

**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 a Python IDE with a script that prints Fibonacci numbers in a triangular pattern. The script uses nested loops and conditional statements to generate the sequence. The output is displayed in the console, showing the first 10 rows of the triangle.

```
1 a = 0
2 b = 1
3
4 for i in range(10):
5     for j in range(i):
6         if j == 0:
7             print(a, end=" ")
8             print(b, end=" ")
9         else:
10            c = a + b
11            a = b
12            b = c
13            print(c, end=" ")
14    print("")
15    a = 0
16    b = 1
```

The console output shows the following pattern:

```
0 1
0 1 1
0 1 1 2
0 1 1 2 3
0 1 1 2 3 5
0 1 1 2 3 5 8
0 1 1 2 3 5 8 13
0 1 1 2 3 5 8 13 21
```

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.

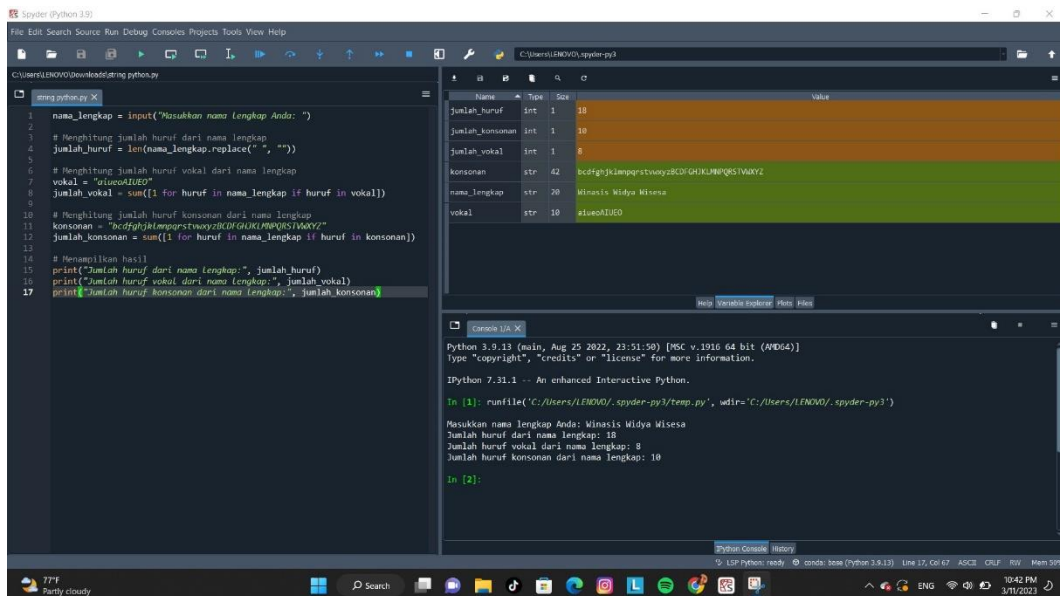
Setelah itu, program menggunakan nested loop for untuk menghasilkan barisan bilangan Fibonacci dalam bentuk segitiga dengan 10 baris. Loop pertama dijalankan sebanyak 10 kali dengan menggunakan fungsi range(10) untuk mengatur banyaknya baris. Loop kedua dijalankan sebanyak i kali (loop dalam loop) dengan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



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.