Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71231005**
Nama Lengkap	**Josephine Marcelia**
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

Pengertian String

String pada Python merupakan sekumpulan karakter yang saling berurutan dan biasanya dikelilingi oleh tanda kutip tunggal (') atau ganda ("). Karakter-karakter tersebut dapat diakses dan diubah sesuai dengan indeksnya, yang dimulai dari 0 sampai panjang string dikurangi 1. String digunakan untuk merepresentasikan teks dan dapat berisi huruf, angka, simbol, atau bahkan karakter khusus seperti baris baru (\n).

Berikut merupakan contoh penggunaan string untuk format teks :

```
nama = "Josephine"
usia = 20
teks = "Halo, nama saya {} , usia saya {} tahun.".format(nama, usia)
print(teks)
```

Output :

```
Halo, nama saya Josephine , usia saya 20 tahun.
```

String juga dapat diolah dengan bantuan berbagai macam method yang disediakan oleh Python, seperti pembuatan kapital (uppercase), membuat huruf kecil (lowercase), mencari substring, dan mengganti substring.

Berikut adalah contoh penggunaan beberapa method tersebut :

```
string= 'Hello, world!'

print(string.upper())
print(string.lower())
print(string.find('world'))
print(string.replace('world', 'Python'))
```

Output :

```
HELLO, WORLD!
hello, world!
7
Hello, Python!
```

Manipulasi String

Pada memori komputer, string disimpan secara urut menggunakan list yang berisi huruf-huruf dengan indeks yang dimulai dari nol. Kita dapat memilih salah satu atau beberapa karakter berdasarkan index yang ada.

Berikut merupakan contoh-contoh manipulasi string :

1. Penggabungan Kalimat

Kita dapat menggabungkan kalimat dengan menggunakan operator +

```
string1 = "Pemrograman"
string2 = "Python"
hasil = string1 + " " + string2
print(hasil)
```

Output :

```
Pemrograman Python
```

1. Pemotongan Kalimat

Pada contoh dibawah ini bisa dilihat bahwa karakter yang diambil adalah karakter dari indeks 0 sampai dengan 9. Sehingga indeks 10 dan seterusnya tidak termasuk.

```
teks = "Josephine Marcelia"
substring = teks[0:9]
print(substring)
```

Output :

```
Josephine
```

3. Penggantian Kalimat

Dengan menggunaskan fungsi replace(), kita dapat mengubah string pada python

```
string = "Saya suka pizza"
new_string = string.replace("pizza", "burger")
print(new_string)
```

Output :

```
Saya suka burger
```

Operator dan Metode String

OPERATOR in

Contoh :

Operator in pada program dibawah ini dapat digunakan untuk memeriksa apakah kata pizza dan pasta ada pada string makanan atau tidak. Jika ada program akan menghasilkan output True, tetapi jika tidak program akan menghasilkan output False.

```
makanan = "pizza, burger, sandwich, hotdog"
if "pizza" in makanan:
    print("True, 'pizza' ada di kalimat makanan")
else:
    print("False, 'pizza' tidak ada di kalimat makanan")

if "pasta" in makanan:
    print("True, 'pasta' ada di kalimat makanan")
else:
    print("False, 'pasta' tidak ada di kalimat makanan")
```

Output :

```
True, 'pizza' ada di kalimat makanan
False, 'pasta' tidak ada di kalimat makanan
```

FUNGSI len

Pada Python, Fungsi len digunakan untuk menghitung panjang dari sebuah string, yaitu jumlah karakter yang ada dalam string , termasuk spasi dan symbol.

```
sentences = [
    "Nama saya Josephine Marcelia.",
    "Saya kuliah di Universitas Krosten Duta Wacana.",
    "Saya tingggal di kos lerem, Klitren, Yogyakarta.",
    "Hobi saya adalah menari dan menyanyi.",
]

for sentence in sentences:
    sentence_length = len(sentence)
    print(f"Panjang dari kalimat '{sentence}' adalah {sentence_length} karakter.")
```

Output :

```
Panjang dari kalimat 'Nama saya Josephine Marcelia.' adalah 29 karakter.
Panjang dari kalimat 'Saya kuliah di Universitas Krosten Duta Wacana.' adalah 47 karakter.
Panjang dari kalimat 'Saya tingggal di kos lerem, Klitren, Yogyakarta.' adalah 48 karakter.
Panjang dari kalimat 'Hobi saya adalah menari dan menyanyi.' adalah 37 karakter.
```

Traversing String

Traversing string pada Python merujuk pada proses pengaksesan setiap karakter dalam sebuah string secara berurutan untuk melakukan operasi tertentu, seperti pencarian, manipulasi, atau analisis. Hal ini melibatkan iterasi melalui setiap karakter dalam string, satu per satu, biasanya menggunakan struktur kontrol seperti loop (for atau while).

