalimat) #True
print("Farisi" in kalimat)  #False
```

Selain itu, pada String juga dapat dilakukan perbandingan (*comparison*) yang juga menghasilkan *True* atau *False*.

```
if "Barnabas" > "Yesus":  
    print("Rakyat minta bebaskan!")  
else:  
    print("Cari cara lain!")
```

### Fungsi len

Cara untuk mengetahui berapa panjang (berapa jumlah karakter) dari sebuah string adalah dengan menggunakan operator `len(<string>)`. Untuk menampilkan huruf terakhir dari sebuah string kita harus menggunakan indeks string yang ke- `len(<string>-1)`, sebab indeks dimulai dari 0.

### SOURCE CODE:

```
kalimat = "Pada mulanya adalah Firman; Firman itu bersama-sama dengan  
Allah dan Firman itu adalah Allah"  
print(len(kalimat))  
  
terakhir = kalimat[len(kalimat)-1]  
terakhir_2 = kalimat[-1]  
print(f"{terakhir} dan {terakhir_2} sama")
```

### Traversing String

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

1. Dilakukan dengan akses terhadap indeks

```
kalimat = "Mata Tuhan ada di segala tempat, mengawasi orang jahat  
dan orang baik"  
i = 0  
while i < len(kalimat):  
    print(kalimat[i], end="")  
    i += 1
```

2. Dilakukan tanpa akses terhadap indeks secara otomatis.

```
kalimat = "Segala perkara dapat kutanggung di dalam Dia yang memberi  
kekuatan kepadaku"  
for i in kalimat:  
    print(i, end="")
```

### String Slice

String slice adalah menampilkan substring pada sebuah string dengan menggunakan indeks dari awal tertentu sampai akhir-1 tertentu. Sintaksnya menggunakan `[awal:akhir]`. Bagian awal atau akhir boleh dikosongkan. Bagian awal dimulai dari 0.

## SOURCE CODE:

```
kalimat = "Satu Bangsa"
kalimat_baru = kalimat[0:5] + "Allah"
print(kalimat_baru) #Satu Allah
```

Nama Method	Kegunaan	Penggunaan
capitalize()	untuk mengubah string menjadi huruf besar	string.capitalize()
count()	menghitung jumlah substring yang muncul dari sebuah string	string.count()
endswith()	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()

## Operasi \* dan + pada String

Pada Python operator + dan \* memiliki kemampuan khusus. Operator + yang biasanya digunakan untuk menjumlahkan bilangan bisa digunakan untuk menggabungkan dua buah string. Sedangkan operator \* yang bisa digunakan untuk mengkalikan bilangan bisa digunakan untuk menampilkan string sejumlah perkaliannya.

## SOURCE CODE:

```
kata_1 = "saya"
kata_2 = "dibasuh"
kata_3 = kata_1 + " " + kata_2 + " bersih "
kata_4 = "selamanya "
print(kata_3 + kata_4 * 7)
```

## OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_7/modul.py
saya dibasuh bersih selamanya selamanya selamanya selamanya selamanya selamanya selamanya
PS D:\71200645_PrakAlpro_7> □
```

## Parsing String

Parsing string adalah cara menelusuri string bagian demi bagian untuk mendapatkan / menemukan / mengubah bagian string yang diinginkan.

### SOURCE CODE:

```
kalimat = "Fans Hololive, pada tanggal 16-03-2024 ada festival ke-5!"
hasil = kalimat.split()
for i in hasil:
    if i[0].isdigit(): #kata yang dimulai dengan angka
        print(i)
        hasil_2 = i.split("-")
        print(hasil_2)
        print(hasil_2[0] + "/" + hasil_2[1] + "/" + hasil_2[2]) #urutan
pada list
```

### OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_7/modul.py
16-03-2024
['16', '03', '2024']
16/03/2024
PS D:\71200645_PrakAlpro_7> █
```

Menghitung jumlah huruf hidup dalam suatu kalimat.

### SOURCE CODE:

