



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230997
Nama Lengkap	Christ Jevicto Ajimas Kirana
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
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

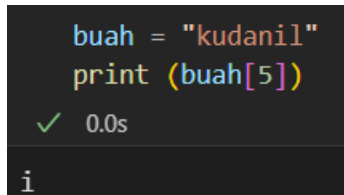
1.1 Pengantar String

String adalah untaian karakter yang menyimpan kalimat dalam komputer. Dengan menggunakan kode ASCII, string bisa menyimpan karakter. Namun tidak semua Bahasa pemrograman memiliki tipe data string contohnya C, tipe data ini memungkinkan penyimpanan lebih dari satu nilai sebagai satu kesatuan. Bahkan beberapa Bahasa pemrograman menganggap string sebagai beberapa Kumpulan karakter

1.2 Pengaksesan String dan Manipulasi String

String adalah tipe data yang dapat diakses sebagai satu kesatuan dengan nama variabelnya atau per huruf dengan indeks dimulai dari 0 seperti pada list. Indeks string harus berupa bilangan bulat, bukan pecahan.

Contoh :



```
buah = "kudaniL"  
print (buah[5])
```

✓ 0.0s

i

Penjelasan =

K = 0

U = 1

D = 2

A = 3

N = 4

I = 5

L = 6

1.3 Operator dan Metode String

Operator in

Pada string kita dapat mengecek apakah True atau False, Contoh :

```
kalimat = "aku suka kamu"
data = "aku"
print(data in kalimat)
print('suka' in kalimat)
print('dia' in kalimat)
```

✓ 0.0s

True
True
False

Selain operator in, kita dapat melakukan perbandingan pada string, Contoh :

```
if 'muach' < 'dia':
    print('Ya')
else:
    print('tidak')

if 'dua' == 'dua':
    print('sama')
```

✓ 0.0s

tidak
sama

Fungsi len

Kita bisa mencari tau berapa banyak jumlah karakter dalam sebuah string dengan menggunakan operator len(<string>), lalu kita bisa menampilkan huruf terakhir dari string dengan cara indeks len(<string>-1) dikarenakan indeks dimulai dari 0. Contoh :

```
• string = "muachhhhhhhhhhhhhhhhh"
panjang_string = len(string)
huruf_terakhir = string[len(string) - 1]

print("Panjang string:", panjang_string)
print("Huruf terakhir:", huruf_terakhir)
```

✓ 0.0s

Panjang string: 21
Huruf terakhir: h

TRAVERSING STRING

Kita dapat menampilkan string dengan menggunakan loop dengan while dan for,

<pre>kalimat = "MUACH AKUH" i = 0 while i < len(kalimat): print(kalimat[i], end="") i += 1</pre> <p>✓ 0.0s</p>	<pre>kalimat = "MUACH AKUH" for i in kalimat: print(i, end="")</pre> <p>✓ 0.0s</p>
contoh: MUACH AKUH	MUACH AKUH

STRING SLICE

String slice adalah teknik untuk menampilkan bagian substring dari sebuah string menggunakan indeks awal dan akhir-1 tertentu. Sintaksisnya adalah <string>[awal:akhir], di mana bagian awal dimulai dari 0. Contoh :

<pre>• kalimat = "muach akuh" awal = 0 akhir = 5 print(kalimat[awal:akhir]) print(kalimat[6:]) print(kalimat[:5]) print(kalimat[5:]) print(kalimat[:])</pre> <p>✓ 0.0s</p>
muach akuh muach akuh muach akuh

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

<pre>kalimat = 'satu' kalimat_baru = kalimat[0] + 'alah' print(kalimat_baru)</pre> <p>✓ 0.0s</p>
salah

Ada beberapa metode 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()
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()

Masih ada banyak lainnya, harus di ingat semua fungsi/method di atas yang mengembalikan string. Mengembalikan string baru dan tidak merubah aslinya, karena string bersifat immutable. Daftar lengkap fungsi/method dapat dilihat di <https://docs.python.org/library/stdtypes.html#string-methods>

