



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230987
Nama Lengkap	Prastha Pradipta Purusa
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). Misalnya, Anda mungkin hanya ingin menampilkan bagian dari nomor seri pada formulir. Atau, Anda mungkin perlu menggabungkan (menggabungkan) beberapa string bersama-sama, seperti nama belakang dan nama depan. Tipe data ini merupakan jenis tipe data yang bukan tipe data dasar, karena tipe data ini pada dasarnya menyimpan lebih dari satu nilai tunggal sebagai satu kesatuan.

Pengakesesan String dan Manipulasi String

String dibuat dengan variabel dan diisi dengan data. String dapat diakses dengan nama variabel, atau perhuruf di dalam index. Index dimulai dari 0 dan harus bilangan bulat. Berikut adalah contohnya:

H	A	L	O
0	1	2	3

Bisa juga seperti ini:

```
1 kata = "aku suka makan ayam"
2
3 print(kata[2])
```

```
u
```

Operator dan metode string

OPERATOR IN

Kita dapat mengecek substring menggunakan operator in. Hasil dari operator in adalah True/False. Contoh:

```
main.py
1 data = "1"
2 jumlah_data = "1,2,3"
3 print(data in jumlah_data)
4 print("4" in jumlah_data)
```

```
True
False
```

Kita dapat melakukan ouput True/False dengan cara perbandingan. Contohnya:

```

1 if 1 < 2:
2     print("Benar")
3 else:
4     print("Salah")
5
6 if 2 == 2:
7     print("Sama")

```

```

Benar
Sama

```

Fungsi len

Fungsi len digunakan untuk berapa panjang jumlah karakter dengan cara len(string). Contohnya:

```

1 kalimat = "wiwuwuwuwuwu"
2 print(len(kalimat))

```

```

14

```

Untuk memanggil kata pertama:

```

1 kalimat = "wiwuwuwuwuwu"
2 pertama = kalimat[0]
3 print(pertama)
4

```

```

w

```

Untuk memanggil kata terakhir:

```

1 kalimat = "wiwuwuwuwuwu"
2 pertama = kalimat[len(kalimat)-1]
3 print(pertama)
4

```

```

i

```

Index dimulai dari 0

Traversing string

Menampilkan huruf demi huruf dengan perulangan. Contoh:

- Dengan akses index:

```

1 angka = "1,2,3,4,5,6,5,4,53,234,2,42,4,4,3"
2 i = 0
3 while i < len(angka):
4     print(angka[i],end='')
5     i += 1

```

```
1,2,3,4,5,6,5,4,53,234,2,42,4,4,3
```

Akses index menggunakan len untuk menghitung baris

- Tanpa akses index

```

1 angka = "1,2,3,4,5,6,5,4,53,234,2,42,4,4,3"
2 for i in angka:
3     print(i,end="")

```

```
1,2,3,4,5,6,5,4,53,234,2,42,4,4,3
```

Dilakukan tanpa menggunakan len

String Slice

Menampilkan substring menggunakan index dari awal tertentu sampai akhir tertentu. Contoh:

```

1 kalimat = "wndfndsfnpds"
2 utama = 5
3 cadangan = 10
4 print(kalimat[utama:cadangan])
5 print(kalimat[3:len(kalimat)])
6 print(kalimat[:3])

```

```
dsfnp
fndsfnpds
wnd
```

Program tersebut membaca kalimat. Jika utama dan Cadangan dia akan membaca dari huruf ke 5 hingga ke 10. 3: akan membaca 3 line dilewati lalu sisany huruf dari kiri ke kanan. :3 maka akan membaca 3 huruf pertama.

