



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231042
Nama Lengkap	Revaldo Fransisco Hohary
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

A. Pengantar String

String adalah tipe data yang menggunakan untuk kalimat atau serangkaian karakter. Dalam string tersimpan beberapa jenis karakter atau huruf yang disimpan dalam kode ASCII. Pada Bahasa C tidak memiliki data string. String biasanya disebut sebagai tipe data karakter / array of character / list of character.

B. Pengaksesan string dan Manipulasi String

String dapat dibuat sederhana sebagai berikut:

```
namasaya = "Revaldo Hohary"
namapeliharaan = "Meoong"
temansaya2 = 'frealy abbyul'
terserah = "jika" + 'jus'

print(terserah)
print(namasaya[0]) #'R'
print(namasaya[9]) #'H'
print(namapeliharaan[1]) #'e'

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

Jadi, string pertama kali dibuat dengan deklarasi variabel dan langsung diisi dengan data. Anda dapat mengakses string sebagai satu kesatuan dengan menyebut nama variabelnya atau sebagai indeks per huruf dengan menyebutkannya. Indeks string dimulai dari 0 seperti indeks list.

Jumlahnya harus bulat, bukan pecahan.

String disimpan secara urut dalam memori komputer melalui daftar huruf dengan indeks yang dimulai dari nol. String yang mengandung kata "Apple" disimpan di memori komputer ditunjukkan pada Tabel 7.1.

A	P	P	L	E
[0]	[1]	[2]	[3]	[4]

Tabel 7.1 : Bentuk String dalam memory

C. Operator dan Metode String

Operator In

Untuk memeriksa kalimat merupakan substring dari suatu kalimat adalah dengan menggunakan operator in.

```
kalimat = "saya mau makan"
data = "saya"
print(data in kalimat) #True
print("mau" in kalimat) #True
print("dia" in kalimat) #False
```

Selain operator in, String memiliki kemampuan untuk melakukan perbandingan, yang menghasilkan True atau False.

```
if "kamu" > "dia":
    print("Ya") #Ya
else:
    print("Tidak")
if "kami" == "kami":
    print("Sama") #Sama
```

Fungsi Len

Operator len() dapat digunakan untuk menentukan panjang string, atau jumlah karakternya. Untuk menampilkan huruf terakhir dari sebuah string, kita harus menggunakan indeks string yang ke-len(-1), karena indeks dimulai dari 0.

Salah satu contoh program Python adalah:

```
kalimat = "universitas kristen satya wacana"
print(len(kalimat)) #output 32

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'
```

Traversing String

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

- Dilakukan dengan akses terhadap indeks

```
kalimat = "indonesia Merdeka"
i = 0
while i < len(kalimat):
    print(kalimat[i],end='')
    i += 1
```

- Dilakukan tanpa akses terhadap indeks secara otomatis

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

String Slice

String slice menampilkan substring pada string dengan indeks dari awal tertentu hingga akhir-1 tertentu. Dalam sintaks, [awal:akhir] digunakan. Awal atau akhir dapat dikosongkan. Bagian pertama dimulai dengan nol.

```
kalimat = "cerita rakyat"
awal = 0
akhir = 6
print(kalimat[awal:akhir]) #cerita
print(kalimat[7:len(kalimat)]) #rakyat
print(kalimat[:5]) #cerit
print(kalimat[5:]) #a rakyat
print(kalimat[:]) #cerita rakyat
```

Data string bersifat immutable karena tidak dapat diubah. Bisa diubah saat program berjalan, contoh:

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

Disimpan dalam variabel yang berbeda agar dapat diubah.

```
kalimat = "satu"
kalimat_baru = kalimat[0] + "alah" #salah
```

Berikut ini adalah beberapa metode string yang paling umum 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()

Tabel 7.2 : Method

dan banyak lagi. Ingat bahwa jika Anda menggunakan salah satu fungsi atau teknik di atas untuk mengembalikan string, Anda hanya dapat mengembalikan string baru daripada mengubah string aslinya karena string tidak dapat diubah. Untuk mendapatkan daftar lengkap dari kategori ini, kunjungi <https://docs.python.org/library/stdtypes.html#string-methods>.

Operator * dan + pada String

Python memiliki kemampuan khusus untuk operator + dan *. Operator +, yang biasanya digunakan untuk menjumlahkan bilangan, dapat digunakan untuk menggabungkan dua buah string, dan operator *, yang biasanya digunakan untuk mengkalikan bilangan, dapat digunakan untuk menampilkan string yang mengandung jumlah perkaliannya. Lihat kode berikut:

```
kata1 = "saya"
kata2 = "makan"
kata3 = kata1 + " " + kata2
print(kata3) #hasil adalah penggabungan: saya makan
kata4 = "ulang"
print(kata* 4) #hasil adalah ulangulangulangulang
kata4 = "ulang "
print(kata4 * 2) #hasil adalah ulang ulang
```

D. Parsing String

Metode untuk mendapatkan, menemukan, atau mengubah bagian string tertentu dikenal sebagai parsing string. Lihat contoh ini: "Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka."

seperti yang ditunjukkan oleh string di atas, diinginkan untuk mengambil tanggal bulan tahun dari kalimat di atas dan diformat kembali pada 08/17/1945. saya dapat memotong string dengan memotong spasi menjadi token, misalnya: "Saudara-saudara", "pada", "tanggal", "17-08-1945", "Indonesia", dan "merdeka".

Selanjutnya, cari token yang dimulai dengan angka dan kemudian pecahkan angka tersebut dengan pemisah "-". Kemudian, susun ulang token dari langkah sebelumnya sehingga berbentuk seperti yang diinginkan. Berikut ini adalah ilustrasi penyelesaiannya:

