



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

| | |
|---------------------------|-------------------------------|
| NIM | 71200609 |
| Nama Lengkap | Airell Aristo Subagia |
| 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

Pengolahan String

Apa Itu String ?

Dalam pemrograman, string adalah sebuah tipe data yang digunakan untuk merepresentasikan teks atau urutan karakter. String biasanya digunakan untuk menyimpan atau memanipulasi data teks (kata-kata, frasa, atau kalimat). Dalam bahasa pemrograman Python, string di representasikan menggunakan tanda kutip ganda (" ") atau tanda kutip tunggal (' '). Contoh :

```
x = 'airell'
y = "airell"
```

Indexing pada String

Indexing pada string adalah cara untuk mengakses karakter dalam sebuah string menggunakan indeks numerik (0,1,2,3,...). Setiap karakter pada string memiliki posisi uniknya tersendiri. Indeks pada string dimulai dari 0 untuk karakter pertama dalam string, kemudian 1, 2, dst. Contoh :

```
#mengakses karakter pertama
x = 'airell'

print(x[0])
outputnya :
```

a

Contoh lainnya :

```
#mengakses karakter kedua
x = 'airell'

print(x[1])
outpunya :
```

i

Penggunaan Operator pada String

Pada string kita dapat menggunakan operator untuk mendapatkan hasil berupa true atau false. Contoh :

1. Operator in :

```
x = 'airell aristo'

print("airell" in x)
print("subagia" in x)
```

outputnya :

```
True
False
```

2. Operator Perbandingan :

```
x = 'zirell'
y = "subagia"

if x > y :
    print("y")
else :
    print("n")
```

outputnya :

```
y
```

```
x = 'aaa'
y = "aaa"

if x == y :
    print("y")
else :
    print("n")
```

outputnya :

```
y
```

Fungsi Len()

Fungsi Len() pada string digunakan untuk mendapatkan panjang dari sebuah string. Selain itu dapat juga digunakan untuk digunakan sebagai indexing. Contoh penggunaan :

```
x = 'airell'

print(len(x))
print(x[len(x)-1])
```

outputnya :

```
6
l
```

String Slicing

String slicing merupakan teknik untuk mengambil sebagian atau potongan dari sebuah string. Cara penggunaan dari string slicing itu seperti berikut, string[start:stop:step].

- Start (inklusif) : indeks awal dari potongan yang ingin kita ambil.
- Stop (eksklusif) : indeks akhir dari potongan yang ingin kita ambil.
- Step : Langkah dalam memindai potongan string.

Contoh :

```
x = "Hello, World!"

# Mengambil potongan dari indeks ke-3 hingga indeks ke-7 (indeks ke-7 tidak termasuk)
print(x[3:7])

# Mengambil potongan dari indeks ke-0 hingga indeks ke-5 dengan langkah 2
print(x[0:5:2])

# Mengambil potongan dari indeks ke-6 hingga akhir string
print(x[7:])

# Mengambil potongan dari indeks ke-2 hingga indeks ke-10 dengan langkah -1 (membalik string)
print(x[10:1:-1])

# Mengambil potongan string secara terbalik
print(x[::-1])
```

Outputnya :

```
lo,  
Hlo  
World!  
lrow ,oll  
!dlrow ,olleH
```

Method yang sering digunakan

String memiliki beberapa method bawaan yang sering digunakan pada bahasa pemrograman python, diantaranya :

1. upper() dan lower()

upper() digunakan untuk mengkonversi string menjadi huruf kapital, sebaliknya lower() untuk huruf kecil.

Contoh :

```
x = "ArIsTo"  
  
print(x.upper())  
print(x.lower())
```

outputnya :

```
ARISTO  
aristo
```

2. count()

count() digunakan untuk menghitung jumlah karakter yang ada pada string.

Contoh :

```
x = "airell"  
  
print(x.count("l"))
```

outputnya :

```
2
```

3. replace()

replace() digunakan untuk mengganti karakter yang lama menjadi karakter yang baru.