Operator * dan + pada String

Python dapat memberi operator + dan * kemampuan khusus, operator + dapat menggabungkan 2 string menjadi 1, sedangkan operator * bisa digunakan untuk menampilkan string sejumlah perkaliannya. Contoh :

```
kata1 = 'aku'
kata2= 'kamu'
kata3 = 'ulang'
kata4 = kata1 + kata2
print(kata4)
print(kata3*2)
```

✓ 0.0s

akukamu
ulangulang

1.4 Parsing String

Parsing string adalah proses menelusuri sebuah string untuk mendapatkan atau mengubah bagian yang diinginkan. Misalnya, jika kita ingin mengambil tanggal dari string "Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka" dan mengubahnya menjadi format 08/17/1945, kita dapat menggunakan beberapa langkah:

1. Split string berdasarkan spasi menjadi token: "Saudara-saudara", "pada", "tanggal", "17-08-1945", "Indonesia", dan "merdeka".
2. Cari token yang diawali dengan angka, lalu split angka tersebut menggunakan pemisah '-'.
Misalnya, token "17-08-1945" akan dipecah menjadi ["17", "08", "1945"].
3. Susun ulang token-token dari langkah sebelumnya menjadi format yang diinginkan.

Contoh:

```
kalimat = "Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka"

hasil = kalimat.split(" ")

for k in hasil:
    if k[0].isdigit():
        hasil2 = k.split("-")
        print(hasil2[1]+"/"+hasil2[0]+"/"+hasil2[2])
```

✓ 0.0s

08/17/1945

MATERI 2

Kegiatan Praktikum

2.1 Buatlah program untuk menghitung berapa huruf hidup pada sebuah kalimat per kata

```
kalimat = str(input('masukan kata atau kalimat='))
hurufkecil = kalimat.lower()
jumlah = 0
for x in "aui eo":
    kata = hurufkecil.count(x)
    jumlah += kata
print(jumlah)
```

✓ 0.0s

7

Penjelasan =

- Meminta input pengguna
- Input dirubah ke huruf kecil semua
- siapkan variabel total jumlah huruf hidup aui eo dan set nilai awal = 0
- carilah ada berapa huruf 'a','i','u','e','o' pada

kalimat yang diinputkan dengan method count()

2.2 Buatlah program untuk membuat mengubah susunan format tanggal dari YYYY-MMDD menjadi DD-MM-YYYY

```
tgl = str(input('masukan tanggal(contoh = 2002-08-23)= '))
hasil = tgl.split('-')
tgl2 = hasil[2]+'-'+hasil[1]+'-'+hasil[0]
print(tgl2)
```

✓ 0.0s

12-12-2020

Penjelasan=

- meminta input pengguna
- input di bagi menjadi 3 dengan .split('-')
- merubah

bentuk menjadi bulan tanggal dan tahun

2.3 Buatlah program untuk mengetahui apakah suatu kalimat adalah palindrom atau bukan!
Palindrom adalah kalimat yang jika dibalik sama saja. Misalnya: Step on no pets, Pull up If I pull up, Some men interpret nine memos, dan Madam, In Eden I'm Adam.

```
tabnine: test | explain | document | ask
def palindrom(kalimat):
    kalimat = kalimat.lower()
    kalimat = ''.join(char for char in kalimat if char.isalpha())
    return kalimat == kalimat[::-1]
tabnine: test | explain | document | ask
def main():
    kalimat = str(input("masukan kalimat= "))
    if palindrom(kalimat):
        print("kalimat tersebut adalah palindrom")
    else:
        print("kalimat tersebut bukan palindrom")
if __name__ == "__main__":
    main()
✓ 0.7s
kalimat tersebut adalah palindrom
```

Penjelasan =

- meminta input dari pengguna
- input dijadikan huruf kecil
- buang semua yang bukan alpha
- Langkah terakhir adalah membandingkan kalimat dengan kalimat[::-1]. Ekspresi kalimat[::-1] digunakan untuk membuat versi terbalik dari kalimat.
- Di dalam fungsi main(), pengguna diminta untuk memasukkan sebuah kalimat.
- Kemudian, fungsi adalah_palindrom() dipanggil dengan parameter kalimat yang dimasukkan pengguna.
- Jika hasil dari pemanggilan fungsi adalah_palindrom() adalah True, program akan mencetak bahwa kalimat tersebut adalah palindrom. Jika hasilnya adalah False, program akan mencetak bahwa kalimat tersebut bukan palindrom.

