



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230972>
Nama Lengkap	<Oktavian Christ Putranto>
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.

String

String sederhananya merupakan Kumpulan 2 atau lebih char yang menjadi suatu tipe data baru yang Bernama string biasanya tipe data string ditandai dengan lambing petik('.....') atau petik dua("....."), String mampu menyimpan huruf / karakter dalam kode ASCII. Tidak semua bahasa pemrograman memiliki tipe data String, seperti misalnya bahasa C.

Karena menyimpan lebih dari satu nilai secara bersamaan, tipe data ini sering dianggap sebagai tipe data yang rumit. Beberapa bahasa pemrograman sering menganggap string sebagai kumpulan tipe data karakter, array karakter, atau daftar karakter.

Cara mengakses string :

```
namasaya = "Antonius Rachmat C"
temansaya1 = "Yuan Lukito"
temansaya2 = 'Laurentius Kuncoro'
temansaya3 = "Matahari" + 'Bakti'

# print(temansaya3)
#setiap string selalu dimulai dari 0
print(namasaya[0]) #'A'
print(namasaya[9]) #'R'
print(namasaya[8]) #" "
print(temansaya1[1]) #'u'
print(namasaya[-1]) #C dari belakang stringnya
namasaya=namasaya + temansaya1[0]
print(namasaya)
huruf = temansaya2[0]
print(huruf) #'L'
```

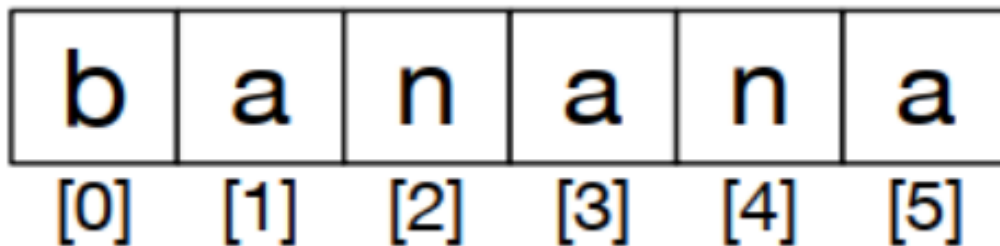
output:

```
A
R

u
C
Antonius Rachmat CY
L
```

Dari data tersebut kita dapat memperoleh pengetahuan baru bahwa dalam tiap data string maka data pertamanya akan dihitung dari 0 dan setiap karakter dalam string tersebut termasuk space(' ') juga dihitung sebagai satu character, dan cara untuk mengakses string dapat dilakukan dengan memasukkan Indeks stringnya dan perlu diperhatikan bahwa dalam mengakses indeks string yang kita inginkan maka kita harus memakai bilangan bulat, bukan pecahan. seperti ini `print(namasaya[0])`.

Pada memory komputer, string disimpan secara urut menggunakan list yang berisi huruf-huruf dengan indeks yang dimulai dari nol. Untuk contoh lebih jelasnya maka kita akan menunjukkan string berisi "banana" yang disimpan pada memory komputer seperti gambar dibawah ini.



Oprator Dalam Tipe Data String

Penggunaan in dalam string :

```
kalimat = "SAYA mau dimakan"
data = "saya"
doraemon="SA"
print(data in kalimat) #True
print(doraemon in kalimat) #True
print("saya" in kalimat) #False
print("mau" in kalimat) #True
print("udi" in kalimat) #False
```

Output :

```
False
True
False
True
False
```

Oprator in dalam pengolahan datastring berguna untuk mengecek apakah terdapat data string yang di ingin dalam string lain nya atau tidak dan akan menghasilkan value Boolean True atau False

Oprator Perbandingan:

```
if "saya" > "dia":  
    print("Ya") #Ya  
else:  
    print("Tidak")  
  
if "dua" == "dua":  
    print("Sama") #Sama
```

output:

```
string/  
Ya  
Sama  
=====
```

Dari contoh tersebut kita dapat melihat bahwa data string dapat dibandingkan untuk menghasilkan nilai True dan false juga yang nantinya dapat dikembangkan dan digunakan dalam berbagai pemrograman