```
def kata_hidup():
    kalimat = input("Masukkan kalimat: ").lower()
    jumlah = 0
    huruf_hidup = "aiueo"
    for i in huruf_hidup: # yang di-looping adalah string "aiueo-nya"
        hasil = kalimat.count(i)
        jumlah = hasil + jumlah
    print("Jumlah huruf hidup:", jumlah)

kata_hidup()
```

### OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_7/modul.py
Masukkan kalimat: Bersama Tuhan Yesus selalu
Jumlah huruf hidup: 10
PS D:\71200645_PrakAlpro_7> █
```

### Menghilangkan karakter dengan index ganjil.

Melakukan *for loop* menggunakan sepanjang kata, sekaligus untuk mengecek *index* dari huruf tersebut. Jika *index* dapat dibagi 2 yang berarti berupa *index* genap, maka kata akan diambil dan disimpan dalam variabel hasil.

#### SOURCE CODE:

```
def string_ganjil():
    kata = input("Masukkan kata: ")
    hasil = ""

    for i in range(len(kata)):
        if i % 2 == 0:
            hasil = hasil + kata[i]

    print(hasil)
```

#### OUTPUT:

```
PS D:\Python3_CodeAcademy> & C:/Users/HP/AppDa
/string_exercise.py
Masukkan kata: Strawberry
Srwer
PS D:\Python3_CodeAcademy> █
```

Melakukan join pada list of strings dengan membuat newline.

#### SOURCE CODE:

```
winter_trees_lines = ['All the complicated details', 'of the attiring
and', 'the disattiring are completed!', 'A liquid moon', 'moves gently
among', 'the long branches.', 'Thus having prepared their buds', 'against
a sure winter', 'the wise trees', 'stand sleeping in the cold.']

winter_trees_full = ("\n".join(winter_trees_lines))
print(winter_trees_full)
```

#### OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP
o_7/modul.py
All the complicated details
of the attiring and
the disattiring are completed!
A liquid moon
moves gently among
the long branches.
Thus having prepared their buds
against a sure winter
the wise trees
stand sleeping in the cold.
PS D:\71200645_PrakAlpro_7> █
```

## 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 7.1** Buatlah sebuah program yang dapat mendeteksi apakah suatu kata adalah anagram dari kata lainnya atau bukan! Anagram adalah kata yang dibolak-balik susunan hurufnya sama. Misal: mata anagram dengan atma, maat, taam, tama, dsb.

Pengerjaan soal dilakukan dengan menerima *input* dari pengguna yang lalu menggunakan fungsi `.lower()` agar kata yang dibolak-balik sama dalam *lowercase*. Lalu dilakukan fungsi `sorted` pada variabel `kata_1` dan `kata_2`, dan percabangan `if-else`, jika `kata_1` dan `kata_2` yang sudah berurutan secara alfabetis sama atau tidak.

### SOURCE CODE:

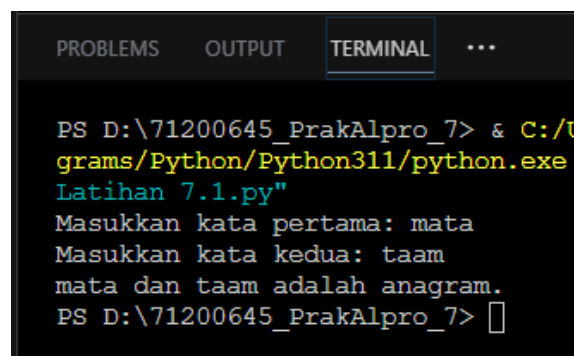
```
def is_anagram():
    kata_1 = input("Masukkan kata pertama: ").lower()
    kata_2 = input("Masukkan kata kedua: ").lower()

    urut_kata_1 = sorted(kata_1)
    urut_kata_2 = sorted(kata_2)

    if urut_kata_1 == urut_kata_2:
        print(f"{kata_1} dan {kata_2} adalah anagram.")
    else:
        print(f"{kata_1} dan {kata_2} bukan anagram.")

is_anagram()
```

### OUTPUT:



```
PROBLEMS  OUTPUT  TERMINAL  ...

