

INPUT DAN OUTPUT PADA BAHASA PEMROGRAMAN PYTHON

(Studi Kasus : STMIK Sumedang)

Akbar Nur Syahrudin (A2.1700010), Tedi Kurniawan (A2.1700119)

Program Studi Teknik Informatika

STMIK Sumedang, Jl. Angkrek Situ No.19, Sumedang, 45323 Indonesia

ABSTRACT

Python adalah salah satu bahasa pemograman yang baru di masa sekarang, pada bahasa pemograman ini kita lebih simpel dalam dan singkat dalam membuat sebuah program, setiap program yang kita buat pasti dan pasti akan membutuhkan inputan dan hasil outputan. Dalam metode penginputannya pun bahasa ini sedikit berbeda, memang terlihat mudah tapi bukan berarti untuk di hiraukan karena pada kenyataan masih banyak yang kesulitan dalam membuat program pada python ini, dengan dibuat nya pembahasan ini semoga dapat membantu bagi pemula yang sedang belajar bahasa pemograman.

Kata kunci : input, output, python

1. Introduction

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa script meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan

bahasa script. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah:

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara **bebas**, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi Open Source maupun **General Public License (GPL)**.

2. Search Metode

a. Input (Memasukan Data)

Setiap program yang kita ciptakan nantinya akan berinteraksi dengan user yang mana program akan meminta data yang diperlukan oleh user misalnya dalam menghitung luas persegi panjang tentu diperlukan data berupa panjang dan lebar dari persegi panjang tersebut.

b. Output (Menampilkan Data)

Mencetak atau menampilkan informasi data yang dihasilkan oleh algoritma, misalnya dalam algoritma penghitung luas persegi panjang, hasil akhir yang diinginkan adalah luas persegi panjang dengan kata lain algoritma tersebut memiliki satu output yaitu luas persegi panjang.

c. Penggunaan input dan output pada python

Input

```
Nama_variabel = Input("masukan data :")
```

Penjelasannya, data yang kita masukan nantinya akan tersimpan dalam Nama_Variabel, namun apabila data yang diinput membutuhkan proses pengolahan kembali atau pengevaluasian maka perintah yang ditulis sebagai berikut:

```
Nama_variabel = eval(input("masukan data :"))
```

Namun ada juga data yang diproses secara otomatis tanpa meminta user untuk menginputkan, yaitu data yang ditulis langsung ketika membuat program, penulisannya sebagai berikut:

```
Nama_variabel = isi data
```

```
Nama_variabel2 = eval(input("masukan data :"))
```

Penjelasannya, "Nama_variabel = isi data" ini artinya ketika program dijalankan maka sistem tidak akan menampilkan perintah tersebut, namun data akan otomatis terbaca oleh sistem. Output

untuk menampilkan hasil program pada python, menggunakan perintah "print", penulisan programnya sebagai berikut:

```
Nama_variabel = Input("masukan data :")
```

```
Print("hasil akhir nya adalah : ", Nama_Variabel)
```

Penjelasannya, perintah "print" akan memanggil data yang tersimpan pada Nama_variabel.

3. Pengaplikasian Program

A. Cara Memasukkan Input

Python sudah menyediakan fungsi `input()` dan `raw_input()` untuk mengambil inputan dari keyboard.