Fungsi len

Sederhananya fungsi len ini berguna untuk menghitung jumlah indeks data string yang ada

Contoh penggunaan len :

```
kalimat = "universitas kristen duta wacana yogyakarta"  
print(len(kalimat)) #output 42  
  
terakhir = kalimat[len(kalimat)-1]  
print(terakhir) #output 'a'  
  
#bisa juga menggunakan indeks -1  
terakhir_versi2 = kalimat[-1]  
print(terakhir_versi2) #output 'a'  
#atau menggunakan indeks -2 untuk huruf terakhir kedua  
terakhir2 = kalimat[-2]  
print(terakhir2) #output 't'
```

Output :

```
string/  
42  
a  
a  
t  
=====
```

Kegunaan len dari contoh code tersebut ialah menghitung total charaktr dalam string tersebut.

TRAVERSING STRING

Traversing berguna untuk mencetak huruh/karakter dalam string memakai looping ada yang mengakses indeks maupun tidak contohnya sebagai berikut:

```
kalimat = "indonesia jaya"
i = 0
while i < len(kalimat):

    print(kalimat[i])
    i += 1
```

Output:

```
string/t
i
n
d
o
n
e
s
i
a

j
a
y
a
```

Contoh lain :

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

output :

```
indonesia jaya
```

String slice

String slice ialah cara untuk mengambil bagian dari string bert=dasarkan indeks nya sintaksnya menggunakan [awal:akhir]. Bagian awal atau akhir boleh dikosongkan. Bagian awal dimulai dari 0.

```
kalimat = "cerita rakyat indonesia"
awal = 0
akhir = 6

# print(kalimat[awal:akhir]) #cerita
print(kalimat[3:8:]) #longkap 2
print(kalimat[8-1:3-1:-1]) #longkap
print(kalimat[7:len(kalimat)]) #rakyat
print(kalimat[:5+1]) #cerit
print(kalimat[5:]) #a rakyat
print(kalimat[:]) #cerita rakyat
```

Output:

```

string/str_sic.py
ita r
r ati
rakyat indonesia
cerita
a rakyat indonesia
cerita rakyat indonesia

```

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()
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()

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

Input:

```
#anagram
kata_1=input("masukan kata pertama:")
kata_2=input("masukan kata kedua:")
def anagram(kata_1,kata_2):
    kata_1=kata_1.lower()
    kata_1= ''.join([i for i in kata_1 if i.isalpha()])
    kata_2=kata_2.lower()
    kata_2= ''.join([i for i in kata_2 if i.isalpha()])
    kata_1=kata_1[::-1]
    if kata_1==kata_2:
        print("merupakan anagram")
    else:
        print("bukan angram")

anagram(kata_1,kata_2)
```

Output:

```
masukan kata pertama:kola
masukan kata kedua:alok
merupakan anagram
```

```
masukan kata pertama:koka
masukan kata kedua:kola
bukan angram
```

Penjelasan:

Pada line pertama code akan langsung meminta 2 input user berupa kata kata yang ingin di cek apakah anagram atau tidak

Kemudian saya mendefine fungsi anagram yang langsung melowercase seluruh karakter dalam input user memakai code `kata_1=kata_1.lower()` dan `kata_2=kata_2.lower()`

```
kata_1= ''.join([i for i in kata_1 if i.isalpha()]) dan kata_2= ''.join([i for i in kata_2 if i.isalpha()])
```

kedua code diatas berguna untuk menghapus semua karakter yang bukan alphabetic atau huruf.

Kemudian

```
kata_1=kata_1[::-1]
if kata_1==kata_2:
    print("merupakan anagram")
else:
    print("bukan angram")
```

code ini berfungsi untuk membalik kata satu kemudian dimasukan ke conditional untuk mengecek ketika kata 1 dibalik urutan nya dari belakang sama atau tidak dnegan kata2.

