

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



<b>NIM</b>	<b>71231006</b>
<b>Nama Lengkap</b>	<b>Fransiskus Tedyanto seran</b>
<b>Minggu ke / Materi</b>	<b>07 / Pengolahan String</b>

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  
2023

## 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.

### MATERI 1

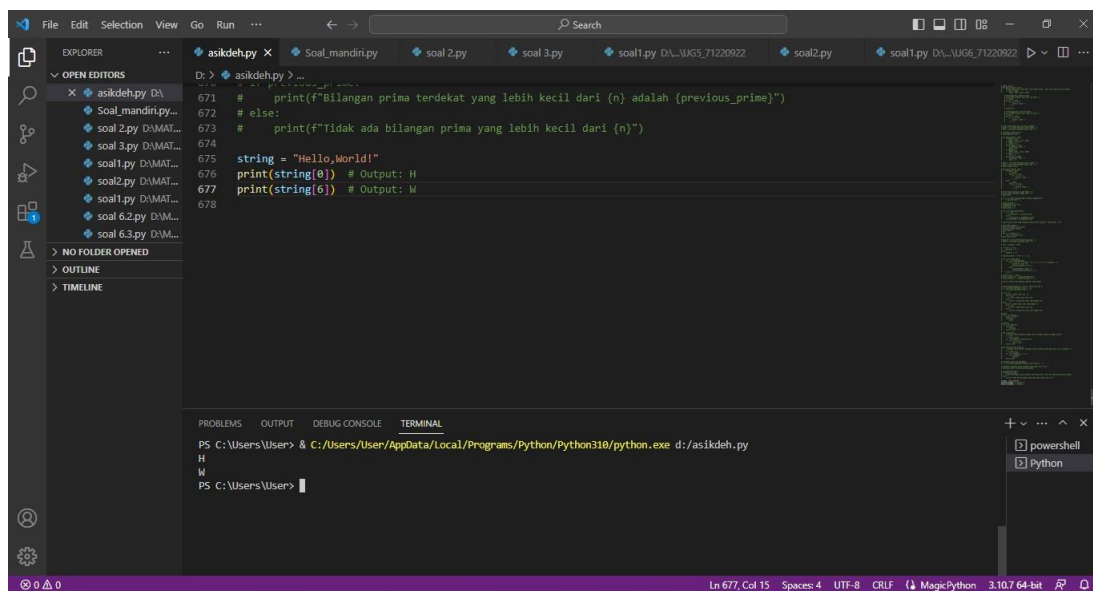
#### Pengantar String

Di python, “string” adalah tipe data yang digunakan untuk merepresentasikan teks atau karakter. String pada python didefinisikan sebagai serangkaian karakter yang diapit oleh tanda kutip tunggal (‘kata’) atau tanda kutip ganda (“kata”). String pada python dapat diakses dan dimanipulasi dengan berbagai cara beberapa operasi dasar pada string termasuk penggabungan string, perulangan string, pengindeksan karakter dalam string, dan memotong atau slicing string.

### MATERI 2

#### Pengaksesan String dan Manipulasi String

Kita dapat mengakses karakter dalam string dengan menggunakan indeks. Indeks dimulai dari 0, sehingga karakter pertama dalam string memiliki indeks 0, karakter kedua memiliki 1, dan seterusnya. Seperti contoh berikut:



```
File Edit Selection View Go Run ... Search
EXPLORER
  OPEN EDITORS
    asikdeh.py x
    Soal_mandini.py
    soal_2.py
    soal_3.py
    soal_1.py
    soal_2.py
    soal_1.py
    soal_6.2.py
    soal_6.3.py
  NO FOLDER OPENED
  OUTLINE
  TIMELINE
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
    PS C:\Users\User> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe d:/asikdeh.py
    H
    W
    PS C:\Users\User>
  Ln 677, Col 15 Spaces: 4 UTF-8 CRLF MagicPython 3.10.7 64-bit
```

The screenshot shows a Python IDE with a file explorer on the left containing several Python files. The main editor displays a script named 'asikdeh.py' with the following code:

```
671 # print(f"Bilangan prima terdekat yang lebih kecil dari {n} adalah {previous_prime}")
672 # else:
673 #     print(f"Tidak ada bilangan prima yang lebih kecil dari {n}")
674
675 string = "Hello,World!"
676 print(string[0]) # Output: H
677 print(string[6]) # Output: W
678
```

The bottom panel shows the terminal output, which displays 'H' and 'W' on separate lines, corresponding to the print statements in the code.

