

LAPORAN PRAKTIKUM PEMROGRAMAN PYTHON

PRAKTIKUM IV STRING LOOP



Disusun oleh :
Winasis Widya Wisesa
V3922048

Dosen
Yusuf Fadlila Rachman. S.Kom., M.Kom

**PS D-III TEKNIK INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS SEBELAS MARET
2023**

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam pemrograman, Loop dan String sering digunakan secara bersamaan. Loop dapat digunakan untuk melakukan operasi pada setiap karakter dalam sebuah String, seperti membalikkan String atau mengubah setiap karakter menjadi huruf besar atau kecil. Sebaliknya, String dapat digunakan dalam Loop untuk menyimpan hasil operasi atau menghasilkan output yang terbentuk dari beberapa elemen. Dalam kombinasi, Loop dan String sangat berguna dalam pengolahan data dan manipulasi teks, yang merupakan bagian penting dari banyak aplikasi dan layanan.

1.2. Tujuan

Memanipulasi teks: String sangat berguna untuk merepresentasikan teks dalam Python. Mahasiswa dapat mempelajari bagaimana mengolah String, seperti memotong, menggabungkan, dan memodifikasi karakter dalam String, untuk memanipulasi teks sesuai kebutuhan.

1.3. Manfaat

Pengembangan aplikasi web: Python banyak digunakan untuk mengembangkan aplikasi web. Dalam mempelajari String dan Loop Python, mahasiswa akan mempelajari cara mengolah teks dan data dalam Python untuk membangun aplikasi web yang efektif.

1.4. Alat dan Bahan

- Laptop
- Spyder Anaconda
- Windows/Linux

BAB II

HASIL DAN PEMBAHASAN

The screenshot shows the Spyder Python IDE interface. On the left, there is a code editor with the file 'loop python.py' open. The code defines a function that prints a Fibonacci triangle. In the center, there is a variable explorer showing the state of variables: a=0, b=1, c=1, i=9, j=8. On the right, there is a Jupyter-style notebook interface with two cells. Cell [1] contains the command 'In [1]: runfile('C:/Users/LENOVO/Downloads/loop python.py', wdir='C:/Users/LENOVO/Downloads')' and its output: 0, 0 1, 0 1 1, 0 1 2 3, 0 1 2 3 5, 0 1 2 3 5 8, 0 1 2 3 5 8 13 21. Cell [2] is currently selected with the command 'In [2]:'. The bottom status bar shows the Python version (3.9.13), memory usage (Mem 5%), and system information (12:11 AM 3/12/2023).

Script di atas adalah program Python yang mencetak barisan bilangan Fibonacci dalam bentuk segitiga dengan 10 baris menggunakan nested loop for.

Pada awal program, terdapat inisialisasi dua variabel ‘a’ dan ‘b’ yang mewakili dua bilangan pertama dalam barisan Fibonacci. ‘a’ diinisialisasi dengan nilai 0 dan ‘b’ diinisialisasi dengan nilai 1.

Setelah itu, program menggunakan nested loop for untuk menghasilkan barisan bilangan Fibonacci dalam bentuk segitiga dengan 10 baris. Loop pertama dijalankan sebanyak 10 kali dengan menggunakan fungsi range(10) untuk mengatur banyaknya baris. Loop kedua dijalankan sebanyak i kali (loop dalam loop) dengan menggunakan fungsi range(i) untuk mengatur banyaknya kolom pada setiap baris.

Pada setiap iterasi loop kedua, program menggunakan sebuah if statement untuk menentukan kondisi awal. Jika ‘j’ sama dengan 0, maka program

mencetak bilangan pertama ‘a’. Jika j sama dengan 1, maka program mencetak bilangan kedua ‘b’. Jika tidak, program melakukan penghitungan bilangan Fibonacci dengan menjumlahkan ‘a’ dan ‘b’ dan menyimpan hasilnya pada variabel ‘c’. Setelah itu, program menggeser nilai variabel ‘a’ dan b dengan memasukkan nilai ‘b’ ke dalam ‘a’ dan nilai ‘c’ ke dalam ‘b’. Program kemudian mencetak hasil penghitungan Fibonacci tersebut dengan mencetak nilai ‘c’.

