Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230976**
Nama Lengkap	**Galih Pramana Chandra Prasetya**
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.

Pengantar String

String atau **str** merupakan kumpulan data char atau karakter yang tersimpan secara urut (*text sequence*). String digunakan dalam program komputer untuk menyimpan kalimat, baik panjang ataupun pendek dalam kode ASCII.

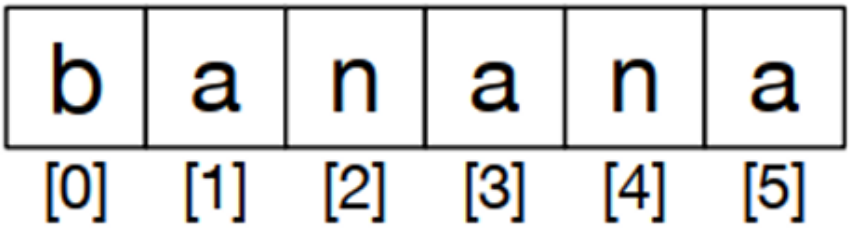
Pengaksesan String dan Manipulasi String

String dapat dibuat dengan sederhana menggunakan variabel:

```
namasaya = "Galih Pramana"
temansaya1 = "Rainie Fanita"
temansaya2 = 'Freire Hanan'
temansaya3 = "Jona" + 'Ruben'
print(temansaya3)
print(namasaya[0]) #'G'
print(namasaya[9]) #'m'
print(namasaya[-1]) #'a'
print(temansaya1[1]) #'a'
huruf = temansaya2[0]
print(huruf) #'F'
```

String dibuat dengan deklarasi variabel dan langsung diisi dengan data, string dapat diakses sebagai satu kesatuan dengan menyebut nama variabelnya, atau per huruf dengan menyebutkan indeksnya. Indeks pada string dimulai dari 0. Indeks string haruslah berupa bilangan bulat, bukan pecahan.

Pada memory komputer, string disimpan secara urut menggunakan list yang berisi huruf-huruf dengan indeks yang dimulai dari nol.



Gambar 1 diambil dari modul praktikum alpro

Operator dan Metode String

OPERATOR in

Pada string kita dapat memeriksa apakah suatu kalimat merupakan substring dari suatu kalimat lain dengan menggunakan operator in dan dengan dilakukan perbandingan (comparison) yang juga menghasilkan output yang sama (True/False).

```
kalimat = "Hari ini cuaca cerah"
data = "Hari"
contoh = "cerah"
print(contoh in kalimat)#True
print(data in kalimat) #True
print("Hari" in kalimat) #True
print("hari" in kalimat) #False
print("ini" in kalimat) #True
print("cerah" in kalimat) #True
print("Mendung" in kalimat) #False
```

Pada pengecekan diatas, data yang tidak ada dalam kalimat akan menghasilkan output false.

String juga dapat dilakukan perbandingan (comparison) yang juga menghasilkan True atau False.

Contoh comparison,

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

FUNGSI len

Fungsi ini digunakan untuk mengetahui berapa panjang (berapa jumlah karakter) dari sebuah string. Untuk menampilkan huruf terakhir dari sebuah string kita harus menggunakan indeks string yang ke- len(<string> -1), sebab indeks dimulai dari 0.

Berikut ini adalah contoh penggunaan fungsi 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'

nama = "Galih Pramana Chandra Prasetya"
print(len(nama)) #output 30

last = nama[len(nama)-1]
print(last) #output 'a

last2 = nama[-1]
print(last2) #output 'a'
```

TRAVERSING STRING

Traversing atau memisahkan string yaitu menampilkan huruf demi huruf dengan menggunakan loop yang dilakukan per huruf dengan 2 cara:

• Dilakukan dengan akses terhadap indeks

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

• Dilakukan tanpa akses terhadap indeks secara otomatis

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

STRING SLICE

String slice adalah cara untuk menampilkan potongan teks dalam sebuah string dengan menentukan indeks mulai dari titik awal tertentu hingga titik akhir -1 tertentu Sintaksnya menggunakan <String >[awal:akhir]. Bagian awal atau akhir boleh dikosongkan. Bagian awal dimulai dari 0.

Berikut ini adalah contoh penggunaan string slice,