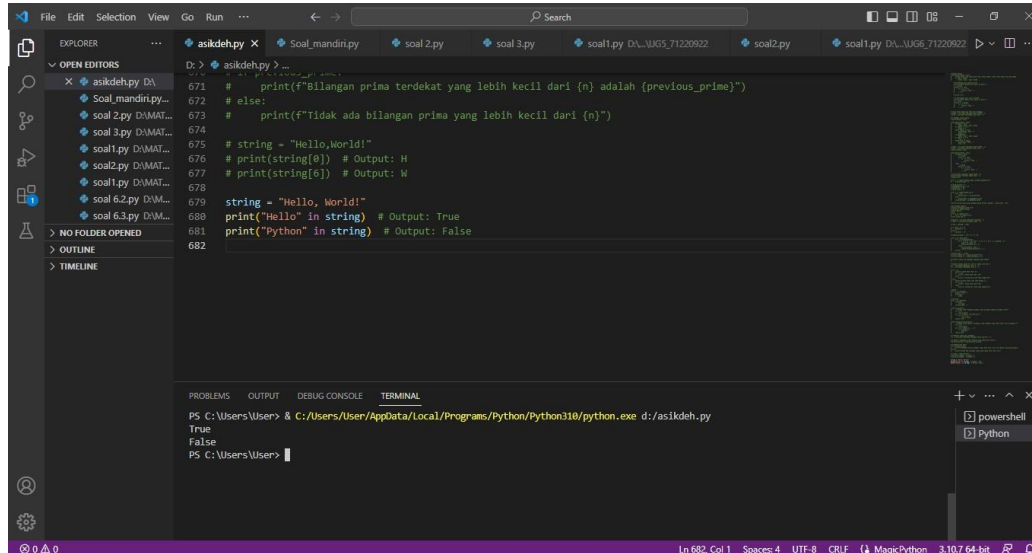
Nah untuk jadi untuk penggunaan string harus juga memakai variabel untuk kita menyimpan suatu nilai. Contohnya saya memakai variabel string dengan nilai (“Hello, World”). Lalu ketikan print dengan memakai tanda kurung “( )” lalu pakai tanda kurung “[ ]” kurung ini digunakan kita mau mengambil nomor index ke berapa.

### MATERI 3

#### Operator dan Metode String

## OPERATOR in

Menggunakan operator in pada python digunakan untuk memeriksa keanggotaan suatu nilai. Pada string operator in digunakan untuk memeriksa keberadaan sebuah substring dalam string. Seperti contoh berikut:



The screenshot shows a Python IDE with a file explorer on the left and a code editor in the center. The code editor displays a Python script with the following lines:

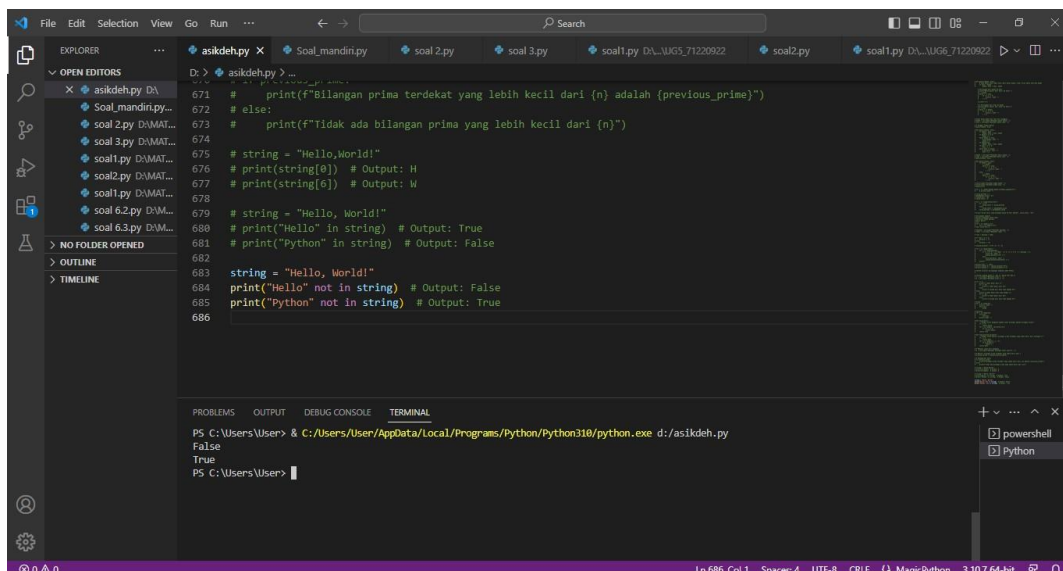
```
671 # print(f"Bilangan prima terdekat yang lebih kecil dari {n} adalah {previous_prime}")
672 # else:
673 #     print(f"Tidak ada bilangan prima yang lebih kecil dari {n}")
674
675 # string = "Hello,World!"
676 # print(string[0]) # Output: H
677 # print(string[6]) # Output: W
678
679 string = "Hello, World!"
680 print("Hello" in string) # Output: True
681 print("Python" in string) # Output: False
682
```