METODE STRING

Method	Kegunaan	Penggunaan
capitalize()	Mengubah string menjadi huruf besar	string.capatilze()
count()	Menghitung 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()	True jika string adalah huruf kecil / huruf besar	string.islower() dan string.isupper()
isdigit()	True jika string adalah digit (angka)	string.isdigit()
strip()	menghapus semua whitespace yang ada di depan dan di akhir string	string.strip()
split()	Memisahkan string menjadi token-token berdasarkan pemisah	string.split()

Operator * dan + pada String

Operator + dapat menggabungkan 2 buah string. Operator * bisa digunakan untuk menampilkan string sejumlah perkaliannya. Contoh:

```
1 angka1 = "9"
2 angka2 = "11"
3 angka3 = angka1 + " " + angka2
4 print(angka3)
5 print(angka1 * 2 + angka2 * 3)
```

```
9 11
99111111
```

Parsing String

Parsing string adalah cara mencari string bagian demi bagian untuk mendapatkan / menemukan / mengubah bagian string yang diinginkan. Contoh:

```
1 kalimat = "tanggal 26-03-2024 uts apirb dan 27-03-2024 libur"
2
3 hasil = kalimat.split(" ")
4
5 for kal in hasil:
6     if "-" in kal:
7         tanggal = kal.split("-")
8         if len(tanggal) == 3 and all(part.isdigit() for part in tanggal):
9             if tanggal[0] == "27" and tanggal[1] == "03" and tanggal[2] == "2024":
10                 tanggal_baru = tanggal[0] + "/" + tanggal[1] + "/" + tanggal[2]
11                 print(tanggal_baru)
12
```

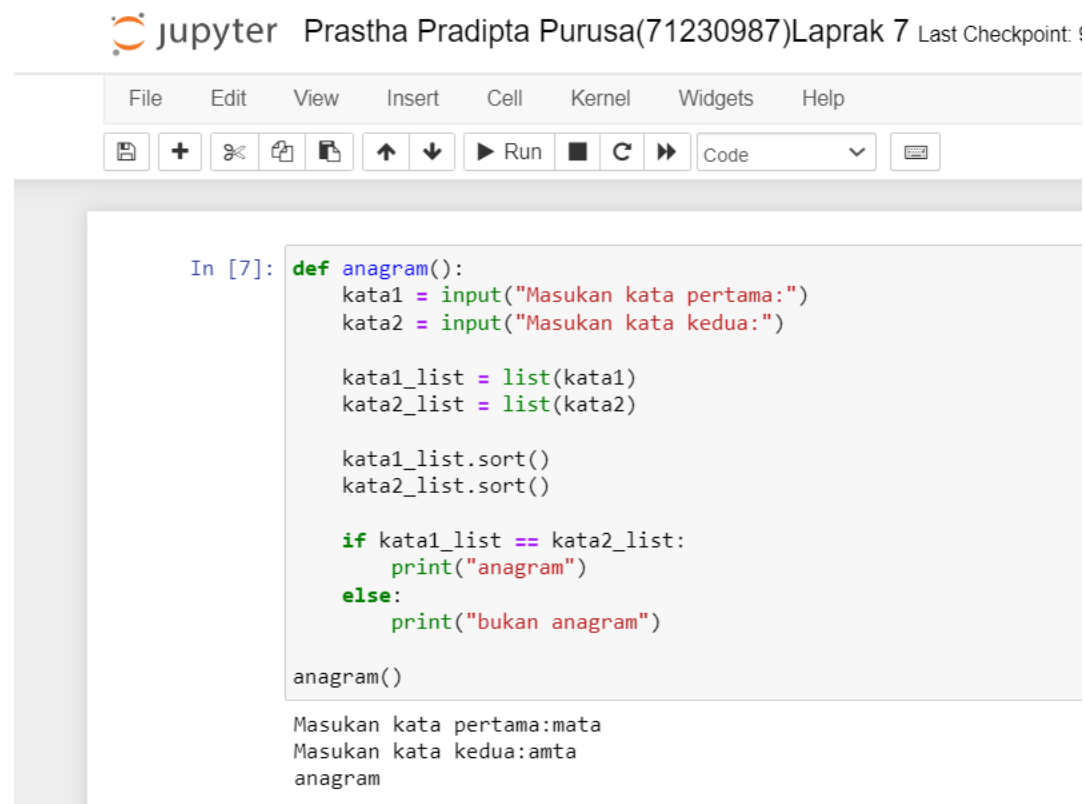
27/03/2024

Program tersebut membuat kita mengambil tanggal 27-03-24 saja dan mengubah dari – menjadi / untuk tiap bulan dan tanggal.

