

Pendahuluan

Dalam pemrograman, tipe data adalah konsep yang fundamental karena menentukan jenis nilai yang dapat disimpan dan dioperasikan dalam suatu variabel. Python, sebagai salah satu bahasa pemrograman yang populer dan serbaguna, menyediakan berbagai tipe data untuk memenuhi kebutuhan yang berbeda. Dua di antaranya yang paling dasar dan sering digunakan adalah **number** (angka) dan **string** (teks). Memahami cara kerja kedua tipe data ini sangat penting bagi setiap programmer yang ingin menguasai Python.

Tujuan dan Manfaat

Tujuan

1. Representasi Data:

- **Number:** Digunakan untuk merepresentasikan data numerik yang dapat digunakan dalam perhitungan matematis, analisis statistik, dan berbagai aplikasi ilmiah serta keuangan.
- **String:** Digunakan untuk merepresentasikan data teks, yang penting untuk berbagai aplikasi seperti pemrosesan bahasa alami, penanganan input/output pengguna, dan pengelolaan data tekstual.

2. Operasi dan Manipulasi Data:

- **Number:** Memungkinkan eksekusi berbagai operasi aritmatika dan fungsi matematika, termasuk penjumlahan, pengurangan, perkalian, pembagian, eksponensial, dan trigonometri.
- **String:** Memungkinkan operasi pengolahan teks seperti penggabungan, pengulangan, pencarian, pemotongan, dan penggantian teks, serta format teks yang kompleks.

3. Penyimpanan dan Pengambilan Data:

- **Number:** Memungkinkan penyimpanan dan pengambilan data numerik dalam berbagai format dan ukuran, termasuk integer, float, dan complex.
- **String:** Memungkinkan penyimpanan dan pengambilan data teks dalam berbagai format dan panjang, termasuk kalimat, paragraf, dan dokumen.

Manfaat

1. Fleksibilitas dan Kemudahan Penggunaan:

- **Number:** Python menyediakan berbagai tipe data numerik dan operasi yang mudah digunakan, memungkinkan penanganan berbagai jenis perhitungan dengan efisien.
- **String:** Python menawarkan berbagai metode manipulasi string yang kuat dan intuitif, memudahkan programmer dalam bekerja dengan teks.

2. Efisiensi dalam Pemrosesan Data:

- **Number:** Operasi numerik di Python dioptimalkan untuk kecepatan dan efisiensi, memungkinkan pemrosesan data dalam skala besar dengan performa tinggi.
- **String:** Operasi string di Python juga sangat efisien, mendukung berbagai tugas pemrosesan teks yang cepat dan akurat.

3. Kemudahan Pemeliharaan Kode:

- **Number:** Kode yang menggunakan tipe data numerik biasanya lebih mudah dibaca dan dipahami, memudahkan pemeliharaan dan pengembangan lebih lanjut.
- **String:** Manipulasi string yang mudah dipahami juga memfasilitasi penulisan dan pemeliharaan kode yang lebih bersih dan terstruktur.

4. Kompatibilitas dan Integrasi:

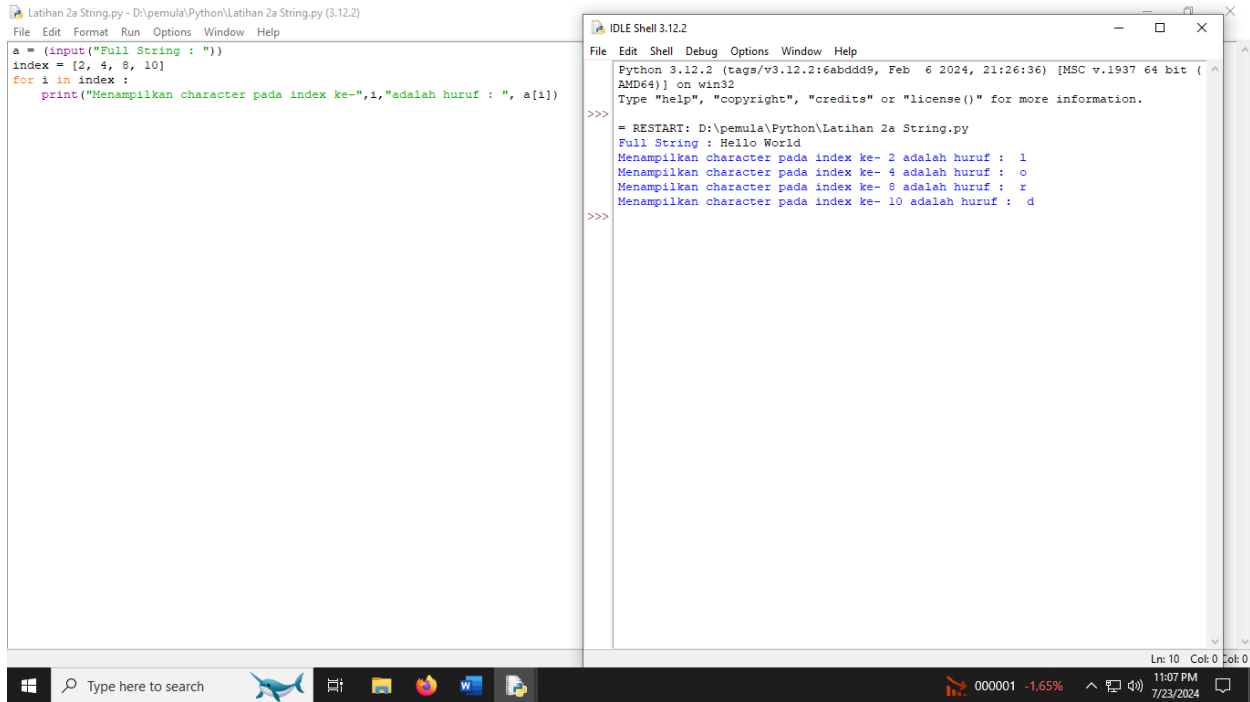
- **Number:** Tipe data numerik di Python kompatibel dengan berbagai perpustakaan dan alat analisis data, seperti NumPy, Pandas, dan SciPy, yang memperluas fungsionalitas pemrograman numerik.
- **String:** Tipe data string di Python dapat diintegrasikan dengan berbagai perpustakaan dan alat pemrosesan teks, seperti re untuk ekspresi reguler, NLTK, dan SpaCy untuk NLP.

