



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231052
Nama Lengkap	GABRIEL SACHIO ATMADJAJA
Minggu ke / Materi	07 / 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%)

Sumber Materi: Modul Pratikum Algoritma dan Pemrograman

MATERI 7

String adalah kumpulan karakter, lebih sederhananya lagi adalah kalimat atau teks. String dapat ditulis dalam variable. String dapat diakses per-karakturnya dengan cara `nama_variable[index]`. Hal yang perlu diketahui adalah karakter pertama dalam string itu dimulai dari index 0. Lalu string juga dapat dimanipulasi elemennya dengan ditambahkan string lain. Berikut contohnya:

```
coba.py > ...
1  namesaya = "Antonius Rachmat C"
2  temansaya1 = "Yuan Lukito"
3  temansaya2 = 'Laurentius Kuncoro'
4  temansaya3 = "Matahari" + 'Bakti'
5
6  print(temansaya3)
7  print(namesaya[0]) #'A'
8  print(namesaya[9]) #'R'
9  print(namesaya[8]) #' '
10 print(temansaya1[-1]) #'C'
11 print(temansaya1[1]) #'u'
12
13 huruf = temansaya2[0]
14 print(huruf) #'L'
15
16 # namesaya[-1] = "Z" # error
17 # print(namesaya) # error
```

Gambar 1.1: Contoh Cara Akses String dan Manipulasinya

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Pro/Pert7/coba.py
MatahariBakti
A
R

o
u
L
```

Gambar 1.2: Output

Operator dan Metode String:

- Operator in -> mengecek apakah suatu kalimat merupakan substring dari kalimat lainnya atau bukan

```
operatorstring.py > ...
1  # OPERATOR IN
2  kalimat = "saya mau makan"
3  kalimat2 = "sayang mau makan"
4  kalimat3 = "SAYA mau makan"
5  data = "saya"
6  print(data in kalimat) #True
7  print("mau" in kalimat) #True
8  print("dia" in kalimat) #False
```

Gambar 1.3: Penggunaan operator in

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Pro/Pert7/operatorstring.py
True
True
False
```

Gambar 1.4: Output

- Perbandingan di String

```
10  # PERBANDINGAN DI STRING
11  # string pertama akan dianggap lebih besar dari string kedua
12  if "saya" > "dianayang":
13  |   print("Ya") #Ya
14  else:
15  |   print("Tidak")
16
17  if "dua" == "dua":
18  |   print("Sama") #Sama
```

Gambar 1.5: Perbandingan di string

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Pro/Pert7/operatorstring.py
Ya
Sama
```

Gambar 1.6: output

- Fungsi Len -> untuk menghitung panjang karakter (spasi juga dihitung)

```
20  # FUNGSI LEN
21  # kalimat = input("Kalimat: ")
22  kalimat = "universitas kristen duta wacana yogyakarta"
23  print(len(kalimat)) #output 42
24
```

Gambar 1.7: Penggunaan fungsi len

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
Pro/Pert7/operatorstring.py  
42
```

Gambar 1.8: Output

- Traversing String -> cara menampilkan huruf demi huruf dalam string dgn loop

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

Gambar 1.9: Dengan while

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

Gambar 1.10: Dengan for

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
Pro/Pert7/operatorstring.py  
i n d o n e s i a   j a y a
```

Gambar 1.11: Kedua output

- String slice -> mengambil bagian tertentu pada string

```
46 # STRING SLICE -> mengambil bagian tertentu  
47 kalimat = "cerita rakyat"  
48 awal = 0  
49 akhir = 6  
50 print(kalimat[awal:akhir]) #cerita  
51 print(kalimat[7:len(kalimat)]) #rakyat  
52 print(kalimat[:5]) #cerit  
53 print(kalimat[5:]) #a rakyat  
54 print(kalimat[:]) #cerita rakyat
```

Gambar 1.12: Cara mengambil bagian tertentu di string

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
Pro/Pert7/operatorstring.py  
cerita  
rakyat  
cerit  
a rakyat  
cerita rakyat
```

Gambar 1.13: Output

- Operator + pada string digunakan untuk menggabungkan string (tipenya harus sama), lalu operator * untuk menampilkan string sejumlah perkaliannya

```
65 # OPERATOR + dan * PADA STRING
66 kata1 = input("Masukkan bil1: ")
67 kata2 = input("Masukkan bil2: ")
68 kata3 = kata1 + kata2
69 print(kata3)
70 print(kata3 * 5)
```

