

Pemrograman Python

pertemuan 2

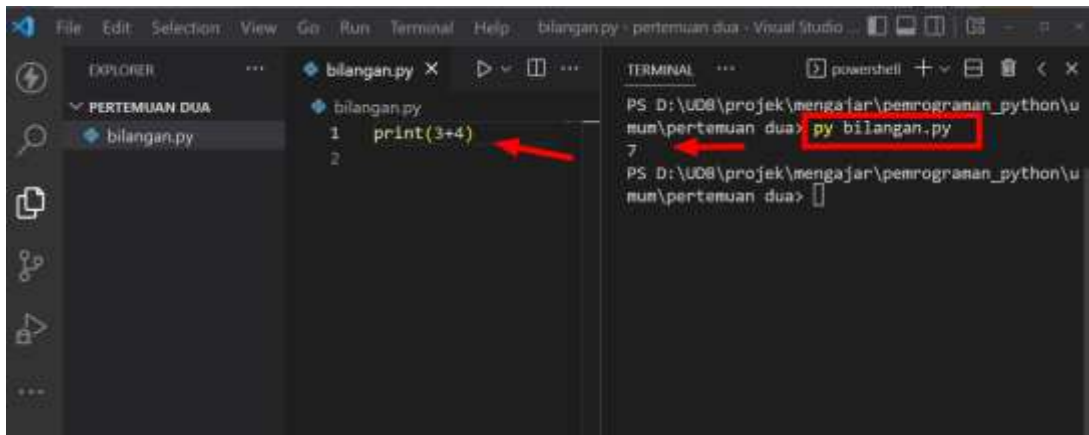
Tipe data

- Integer / bilangan /numerik
- String

Integer / Bilangan / Numerik

Angka	Type	Angka	Type
17	Int	77.77	Float
17.	float	-77	int

- | Operator | Deskripsi | Contoh |
|----------|--------------------------------|--------------------|
| + | Penjumlahan | 3 + 4 bernilai 7 |
| - | Pengurangan | 8 - 1 bernilai 7 |
| * | Perkalian | 1 * 7 bernilai 7 |
| / | Pembagian | 7 / 1 bernilai 7 |
| // | Pembagian (dibulatkan kebawah) | 15 // 2 bernilai 7 |
| % | Sisa Bagi / Modulo | 13 % 5 bernilai 3 |



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a file named 'bilangan.py'. The main editor area displays the code: `1 print(3+4)`. A red arrow points to the code. On the right, the Terminal pane shows the command prompt running `python3 bilangan.py`, which outputs `7`. A red box highlights the command in the terminal, and a red arrow points to the output.

Operator Aritmatika

Operator	Contoh	Penjelasan
Penjumlahan <code>+</code>	<code>1 + 3 = 4</code>	Menjumlahkan nilai dari masing-masing operan atau bilangan
Pengurangan <code>-</code>	<code>4 - 1 = 3</code>	Mengurangi nilai operan di sebelah kiri menggunakan operan di sebelah kanan
Perkalian <code>*</code>	<code>2 * 4 = 8</code>	Mengalikan operan/bilangan
Pembagian <code>/</code>	<code>10 / 5 = 2</code>	Untuk membagi operan di sebelah kiri menggunakan operan di sebelah kanan
Sisa Bagi <code>%</code>	<code>11 % 2 = 1</code>	Mendapatkan sisa pembagian dari operan di sebelah kiri operator ketika dibagi oleh operan di sebelah kanan
Pangkat <code>**</code>	<code>8 ** 2 = 64</code>	Memangkatkan operan disebelah kiri operator dengan operan di sebelah kanan operator
Pembagian Bulat <code>//</code>	<code>10 // 3 = 3</code>	Sama seperti pembagian. Hanya saja angka dibelakang koma dihilangkan

operator abs, int , round

- **abs()** untuk membuat nilai absolut atau positif
- **int()** untuk membuat nilai int
- **round()** untuk membulatkan nilai
 - round(7.7177, 2) koma menunjukkan berapa angka di belakang koma

input output

abs(7)	7
abs(0)	0
abs(-7)	7
int(7.7)	7
int(7)	7

int(-7.7)	-7
round(7.7)	8
round(7.7177, 2)	7.72
round(7.7177, 1)	7.7

operator penugasan / assignment

Tabel 1.2. Operator Assignment

Operator	Deskripsi	Contoh
=	Assignment	N = 7
+=	Penjumlahan	N += 7, N akan ditambah 7.
-=	Pengurangan	N -= 7, N akan dikurangi 7.
*=	Perkalian	N *= 7, N akan dikali 7.
//=	Pembagian (dibulatkan kebawah)	N //= 7, N akan dibagi 7 (dibulatkan kebawah)
%=	Sisa Bagi / Modulo	N %= 7, N akan dimodulo 7.
=	Assignment	N = 7

operator penugasan / assignment

```
operator_penugasan.py > ...
1  angkaPertama = 4
2  angkaPertama += 1
3
4  print(angkaPertama)
5
```

TERMINAL ... powershell + v [] [] ^ x

PS D:\UDB\projek\mengajar\pemrograman_python\umum\pertemuan dua> py operator_penugasan.py

5

PS D:\UDB\projek\mengajar\pemrograman_python\umum\pertemuan dua> []

```
1  angkaPertama = 4
2  angkaPertama -= 1
3
4  print(angkaPertama)
5
```

TERMINAL ... powershell + v [] [] ^ x

PS D:\UDB\projek\mengajar\pemrograman_python\umum\pertemuan dua> py operator_penugasan.py