```
kalimat = "Kampung halaman"
awal = 0
akhir = 6
```

```
print(kalimat[awal:akhir]) #kampun
print(kalimat[7:len(kalimat)]) # halaman
print(kalimat[:5]) #kampu
print(kalimat[5:]) #ng halaman
print(kalimat[:]) #kampng halaman
```

String merupakan data yang bersifat immutable, yaitu data tersebut tidak bisa diubah saat program berjalan, hanya bisa diinisialisasi saja.

Contoh,

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

Apabila program diatas dijalankan akan erorr (#TypeError: 'str' object does not support item assignment). Maka agar bisa diubah harus disimpan dalam variable yang berbeda.

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

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 dinputkan	string.endswith()
startswith()	mengetahui apakah suatu string diawali dengan string yang dinputkan	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()

Gambar 2 diambil dari modul praktikum alpro

Operator * dan + pada String

Pada Python operator **+** dan ***** memiliki kemampuan khusus. Operator * yang biasa digunakan untuk mengkalikan bilangan bisa digunakan untuk menampilkan string sejumlah perkaliannya. Sedangkan operator + yang biasanya digunakan untuk menjumlahkan bilangan bisa digunakan untuk menggabungkan dua buah string.

Contoh pengguaan operator * dan + pada String,

```
kata1 = "Minum"
kata2 = "Air"
kata3 = kata1 + " " + kata2
print(kata3)#hasil adalah penggabungan: Minum Air

kata4 = "loop"
print(kata4 * 4) #hasil adalah looplooploploop
kata4 = "loop "
print(kata4 * 2) #hasil adalah loop lop
```

Parsing String

Parsing string adalah cara menelurusi string bagian demi bagian untuk mendapatkan / menemukan / mengubah bagian string yang diinginkan.

Contoh parsing string,

```
kalimat = "Aku lahir pada tanggal 22-03-2005"
hasil = kalimat.split(" ")
print(hasil)

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

```
['Aku', 'lahir', 'pada', 'tanggal', '22-03-2005']
22-03-2005
['22', '03', '2005']
03/22/2005
```

Pada program diatas yang ditelusuri adalah angka, dan Menyusun ulang angka tersebut menjadi format tanggal MM/DD/YYYY dan mengeluarkan hasil 3/22/2005.

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

Source Code:

```
#Latihan Mandiri 7.1
def anagram(h1, h2):
    h1 = h1.lower()  # Mengubah semua huruf dalam h1 menjadi huruf kecil
    h2 = h2.lower()  # Mengubah semua huruf dalam h2 menjadi huruf kecil
    if sorted(h1) == sorted(h2):
        print("anagram")
    else:
        print("Tidak anagram")

h1 = str(input("Masukkan kata 1: "))
h2 = str(input("Masukkan kata 2: "))

anagram(h1, h2)
```

Output:

```
Masukkan kata 1: mata
Masukkan kata 2: atma
anagram

Masukkan kata 1: kasur
Masukkan kata 2: rusak
Anagram

Masukkan kata 1: makan
Masukkan kata 2: minum
Tidak anagram
```

Penjelasan:

- Input dari Pengguna:
  - Program meminta user untuk menginput dua kata yang ingin diperiksa apakah kedua kata tersebut anagram atau tidak.
- Pemrosesan Input:
  - Kata-kata yang diinput oleh user disimpan dalam variabel **h1** dan **h2**.
- Fungsi anagram:
  - Fungsi ini digunakan untuk memeriksa apakah dua string adalah anagram satu sama lain.
- Konversi ke Huruf Kecil:
  - menggunakan metode **lower()** untuk mengubah Kedua kata (**h1** dan **h2**) menjadi huruf kecil.
- Pengecekan Anagram:

- Kedua kata yang telah diubah menjadi huruf kecil diurutkan secara alfabetis.
- Hasil pengurutan kedua kata tersebut dibandingkan. Jika hasil pengurutan sama, maka kedua kata tersebut adalah anagram.

