



TIPE DATA TEKS

Dasar-Dasar Pemrograman

Semester Gasal 2020/2021

Tujuan

- Mengenal tipe data string dan operasi-operasi yang dapat dikenakan terhadapnya
- Memahami cara mengakses bagian string menggunakan indexing dan slicing
- Mengenal fungsi-fungsi bawaan untuk tipe data string

String

Tipe data String

- Membuat variabel bertipe String

```
r = 'Ini sebuah string'  
s = "Ini juga sebuah string"  
t = """Ini sebuah string panjang.  
Bisa ditulis lebih dari satu baris."""  
u = input("Masukkan sebuah teks: ")
```

- String kosong

```
x = ''
```

- Panjang string

```
len("Menghitung panjang string")
```

Konkatenasi dan Repetisi

- Konkatenasi (operator `+`) → menggabung suatu string dengan string lain
- Repetisi (operator `*`) → mengulang string sebanyak jumlah tertentu

Ekspresi	Hasil
<code>'AB' + 'cd'</code>	<code>'Abcd'</code>
<code>'A' + '7' + 'b'</code>	<code>'A7b'</code>
<code>'A' + str(7) + 'b'</code>	<code>'A7b'</code>
<code>'Hello ' * 3</code>	<code>'Hello Hello Hello '</code>

- String bersifat *immutable*, artinya setelah diciptakan, suatu string tidak dapat dimodifikasi
 - Modifikasi suatu string pada dasarnya adalah membuat string baru dan menyimpannya ke dalam variabel asal

Operator in pada string

- Operator `in` digunakan untuk memeriksa apakah suatu string mengandung sesuatu
- Operator `in` mengembalikan `True` atau `False`

```
'i' in 'aiueo' # menghasilkan True  
't' in 'aiueo' # menghasilkan False
```

```
s = 'Assalamualaikum'  
if 'salam' in s:  
    print("String mengandung salam")  
if 'x' not in s:  
    print("String tidak mengandung x")
```

Operator in pada string (lanjutan)

- Operator `in` dapat digunakan untuk melakukan iterasi setiap karakter pada string

```
s = 'Selamat datang'  
for c in s:  
    print(c)
```

S
e
l
a
m
a
t

d
a
t
a
n
g

Escape character

- Penulisan karakter khusus pada string dapat memanfaatkan karakter *backslash* ('\\')

Escape character	Deskripsi	Contoh	Hasil
\n	Menyisipkan baris baru (<i>newline</i>)	s = 'pertama\nkedua'	pertama kedua
\' atau \"	Menyisipkan apostrof	t = 'Assalamu\'alaikum'	Assalamu'alaikum
\\	Menyisipkan <i>backslash</i>	u = 'D:\\program.py'	D:\program.py
\t	Menyisipkan tab	v = 'pertama\t\tkedua'	pertama kedua

String Indexing dan Slicing

String Indexing

- Mengakses karakter pada string s di posisi x

`s[x]`

s	h	e	l	l	o
Index	0	1	2	3	4
s[0]	h	e	l	l	o
s[1]	h	e	l	l	o

- Backward indexing* → menggunakan index bilangan negatif

Negative Index	-5	-4	-3	-2	-1
s	h	e	l	l	o
Index	0	1	2	3	4
s[-1]	h	e	l	l	o
s[-2]	h	e	l	l	o

String Slicing

- *Slicing* → mengakses substring pada suatu string
- Tanpa selang: `s[idx_awal:idx_akhir]`

```
s = 'hello'
print(s[1:4])
print(s[1:-1])
print(s[:4])
print(s[1:])
```

- Dengan selang: `s[idx_awal:idx_akhir:selang]`

```
s = 'hello'
print(s[1:4:2])
print(s[:4:3])
print(s[1::2])
print(s[::-1])
```

String methods

Fungsi-fungsi pada string

- String pada Python dilengkapi sejumlah fungsi/method untuk memudahkan pengolahannya

Method	Deskripsi
<code>lower()</code>	Mengembalikan string yang berisi huruf kecil dari string awal
<code>upper()</code>	Mengembalikan string yang berisi huruf kapital dari string awal
<code>replace(x,y)</code>	Mengembalikan string di mana kemunculan x diubah menjadi y
<code>count(x)</code>	Mengembalikan jumlah kemunculan x pada string
<code>startswith(x)</code>	Mengembalikan True jika string diawali dengan x, False jika tidak
<code>endswiths(x)</code>	Mengembalikan True jika string diakhiri dengan x, False jika tidak

Fungsi-fungsi pada string (lanjutan)

- Contoh pemanggilan fungsi pada string

```
s = 'Hello'
print(s.lower())           # mencetak 'hello'
print(s.upper())          # mencetak 'HELLO'
print(s.replace('o', 'yyy')) # mencetak 'Hellyyy'
print(s.count('el'))       # mencetak 1
print(s.startswith('h'))   # mencetak False
print(s.endswith('o'))     # mencetak True
```

Selamat Belajar!!