Berikut merupakan contoh program traversing string dalam Python yang mencetak setiap karakter string dalam urutan terbalik:

```python
def traverse_reverse(string):
    reversed_string = ""
    for char in reversed(string):
        reversed_string += char
    return reversed_string

input_string = input("Masukkan sebuah string: ")
reversed_string = traverse_reverse(input_string)
print("String terbalik:", reversed_string)
```

Output :

```
Masukkan sebuah string: python
String terbalik: nohtyp
```

String Slice

String slice pada Python digunakan untuk memanipulasi string dengan cara mengambil beberapa bagian atau karakter dari string. Hal ini dilakukan dengan menggunakan indeks yang ditentukan, dan menetapkan nilai awal dan akhir dari slicing. Slice string akan mengembalikan substring yang dimulai dari indeks awal dan diakhiri sebelum indeks akhir.

Contoh :

```
string = "Learn Python!"

print(string[0:5])
print(string[6:12])
print(string[::2])
print(string[-1])
print(string[:-1])
```

Output :

```
Learn
Python
LanPto!
!
Learn Python
```

Beberapa Method String yang sering digunakan :

Nama Fungsi	Kegunaan	Penggunaan
Capitalize	Mengubah string menjadi huruf besar	String.capitalize()
Count	Menghitung jumlah substring yang muncul dari sebuah string	String.count()
Starts With	Mengetahui apakah suatu string diawali dengan string yang diinputkan	String.startswith()
Ends With	Mengetahui apakah suatu string diakhiri dengan string yang diinputkan	String.endswith()
Find	Mengembalikan indeks pertama string jika ditemukan string yang dicari	String.find()
Isupper/lower	Mencari tahu apakah sebuah string kapital atau kecil	String.isupper() String.islower()
Is Digit	Mencari tahu apakah sebuah string sebuah digit atau tidak	String.isdigit()
Strip	Menghapus whitespace didepan dan dibelakang string	String.strip()
Split	Memisah sebuah string berdasarkan sebuah parameter	String.split(parameter)

BAGIAN 2: LATIHAN MANDIRI

Lat 7.1

```
def check_anagram(word1, word2):
    if sorted(word1) == sorted(word2):
        print(f"{word1} anagram dengan {word2}.")
    else:
        print(f"{word1} tidak anagram dengan {word2}.")
word1 = input("Masukkan kata pertama: ")
word2 = input("Masukkan kata kedua: ")

check_anagram(word1, word2)
```

```
Masukkan kata pertama: mata
Masukkan kata kedua: atma
mata anagram dengan atma.
```

```
Masukkan kata pertama: mata
Masukkan kata kedua: mana
mata tidak anagram dengan mana.
```

Penjelasan :

- Pada fungsi bernama check_anagram saya menggunakan dua parameter, yaitu word1 dan word2, yang akan digunakan untuk memeriksa apakah kata yang diinput merupakan anagram atau bukan

- Pada baris if sorted(word1) == sorted(word2), kedua kata yang dimasukkan (word1 dan word2) diurutkan secara alfabetis menggunakan fungsi sorted(). Jika kedua kata tersebut memiliki susunan huruf yang sama setelah diurutkan, maka mereka dianggap sebagai anagram.

- Jika kata pada word1 anagram dengan word2 maka output akan menghasilkan “(word1) anagram dengan (word2)”

- Jika kata pada word1 tidak anagram dengan word2, maka output akan menghasilkan “(word1) tidak anagram dengan (word2)”

Lat 7.2

```
teks = input("Masukkan Kalimat: ")
kata = input("Masukkan kata: ")

teks = ''.join(ch for ch in teks if ch not in ('!', '.', '?', ':', ';', ',','"')).lower()

def hitung_kata(teks, kata):

    words = teks.split()

    hitung = 0
    for j in words:
        if j == kata:
            hitung += 1

    print(f"{kata} ada {hitung} buah.")

hitung_kata(teks, kata)
```

```
Masukkan Kalimat: "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan"
Masukkan kata: makan
makan ada 3 buah.
```

Penjelasan :

- Saya menggunakan fungsi bernama hitung_kata yang menerima dua parameter, yaitu teks (kalimat yang dimasukkan oleh pengguna) dan kata (kata yang ingin dicari jumlah kemunculannya).