Atau bisa juga untuk menghilangkan spasi/whitespace.

Contoh :

```
x = "airell"  
print(x.replace("a","o"))
```

outputnya :

```
oirell
```

4. split()

split() digunakan untuk memisahkan sebuah string berdasarkan suatu karakter menjadi bentuk list.

Contoh :

```
x = "airell aristo subagia"  
print(x.split(" "))
```

outputnya :

```
['airell', 'aristo', 'subagia']
```

String Parsing

Merupakan cara untuk menelusuri karakter per karakter (satu-satu) pada sebuah string.

Contoh penggunaanya :

```
x = "subagia"  
for karakter in x :  
    print(karakter)
```

outputnya :

```
s  
u  
b  
a  
g  
i  
a
```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

#Latihan 6.1

```
def checkAnagram(kata, checkKata) :  
    noSpaceKata = kata.replace(" ", "")  
    noSpaceCheckKata = checkKata.replace(" ", "")  
    hitung = 0  
    if len(noSpaceKata) == len(noSpaceCheckKata) :  
        for huruf in noSpaceCheckKata :  
            if noSpaceCheckKata.count(huruf) == noSpaceKata.count(huruf) :  
                hitung += 1  
        if len(noSpaceKata) == hitung :  
            print(checkKata, "anagram dengan", kata)  
        else :  
            print(checkKata, "bukan anagram dari", kata)  
    else :  
        print(checkKata, "bukan anagram dari", kata)  
  
checkAnagram("mata", "atma")  
checkAnagram("mata", "maat")  
checkAnagram("mata", "taam")  
checkAnagram("mata", "tama")
```

Outputnya :

```
atma anagram dengan mata  
maat anagram dengan mata  
taam anagram dengan mata  
tama anagram dengan mata
```

Penjelasannya :

1. Pertama-tama, akan dibuat function bernama **checkAnagram** dengan 2 parameter bernama **kata**, **checkKata**. parameter tersebut akan digunakan untuk menyimpan input dari user.
2. Setelah itu, dilakukan penghilangan whitespace/spasi pada kedua string, dilakukan dengan cara menggunakan **replace()**.
3. Kemudian, dibuat variable **hitung** yang nantinya akan digunakan untuk checking apakah jumlah huruf masing-masing yang terdapat pada string yang akan dicek itu sama dengan jumlah huruf kata anagramnya.

4. Kemudian, dibuat **if** pertama yang digunakan untuk checking, apakah panjang karkter dari kata yang akan dicek anagramnya itu sama atau tidak.
5. Setelah itu, **for loop** yang pertama digunakan untuk mengakses setiap karakter yang ada di dalam variable **noSpaceCheckKata**.
6. Kemudian, **if** kedua digunakan untuk check apakah jumlah huruf yang tersedia pada **noSpaceCheckKata** dan **noSpaceKata** itu sama atau tidak. Jika sama maka akan masuk ke perhitungan.

Jadi, pada dasarnya kata yang anagram itu **sudah pasti sama jumlah masing-masing hurufnya** dengan kata yang akan dicek. Misalnya pada kasus diatas itu mata (m = 1, a = 2, t = 2) sama dengan kata yang dicek atma (a = 2, t =1, m = 1).

SOAL 2

#Latihan 7.2

```
kalimat = "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan"
cari = input("Kata Yang Mau Dihitung : ")

def hitungFrekuensi(kalimat,kata) :
    lowerKalimat = kalimat.lower()
    return lowerKalimat.count(kata.lower())

print(hitungFrekuensi(kalimat,cari))
```

Outputnya :

```
Kata Yang Mau Dihitung : makan
3
```

Penjelasannya :

1. Pertama-tama, diinisiasi variable bernama **cari** yang digunakan untuk menyimpan inputan user.

2. Setelah itu, akan dibuat function bernama **hitungFrekuensi** dengan 2 parameter bernama **kalimat** dan **kata**. parameter tersebut akan digunakan untuk menyimpan input dari user.
3. Kemudian, kalimat dibuat huruf kecil terlebih dahulu menggunakan **lowerKalimat = kalimat.lower()** agar menghitungnya mudah.
4. Terakhir, tinggal dilakukan pengembalian jumlah menggunakan fungsi **return** yang didalamnya terdapat **lowerKalimat.count(kata.lower())**. Digunakan method **count()** untuk menghitung berapa kali kata tersebut muncul, dan sebelum menghitung akan kata checkingnya akan dijadikan huruf kecil terlebih dahulu.

