



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230978
Nama Lengkap	Jonathan Satriani Gracio Andrianto
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%)

Pengantar String

String adalah salah satu tipe data yang paling sering digunakan dalam python. String adalah kumpulan dari karakter yang menjadi kesatuan. Tipe data string mampu menyimpan huruf atau karakter dan disimpan dalam kode ASCII (American Standard Code for Information Interchange). Tidak semua Bahasa pemrograman memiliki tipe data string, misalnya Bahasa C. Tipe data string juga bukan merupakan tipe data dasar, karena tipe data ini menyimpan kumpulan nilai tunggal menjadi satu kesatuan. Ada beberapa Bahasa pemrograman yang menyatakan string sebagai list of characters / array of characters.

Pengaksesan String dan Manipulasi String

Sama seperti integer, string dapat dibuat dengan variable :

```
namasaya = "Jonathan Andrianto"
temansaya1 = "Cristiano Siuu"
temansaya2 = "Alfredo"
temansaya3 = "Yohanes" + "Nevan"

print(temansaya3) #YohanesNevan
print(namasaya[0]) #'J'
print(namasaya[9]) #'A'
print(temansaya1[1]) #'r'

huruf = temansaya2[0]
print(huruf) #'A'
```

Pengaksesan dan manipulasi string

Kita bisa membuat string dengan mendeklarasikan variable dan diisi dengan data string. String dapat diakses secara lengkap atau dapat diakses karakter per karakter dengan menyebutkan indexnya. Index harus bilangan bulat dan dimulai dari 0.

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

Bentuk string dalam memori

[i] → index = i. Kita dapat lihat kalau index dimulai dari 0.

Operator dan Metode String

Operasi in

```
kalimat = "saya ganteng"  
data = "saya"  
print(data in kalimat) #True  
print("ganteng" in kalimat) #True  
print("jelek" in kalimat) #False
```

Operasi in

Operasi ini memeriksa apakah suatu kata atau kalimat adalah substring dari kalimat lain. Operasi ini mirip dengan kondisional, karena memberi hasil True atau False. String juga dapat melakukan perbandingan yang menghasilkan True atau False.

```
if "real madrid" > "barcelona":  
    print("Ya") #Ya  
else:  
    print("Tidak")  
  
if "gue" == "gue":  
    print("Sama") #Sama
```

Perbandingan string

Fungsi len

```
kalimat = "Joe kok ganteng banget ya"  
print(len(kalimat)) #output 25  
  
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 'y'
```

Fungsi len string

Fungsi len adalah fungsi yang dapat membantu kita dalam mencari tahu Panjang string atau Panjang karakter. Untuk menampilkan karakter terakhir kita harus menggunakan index [len – 1] atau [-1] karena ingat index itu dimulai dari 0.

Traversing string

Untuk menampilkan string dengan cara huruf per huruf bisa dengan dua cara :

```
kalimat = "hala madrid"
i = 0
while i < len(kalimat):
    print(kalimat[i], end=" ")
    i += 1

# cara 2 (tanpa akses index)
kalimat = "hala madrid"
for kal in kalimat:
    print(kal, end=" ")
```

Traversing string

String slice

```
kalimat = "hala madrid"
awal = 0
akhir = 4
print(kalimat[awal:akhir]) #hala
print(kalimat[5:len(kalimat)]) #madrid
print(kalimat[:3]) #hal
print(kalimat[3:]) #a madrid
print(kalimat[:]) #hala madrid
```

String slice

String slice menampilkan substring sebuah string dengan menggunakan index dimana awal dan akhirnya dapat dimanipulasi. Kita dapat menggunakan [awal:akhir]. Bagian awal atau akhir boleh dikosongkan. Bagian awal dimulai dari 0. Bagian akhir defaultnya adalah index terakhir.

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

```
kalimat = "lima"
kalimat[0] = "rumpot" #TypeError: 'str' object does not support item
                        assignment

#agar dapat diubah, kita harus meletakkannya dalam variabel berbeda
kalimat = "lima"
kalimat_baru = kalimat[0] + "ama" #lama
```

Contoh kasus immutable string

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()

Daftar lengkap dapat dilihat di <https://docs.python.org/library/stdtypes.html#string-methods>

Operator * dan + pada String

Pada string, operator '+' dapat digunakan untuk menggabungkan dua string. Sedangkan operator '*' dapat digunakan untuk menampilkan string sejumlah perkaliannya.

```
kata1 = "hala"
kata2 = "madrid"
kata3 = "haleluya"

print(kata1 + kata2) #halamadrid
print(kata3 * 3) #haleluyahaleluyahaleluya
```

Parsing String

Parsing string adalah cara menelusuri string bagian demi bagian untuk menemukan dan mengubah bagian string yang diinginkan. Contohnya :

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

hasil = kalimat.split(" ")

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

Contoh parsing string

Output program adalah **08/17/1945**

Jadi pertama kita split string berdasarkan spasi, kemudian kita cari token yang diawali dengan angka, dan kita split angka tersebut dengan pemisah '-'. Kemudian susun ulang token sesuai dengan keinginan.