Jika True maka akan mengeluarkan output “ merupakan anagram” kalau False maka keluar input “Bukan anagram”

SOAL 2

Input:

```
kata= "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan"
kata=kata.lower()
kata=''.join([i for i in kata if i.isspace() or i.isalpha()])
kata=kata.split(' ')
total=0
for x in ['makan']:
    jml = kata.count(x)
    # jml=a_string.count(x)
    total += jml
print(total) #hasil = 4
```

Output :

3

Penjelasan :

Code ini berfungsi untuk menghitung ada berapa kata makan dalam satu data string/satu kalimat

pertama tama dalam aturan pengecekan maka dalam data string tersebut harus di jadikan lowercase dahulu memakai code `kata=kata.lower()`

kemudian di hilangkan semua yang tidak terasuk huruf termasuk tiap tanda baca. Memakai code

```
kata=''.join([i for i in kata if i.isspace() or i.isalpha()])
```

kemudian kata kita split berdasarkan spasinya sehingga terbagi menjadi token token yang terpisah code yang dipakai ialah `kata=kata.split(' ')`


```
total=0
for x in ['makan']:
    jml = kata.count(x)
    # jml=a_string.count(x)
    total += jml
print(total) #hasil = 4
```

keseluruhan code diatas berfungsi untuk menambahkan jumlah total ketika dalam token token tersebut terdapat kata 'makan' kemudian setelah dimasukan dalam value total setiap perulangan berjalan dan mengecek.

SOAL 3

Input:

```
kata = "saya tidak      suka memancing      ikan      "
# kata=kata.split(" ")
print(kata)
kata=kata.rstrip()
kata=' '.join(kata.split())
print(kata)
```

Output:

```
saya tidak      suka memancing      ikan
saya tidak suka memancing ikan
```

pada soal ini kita diminta untuk membuat fungsi yang mampu menghapus spasi berlebih.

Penjelasan code:

```
kata=kata.rstrip()
```

code itu berfungsi untuk menghapus keterlebihan spasi pada awal atau akhir kalimat.

```
kata=' '.join(kata.split())
```

code tersebut berfungsi untuk menyambungkan token yang dipisah memakai oprator split karena digabung/di join memakai spasi maka hasil akhir pastinya akan memiliki jumlah spasi yang tepat antar katanya.

SOAL 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
kalimat ="red snakes and a black frog in the pool"
kata=kalimat.split(" ")
print (kata)
kata_pendek=kata_panjang=kata[0]
```

```

for kata in kata:
    if len(kata)<len(kata_pendek):
        kata_pendek=kata
    if len(kata)>len(kata_panjang):
        kata_panjang=kata
print(f" kata terpanjang adalah {kata_panjang} dan kata terpendek adalah {kata_pendek} ")

```

Output:

```

['red', 'snakes', 'and', 'a', 'black', 'frog', 'in', 'the', 'pool']
kata terpanjang adalah snakes dan kata terpendek adalah a

```

```

kalimat ="red snakes and a black frog in the pool"

```

menentukan kalimat yang di tes dalam soal contoh kalimatnya ialah red snakes and a black frog in the pool kemudian setelah membuat variable kalimat yang berisi keseluruhan kalimat kalimat langsung di split menjadi token kata satu persatu

visual :

```

['red', 'snakes', 'and', 'a', 'black', 'frog', 'in', 'the', 'pool']

```

Setelah terpisah tiap token maka kita define 2 variable baru yaitu

`kata_pendek=kata_panjang=kata[0]` keduanya akan memiliki value red apada awalnya karena memiliki indeks 0 kata yaitu 'red' setelah itu barulah valuenya dibandingkan memakai len satu persatu untuk menemukan kata yang tependek dan terpanjang code:

```

if len(kata)<len(kata_pendek):
    kata_pendek=kata
    if len(kata)>len(kata_panjang):
        kata_panjang=kata

```

setelah itu maka akan dikeluarkan output sebagai berikut :

```

print(f" kata terpanjang adalah {kata_panjang} dan kata terpendek adalah {kata_pendek} ")

```

Source:

https://github.com/vianchr/laporan_7_vian