



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
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.

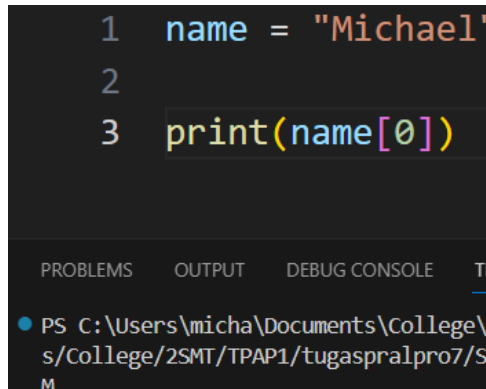
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Pengaksesan dan Manipulasi String

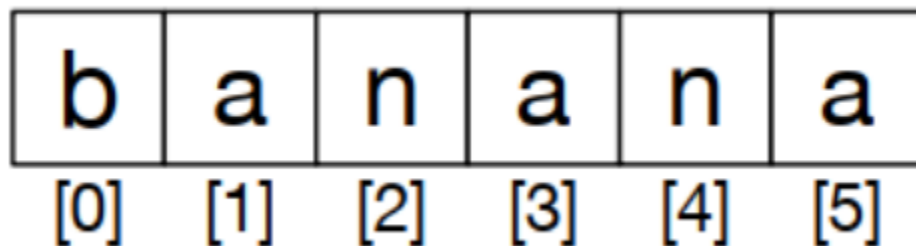
Dalam sebuah string, semua jenis tulisan yang berada di dalam tanda kutip (") akan terhitung sebagai 1 angka indeks. Angka indeks ini bisa digunakan untuk mengambil tulisan tertentu itu. Contohnya adalah:

```
1 name = "Michael"
2
3 print(name[0])
```

The image shows a screenshot of a code editor with a dark background. The code consists of three lines: '1 name = "Michael"', '2', and '3 print(name[0])'. Below the code, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'OUTPUT' tab is selected, showing a single line of output: 'M'. The file path in the terminal is partially visible: 'PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralpro7\Str'.

Gambar string dan penggunaan indeks string

Dari contoh diatas diberi string "name" yang berisi "Michael", dan jika diprint `name[0]` ← untuk memanggil string "name" indeks variabel ke-0, maka output akan diberikan M yaitu huruf pertama dari isi string. Dengan cara ini dapat dengan mudah mengambil, mengecek, dan mengeluarkan bagian tertentu sebuah string. Untuk memperjelas lihat gambar di bawah ini:



Gambar contoh visualisasi nomor indeks sebuah string

Dari gambar di atas sudah dapat diketahui posisi huruf dan angka indeks yang berkorelasi, sehingga jika ingin mengetahui apakah kata tersebut memiliki huruf tertentu dapat menggunakan for loop untuk mengecek angka indeks huruf tersebut. Untuk contoh singkat:

```
11 food = "banana"
12
13 for i in range(len(food)):
14     if food[i] == "a":
15         print("Yes")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspr
• s/College/2SMT/TPAP1\tugaspralpro7\SrcCd.py
Yes
Yes
Yes

Gambar penggunaan loop untuk mengecek alfabet kata

Dari contoh di atas, diketahui kata banana dan digunakan for loop untuk mengeluarkan indeks dari 0 sampai 5 menggunakan range() dan len(food) untuk mengetahui panjang sebuah string. Lalu dengan menggunakan i sebagai angka indeks dapat dikeluarkan huruf satu per satu dan jika ditemukan huruf "a" maka "Yes" akan di print. Dari kode di atas "Yes" di print 3 kali sesuai dengan jumlah huruf a yang ada di kata banana.

MATERI 2: Operator dan Metode String

String juga dapat digunakan untuk mengecek apakah sebuah kata ada dalam sebuah kalimat string, Contohnya:

```
5 work = "i am going to work"
6 obj = "work"
7 print(obj in work)
8 print("going" in work)
9 print("me" in work)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

• PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspral
s/College/2SMT/TPAP1\tugaspralpro7\SrcCd.py
True
True
False

Gambar contoh penggunaan Logic dengan string

kata in pada "obj in work" dapat ditafsirkan sebagai apakah obj ada di dalam work, berarti apakah kata "work" ada di dalam variabel kalimat "work", Jika ada maka output akan menghasilkan True, jika tidak maka output akan False.