- Pada baris ini teks = ''.join(ch for ch in teks if ch not in ('!', '.', '?', ':', ';', ',','')).lower() saya gunakan untuk menghilangkan tanda baca seperti tanda seru, titik, tanya, dua titik, titik koma, dan koma dari kalimat menggunakan ekspresi generator. Setelah itu, semua huruf dalam kalimat tersebut dikonversi menjadi huruf kecil agar pencarian kata menjadi case-insensitive.

- Pada baris hitung = 0, variabel hitung saya gunakan untuk menyimpan jumlah kemunculan kata yang dicari.

- Pada baris for j in words:, program melakukan iterasi pada setiap kata dalam list words.

- Pada baris if j == kata: berarti jika kata yang sedang dievaluasi sama dengan kata yang ingin dicari, maka hitung akan bertambah satu.

- Program diatas kemudian akan mencetak hasil jumlah kata yang ada pada kalimat pada fungsi print(f" {kata} ada {hitung} buah.")

Lat 7.3

```
def remove_extra_spaces(string):
    return ' '.join(string.split())

input_string = input("Masukan kalimat: ")

normalized_string = remove_extra_spaces(input_string)

print("hasil:", normalized_string)
```

```
Masukan kalimat: saya tidak    suka memancing ikan
hasil: saya tidak suka memancing ikan
```

Penjelasan :

- Fungsi bernama remove_extra_spaces akan mengambil satu parameter, yaitu string, yang merupakan string yang akan diproses.
- Pada baris return ' '.join(string.split()): , string digunakan untuk menghapus spasi ekstra. Dimana string akan dipecah menjadi kata-kata menggunakan metode split(). Kemudian, kata-kata tersebut digabungkan kembali menggunakan metode join() dengan spasi sebagai pemisah.
- Pada baris normalized_string = remove_extra_spaces(input_string):, program akan memanggil fungsi remove_extra_spaces untuk menghapus spasi ekstra dari kalimat yang dimasukkan oleh pengguna.
- Kemudian program akan mencetak hasil dari proses penghapusan spasi ekstra, yaitu kalimat yang sudah dinormalisasi. Melalui fungsi print("hasil:", normalized_string):

Lat 7.4

```
def cari_kata_terpendek_dan_terpanjang(kalimat):
    kata_kalimat = kalimat.split()
    kata_terpendek = kata_kalimat[0]
    kata_terpanjang = kata_kalimat[0]
    for kata in kata_kalimat:
        if len(kata) < len(kata_terpendek):
            kata_terpendek = kata
        if len(kata) > len(kata_terpanjang):
            kata_terpanjang = kata
    return kata_terpendek, kata_terpanjang
kalimat = input("Masukkan kalimat: ")
kata_terpendek, kata_terpanjang = cari_kata_terpendek_dan_terpanjang(kalimat)

print("kata terpendek:", kata_terpendek)
print("kata terpanjang:", kata_terpanjang)
```

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

Penjelasan :

- Pada fungsi bernama cari_kata_terpendek_dan_terpanjang kita menerima satu parameter yaitu kalimat, yang merupakan kalimat yang akan diproses.
- Pada program diatas kalimat akan diubah menjadi daftar kata menggunakan split() dan disimpan dalam variabel kata_kalimat
- kata_terpendek dan kata_terpanjang kemudiandiinisialisasi dengan kata_kalimat[0]
- pada baris for kata in kata_kalimat:, program melakukan iterasi pada setiap kata dalam list kata_kalimat.
- if len(kata) < len(kata_terpendek)::artinya jika panjang kata yang sedang dievaluasi lebih kecil dari panjang kata terpendek yang saat ini, maka kata terpendek akan diperbarui dengan kata yang sedang dievaluasi.
- if len(kata) > len(kata_terpanjang): artinya jika panjang kata yang sedang dievaluasi lebih besar dari panjang kata terpanjang yang saat ini, maka kata terpanjang akan diperbarui dengan kata yang sedang dievaluasi.
- fungsi return kata_terpendek, kata_terpanjang: saya gunakan untuk mengembalikan dua nilai, yaitu kata terpendek dan kata terpanjang.
- Pada baris kata_terpendek, kata_terpanjang = cari_kata_terpendek_dan_terpanjang(kalimat):, program akan memanggil fungsi cari_kata_terpendek_dan_terpanjang untuk mencari kata terpendek dan terpanjang dalam kalimat yang dimasukkan oleh pengguna.
- Kemudian Program akan mencetak kata terpendek yang ditemukan dalam kalimat melalui fungsi print("kata terpanjang:", kata_terpanjang)

Link Github:

https://github.com/Josephinemrc/Tugas7-PrakAlpro.git