2.4 Buatlah program untuk mengambil beberapa kata dari suatu kalimat yang diinputkan. Misal kalimat: "saya akan pergi sekolah". Akan diambil 1 kata, maka hasilnya: "saya", "akan", "pergi", "sekolah". Sedangkan jika 2 kata maka hasilnya: "saya akan", "akan pergi", "pergi sekolah".

```
tabnine: test | explain | document | ask
def ambil_beberapa_kata(kalimat, jumlah_kata):
    kata_kalimat = kalimat.split()
    hasil = []
    for i in range(len(kata_kalimat) - jumlah_kata + 1):
        hasil.append(" ".join(kata_kalimat[i:i+jumlah_kata]))
    return hasil

tabnine: test | explain | document | ask
def main():
    kalimat = input("Masukkan kalimat: ")
    jumlah_kata = int(input("Masukkan jumlah kata yang ingin diambil: "))
    hasil = ambil_beberapa_kata(kalimat, jumlah_kata)
    for kata in hasil:
        print(kata)

if __name__ == "__main__":
    main()
```

✓ 6.5s

```
saya akan
akan pergi
pergi ke
ke pasar
```

Penjelasan =

- Memasukan input dari pengguna, input berisi kalimat dan berapa banyak pengguna ingin di ambil
- Lalu fungsi def ambil_beberapa_kata(kalimat, jumlah_kata) akan menentukan berapa banyak kata yang akan diambil
- Pertama-tama kalimat akan di bagi bagi dengan menggunakan operator .split()
- Selanjutnya, program akan membuat jendela geser dengan lebar sejumlah jumlah_kata. Dalam setiap iterasi, program akan menggabungkan sejumlah jumlah_kata kata ke dalam satu string menggunakan metode join() dan menambahkannya ke dalam list hasil.
- Akhirnya, list hasil berisi semua kombinasi kata sesuai dengan jumlah_kata yang diminta.
- Hasilnya kemudian dicetak satu per satu.

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

```
tabnine: test | explain | document | ask
def anagram(kata1, kata2):
    kata1 = kata1.lower()
    kata2 = kata2.lower()

    mengurutkan1 = sorted(kata1)
    mengurutkan2 = sorted(kata2)

    return mengurutkan1 == mengurutkan2

tabnine: test | explain | document | ask
def main():
    kata1 = str(input("masukan kata 1 ="))
    kata2 = str(input("masukan kata 2 ="))

    if anagram(kata1, kata2):
        print(f"kata1 = {kata1}, kata2 = {kata2} adalah anagram")
    else:
        print(f"kata1 = {kata1}, kata2 = {kata2} bukan anagram")

if __name__ == "__main__":
    main()
```

```
kata1 = kiri, kata2 = riki adalah anagram
```

```
kata1 = bukan , kata2 = kubek bukan anagram
```

Penjelasan = user menginputkan kata1 dan kata2 lalu input user di ubah menjadi huruf kecil semua, lalu kata1 dan kata2 diurutkan menggunakan fungsi sorted, lalu fungsi membandingkan mengurutkan1 apakah sama dengan mengurutkan2, jika sama maka hasilnya True jika salah hasilnya False, ketika hasil True maka akan mengprint "kata1 = {kata1}, kata={kata2} adalah anagram", jika hasilnya False maka akan menprint "kata1 = {kata1}, kata={kata2} bukan anagram"