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



The screenshot shows a Jupyter Notebook interface. At the top, it says "Jupyter Prastha Pradipta Purusa(71230987)Laprak 7 Last Checkpoint :". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, running, and a dropdown menu currently set to "Code". The main area of the notebook contains a code cell with the following Python code:

```
In [7]: def anagram():
        kata1 = input("Masukan kata pertama:")
        kata2 = input("Masukan kata kedua:")

        kata1_list = list(kata1)
        kata2_list = list(kata2)

        kata1_list.sort()
        kata2_list.sort()

        if kata1_list == kata2_list:
            print("anagram")
        else:
            print("bukan anagram")

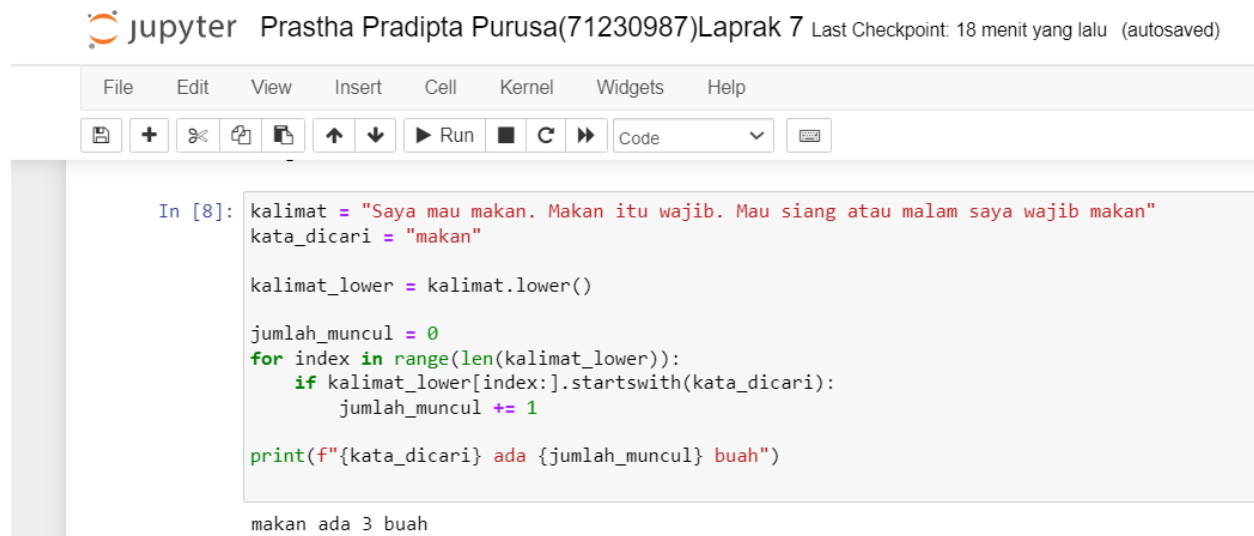
        anagram()
```

Below the code cell, the output of the function is displayed:

```
Masukan kata pertama:mata
Masukan kata kedua:amta
anagram
```

Kita mencari anagram kata. Kita buat input untuk kata 1 dan kata 2. Kita membuat variabel baru yaitu kata1_list dan kata2_list dengan diberi list. Lalu di sort. Jika kata1_list sama dengan kata2_list maka anagram dan jika tidak sama maka bukan anagram. Anagram() digunakan untuk memanggil fungsi

