**ARTIKEL**

**INPUT DAN OUTPUT STRING**



**DOSEN PENGAMPU** :

**Alun Sujjada,S.Kom,M.T**

**Di susun oleh** :

**Mifta siti mariam**

**PRODI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK KOMPUTER DAN DESAIN**

**UNIVERSITAS NUSA PUTRA**

**TAHUN AKADEMIK 2025/2026**

**PENDAHULUAN**

Python adalah bahasa pemrograman tingkat tinggi yang dirancang untuk kemudahan penggunaan dan membaca kode. Python sangat populer di kalangan pengembang karena sintaksisnya yang jelas dan mudah dipahami, serta dukungan untuk berbagai paradigma pemrograman, seperti pemrograman berorientasi objek, fungsional, dan prosedural.

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja.

Karena sifatnya yang serba guna dan mudah digunakan, ia menjadi bahasa pemrograman yang paling banyak digunakan. Terutama untuk mereka yang masih pemula.

Berdasarkan survei pengembang [Stack Overflow tahun 2022](https://survey.stackoverflow.co/2022/), Python menjadi bahasa pemrograman terpopuler keempat. Sebanyak hampir 50% dari responden mengatakan bahwa mereka menggunakan hampir setengah dari waktu kerja mereka dengan menggunakan bahasa pemrograman ini.

Nama Python sendiri berasal dari Monty Python. Ketika Guido van Rossum membuatnya, dia juga sedang membaca skrip Sirkus Terbang Monty Python BBC. Menurutnya nama itu singkat dan sedikit misterius. Karena itulah, sang kreator memilih menggunakan nama tersebut untuk bahasa pemrograman yang dibuatnya itu.

Berikut adalah penjelasan lengkap tentang Python, termasuk fitur-fitur utamanya:

**1. Sejarah dan Pengembangan Python**

Python pertama kali diciptakan oleh **Guido van Rossum** pada akhir 1980-an di Belanda dan pertama kali dirilis pada tahun 1991. Nama "Python" diambil dari serial televisi komedi Inggris **Monty Python's Flying Circus**. Python dirancang dengan filosofi untuk membuat kode yang mudah dibaca dan dipahami.

**2. Fitur Utama Python**

* **Sintaksis yang Mudah Dipahami**: Python memiliki sintaksis yang lebih dekat dengan bahasa manusia (seperti bahasa Inggris), yang menjadikannya mudah untuk dipelajari, bahkan bagi pemula.
* **Interpreted Language**: Python adalah bahasa pemrograman yang diinterpretasikan, yang berarti kode dieksekusi baris demi baris oleh interpreter, bukan dikompilasi menjadi kode mesin terlebih dahulu.
* **Cross-platform**: Python dapat dijalankan di berbagai platform, seperti Windows, Linux, macOS, dan lainnya, tanpa perlu modifikasi kode.
* **Dukungan untuk Pemrograman Berorientasi Objek**: Python mendukung pemrograman berorientasi objek (OOP), yang memungkinkan pengguna untuk mendefinisikan kelas dan objek, serta prinsip pewarisan, enkapsulasi, dan polimorfisme.
* **Libraries dan Frameworks yang Banyak**: Python memiliki banyak pustaka dan framework yang kuat seperti NumPy, Pandas, Matplotlib (untuk analisis data), Django, Flask (untuk pengembangan web), dan banyak lagi.
* **Dukungan untuk Pemrograman Fungsional**: Python juga mendukung paradigma pemrograman fungsional, dengan fitur seperti lambda, map, filter, dan reduce.
* **Manajemen Memori Otomatis**: Python memiliki pengelolaan memori otomatis dengan sistem pemungutan sampah (garbage collection) yang menghapus objek yang tidak digunakan untuk menghemat memori.