Setelah selesai melakukan penghitungan pada satu baris, program mencetak baris baru dengan menggunakan fungsi print(""). Setelah itu, program mengembalikan nilai variabel ‘a’ dan ‘b’ ke nilai awal yaitu 0 dan 1 untuk memulai penghitungan pada baris selanjutnya.

Dengan demikian, program menghasilkan segitiga bilangan Fibonacci dengan 10 baris, di mana setiap bilangan dalam barisan Fibonacci dihitung menggunakan teknik pengulangan dan penugasan nilai variabel.

```

string.py
```

```

1  nama_lengkap = input("Masukkan nama Lengkap Anda: ")
2  # Menghitung jumlah huruf dari nama lengkap
3  jumlah_huruf = len(nama_lengkap.replace(" ", ""))
4
5  # Menghitung jumlah huruf vokal dari nama lengkap
6  vokal = "aeiouAEIOU"
7  jumlah_vokal = sum([1 for huruf in nama_lengkap if huruf in vokal])
8
9  # Menghitung jumlah huruf konsonan dari nama lengkap
10 konsonan = "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"
11 jumlah_konsonan = sum([1 for huruf in nama_lengkap if huruf in konsonan])
12
13 # Menampilkan hasil
14 print("Jumlah huruf dari nama lengkap:", jumlah_huruf)
15 print("Jumlah huruf vokal dari nama lengkap:", jumlah_vokal)
16 print("Jumlah huruf konsonan dari nama lengkap:", jumlah_konsonan)
17

```

Variable Explorer:

Name	Type	Size	Value
jumlah_huruf	int	1	18
jumlah_konsonan	int	1	10
jumlah_vokal	int	1	8
konsonan	str	42	bcd...uvwxyzBCDFGHJKLMNPQRSTVWXYZ
nama_lengkap	str	20	Winasia Widya Wisesa
vokal	str	10	aeiouAEIOU

Console:

```

Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/LENOVO/.spyder-py3/temp.py', wdir='C:/Users/LENOVO/.spyder-py3')
Masukkan nama lengkap Anda: Winasia Widya Wisesa
Jumlah huruf dari nama lengkap: 18
Jumlah huruf vokal dari nama lengkap: 8
Jumlah huruf konsonan dari nama lengkap: 10

In [2]:

```

Script di atas adalah sebuah program Python yang menghitung jumlah huruf, huruf vokal, dan huruf konsonan dari sebuah nama lengkap yang dimasukkan oleh pengguna melalui input.

Pertama-tama, program meminta pengguna untuk memasukkan nama lengkap mereka menggunakan fungsi `input()` dan menyimpan hasilnya dalam variabel `nama_lengkap`.

Selanjutnya, program menghitung jumlah huruf dari nama lengkap dengan mengurangi jumlah spasi dari total panjang nama menggunakan fungsi `len()` dan `count()`. Ini dilakukan dengan menghitung jumlah spasi dalam nama dan mengurangi jumlah spasi tersebut dari total panjang nama lengkap. Hasilnya disimpan dalam variabel `jumlah_huruf`.

Selanjutnya, program menghitung jumlah huruf vokal dari nama lengkap menggunakan variabel `vokal` yang berisi string yang berisi huruf vokal (kecil dan besar). Program kemudian menghitung jumlah kemunculan setiap huruf vokal dalam nama lengkap menggunakan fungsi `count()`, dan menjumlahkan hasilnya

menggunakan fungsi sum(). Hasilnya disimpan dalam variabel jumlah_vokal.

Setelah itu, program menghitung jumlah huruf konsonan dari nama lengkap dengan mengurangi jumlah huruf vokal dari total jumlah huruf. Hasilnya disimpan dalam variabel jumlah_konsonan.

Terakhir, program menampilkan hasil dengan menggunakan fungsi print(). Program mencetak jumlah huruf, huruf vokal, dan huruf konsonan dari nama lengkap yang dimasukkan oleh pengguna dengan memasukkan variabel ke dalam string f-strings.