

# PEMBELAJARAN DASAR PYTHON

## BAB 7 : STRING

---

### 1.1 Tujuan & Manfaat

1. Mahasiswa mengerti konsep string pada bahasa pemrograman python

### 1.2 Ulasan Materi

#### STRING

##### 1. Pengertian String

String (atau str) merupakan kumpulan data char atau karakter yang tersimpan secara urut (text sequence). String di Python mengadopsi standar Unicode dengan default encoding adalah UTF-8.

Python mendesain tipe data string dalam bentuk yang sangat sederhana dan mudah digunakan. Untuk membuat string cukup tulis saja text yang diinginkan dengan diapit tanda petik satu atau petik dua.

Contoh:

```
print("hello python")  
print('hello python')
```

Output :

```
hello python
```

##### 2. Penggunaan String pada Python

###### 1. Multiline string

Untuk string multiline atau lebih dari satu baris, cara penulisannya bisa dengan:

- a. Menggunakan karakter spesial \n

Contoh:

```
text = "a multiline string\nin python"  
print(text)
```

Output :

```
a multiline string
in python
```

b. menggunakan tanda `"""` atau `'` untuk mengapit text

Contoh:

```
text = """a multiline string
in python"""
print(text)
```

Output :

```
a multiline string
in python
```

## 2. Escape character

Python mengenal escape character umum yang ada di banyak bahasa pemrograman, contohnya seperti `\` digunakan untuk menuliskan karakter `"` (pada string yang dibuat menggunakan literal `" "`). Penambahan karakter `\` adalah penting agar karakter `"` terdeteksi sebagai penanda string.

## 3. String Length

Untuk mendapatkan string length, gunakan fungsi `len()`. Fungsi `len()` mengembalikan panjang string

Contoh:

```
a = "Hello, Python!"
print(len(a))
```

Output :

```
14
```

## 4. Slicing String\

Anda dapat mengembalikan rentang karakter dengan menggunakan sintaks slice. Tentukan indeks awal dan indeks akhir, dipisahkan oleh titik dua, untuk mengembalikan bagian dari string.

Contoh:

```
a = "Hello, Python!"  
print(a[2:5])
```

Output :

```
llo
```

## 5. Upper Case

Metode upper() mengembalikan string dalam huruf besar.

Contoh:

```
a = "Hello, Python!"  
print(a.upper())
```

Output :

```
HELLO, PYTHON!
```

## 6. Lower Case

Metode lower() mengembalikan string dalam huruf kecil.

Contoh:

```
a = "Hello, Python!"  
print(a.lower())
```

Output :

```
hello, python!
```

## 7. Replace String

Metode replace() menggantikan string dengan string lain.

Contoh:

```
a = "Hello, Python!"  
print(a.replace("T", "S"))
```

Output :

```
Hello, Pyshon!
```

## 8. Penggabungan String (String Concatenation)

Untuk menggabungkan, atau menggabungkan, dua string dapat menggunakan operator +. Gabungkan variabel a dengan variabel b menjadi variabel c

Ada beberapa metode yang bisa digunakan untuk string concatenation atau operasi penggabungan string.

- a. Menggunakan teknik penulisan string literal sebaris.

Caranya dengan langsung tulis saja semua string-nya menggunakan separator karakter spasi.

Contoh:

```
text = "Hello" "Python"
print(text)
```

Output :

```
Hello Python
```

- b. Menggunakan operator +

Operator + jika diterapkan pada string menghasilkan penggabungan string.

Contoh:

```
a = "Hello"
b = "Python"
text = a + " " + b
print(text)
```

Output :

```
Hello Python
```

- c. Untuk data non-string jika ingin digabung harus dibungkus dengan fungsi str() terlebih dahulu. Fungsi str() digunakan untuk mengkonversi segala jenis data ke bentuk string.

Contoh:

```
text = "Hello"  
number = 123  
yes = True  
message = text + " " + str(number) + " " + str(yes)  
print(message)
```

Output :

```
Hello 123 True
```

d. Menggunakan method join() milik string.

Pada penerapannya, karakter pembatas atau separator ditulis terlebih dahulu, kemudian di-chain dengan method join dengan isi argument adalah list yang ingin digabung.

Contoh:

```
text = " ".join(["Hello", "Python"])  
print(text)
```

Output :

```
Hello Python
```

## 9. Split string

Split string adalah operasi pemisahan karakter string menjadi list dengan element berisi string. Operasi ini merupakan kebalikan dari penggabungan string.

Cara penggunaannya adalah dengan mengakses method split() langsung dari data string. Pemanggilan method tersebut perlu disertai argument berisi separator atau karakter pemisah string.

Contoh:

```
text = "Hello Python 12345"  
res = text.split(" ")  
print(res)
```

Output :

```
['Hello', ' Python', '12345']
```

## 10. Format String

Seperti yang dipelajari pada bab Variabel Python, kita tidak dapat menggabungkan string dan angka. Tetapi kita dapat menggabungkan string dan angka dengan menggunakan method format()!. Method format() mengambil argumen yang diteruskan, memformatnya, dan menempatkannya dalam string tempat placeholder {} berada. Gunakan metode format() untuk memasukkan angka ke dalam string

Contoh:

```
age = 25
txt = "Namaku Budi, dan umurku {}"
print(txt.format(age))
```

Output :

```
Namaku Budi, dan umurku 25
```

## 1.3 Praktikum

**Contoh : Mengubah string ke dalam huruf besar menggunakan method upper()**

Pada contoh ini kita akan membuat variabel dengan nama a dan bernilai string "Hello, World!". Selanjutnya kita akan mengubah nilai variabel a menjadi huruf besar menggunakan method upper(). Berikut ini adalah langkah-langkahnya :

1. Tuliskan kode yang mendefinisikan variabel dengan nama a dengan nilai string "Hello, World!"

```
a = "Hello World!"
```

2. Buat fungsi dengan nama kapital yang memiliki parameter variable a

```
def kapital(a):
```

3. Kemudian tambahkan perintah return dan perintah upper() agar isi dari variabel a menjadi kapital

```
return a.upper()
```

4. Tampilkan hasil upper menggunakan perintah `print()` dan panggil fungsi kapital yang memiliki parameter variabel `a`

```
print(kapital(a))
```

#### 1.4 Percobaan 1

Sekarang kerjakan percobaan 1 dengan mengikuti langkah-langkah dibawah ini !

1. Buatlah variabel tulis yang memiliki nilai “Belajar Python”
2. Buat fungsi kapital yang memiliki parameter variabel tulis
3. Berikan sintaks untuk mengubah nilai variabel tulis sehingga menjadi huruf kapital menggunakan method `upper()`
4. Tampilkan hasil perubahan nilai dari variabel tulis menggunakan fungsi `print()`
5. Jalankan kode program percobaan 2 diatas dengan menekan tombol Check Code Validity

#### 1.5 Percobaan 2

Sekarang kerjakan percobaan 2 dengan mengikuti langkah-langkah dibawah ini !

1. Buatlah variabel tulis yang memiliki nilai “Belajar Python”
2. Buat fungsi kecil yang memiliki parameter variabel tulis
3. Berikan sintaks untuk mengubah nilai variabel tulis sehingga menjadi huruf kecil menggunakan method `lower()`
4. Tampilkan hasil perubahan nilai dari variabel tulis menggunakan fungsi `print()`
5. Jalankan kode program percobaan 3 diatas dengan menekan tombol Check Code Validity