**3. Struktur Dasar dalam Python**

Berikut adalah beberapa struktur dasar dalam Python:

**a. Variabel dan Tipe Data**

Python mendukung berbagai tipe data, seperti:

* **Integer**: Angka bulat, contohnya x = 5
* **Float**: Angka desimal, contohnya y = 3.14
* **String**: Teks atau karakter, contohnya name = "Python"
* **Boolean**: Nilai benar atau salah, contohnya is\_active = True
* **List**: Struktur data yang dapat menampung berbagai tipe data, contohnya numbers = [1, 2, 3, 4]
* **Tuple**: Mirip dengan list, tetapi bersifat immutable (tidak dapat diubah), contohnya coordinates = (10, 20)
* **Dictionary**: Struktur data yang menyimpan pasangan key-value, contohnya person = {"name": "Alice", "age": 25}

**b. Operator**

Python memiliki berbagai jenis operator:

* **Operator Aritmatika**: +, -, \*, /, // (pembagian bulat), % (modulus), \*\* (pangkat)
* **Operator Perbandingan**: ==, !=, <, >, <=, >=
* **Operator Logika**: and, or, not
* **Operator Penugasan**: =, +=, -=, \*=, /=
* **Operator Bitwise**: &, |, ^, ~, <<, >>

**c. Kontrol Alur**

* **If-else**: Digunakan untuk membuat percabangan dalam program.

if x > 10:

print("x lebih besar dari 10")

else:

print("x tidak lebih besar dari 10")

* **Looping**: Python mendukung for dan while loops untuk iterasi.

for i in range(5):

print(i)

**d. Fungsi**

Fungsi digunakan untuk mengorganisasi dan mendefinisikan blok kode yang bisa digunakan kembali.

def greet(name):

return f"Hello, {name}!"

**e. Kelas dan Objek (OOP)**

Python mendukung pemrograman berorientasi objek, memungkinkan kita mendefinisikan kelas dan objek:

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def greet(self):

return f"Hello, my name is {self.name} and I'm {self.age} years old."

person1 = Person("Alice", 30)

print(person1.greet())

**4. Modul dan Pustaka**

Python memiliki pustaka standar yang sangat luas dan banyak modul yang dapat digunakan untuk berbagai keperluan:

* Matematika: math, cmath
* Operasi File: os, shutil
* Jaringan: socket, requests
* Pemrograman Web: Flask, Django
* Analisis Data: Pandas, NumPy
* Visualisasi: Matplotlib, Seaborn

**Instalasi Pustaka dengan pip**

**Python juga menggunakan manajer paket bernama pip untuk menginstal pustaka eksternal.**

Bash

pip install numpy

**5. Pemrograman Asinkron**

Python mendukung pemrograman asinkron melalui modul asyncio dan async/await. Hal ini memungkinkan eksekusi kode secara paralel, yang sangat berguna untuk aplikasi yang membutuhkan I/O tinggi, seperti aplikasi web.

**6. Pengelolaan Memori**

Python menggunakan pengumpulan sampah otomatis untuk mengelola memori, yang berarti bahwa programmer tidak perlu khawatir tentang alokasi dan dealokasi memori secara manual.

**7. Python di Berbagai Bidang**

Python banyak digunakan dalam berbagai bidang, antara lain:

* Pengembangan Web: Dengan framework seperti Django dan Flask.
* Ilmu Data dan Pembelajaran Mesin: Dengan pustaka seperti NumPy, Pandas, dan scikit-learn.
* Otomatisasi dan Skriping: Untuk menulis skrip dan tugas otomatisasi.
* Aplikasi GUI: Menggunakan pustaka seperti Tkinter dan PyQt.
* Pengembangan Game: Menggunakan pustaka seperti Pygame.