SOAL 2



Jupyter Prastha Pradipta Purusa(71230987)Laprak 7 Last Checkpoint: 18 menit yang lalu (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run

```
In [8]: kalimat = "Saya mau makan. Makan itu wajib. Mau siang atau malam saya wajib makan"
kata_dicari = "makan"

kalimat_lower = kalimat.lower()

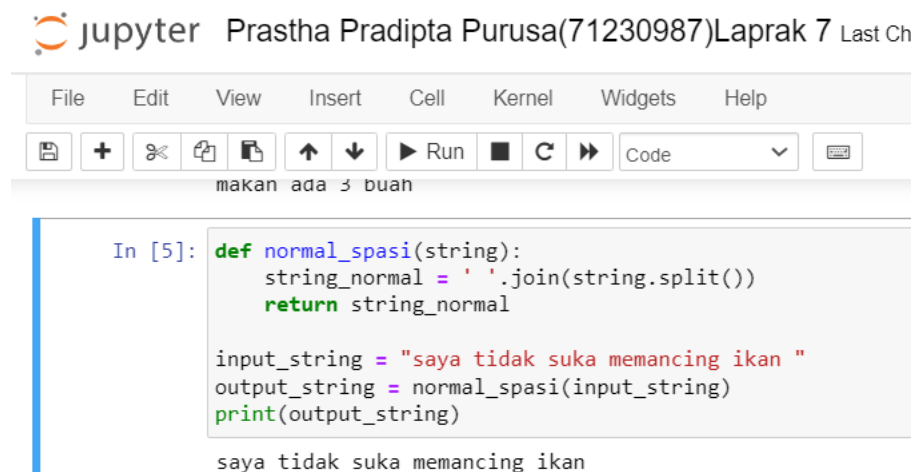
jumlah_muncul = 0
for index in range(len(kalimat_lower)):
    if kalimat_lower[index:].startswith(kata_dicari):
        jumlah_muncul += 1

print(f"{kata_dicari} ada {jumlah_muncul} buah")
```

makan ada 3 buah

Kita buat mencari frekuensi kata. Buat kalimat dan kata yg mau dicari terlebih dahulu. Untuk memudahkan mencari maka `kalimat.lower()` supaya semua tulisanya huruf kecil. `Jumlah_muncul = 0` digunakan untuk menghitung jumlah yang muncul. `index = kalimat_lower.find(kata_dicari)` metode `.find()` digunakan untuk mencari indeks pertama dari kemunculan pertama kata yang dicari dalam teks. Nilai indeks ini akan digunakan sebagai titik awal untuk pencarian berikutnya. `for index in range(len(kalimat_lower))` loop `for` digunakan untuk melintasi setiap indeks dalam teks `kalimat_lower`. `if kalimat_lower[index:].startswith(kata_dicari)` digunakan untuk jika pernyataan `if` memeriksa bagian teks mulai dari posisi `index` sesuai dengan kata yang dicari (`kata_dicari`) menggunakan metode `.startswith()`. Jika ya, artinya kita menemukan kemunculan kata yang dicari, maka `jumlah_muncul` akan bertambah satu. Terakhir menggunakan `print` untuk memunculkan output dengan format `string{kata_dicari}` dan `{jumlah_muncul}`