Ada juga fungsi len() dari contoh-contoh sebelumnya yang digunakan untuk menghitung panjang sebuah string. Fungsi ini digunakan untuk mengeluarkan huruf per huruf dari sebuah kata dalam 2 cara, dengan indeks dan tanpa indeks. Dengan indeks tampak seperti di bawah:

```
word = "hello"
for i in range(len(word)):
    print(word[i])

word = "hello"
i = 0
while i < len(word):
    print(word[i])
    i += 1
```

Gambar contoh print huruf per huruf dengan indeks

Dengan indeks, perlu kurung kotak agar dapat mengeluarkan huruf dengan angka indeks yang berkolerasi. Sedangkan tanpa indeks tampak seperti di bawah:

```
sentence = "My family"

for item in sentence:
    print(item)
```

Gambar contoh print tanpa indeks

Contoh di atas langsung menggunakan huruf per huruf yang ada di dalam variabel kata "sentence". Ada juga yang terakhir yaitu operator + dan * untuk sebuah string. Contohnya seperti berikut:

```
word = "pizza"

print(word + word)
print(word * 2)
```

```
pizzapizza
pizzapizza
```

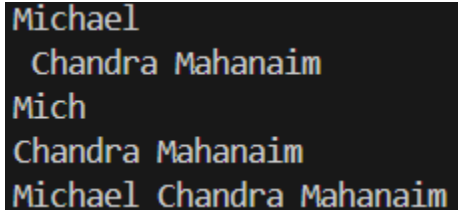
Gambar contoh penggunaan operator + dan * dalam sebuah string dan outputnya

Dari contoh di atas, diketahui bahwa jika string + string maka hasilnya hanyalah penggabungan dari kedua string tersebut. Jika string * 2 maka hasilnya hanya string itu di print 2 kali tanpa spasi. Mengapa string * string tidak dimasukkan contoh? Karena tidak bisa, perkalian antar string tidak dapat dilakukan.

MATERI 3: String Slice

String slice adalah pemotongan sebuah string relatif dengan angka indeksinya. Langsung ke contohnya adalah:

```
name = "Michael Chandra Mahanaim"
start = 0
end = 7
print(name[start:end])
print(name[end:len(name)])
print(name[:4])
print(name[8:])
print(name[:])
```

The output of the Python code is displayed in a separate box. It shows five lines of text corresponding to the five print statements: 'Michael' (characters 0-7), 'Chandra Mahanaim' (characters 7-19), 'Mich' (characters 0-4), 'Chandra Mahanaim' (characters 8-19), and 'Michael Chandra Mahanaim' (characters 0-19, the full string).

Gambar contoh string slicing dan outputnya

Mulai dari print pertama, slice hanya akan mengeluarkan output untuk huruf indeks ke 0 sampai 7, print kedua akan mengeluarkan output dari indeks ke 7 sampai akhir / panjang dari string "name". Perlu dicatat bahwa spasi terhitung sebagai 1 indeks, maka harus berhati-hati jika menggunakan indeks agar tidak lupa menghitung spasi. Print ketiga hanya memberi akhir / stop maka akan mengeluarkan output dari awal sampai indeks ke-4. Print keempat hanya memberikan awal / start, maka hanya akan menampilkan dari indeks ke-8 sampai akhir. Print kelima tidak diberikan start dan stop, maka hanya akan menampilkan kalimat apa adanya.

MATERI 4: String Method

String method adalah fungsi khusus pengolahan string yang sangat berguna. Beberapa contoh fungsi ini yaitu:

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()

Semua fungsi ini berguna pada saat tertentu dan penting untuk dipahami.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Question1.py U X Question2.py U Question3.py U Question4.py U SrcCd.py U
Question1.py > ...
1 word = "mata"
2 chklist = ["ama", "maat", "taas", "tama"]
3
4 sortword = sorted(word)
5
6 for item in chklist:
7     chkw = sorted(item)
8     if chkw == sortword:
9         print(f"{item} is an Anagram of {word}")
10    else:
11        print(f"{item} is not an Anagram of {word}")
```

B. Output Result