**8. Kelebihan dan Kekurangan Python**

*Kelebihan***:**

* Mudah dipelajari dan digunakan
* Sintaksis yang jelas dan elegan
* Dukungan komunitas yang besar
* Banyak pustaka dan framework yang tersedia
* Cross-platform dan portabilitas tinggi

*Kekurangan:*

* Kecepatan eksekusi lebih lambat dibandingkan bahasa seperti C atau C++
* Tidak cocok untuk aplikasi dengan kebutuhan performa tinggi (misalnya game real-time)
* Pengelolaan memori bisa mempengaruhi kinerja dalam aplikasi besar

**9. Contoh Program Sederhana**

Berikut adalah contoh program sederhana yang mencetak "Hello, World!" di Python:

**Python**

**print("Hello, World!")**

Python adalah bahasa pemrograman yang sangat fleksibel, mudah dipelajari, dan dapat digunakan untuk berbagai keperluan, dari pemrograman web hingga analisis data dan pengembangan perangkat lunak.

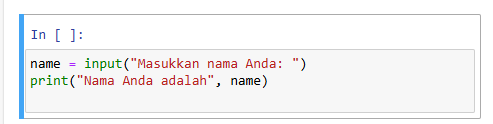
**Input**

Di Python, fungsi input() digunakan untuk mengambil input dari pengguna melalui keyboard. Fungsi ini mengembalikan nilai dalam bentuk string, yang berarti segala sesuatu yang dimasukkan oleh pengguna akan dianggap sebagai teks, meskipun angka atau jenis data lainnya dimasukkan.

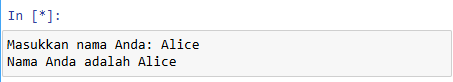
* 1. **Sintaks Dasar input()**

input(prompt)

* **prompt** (opsional): Merupakan teks yang ditampilkan kepada pengguna sebelum meminta input. Ini bisa berupa string yang memberikan instruksi atau penjelasan tentang apa yang harus dimasukkan oleh pengguna.



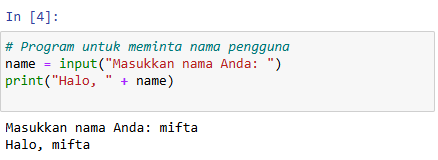
Output (tergantung pada input pengguna):



**2. Contoh Penggunaan input()**

**Berikut beberapa contoh penggunaan input() untuk meminta input dari pengguna.**

**a. Mengambil Input Teks**

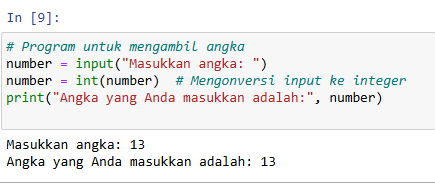
****

jika pengguna memasukkan Alice, outputnya adalah:

Halo, Alice

**b. Mengambil Input Angka**

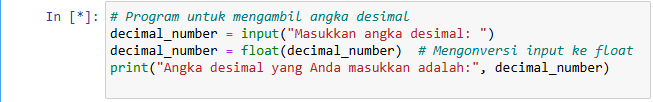
Karena input() mengembalikan string, jika Anda ingin mengambil angka, Anda perlu mengonversinya ke tipe data yang sesuai, seperti int() untuk bilangan bulat atau float() untuk bilangan desimal.



Jika pengguna memasukkan 10, outputnya adalah:

Angka yang Anda masukkan adalah: 10

c. **Mengambil Input Bilangan Desimal**

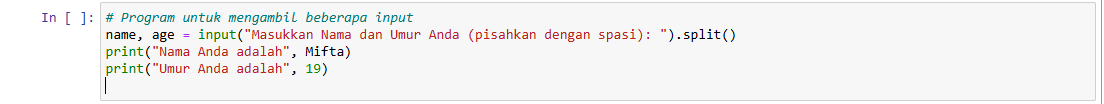


Jika pengguna memasukkan 3.14, outputnya adalah:

Angka desimal yang Anda masukkan adalah: 3.14

**3. Multiple Inputs**

Anda juga bisa meminta beberapa input sekaligus dalam satu baris dengan memisahkan input menggunakan spasi, dan kemudian membaginya menjadi beberapa bagian dengan metode split().



**4. Menangani Input dengan Validasi**

Saat menggunakan input(), penting untuk menangani kesalahan atau validasi input jika pengguna memasukkan data yang tidak sesuai. Berikut adalah contoh validasi untuk memastikan pengguna hanya memasukkan angka.