Kegiatan Praktikum

Kasus 7.1 Buatlah program untuk menghitung berapa huruf hidup pada sebuah kalimat per kata

```
a_string = "JoNATHAN"
lowercase = a_string.lower()
total = 0
for x in "aiueo":
    jml = lowercase.count(x)
    total += jml
print(total) #hasil = 3
```

Jadi pertama kita ubah semua huruf menjadi huruf kecil, kemudian kita tetapkan total = 0. Setelah itu kita cari huruf hidup dalam string, dan jika ada huruf hidup maka total yang tadinya 0 akan bertambah 1 terus.

Kasus 7.2 Buatlah program untuk membuat mengubah susunan format tanggal dari YYYY-MMDD menjadi DD-MM-YYYY

```
tgl = "2007-05-12"
hasil = tgl.split("-")
tgl2 = hasil[2]+"-"+hasil[1]+"-"+hasil[0]
print(tgl2) #12-05-2007
```

Kita menggunakan metode yang sama seperti pada parsing string.

Kasus 7.3 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.

```
satu = "kasur rusak"
satu = satu.lower()

satu = ''.join([i for i in satu if i.isalpha()]) #buang semua yang
        bukan alfabet

dua = satu[::-1]
if dua == satu:
    print("palindrom")
else:
    print("bukan")
```

Kita membalik kata dan mencocokkannya dengan kata aslinya. Jika sama maka palindrom, dan sebaliknya.

Kasus 7.4 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".

```
def ambil_kata_kalimat(kalimat, n):
    kalimat = kalimat.lower() #ubah huruf kecil
    hasil_akhir = [] #siapkan tempat hasil
    hasil = kalimat.split() #pecah berdasarkan spasi
    for i in range(0, len(hasil)): #loop per hasil pecah
        tmp = ' '.join(hasil[i:i + n]) #ambil per indeks
        hasil_akhir.append(tmp) #tambahkan ke list

    return hasil_akhir #return

print(ambil_kata_kalimat(text, 2))
```

Hasil = ['ganteng banget', 'banget gweh', 'gweh bro.', 'bro.']

Pertama kita ubah kalimat jadi huruf kecil semua, kemudian kita split. Untuk setiap kata yang sudah dipisah tersebut, maka ambil indeks kata per n buah, misal 1 buah jika n=1, 2 jika n=2. Indeks yang diambil adalah dimulai dari 0, jika n=2 maka ambil 0 dan 1, lalu ambil 1 dan 2, 2 dan 3, 3 dan 4, dst...dilakukan dalam perulangan.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Source Code

```
def anagram(kata1, kata2) :  
    for i in range(0, len(kata2) - 1):  
        if kata1[i] in kata2 :  
            hasil = "benar"  
        else :  
            hasil = "salah"  
    if hasil == "benar" :  
        print("anagram")  
    else :  
        print("bukan")  
  
kata1 = input("masukkan kata pertama = ")  
kata2 = input("masukkan kata kedua = ")  
anagram(kata1, kata2)
```

Output

```
masukkan kata pertama = mata  
masukkan kata kedua = atma  
anagram
```

Penjelasan

Jadi saya mengecek apakah karakter pada kata pertama sama seperti kata kedua menggunakan perulangan dengan mengecek masing-masing index. Jika ada karakter yang berbeda maka hasil == "salah" dan akan mencetak output "bukan".

SOAL 2

Source Code

```
def kata2(kalimat, kata) :  
    kalimat = kalimat.lower()  
    kata = kata.lower()  
    hasil = kalimat.count(kata)  
    return hasil  
  
kalimat = "saya harus makan, makan, dan makan"  
kata = "makan"  
hasil = kata2(kalimat, kata)  
print(f"kata {kata} muncul {hasil} kali ")
```


Output

```
kata makan muncul 3 kali
```

Penjelasan

Pertama kita ubah kalimat dan kata agar huruf kecil semua, kemudian kita hitung ada berapa kata dalam kalimat menggunakan metode **count()**. Dan kita tinggal mencetak hasilnya.

SOAL 3

Source Code

```
def hapus_spasi(kalimat) :  
    kalimat = kalimat.split()  
    kalimat_new = " ".join(kalimat)  
    print(kalimat_new)  
  
kalimat = "saya    tidak suka    memancing ikan    "  
hapus_spasi(kalimat)
```

Output

```
saya tidak suka memancing ikan
```

Penjelasan

Pertama kalimatnya kita split, kemudian kita buat variable baru yang akan berisi kalimat dengan spasi tidak berlebih dengan perintah **join()** dengan satu spasi diantara setiap kata. Setelah itu kita oleh mencetak output

SOAL 4

Source Code

```
def perbandingan_kata(kalimat) :  
    hasil = kalimat.split(" ")  
    hasil.sort(key = len)  
    terpendek = hasil[0]  
    terpanjang = hasil[len(hasil) - 1]  
    print(f"terpendek = {terpendek}")  
    print(f"terpanjang = {terpanjang}")  
  
kalimat = input("masukkan kalimat = ")  
perbandingan_kata(kalimat)
```

Output

```
masukkan kalimat = red snakes and a black frog in the pool  
terpendek = a  
terpanjang = snakes
```

Penjelasan

Pertama seperti biasa kita split dulu stringnya menjadi substring. Kemudian kita bisa mengurutkan substring menurut Panjang karakternya menggunakan metode **sort(key = len)**. Kemudian variable terpendek bisa dibuat yaitu index = 0, dan variable terpanjang dengan index = len - 1. Setelah itu kita bisa mencetak output.

Link Github

https://github.com/JonathanAndrianto123/laporan_AIPro7_71230978.git