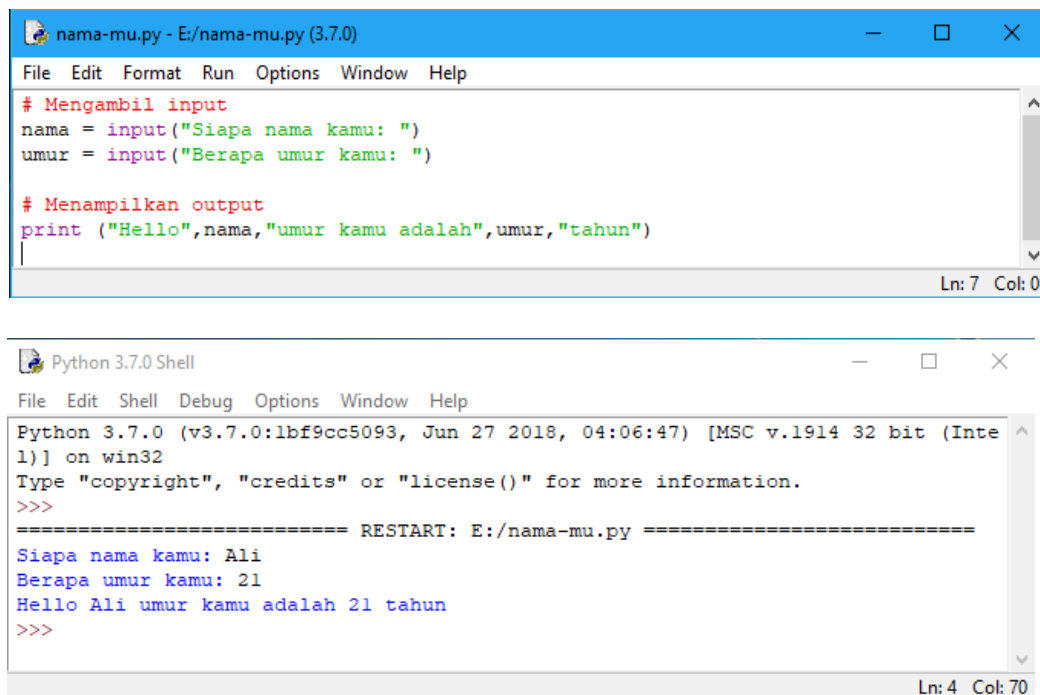
Cara pakai:

```
Nama_Variabel = input("masukkan text")
```

Artinya, teks yang kita inputkan dari keyboard akan disimpan ke dalam nama_variabel.

```
# Mengambil input nama =  
input("Siapa nama kamu: ") umur =  
input("Berapa umur kamu: ")  
  
# Menampilkan output print  
("Hello",nama,"umur kamu adalah",umur,"tahun")
```

Hasilnya:



The image shows two windows from a Python 3.7.0 IDE. The top window, titled 'nama-mu.py - E:/nama-mu.py (3.7.0)', displays the source code of a script that takes a name and age as input and prints a greeting. The bottom window, titled 'Python 3.7.0 Shell', shows the output of running the script. It displays the prompt, the input 'Ali' and '21', and the resulting output 'Hello Ali umur kamu adalah 21 tahun'.

```
nama-mu.py - E:/nama-mu.py (3.7.0)  
File Edit Format Run Options Window Help  
# Mengambil input  
nama = input("Siapa nama kamu: ")  
umur = input("Berapa umur kamu: ")  
  
# Menampilkan output  
print ("Hello",nama,"umur kamu adalah",umur,"tahun")  
Ln: 7 Col: 0  
  
Python 3.7.0 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: E:/nama-mu.py =====  
Siapa nama kamu: Ali  
Berapa umur kamu: 21  
Hello Ali umur kamu adalah 21 tahun  
>>>  
Ln: 4 Col: 70
```

Fungsi `input()` digunakan untuk mengambil data angka. Sedangkan `raw_input()` untuk mengambil teks.

Pada Python3 cukup menggunakan fungsi `input()`, karena fungsi `raw_input()` telah disederhanakan menjadi `input()` dan fungsi `raw_input()` dihilangkan pada versi ini.

B. Cara Menampilkan Output

Contoh:

```
print "Hello World!" print nama_variabel print "Gabung dengan", variabel
```

- Menampilkan Variabel dan Teks

Pada contoh di atas kita menggunakan tanda koma (,) untuk menggabungkan teks dan variabel yang akan ditampilkan. `nama = "Ali" print "Hello",nama`

Hasil: Hello Ali

Antara kata Hello dan Ali terdapat spasi sebagai pemisah, karena kita menggunakan tanda koma atau bisa juga dengan tanda plus (+).

- Menggunakan Fungsi format()

Fungsi format() akan menggabungkan isi variabel dengan teks.