PS D:\71200645_PrakAlpro_7> & C:/Users/PrakAlpro_7/Python/Python311/python.exe
Latihan 7.1.py
Masukkan kata pertama: mata
Masukkan kata kedua: taam
mata dan taam adalah anagram.
PS D:\71200645_PrakAlpro_7> 
```

## SOAL 2

**Latihan 7.2** Buatlah suatu program yang dapat menghitung frekuensi kemunculan suatu kata yang ada pada String. Misal terdapat kalimat "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan". Ditanyakan kata "makan". Output: makan ada 3 buah

Pengerjaan soal dilakukan dengan menerima inputan berupa kalimat dan kata yang frekuensi kemunculannya pada string ingin diketahui. Dilakukan fungsi if kata tersebut ditemukan di dalam kalimat, maka dengan fungsi .count, banyaknya kemunculan kata tersebut dapat dihitung.

### SOURCE CODE:

```
def hitung_string():
    kalimat = input("Masukkan kata: ").lower()
    kata = input("Jumlah kemunculan kata yang diinginkan: ")
    if kata in kalimat:
        hitung = kalimat.count(kata)
        print(f"{kata} ada {hitung} buah")

hitung_string()
```

### OUTPUT:

```
on311/python.exe "d:/71200645_PrakAlpro_7/Latihan 7.2.py"
Masukkan kata: Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan.
Jumlah kemunculan kata yang diinginkan: makan
makan ada 3 buah
PS D:\71200645_PrakAlpro_7>
```

## SOAL 3

**Latihan 7.3** Buatlah suatu program yang dapat menghapus semua spasi yang berlebih pada sebuah string, dan menjadikannya satu spasi normal! Misal: "saya tidak suka memancing ikan" Output: "saya tidak suka memancing ikan"

Pengerjaan soal dilakukan menggunakan fungsi input untuk menerima masukan kata dari pengguna, lalu menggunakan split untuk memisahkan setiap kata berdasarkan spasi yang tersimpan dalam bentuk list, lalu menggabungkan string kembali dengan fungsi join menggunakan satu spasi normal, sehingga menghapus jumlah spasi yang berlebih.

### SOURCE CODE:

```
def hapus_spasi():
    kalimat = input("Masukkan kalimat: ")
    hasil = kalimat.split()
    gabung = " ".join(hasil)
    print(gabung)

hapus_spasi()
```



## OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_7/Latihan 7.3.py"
Masukkan kalimat: Tuhan Yesus adalah jalan, kebenaran, dan hidup
Tuhan Yesus adalah jalan, kebenaran, dan hidup
PS D:\71200645_PrakAlpro_7> █
```

## SOAL 4

**Latihan 7.4** Buatlah suatu program mengetahui kata terpendek dan terpanjang dari suatu kalimat yang diinputkan! Misal: "red snakes and a black frog in the pool" Output: terpendek: a, terpanjang: snakes

Pengerjaan soal dilakukan dengan menerima masukan kalimat dari pengguna dengan fungsi *input*, lalu menggunakan fungsi *split* untuk memecah tiap kata pada kalimat yang dipisahkan oleh spasi. Dibuat variabel terpendek dan terpanjang yang menyimpan kata pertama pada *list* kata. Dilakukan *for loop* untuk memeriksa panjang dari setiap kata yang ada pada *list*. Ada pemeriksaan panjang dari kata yang di-*looping* dengan panjang kata yang tersimpan di variabel terpendek dan terpanjang, sehingga yang terpendek terbaru disimpan dan terpanjang terbaru yang disimpan sampai akhir *looping*.

## SOURCE CODE:

```
def panjang_pendek_kata():
    kalimat = input("Masukkan kalimat: ")
    kata = kalimat.split()
    terpendek = kata[0]
    terpanjang = kata[0]

    for i in kata:
        if len(i) < len(terpendek):
            terpendek = i
        if len(i) > len(terpanjang):
            terpanjang = i
    print(f"terpendek: {terpendek}, terpanjang: {terpanjang}")

panjang_pendek_kata()
```

## OUTPUT:

```
PS D:\71200645_PrakAlpro_7> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_7/Latihan 7.4.py"
Masukkan kalimat: red snakes and a black frog in the pool
terpendek: a, terpanjang: snakes
PS D:\71200645_PrakAlpro_7> █
```

## Link Github:

[https://github.com/EdithFelicia/71200645\\_Guided/tree/main/71200645\\_PrakAlpro\\_7](https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_7)