SOAL 2

```
tabnine: test | explain | document | ask
def mencarikata(kalimat, kata):
    kalimat = kalimat.lower()
    kata = kata.lower()

    kemunculan = 0
    kdk = kalimat.split()
    for kk in kdk:
        if kk == kata:
            kemunculan += 1
    return kemunculan

tabnine: test | explain | document | ask
def main():
    kalimat = str(input("masukan kalimat= "))
    kata = str(input("masukan kata= "))

    jumlah = mencarikata(kalimat, kata)
    if jumlah > 0:
        print(f"kata = '{kata}' muncul sebanyak {jumlah} buah di kalimat = {kalimat}")
    elif jumlah <= 0:
        print(f"kata = '{kata}' tidak ada dalam kalimat = '{kalimat}' ")

if __name__ == "__main__":
    main()
```

kata = 'suka' muncul sebanyak 3 buah di kalimat = aku suka dia dia suka aku dia ga suka aku

kata = 'makan' tidak ada dalam kalimat = 'aku suka dia dia suka aku aku ga suka dia'

Penjelasan = user akan menginput suatu kalimat lalu user akan menginput kata apa yang ingin di cari, lalu kalimat dan kata akan diubah ke huruf kecil dengan fungsi .lower(), lalu menghitung kemunculan dengan menggunakan fungsi for. Jika jumlah kata lebih besar daripada nol maka akan mengprint "kata = '{kata}' muncul sebanyak {jumlah} buah di kalimat = {kalimat}". Namun jika jumlah kata kurang atau sama dengan nol maka akan mengprint "kata = '{kata}' tidak ada dalam kalimat = '{kalimat}' ".

SOAL 3

```
tabnine: test | explain | document | ask
def hsb(kalimat):
    kalimat_split = kalimat.split()
    kalimat_normal = " ".join(kalimat_split)
    return kalimat_normal

tabnine: test | explain | document | ask
def main():
    kalimat = str(input("masukan suatu kalimat= "))
    hasil = hsb(kalimat)

    print(f" '{kalimat}' setelah dihapus spasi berlebih='{hasil}' ")

if __name__ == "__main__":
    main()
```

```
'akuu          suka          kamu          hehe' setelah dihapus spasi berlebih='akuu suka kamu hehe'
```

Penjelasan =

User memasukan input sebuah kalimat dengan spasi yang tidak beraturan, lalu setiap kata akan di pisah pisah dengan fungsi .split() lalu setiap kata digabungkan Kembali dengan menambah 1 spasi dengan fungsi .join(), lalu akan menghasilkan output '{kalimat}' setelah dihapus spasi berlebih='{hasil}'.

SOAL 4

```
tabnine: test | explain | document | ask
def panjang_pendek(kalimat):
    kata = kalimat.split()

    terpendek = kata[0]
    terpanjang = kata[0]

    for i in kata:
        if len(i) < len(terpendek):
            terpendek = i
        elif len(i) > len(terpanjang):
            terpanjang = i

    return terpendek, terpanjang

tabnine: test | explain | document | ask
def main():
    kalimat = str(input('masukan suatu kalimat= '))
    terpendek, terpanjang = panjang_pendek(kalimat)
    print(f"dari kalimat = '{kalimat}', kata terpendek = '{terpendek}', terpanjang = '{terpanjang}' ")

if __name__ == "__main__":
    main()
```

```
dari kalimat = 'red snakes and a black frog in the pool', kata terpendek = 'a', terpanjang = 'snakes'
```

Penjelasan =

User memasukkan input suatu kalimat, lalu kalimat akan dipisah pisahkan dengan fungsi `.split()`, lalu program akan melakukan iterasi pada setiap kata dalam setiap iterasi, kata akan dibandingkan dengan terpendek dan terpanjang, jika kata saat ini lebih kecil daripada terpendek maka kata saat ini menjadi terpendek, dan jika kata saat ini lebih besar daripada terpanjang maka kata saat ini menjadi terpanjang. Lalu program akan memberikan output “dari kalimat = ‘{kalimat}’, kata terpendek = ‘{terpendek}’, terpanjang = ‘{terpanjang}’ “

LINK GITHUB = <https://github.com/Yeemeki/praktikum-algoritma-dan-pemograman.git>