Contoh :

```
nama = input("Nama: ")
```

```
print("Hello {} apa kabar?".format(nama))
```

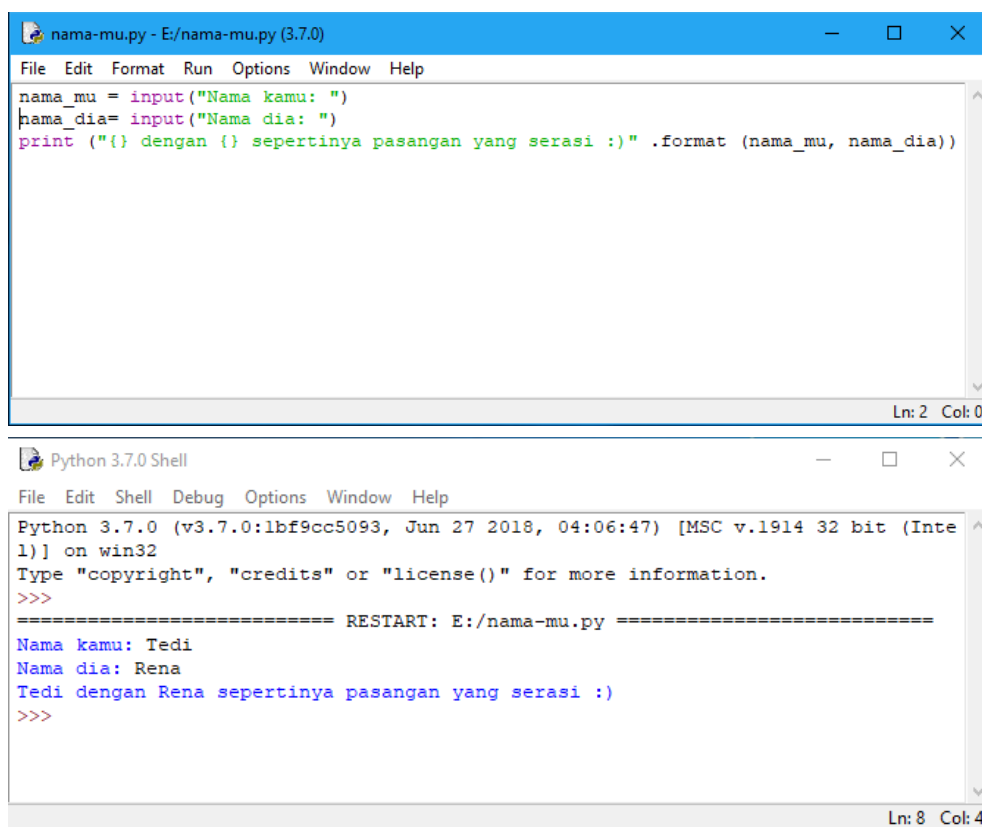
Tanda {} akan otomatis diganti sesuai dengan nilai yang diinput ke variabel nama.

Contoh lainnya:

```
nama_mu = input("Nama kamu: ")
```

```
nama_dia= input("Nama dia: ")
```

```
print("{} dengan {} sepertinya pasangan yang serasi :)".format (nama_mu, nama_dia))
```



The screenshot displays two windows from a Python 3.7.0 IDE. The top window, titled 'nama-mu.py - E:/nama-mu.py (3.7.0)', contains the following code:

```
File Edit Format Run Options Window Help
nama_mu = input("Nama kamu: ")
nama_dia= input("Nama dia: ")
print("{} dengan {} sepertinya pasangan yang serasi :)".format (nama_mu, nama_dia))
```

The bottom window, titled 'Python 3.7.0 Shell', shows the execution output:

```
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/nama-mu.py =====
Nama kamu: Tedi
Nama dia: Rena
Tedi dengan Rena sepertinya pasangan yang serasi :)
>>>
```

C. Menggunakan String Formatting Cara Lama

Penggabungan teks dan variabel cara lama menggunakan simbol persen (%). Contoh:

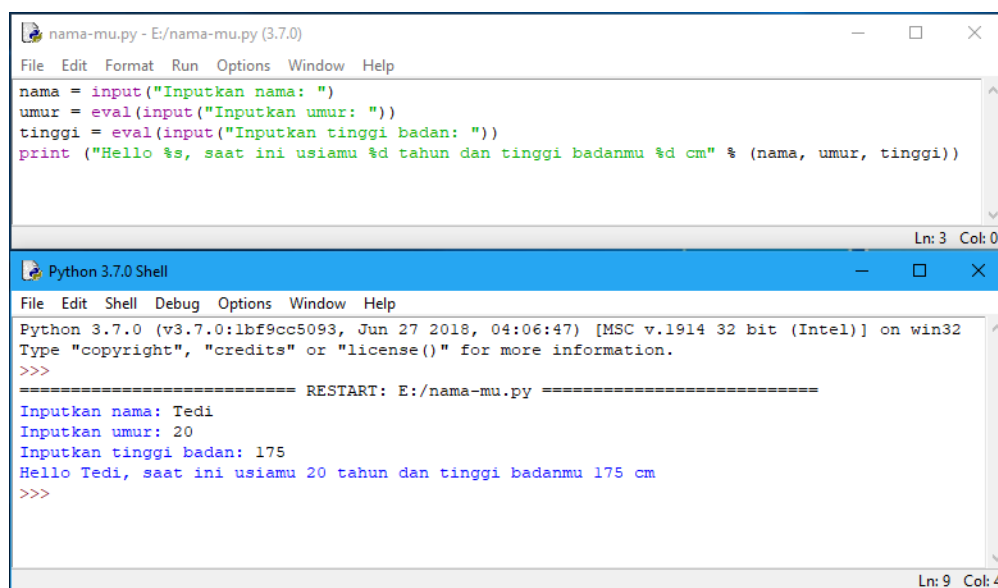
```
nama = input("Inputkan nama: ")
print ("Selamat datang %s" % nama)
```

Tanda %s akan otomatis terganti dengan nilai yang kita inputkan ke variabel nama.

Contoh:

```
nama = input("Inputkan nama: ")
umur = eval(input("Inputkan umur: "))
tinggi = eval(input("Inputkan tinggi badan: "))

print ("Hello %s, saat ini usiamu %d tahun dan tinggi badanmu %d
cm" % (nama, umur, tinggi))
```



The screenshot displays a Python 3.7.0 IDE window titled 'nama-mu.py - E:/nama-mu.py (3.7.0)'. The code in the editor is as follows:

```
nama = input("Inputkan nama: ")
umur = eval(input("Inputkan umur: "))
tinggi = eval(input("Inputkan tinggi badan: "))
print ("Hello %s, saat ini usiamu %d tahun dan tinggi badanmu %d cm" % (nama, umur, tinggi))
```

Below the editor is a 'Python 3.7.0 Shell' window showing the execution of the script. The output is:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/nama-mu.py =====
Inputkan nama: Tedi
Inputkan umur: 20
Inputkan tinggi badan: 175
Hello Tedi, saat ini usiamu 20 tahun dan tinggi badanmu 175 cm
>>>
```

Penjelasan : Tanda %s untuk tipe data teks, %d untuk angka (desimal).

4. Conclusion

input dan output adalah proses dasar dalam sebuah pemograman, tanpa adanya input dan output mustahil program dapat berjalan dan berinteraksi dengan user, dalam mempelajari bahasa pemograman yang harus kita pahami yaitu struktur pemograman bukan menghafal coding, karena setiap adanya pembaruan/*update* terkadang terdapat rumus/coding yang berubah.

REFERENCES

- [1]. Kontributor Wikipedia. "Python (bahasa pemrograman)." *Wikipedia, Ensiklopedia Bebas*. Wikipedia, Ensiklopedia Bebas, 29 mei 2018.
- [2]. Cahyono. "Input dan Output" *Workshop Python 101*. <http://sakti.github.io/python101/io.html>, 28 mei 2018