Gambar 1.14: Penggunaan operator + dan * pada string

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Pro/Pert7/operatorstring.py
Masukkan bil1: sayang
Masukkan bil2: kamu
sayangkamu
sayangkamusayangkamusayangkamusayangkamusayangkamu
```

Gambar 1.15: Output

Parsing String:

Parsing adalah cara menelusuri string untuk mengubah atau memodifikasi bagian yang diinginkan. Berikut contohnya:

```
parsingstring.py > ...
1 # parsing utk mengubah atau memodifikasi
2 kalimat = "Saudara-saudara, pada tanggal 17-08-1945 Indonesia merdeka"
3
4 hasil = kalimat.split(" ")
5 # print(hasil)
6
7 for kal in hasil:
8     # print(kal[0])
9     if kal[0].isdigit():
10         # print(kal)
11         hasil2 = kal.split("-")
12         # print(hasil2)
13         print(hasil2[1]+"/"+hasil2[0]+"/"+hasil2[2])
```

Gambar 1.16: Contoh parsing

Perhatikan baris 4, disana terdapat kalimat.split(" ") yang berfungsi untuk membagi string berdasarkan spasi.

Output:

```
PS D:\Sachio\Kampus\PrakAlPro\Pert7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Pro/Pert7/parsingstring.py
08/17/1945
```

Gambar 1.17: Output

BAGIAN 2: LATIHAN MANDIRI (60%)

Source: <https://github.com/gabrielsachioa/PrakAIPro7.git>

SOAL 1

Kode program untuk mengecek apakah kata tersebut anagram dengan kata lainnya:

```
latihan7-1.py > cek_anagram
1 def jumlah_huruf_kata1(kata1):
2     hasil = {}
3
4     for huruf in kata1:
5         if huruf in hasil:
6             hasil[huruf] += 1
7         else:
8             hasil[huruf] = 1
9     return hasil
10
11 def jumlah_huruf_kata2(kata2):
12     hasil = {}
13
14     for huruf in kata2:
15         if huruf in hasil:
16             hasil[huruf] += 1
17         else:
18             hasil[huruf] = 1
19     return hasil
20
21 def cek_anagram(kata1, kata2):
22     if len(kata1) != len(kata2):
23         print("Bukan Anagram")
24     else:
25         huruf_kata1 = jumlah_huruf_kata1(kata1)
26         huruf_kata2 = jumlah_huruf_kata2(kata2)
27
28         if huruf_kata1 == huruf_kata2:
29             print("Anagram")
30         else:
31             print("Bukan Anagram")
32
33 # PROGRAM UTAMA
34 kata1 = input("Kata pertama: ")
35 kata2 = input("Kata kedua: ")
36 cek_anagram(kata1, kata2)
```

Gambar 2.1: Kode Program Anagram

Keterangan:

Anagram adalah kata yang dibolak-balik susunannya sama, yang berarti anagram itu panjang stringnya harus sama, serta jumlah dan jenis hurufnya harus sama.

- Baris 1 s.d. 9 -> fungsi untuk menghitung jumlah huruf pada variable kata1
- Baris 11 s.d. 19 -> fungsi untuk menghitung jumlah huruf pada variable kata2
- Baris 21 s.d. 31 -> fungsi untuk mengecek apakah itu anagram atau bukan dari kedua variable. Pertama, kondisi if untuk mengecek jika panjang string dari tiap inputan berbeda, maka tampilkan "Bukan Anagram". Lalu Pada kondisi else, buat variable huruf_kata1 dan huruf_kata2 untuk memanggil masing-masing fungsi jumlah_huruf_kata. Masih di kondisi else, kondisi if disitu berfungsi untuk mengecek apakah jumlah karakter yang muncul sama, jika True maka cetak "Anagram", else cetak "Bukan Anagram"
- Baris 34, 35 -> variable yang meminta input dari user, digunakan sebagai parameter pada fungsi
- Baris 36 -> memanggil fungsi cek_anagram

Output:

```
PS D:\Sachio\Tugas\prakalpro7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe an7-1.py
Kata pertama: mata
Kata kedua: atam
Anagram
```

Gambar 2.2: Output

SOAL 2

Kode:

```
latihan7-2.py > ...
1  # CARA KE 1
2  def frekuensi_kata(kalimat, kata):
3      kalimat = kalimat.lower()
4      kata = kata.lower()
5
6      kalimat = "".join(i for i in kalimat if i.isalpha()) for kalimat in kalimat.split()
7
8      frek_kata = kalimat.count(kata)
9      print(frek_kata)
10
11
12 # PROGRAM UTAMA
13 kalimat = input("Kalimat: ")
14 kata = input("Kata: ")
15 frekuensi_kata(kalimat, kata)
16 # frekuensi_kata("Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan", "makan")
17
```

Gambar 2.3: Kode Program menghitung frekuensi kemunculan kata

Keterangan:

- Baris 1 s.d. 9 -> fungsi untuk menghitung frekuensi kemunculan suatu kata pada kalimat
- Baris 3,4 -> mengubah string yang di-input-kan menjadi lowercase
- Baris 6 -> cara membuang semua yang bukan alfabet selain spasi
- Baris 8 -> menghitung frekuensi katanya
- Baris 9 -> menampilkan hitungan dari variable frek_kata
- Baris 13, 14 -> variable yang meminta input user, digunakan di parameter fungsi frekuensi_kata
- Baris 15 -> memanggil fungsi frekuensi_kata

Output:

```
PS D:\Sachio\Tugas\prakalpro7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
tihan7-2.py
Kalimat: Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan
Kata: makan
3
```

Gambar 2.4: Output

SOAL 3

Kode:

```
latihan7-3.py > ...
1 def hapus_kelebihan_spasi(kalimat):
2     kalimat = kalimat.lower()
3     kalimat = " ".join(kalimat.split())
4     print(kalimat)
5
6 # PROGRAM UTAMA
7 kalimat = input("Kalimat: ")
8 hapus_kelebihan_spasi(kalimat)
9 # hapus_kelebihan_spasi("saya tidak suka memancing ikan ")
```

Gambar 2.5: Kode Program menghapus kelebihan spasi

Keterangan:

- Baris 1 s.d. 4 -> fungsi untuk menghapus kelebihan spasi
- Baris 2 -> mengubah string yang di-input-kan menjadi lowercase
- Baris 3 -> Membagi string berdasarkan spasi otomatis spasinya akan hilang. Setelah itu tambahkan " ".join supaya terdapat spasi antar kata
- Baris 7 -> variable yang meminta input user, digunakan di parameter fungsi hapus_kelebihan_spasi
- Baris 8 -> memanggil fungsi hapus_kelebihan_spasi

Output:

```
PS D:\Sachio\Tugas\prakalpro7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
tihan7-3.py
Kalimat: saya tidak suka memancing ikan
saya tidak suka memancing ikan
```

Gambar 2.6: Output

SOAL 4

Kode:

```
latihan7-4.py > ...
1  def ukuran_kata(kalimat):
2      kalimat = kalimat.lower()
3      kata = kalimat.split(" ")
4
5      terpendek = len(kata[0])
6      terpanjang = len(kata[0])
7
8      for i in range(len(kata)):
9          if len(kata[i]) < terpendek:
10             terpendek = len(kata[i])
11             kata_terpendek = kata[i]
12          elif len(kata[i]) > terpanjang:
13             terpanjang = len(kata[i])
14             kata_terpanjang = kata[i]
15
16      print(f"terpendek: {kata_terpendek} ")
17      print(f"terpanjang: {kata_terpanjang}")
18
19  # PROGRAM UTAMA
20  kalimat = input("Kalimat: ")
21  ukuran_kata(kalimat)
22  # ukuran_kata("red snakes and a black frog in the pool")
```

Gambar 2.7: Program mencari kata terpendek dan terpanjang

Keterangan:

- Baris 1 s.d. 17 -> fungsi untuk menentukan kata terpendek dan terpanjang pada sebuah kalimat
- Baris 2 -> mengubah input string dari user menjadi lowercase
- Baris 3 -> memecah string berdasarkan spasi dan menyimpannya di variable kata
- Baris 5, 6 -> menghitung panjang string pada kata[0] untuk digunakan pada perbandingan di perulangan nantinya
- Baris 8 s.d. 14 -> perulangan for untuk menentukan kata terpendek dan terpanjang. Pada kondisi if, jika panjang string dari i kurang dari panjang variable terpendek, maka terpendek diganti menjadi panjang string dari i tersebut dan kata_terpendek = kata[i] untuk mencari kata berdasarkan index itu. Begitu juga dalam kondisi elif, bedanya operatornya lebih dari. Lalu jika True, maka variable terpanjang akan diganti dengan panjang string i dan kata_terpanjang akan mengakses kata pada index i
- Baris 16, 17 -> menampilkan kata yang terpendek dan terpanjang
- Baris 20 -> variable yang meminta input string dari user, digunakan sebagai parameter pada fungsi ukuran_kata
- Baris 21 -> memanggil fungsi ukuran_kata

Output:

```
PS D:\Sachio\Tugas\praka1pro7> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
tihan7-4.py  
Kalimat: red snakes and a black frog in the pool  
terpendek: a  
terpanjang: snakes
```

Gambar 2.8: Output