```
PS C:\Users\micha\Documents\Col
s/College/2SMT/TPAP1/tugaspralp
ama is not an Anagram of mata
maat is an Anagram of mata
taas is not an Anagram of mata
tama is an Anagram of mata
```

C. Explanation

Kode di atas menggunakan list, for loop, string method, dan percabangan untuk mengeluarkan output berupa apakah kata yang dari dalam list termasuk anagram dari kata tertentu atau bukan.

- Pertama variabel "word" digunakan untuk menjadi kata basis pengecekan anagram
- Lalu ada list untuk kata-kata lain yang akan di cek sebagai anagram "word" atau bukan
- Setelah itu fungsi sorted(word) digunakan untuk mengurutkan kata yang di dalam variabel "word" dan diubah menjadi Char List

(Contoh perubahan akan menjadi seperti, "mata" akan menjadi list
[("a"),("a"),("m"),("t")])

- Lalu masuk ke dalam for loop yang akan mengeluarkan kata di dalam list satu per satu sebagai variabel "item"
- Variabel "item" akan di sorted

- Hasil sorted dari variabel "word" dan "item" akan di cek dengan percabangan IF ELSE
- Jika Char list hasil sorted sama, maka kata tersebut adalah sebuah anagram
- Jika Char list beda maka bukan anagram
- Proses akan dilakukan untuk semua kata di dalam variabel "chklist"

SOAL 2

A. Source Code

```

Question1.py U  Question2.py U x  Question3.py U  Question4.py U  SrcCd.py U
Question2.py > ...
1  snt = "Saya mau makan. Makan itu wajib, Mau siang atau malam saya wajib makan."
2  search = "makan"
3
4  snt = snt.lower()
5  sntlist = snt.split()
6  count = 0
7
8  for item in sntlist:
9      word = ''.join([j for j in item if j.isalpha()])
10     if word == search:
11         count += 1
12
13  print(f"{search} Muncul {count} kali")

```

B. Output Result

```

PS C:\Users\micha\Desktop> python3 srcCd.py
s/College/2SMT/TPAP2021/Question2.py
makan Muncul 3 kali

```

C. Explanation

Kode di atas menggunakan string method, for loop, dan percabangan untuk mencari berapa kali sebuah string kata muncul dalam sebuah string kalimat. String method yang digunakan adalah lower() untuk membuat string dalam huruf kecil semua, split() untuk memisahkan kata-kata yang ada di kalimat agar lebih mudah di cek, dan ".join()" untuk menghapus tanda baca dari masing-masing kata. Setelah itu hanya dengan menggunakan for loop untuk mengecek kata satu per satu dan jika kata itu ditemukan maka "count" akan bertambah 1 sebagai penghitung berapa kali kata itu muncul.

SOAL 3

A. Source Code

```
Question1.py U Question2.py U Question3.py U x Question4.py U SrcCd.py U
Question3.py > ...
1  snt = "Saya      tidak  suka,   memancing           ikan "
2
3  sntlist = snt.split()
4
5  for i in range(len(sntlist)):
6      if i == 0:
7          csnt = sntlist[i]
8      else:
9          csnt = csnt + " " + sntlist[i]
10
11  print(csnt)
```

B. Output Result

```
PS C:\Users\micha\Documents\Coll
s/College/2SMT/TPAP1/tugaspralpr
Saya tidak suka, memancing ikan
```

C. Explanation

Kode di atas relatif mudah, hanya menggunakan string method, len, for loop dan percabangan untuk menyusun ulang kalimat yang memiliki terlalu banyak spasi. split() digunakan untuk menghapus spasi antar kata dan mengubah kalimat menjadi sebuah list kata-kata dalam kalimat. Lalu menggunakan for loop dengan i sebagai angka indeks untuk mengambil kata-kata dalam list kalimat sebelumnya dan percabangan singkat untuk kata pertama yang masuk. Kata setelah kata pertama akan diberikan spasi dan di gabung kembali dengan kata lain agar menjadi kalimat utuh lagi.

SOAL 4

A. Source Code

```
Question1.py U Question2.py U Question3.py U Question4.py U X SrcCd.py U
Question4.py > ...
1  snt = "red snakes and a black frog in the pool"
2
3  sntlist = snt.split()
4  long = 0
5  short = 1000
6  res = []
7
8  for item in sntlist:
9      word = ''.join([i for i in item if i.isalpha()])
10     if len(word) > long:
11         long = len(word)
12         lword = word
13     elif len(word) < short:
14         short = len(word)
15         sword = word
16
17  print(f"Shortest word is {sword}")
18  print(f"Longest word is {lword}")
```

B. Output Result

```
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tu
Shortest word is a
Longest word is snakes
```

C. Explanation

Kode di atas relatif mudah dan hanya menggunakan string method, for loop, dan percabangan untuk mencari kata yang terpendek dan terpanjang dari sebuah kalimat. `split()` digunakan untuk membuat list kata-kata dari kalimat, `"".join()` digunakan untuk menghapus tanda baca jika ada, dan `len()` digunakan untuk mengetahui mana kata terpanjang dan terpendek.