The terminal window at the bottom shows the output of the script:

```
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe d:/asikdeh.py
True
False
PS C:\Users\User>
```

Pada contoh di atas, kita menggunakan operator in untuk memeriksa apakah substring hello dan python terdapat dalam string. Hasilnya substring hello ditemukan dalam string sehingga operator in mengembalikan nilai true. Sedangkan substring python tidak ditemukan dalam string, sehingga operator in mengembalikan nilai false.

Selain itu ada juga operator yang bernama not in gunanya untuk memeriksa kebalikan dari operator in yaitu apakah substring tidak terdapat dalam string. Seperti contoh:



The screenshot shows a Python IDE with a file explorer on the left and a code editor in the center. The code editor displays a Python script with the following lines:

```
683 string = "Hello, World!"
684 print("Hello" not in string) # Output: False
685 print("python" not in string) # Output: True
686
```

The terminal window at the bottom shows the output of the script:

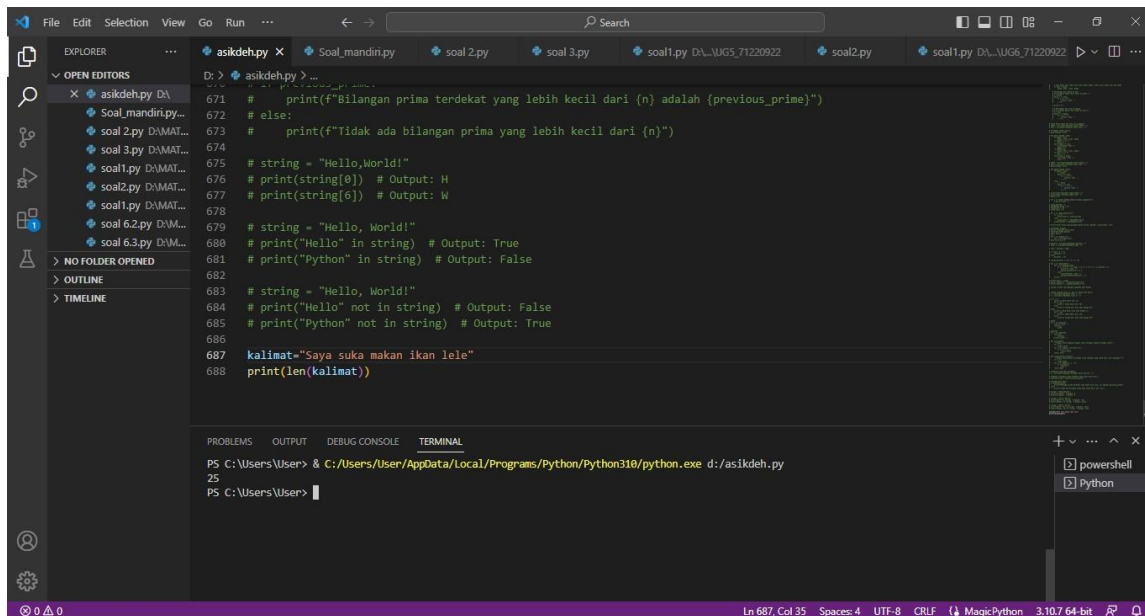
```
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe d:/asikdeh.py
False
True
PS C:\Users\User>
```

Pada contoh di atas kita menggunakan operator not in untuk memeriksa apakah substring hello dan python tidak terdapat dalam string. Hasilnya substring hello terdapat dalam string, sehingga

operator not in mengembalikan nilai false. Sedangkan substring python tidak ditemukan dalam string sehingga operator not in mengembalikan nilai true.