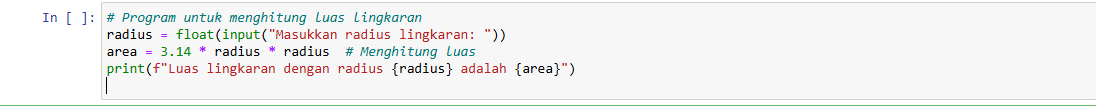


Jika pengguna memasukkan teks selain angka, misalnya abc, outputnya akan menjadi:



**5. Menggunakan input() dalam Program yang Lebih Kompleks**

Berikut adalah contoh program lebih kompleks yang menggunakan beberapa input untuk melakukan perhitungan atau operasi lainnya.



Jika pengguna memasukkan 5, outputnya adalah:



**Kesimpulan**

* Fungsi **input()** digunakan untuk mengambil data dari pengguna dalam bentuk string.
* Anda bisa mengonversi input ke tipe data lain (seperti int() atau float()) jika diperlukan.
* Anda bisa menambahkan prompt untuk memberikan instruksi atau informasi sebelum mengambil input.
* Penanganan kesalahan (seperti validasi) dapat dilakukan dengan menggunakan **try-except** untuk memastikan input yang diterima sesuai dengan yang diinginkan.

**String**

Output string dalam Python mengacu pada cara menampilkan atau mencetak string (teks) ke layar atau ke perangkat keluaran lainnya, seperti file. Python memiliki beberapa cara untuk menghasilkan output string, salah satunya adalah dengan menggunakan fungsi print(). Selain itu, ada beberapa metode lain untuk mengolah string dan menampilkannya.

Berikut adalah penjelasan lengkap tentang output string di Python.

**Fungsi string**

1. len()

Menghitung panjang string.

Contoh: len("hello") akan menghasilkan 5.

2. upper() dan lower()

Mengubah seluruh huruf dalam string menjadi huruf besar atau kecil.

Contoh: "hello".upper() menghasilkan "HELLO", sedangkan "HELLO".lower() menghasilkan "hello".

3. capitalize() dan title()

capitalize() mengubah huruf pertama menjadi kapital dan sisanya menjadi huruf kecil, sedangkan title() mengubah huruf pertama dari setiap kata menjadi kapital.

Contoh: "hello world".capitalize() menghasilkan "Hello world", dan "hello world".title() menghasilkan "Hello World".

4. strip(), lstrip(), dan rstrip()

Menghapus spasi atau karakter tertentu dari awal dan/atau akhir string.

Contoh: " hello ".strip() menghasilkan "hello".

5. split() dan join()

split() memecah string menjadi list berdasarkan separator tertentu, sedangkan join() menggabungkan elemen list menjadi satu string dengan pemisah yang ditentukan.

Contoh: "a,b,c".split(",") menghasilkan ['a', 'b', 'c'], dan ",".join(['a', 'b', 'c']) menghasilkan "a,b,c".

6. replace()

Mengganti bagian tertentu dari string dengan substring lain.

Contoh: "hello world".replace("world", "Python") menghasilkan "hello Python".

7. find() dan index()

Mencari posisi (index) dari substring di dalam string.

Contoh: "hello".find("e") menghasilkan 1.

(Perbedaan utamanya, jika substring tidak ditemukan, find() mengembalikan -1 sedangkan index() akan menghasilkan error.)

Fungsi pengecekan karakter

Seperti isalpha(), isdigit(), dan isalnum() untuk memeriksa apakah string hanya berisi huruf, angka, atau kombinasi keduanya.

Contoh: "hello".isalpha() menghasilkan True, sedangkan "123".isdigit() menghasilkan True.

Selain fungsi-fungsi tersebut, ada juga metode lain untuk pengolahan string seperti format string (misalnya dengan metode .format() atau f-string di Python) yang memudahkan penyisipan variabel ke dalam string.