- Output:
  - Jika kedua kata adalah anagram, program mencetak "anagram".
  - Jika tidak, maka mencetak "Tidak anagram"

SOAL 2

Source Code:

```
#Latihan Mandiri 7.2
def hitung_frekuensi(kalimat, kata) :
    kalimat = kalimat.lower()
    kata = kata.lower()
    hasil = kalimat.count(kata)
    print(f'{kata} ada {hasil} buah.')

kalimat = input("masukkan kalimat : ")
kata = input("masukkan kata : ")

hitung_frekuensi(kalimat, kata)
```

Output:

```
masukkan kalimat : Saya mau makan. Makan itu wajib. Mau siang  atau malam
saya wajib makan
masukkan kata : makan
makan ada 3 buah.
```

Penjelasan:

Kode ini adalah program untuk menghitung frekuensi kemunculan suatu kata dalam sebuah kalimat. Fungsi **hitung_frekuensi** digunakan untuk melakukan perhitungan tersebut. User diminta untuk menginput 2 nilai variabel, yaitu kalimat dan kata, kemudian diubah menjadi huruf kecil menggunakan metode **lower()**. Kemudian, menggunakan metode **count()** untuk menghitung jumlah kemunculan kata. Setelah itu, program akan menampilkan jumlah kata dalam kalimat tersebut

SOAL 3

Source Code:

```
#Latihan Mandiri 7.3
def hapus_spasi(kalimat):
    kalimat_tanpa_spasi = " ".join(kalimat.split())
    return kalimat_tanpa_spasi

kalimat_input = input("Masukkan kalimat dengan spasi yang ingin dihapus:
")
print(hapus_spasi(kalimat_input))
```

Output:

```
Masukkan kalimat dengan spasi yang ingin dihapus: saya     makan    ikan
saya makan ikan


Masukkan kalimat dengan spasi yang ingin dihapus: saya      makan
nasi
saya makan nasi
```

Penjelasan:

- Input dari Pengguna:
  - Program meminta pengguna untuk memasukkan kalimat yang ingin dihapus spasi berlebihnya.
- Fungsi hapus_spasi:
  - digunakan untuk menghapus spasi berlebih dalam sebuah kalimat.
- Penghapusan Spasi Berlebih:
  - menggunakan metode **split()** untuk memisahkan kata-kata, kemudian **join()** untuk menggabungkan kata-kata tersebut kembali tanpa spasi berlebih.

SOAL 4

Source Code:

```
#Latihan Mandiri 7.4
def cek_kata(kalimat):
    kata = kalimat.split()
    terpendek = min(kata, key=len)
    terpanjang = max(kata, key=len)
    print("Kata terpendek:", terpendek)
    print("Kata terpanjang:", terpanjang)
```

```
kalimat = input("Masukkan kalimat: ")
cek_kata(kalimat)
```

Output:

```
Masukkan kalimat: "red snakes and a black frog in the pool
Kata terpendek: a
Kata terpanjang: snakes
```

Penjelasan:

- Meminta user untuk menginput kalimat
- Memecah kalimat yang diinput menjadi kata-kata menggunakan metode **split()**
- Menggunakan fungsi **min()** dan **max()** untuk mencari kata terpendek dan terpanjang dalam kalimat tersebut berdasarkan panjangnya.
- Program akan menampilkan kata terpendek dan terpanjang

Link Github: https://github.com/GalihPramana/Praktikum-Alpro-71230976.git