## Funksi Len

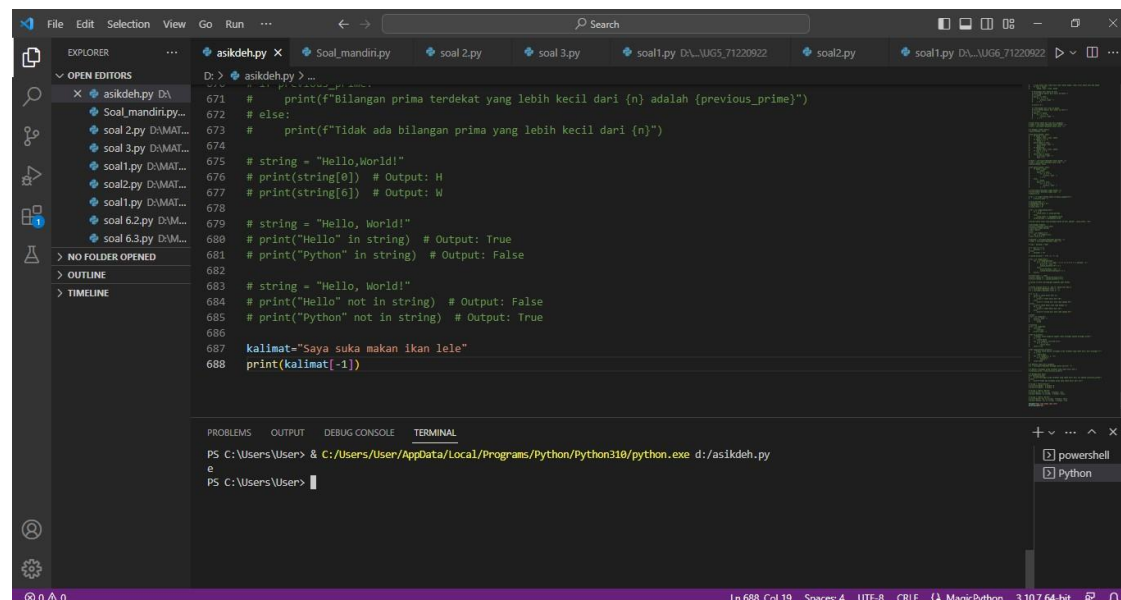
Funksi len untuk mengetahui berapa panjang jumlah karakter dari sebuah string. Untuk menampilkan huruf terakhir dari sebuah karakter menggunakan operator len(kalimat)-1, karena indeks dimulai dari 0. Seperti contoh berikut:



```
File Edit Selection View Go Run ... Search
EXPLORER
  OPEN EDITORS
    askideh.py
    Soal_mandiri.py
    soal 2.py
    soal 3.py
    soal1.py
    soal2.py
    soal1.py
    soal 6.2.py
    soal 6.3.py
  NO FOLDER OPENED
  OUTLINE
  TIMELINE
D:\> askideh.py
671 # print(f"Bilangan prima terdekat yang lebih kecil dari {n} adalah {previous_prime}")
672 # else:
673 # print(f"Tidak ada bilangan prima yang lebih kecil dari {n}")
674
675 # string = "Hello,World!"
676 # print(string[0]) # Output: H
677 # print(string[6]) # Output: W
678
679 # string = "Hello, World!"
680 # print("Hello" in string) # Output: True
681 # print("Python" in string) # Output: False
682
683 # string = "Hello, World!"
684 # print("Hello" not in string) # Output: False
685 # print("Python" not in string) # Output: True
686
687 kalimat="Saya suka makan ikan lele"
688 print(len(kalimat))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe d:/askideh.py
25
PS C:\Users\User>
```

Pada contoh di atas saya tetap memakai variabel untuk menyimpan sebuah nilai. disini saya memakai operator len untuk melihat jumlah karater yang ada. Ada satu contoh lagi menggunakan len tapi kita hanya mengambil kata bagian belakang. Contoh:



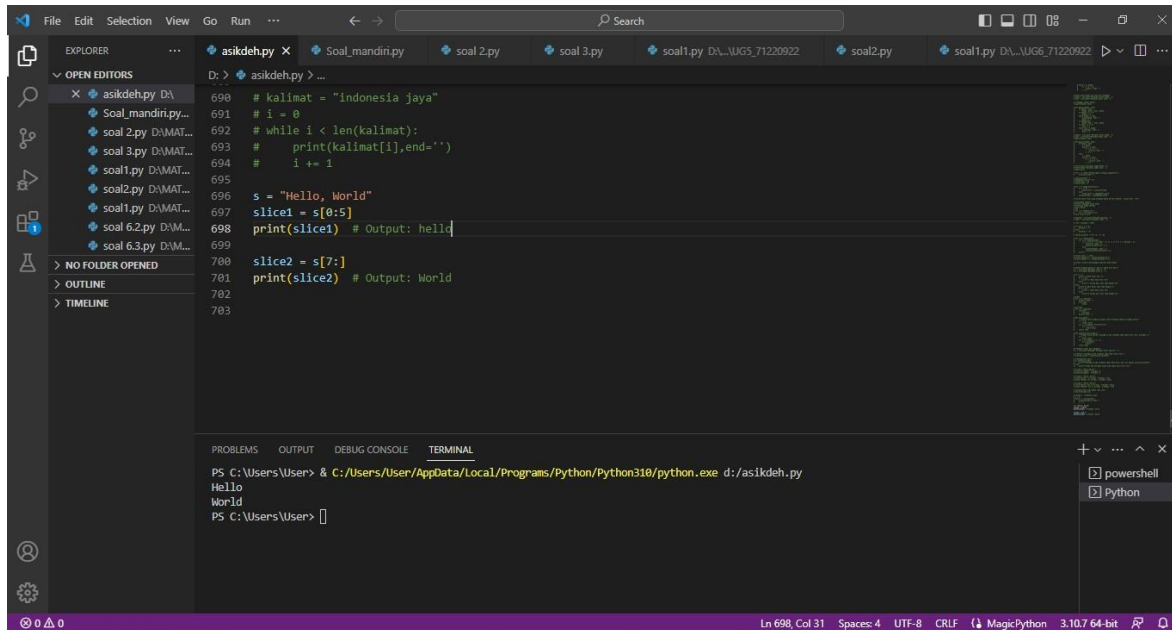
```
File Edit Selection View Go Run ... Search
EXPLORER
  OPEN EDITORS
    askideh.py
    Soal_mandiri.py
    soal 2.py
    soal 3.py
    soal1.py
    soal2.py
    soal1.py
    soal 6.2.py
    soal 6.3.py
  NO FOLDER OPENED
  OUTLINE
  TIMELINE
D:\> askideh.py
671 # print(f"Bilangan prima terdekat yang lebih kecil dari {n} adalah {previous_prime}")
672 # else:
673 # print(f"Tidak ada bilangan prima yang lebih kecil dari {n}")
674
675 # string = "Hello,World!"
676 # print(string[0]) # Output: H
677 # print(string[6]) # Output: W
678
679 # string = "Hello, World!"
680 # print("Hello" in string) # Output: True
681 # print("Python" in string) # Output: False
682
683 # string = "Hello, World!"
684 # print("Hello" not in string) # Output: False
685 # print("Python" not in string) # Output: True
686
687 kalimat="Saya suka makan ikan lele"
688 print(kalimat[-1])

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe d:/askideh.py
e
PS C:\Users\User>
```

Pada contoh di atas saya menggunakan indeks -1 untuk mengakses kata terakhir dalam kalimat, dan karakter terakhir dalam string. Dengan menggunakan indeks -1 kita tidak perlu menggunakan fungsi len untuk menghitung jumlah kata yang ada.

## String Slice

String slice pada python adalah cara untuk memotong atau mengambil potongan dari string yang lebih panjang. Potongan string ini dapat diambil dengan menentukan indeks awal dan akhir potongan, dan juga dapat disertai dengan step size untuk mengambil setiap karakter ke-n dari potongan. Contohnya seperti ini:



```
690 # kalimat = "Indonesia jaya"
691 # i = 0
692 # while i < len(kalimat):
693 #     print(kalimat[i],end='')
694 #     i += 1
695
696 s = "Hello, World"
697 slice1 = s[0:5]
698 print(slice1) # Output: hello
699
700 slice2 = s[7:]
701 print(slice2) # Output: World
702
703
```

Terminal Output:

```
PS C:\Users\User> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe d:/asikdeh.py
Hello
World
PS C:\Users\User>
```

Pada contoh gambar di atas untuk variabel slice1 start mengambil potongan string dari indeks ke-0 hingga ke-5, sedangkan untuk slice 2 mengambil potongan string dari indeks ke-7 hingga akhir string.

Berikut merupakan beberapa method yang biasa 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 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()

## MATERI 4

Parsing string merupakan proses mengubah string menjadi tipe data yang lebih kompleks atau terstruktur, seperti list, tuple, atau dictionary. Proses parsing umumnya dilakukan dengan cara membagi string menjadi beberapa bagian yang lebih kecil, lalu mengkonversi bagian bagian tersebut ke tipe data yang diinginkan.