SOAL 3

#Latihan 7.3

```
def spasiBerlebih(kalimat) :  
    x = kalimat.lower()  
    x1 = kalimat.split(" ")  
    hitung = 0  
    print(' ',end="")  
    for kata in x1 :  
        if kata != "" :  
            hitung += 1  
            print(kata,end=" " if hitung != len(x1) - x1.count("") else '')  
  
kalimat = "saya    tidak suka memancing ikan "  
spasiBerlebih(kalimat)
```

Outputnya :

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

Penjelasannya :

1. Pertama-tama, dibuat function bernama **spasiBerlebih**. Dengan 1 parameter bernama **kalimat**. parameter tersebut akan digunakan untuk menyimpan input dari user.
2. Kemudian, variabelnya dilower terlebih dahulu menggunakan method **lower()**.
3. Setelah itu, kalimat tersebut akan di split berdasarkan whitespace menggunakan method **split(" ")**.

4. Kemudian, variable **hitung** nantinya akan digunakan untuk menghitung berapa banyak nilai yang isinya bukan whitespace.
5. Setelah itu, dilakukan parsing pada setiap value yang ada di list bernama **x1**.
6. Kemudian, dilakukan checking menggunakan **if** kalau **kata** itu **bukan whitespace**. Maka dia akan dilakukan fungsi print.
7. Setelah itu, variable **hitung** ditambahkan **1**.
8. Terakhir, dilakukan printing dimana didalamnya terdapat ternary operator untuk fungsi **endnya** yang dimana jika dia semua kata sudah dikeluarkan maka akan diakhir dengan tanda kutip dua(") bukan dengan whitespace lagi. Cara ini dilakukan dengan komparasi jumlah kata yang ada pada list **x1**, dan nilai whitespace dikurangi dengan panjang list **x1**.

SOAL 4

#Latihan 7.3

```
def lowlong(kalimat) :  
    split = kalimat.split(" ")  
    terpanjang = split[0]  
    terpendek = split[0]  
    for kata in split :  
        if len(kata) > len(terpanjang) :  
            terpanjang = kata  
        if len(kata) < len(terpendek) :  
            terpendek = kata  
    print(terpanjang)  
    print(terpendek)  
  
kalimat = "red snakes and a black frog in the pool"  
lowlong(kalimat)
```

Outputnya :

```
snakes  
a
```

Penjelasannya :

1. Pertama-tama, dibuat function bernama **lowlong**. Dengan 1 parameter bernama **kalimat**. parameter tersebut akan digunakan untuk menyimpan input dari user.

2. Kemudian, displit terlebih dahulu berdasarkan whitespace/spasi menggunakan **.split(" ")**
3. Setelah itu dilakukan insisiasi awal untuk kata terpanjang dan terpendek yang isinya merupakan value pertama pada list **split**.
4. Kemudian, dilakukan parsing untuk mengakses nilai yang berada pada list menggunakan **for loop**.
5. Setelah itu tinggal dilakukan checking saja menggunakan dua **if**. Yang dimana yang pertama digunakan untuk cek apakah kata sekarang itu lebih besar dari variable bernama **terbesar** yang sudah diinsiasi diawal tadi. Dan juga **if** kedua digunakan untuk untuk cek apakah kata sekarang itu lebih kecil dari variable bernama **terkecil** yang sudah diinsiasi diawal tadi.

Link Github :

https://github.com/AirellAristo/TugasPrakAlpro/tree/main/71200609_Pertemuan7Guided