SOAL 3



Jupyter Prastha Pradipta Purusa(71230987)Laprak 7 Last Ch

File Edit View Insert Cell Kernel Widgets Help

Run

makan ada 3 buah

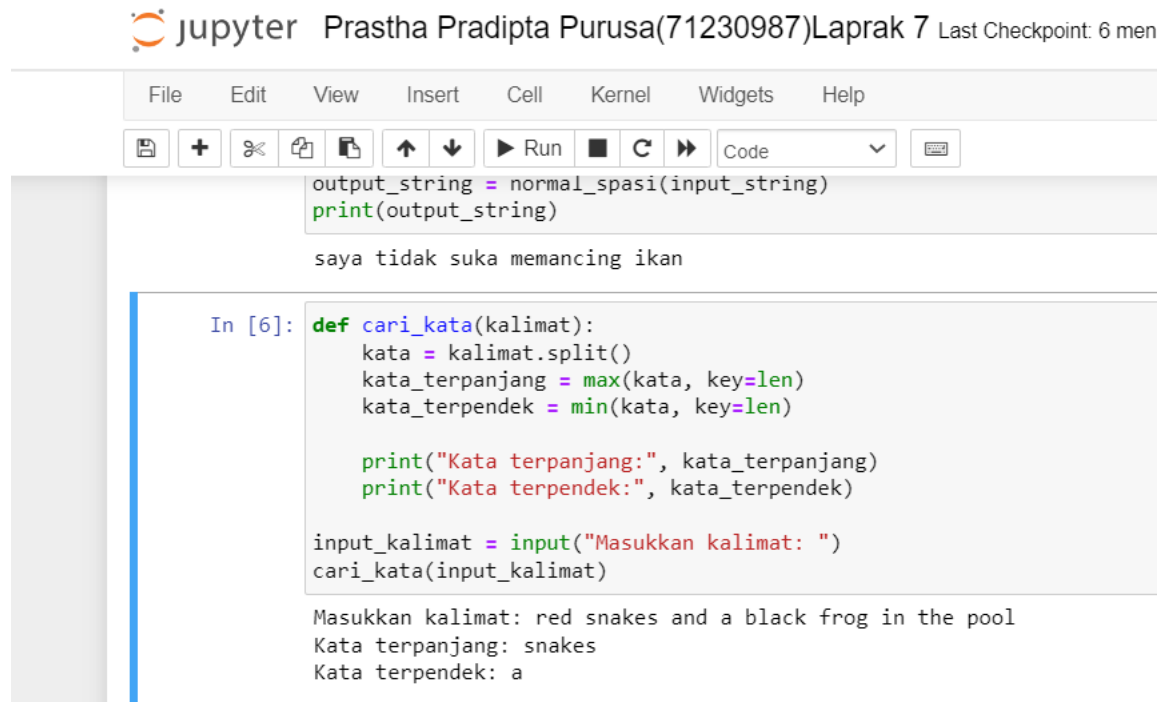
```
In [5]: def normal_spasi(string):
        string_normal = ' '.join(string.split())
        return string_normal

input_string = "saya tidak suka memancing ikan "
output_string = normal_spasi(input_string)
print(output_string)
```

saya tidak suka memancing ikan

Menggunakan fungsi `normal_spasi` dengan parameter string. `String_normal` akan mengambil parameter string dan membaginya berdasarkan spasi menggunakan metode `split()` kemudian menggabungkan kembali kata-kata dengan menggunakan spasi menggunakan metode `join()` digunakan untuk menghapus spasi berlebih di antara kata-kata. Return `string_normal` untuk mengembalikan nilai string. `Input_string` untuk nilai string. `output_string = normal_spasi(input_string)` digunakan untuk pemanggilan fungsi `normal_spasi` dengan `input_string` sebagai argumen. `Print(output_string)` digunakan untuk memanggil nilai `output_string` yang variabel sudah diolah.

SOAL 4



The screenshot shows a Jupyter Notebook interface. At the top, the header reads "Jupyter Prastha Pradipta Purusa(71230987)Laprak 7 Last Checkpoint: 6 men". Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding, deleting, copying, pasting, undo, redo, and running code. The main area contains a code cell with the following code:

```
output_string = normal_spasi(input_string)
print(output_string)

saya tidak suka memancing ikan
```

Below the code cell is an input field with the text "saya tidak suka memancing ikan". Below the input field is a code cell with the following code:

```
In [6]: def cari_kata(kalimat):
        kata = kalimat.split()
        kata_terpanjang = max(kata, key=len)
        kata_terpendek = min(kata, key=len)

        print("Kata terpanjang:", kata_terpanjang)
        print("Kata terpendek:", kata_terpendek)

        input_kalimat = input("Masukkan kalimat: ")
        cari_kata(input_kalimat)
```

Below the code cell is an input field with the text "Masukkan kalimat: red snakes and a black frog in the pool". Below the input field is the output of the code:

```
Kata terpanjang: snakes
Kata terpendek: a
```

Menggunakan fungsi `cari_kata` dengan parameter `kalimat`. `Kata = kalimat.split()` untuk memecah kata-kata tiap kalimat. `Kata_terpanjang = max` untuk mencari nilai terbesar di kata dan `kata_terpendek` dengan `min` untuk mencari nilai terkecil dan menggunakan `key = len` untuk membandingkan panjang (jumlah karakter) dari setiap kata. Lalu di `print` untuk memunculkan kata terpendek dan `kata_terpanjang`. Lalu `input` kalimat. Untuk memanggil nilai menggunakan fungsi `cari_kata(input_kalimat)`.

Link github:

<https://github.com/prasss020904/Laprak7-71230987.git>