5. Pengembangan Aplikasi yang Beragam:

- **Number:** Aplikasi yang membutuhkan perhitungan matematis dan analisis data dapat dikembangkan dengan lebih mudah dan efisien.

- **String:** Aplikasi yang membutuhkan pemrosesan dan manipulasi teks, seperti pengolahan dokumen, chatbots, dan aplikasi web, dapat dikembangkan dengan lebih cepat dan efisien.

Program 2A.



```
Latihan 2a String.py - D:\pemula\Python\Latihan 2a String.py (3.12.2)
File Edit Format Run Options Window Help

a = (input("Full String : "))
index = [2, 4, 8, 10]
for i in index :
    print("Menampilkan character pada index ke-",i,"adalah huruf : ", a[i])

IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help

Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\pemula\Python\Latihan 2a String.py
Full String : Hello World
Menampilkan character pada index ke- 2 adalah huruf : l
Menampilkan character pada index ke- 4 adalah huruf : o
Menampilkan character pada index ke- 8 adalah huruf : r
Menampilkan character pada index ke- 10 adalah huruf : d
>>>
```

The screenshot shows a Windows desktop with a taskbar at the bottom. Two windows are open. The left window is a text editor titled 'Latihan 2a String.py' containing Python code. The right window is the 'IDLE Shell 3.12.2' terminal, which shows the execution of the code. The code prompts for a 'Full String' (input: 'Hello World') and then prints characters at indices 2, 4, 8, and 10, resulting in 'l', 'o', 'r', and 'd' respectively.

2B.

The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar includes the Start button, a search bar, and several application icons. The main window is a Python script editor titled "Latihan 2b string.py - D:\pemula\Python\Latihan 2b string.py (3.12.2)". The script contains the following code:

```
a = (input("Full String : "))

print("Menampilkan character pada index ke 1 sampai 3 adalah huruf : ", a[1:3])
print("Menampilkan character pada index ke 1 sampai 3 adalah huruf : ", a[2:5])
print("Menampilkan character pada index ke 1 sampai 3 adalah huruf : ", a[7:11])
```

To the right, the IDLE Shell 3.12.2 window displays the output of the script. It shows the prompt "Full String : Hello World" and the subsequent print statements:

```
= RESTART: D:\pemula\Python\Latihan 2b string.py
Full String : Hello World
Menampilkan character pada index ke 1 sampai 3 adalah huruf : el
Menampilkan character pada index ke 1 sampai 3 adalah huruf : llo
Menampilkan character pada index ke 1 sampai 3 adalah huruf : orld
```

The taskbar at the bottom shows the system clock as 11:07 PM on 7/23/2024, along with other system icons like network, volume, and battery.

2C.

The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar includes the Start button, a search bar, and several application icons. The main window is a Python script editor titled "Latihan 2c string.py - D:\pemula\Python\Latihan 2c string.py (3.12.2)". The script contains the following code:

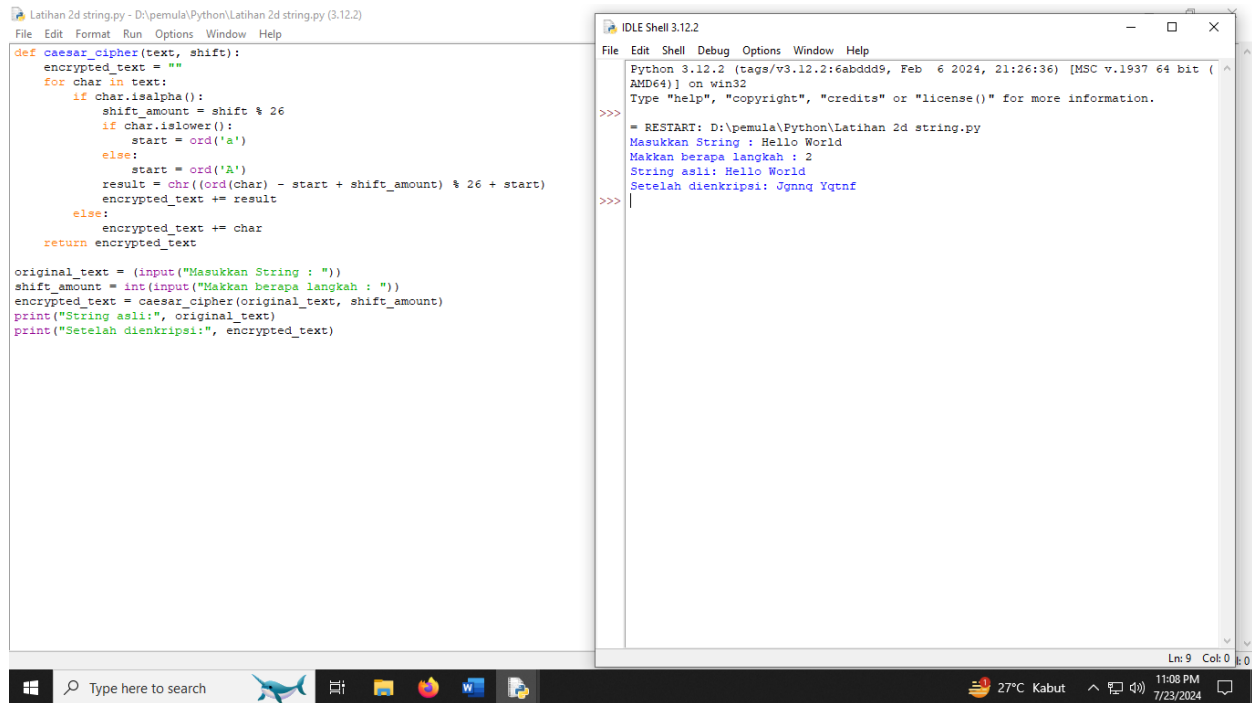
```
a = "Hello, world"
print("Original string:",a)
print("lowercase string:",a.lower())
print("uppercase string:",a.upper())
a = a.replace(a,"orang jahat adalah orang baik yang tersakiti")
print("Original string:",a)
print("menggantikan karakter a,o,e menjadi i : ",a.replace('a','i').replace('o','i').replace('e','i'))
```

To the right, the IDLE Shell 3.12.2 window displays the output of the script. It shows the prompt "Original string: Hello, world" and the subsequent print statements:

```
= RESTART: D:\pemula\Python\Latihan 2c string.py
Original string: Hello, world
lowercase string: hello, world
uppercase string: HELLO, WORLD
Original string: orang jahat adalah orang baik yang tersakiti
menggantikan karakter a,o,e menjadi i : iring jihat idilih iriik yang tirsai kiti
```

The taskbar at the bottom shows the system clock as 11:07 PM on 7/23/2024, along with other system icons like network, volume, and battery.

2D.



The image shows a screenshot of a Windows desktop with two windows open. The left window is a Python script editor titled 'Latihan 2d string.py - D:\pemula\Python\Latihan 2d string.py (3.12.2)'. It contains a Python program that implements a Caesar cipher. The program defines a function 'caesar_cipher(text, shift)' that takes a string and a shift amount, and returns the encrypted string. It then prompts the user for a string and a shift amount, and prints the original and encrypted strings. The right window is an 'IDLE Shell 3.12.2' window showing the execution of the program. It displays the prompt 'Masukkan String :', the user input 'Hello World', the prompt 'Makkan berapa langkah :', the user input '2', and the output 'String asli: Hello World' and 'Setelah dienkripsi: Uggnq Yqcnf'.

```
def caesar_cipher(text, shift):
    encrypted_text = ""
    for char in text:
        if char.isalpha():
            shift_amount = shift % 26
            if char.islower():
                start = ord('a')
            else:
                start = ord('A')
            result = chr((ord(char) - start + shift_amount) % 26 + start)
            encrypted_text += result
        else:
            encrypted_text += char
    return encrypted_text

original_text = (input("Masukkan String : "))
shift_amount = int(input("Makkan berapa langkah : "))
encrypted_text = caesar_cipher(original_text, shift_amount)
print("String asli:", original_text)
print("Setelah dienkripsi:", encrypted_text)
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\pemula\Python\Latihan 2d string.py
Masukkan String : Hello World
Makkan berapa langkah : 2
String asli: Hello World
Setelah dienkripsi: Uggnq Yqcnf
>>>
```

Analisis program

Number dan string adalah dua tipe data dasar yang esensial dalam pemrograman Python. Number digunakan untuk berbagai jenis operasi numerik, sementara string digunakan untuk memanipulasi teks. Pemahaman yang kuat tentang bagaimana cara kerja kedua tipe data ini, serta operasi dan metode yang tersedia, akan sangat membantu dalam menulis kode Python yang efektif dan efisien. Dengan memanfaatkan fitur-fitur ini, programmer dapat menyelesaikan berbagai masalah pemrograman dengan lebih mudah dan cepat.

Referensi

<https://www.w3schools.com/python/>