```
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])
```

Hasil dari keluaran program adalah 08/17/1945

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

Pada soal pertama kita diminta untuk membuat program anagram yang menentukan apakah kata itu anagram atau bukan. Berikut source code dan outputnya :

```
latihan7.1.py > ...
1 # Soal 1
2 def anagram():
3     x = sorted(input1)
4     y = sorted(input2)
5
6     if x == y or y == x:
7         print("kata tersebut anagram")
8     else:
9         print("kata itu bukanlah anagram")
10
11 input1 = input("Masukkan kata pertama: ").lower()
12 input2 = input("Masukkan kata kedua: ").lower()
13 anagram()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> & C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh\latihan7.1.py"

Masukkan kata pertama: mata

Masukkan kata kedua: atam

Kata tersebut anagram

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> |

Gambar 7.3 : Source code dan Output

SOAL 2

Pada soal kedua kita diminta menemukan kata yang muncul disetiap kata yang dihitung. Berikut source code dan outputnya:

```
latihan7.2.py > ...
1 # Soal 2
2 def frekuensi_kata(input1, input2):
3     kalimat_murni = input1.replace(".", "").replace("!", "").replace("?", "").replace(":", "").replace(";", "").replace(",", "").replace(" ", "")
4     kalimat_new = kalimat_murni.lower()
5     input2 = input2.lower()
6     input1 = input1.strip('.,!?:;')
7     input2 = input2.strip('.,!?:;')
8     kata1 = kalimat_new.split()
9
10    frekuensi = 0
11    for kata2 in kata1:
12        if kata2 == input2:
13            frekuensi += 1
14    return frekuensi
15
16 input1 = input("Masukkan kalimat: ")
17 input2 = input("Kata yang dicari: ")
18 frekuensi = frekuensi_kata(input1, input2)
19 print(f"{input2} ada {frekuensi} pada kalimat tersebut")
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> & C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh\latihan7.2.py"

Masukkan kalimat: kata ini adalah frekuensi kata

Kata yang dicari: kata

kata ada 2 pada kalimat tersebut

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> |

Gambar 7.4 : Hasil Output dan source codenya

SOAL 3

Pada soal ke tiga kita diminta perbaiki kalimat yang spasinya berlebihan. Berikut source code dan hasilnya:

```
2 def spasi_berlebih(): # Mendefinisikan
3     x = input.lower()
4     x = input.split() # Memecahkan atau memisahkan
5     kalimatnormal = " ".join(x)
6     return kalimatnormal
7
8 def spasi_berlebih(): # Mendefinisikan
9     x = input.lower()
10    x = input.split() # Memecahkan dan memisahkan kalimat pada inputan
11    hasil = [] # Menyediakan Hasil buat di isi nantinya
12    for kata in x: # Perulangan
13        if kata[0] == " ": # Jika kata dipanggil 0 sama dengan kosong / spasi
14            continue # Continue atau lanjutkan
15        else: # Jika tidak maka diisi ke hasil
16            hasil.append(kata)
17    return " ".join(hasil)
18
19 input = input("Masukkan kalimat yang mau diperbaiki: ") # Meminta pengguna memasukan inputan
20 kalimatnormal = spasi_berlebih()
21 print(f"Setelah diperbaiki : {kalimatnormal}") # Memunculkan kalimatnormal yang sudah diisi dengan def spasi_berlebih
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Asus/Documents/UKDM/PrakAlPro/Pertemuan Ketujuh/latihan7.3.py"

Masukkan kalimat yang mau diperbaiki: Jika kamu masih ragu untuk menerima

Setelah diperbaiki : Jika kamu masih ragu untuk menerima

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh>

Gambar 7.5 : Hasil source code dan outputnya

SOAL 4

Pada soal ke empat kita diminta untuk mencari kata terkecil dan terbesar dalam sebuah kalimat yang di inputkan. Berikut source code dan hasilnya:

```
1 def terpendek_terpanjang(): # Mendefinisikan
2     x = input1.split() # Memecahkan dan memisahkan kalimat
3     pendek = x[0]
4     panjang = x[0]
5     for kata in x: # Untuk kata di dalam x akan di ulang jika kat lebih kecil dari pendek
6         if len(kata) < len(pendek):
7             pendek = kata # Maka pendek diisi dengan kata
8         elif len(kata) > len(panjang): # Dan jika kata lebih besar dari panjang
9             panjang = kata # Maka panjang diisi dengan kata
10    return pendek, panjang
11
12 input1 = input("Masukkan kalimat anda: ") # Inputan dari user
13 pendek, panjang = terpendek_terpanjang() # pendek dan panjang diisi def terpendek_panjang
14 print(f"Terpendek adalah : {pendek} dan Terpanjang adalah : {panjang}") # Memunculkan hasil
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh> C:\Users\Asus\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/Asus/Documents/UKDM/PrakAlPro/Pertemuan Ketujuh/latihan7.4.py"

Masukkan kalimat anda: membaca adalah kunci untuk masa depan yang cerah

Terpendek adalah : masa dan Terpanjang adalah : membaca

PS C:\Users\Asus\Documents\UKDM\PrakAlPro\Pertemuan Ketujuh>

Gambar 7.6 : Hasil source code dan outputnya

E. Link Github

<https://github.com/FrealY0901/PraAlPro7.git>