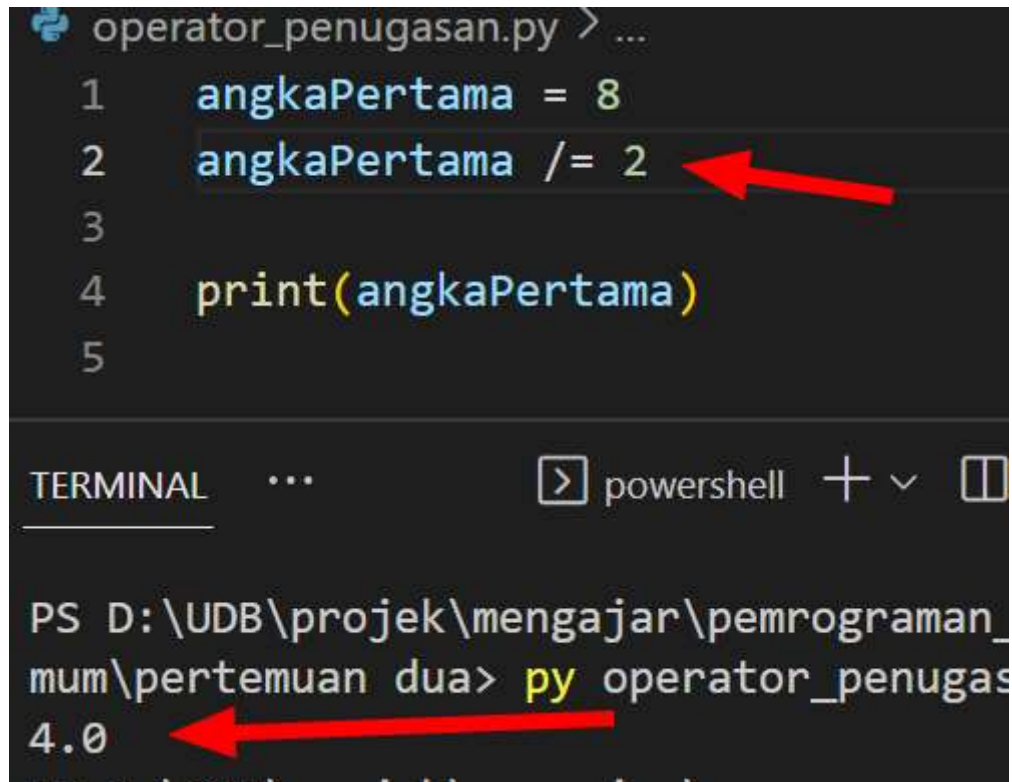
3

operator penugasan / assignment

```
operator_penugasan.py > ...
1  angkaPertama = 8
2  angkaPertama /= 2
3
4  print(angkaPertama)
5
```

TERMINAL ... powershell + v

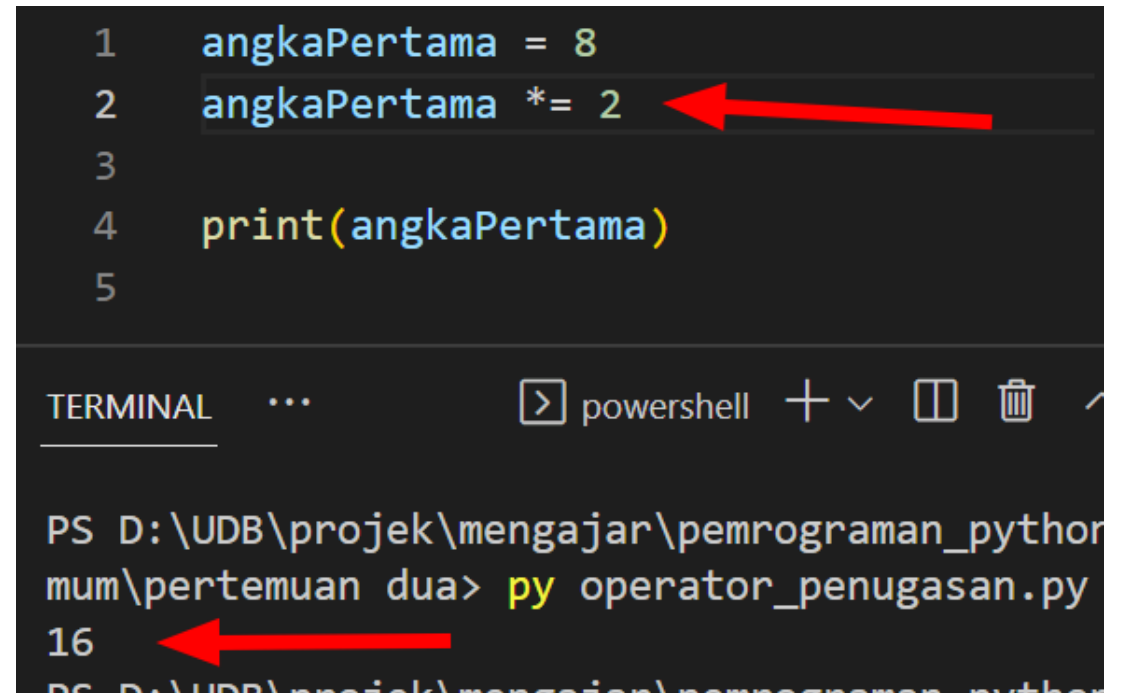
```
PS D:\UDB\projek\mengajar\pemrograman_pythor
mum\pertemuan dua> py operator_penugas
4.0
```



```
1  angkaPertama = 8
2  angkaPertama *= 2
3
4  print(angkaPertama)
5
```

TERMINAL ... powershell + v

```
PS D:\UDB\projek\mengajar\pemrograman_pythor
mum\pertemuan dua> py operator_penugas
16
```



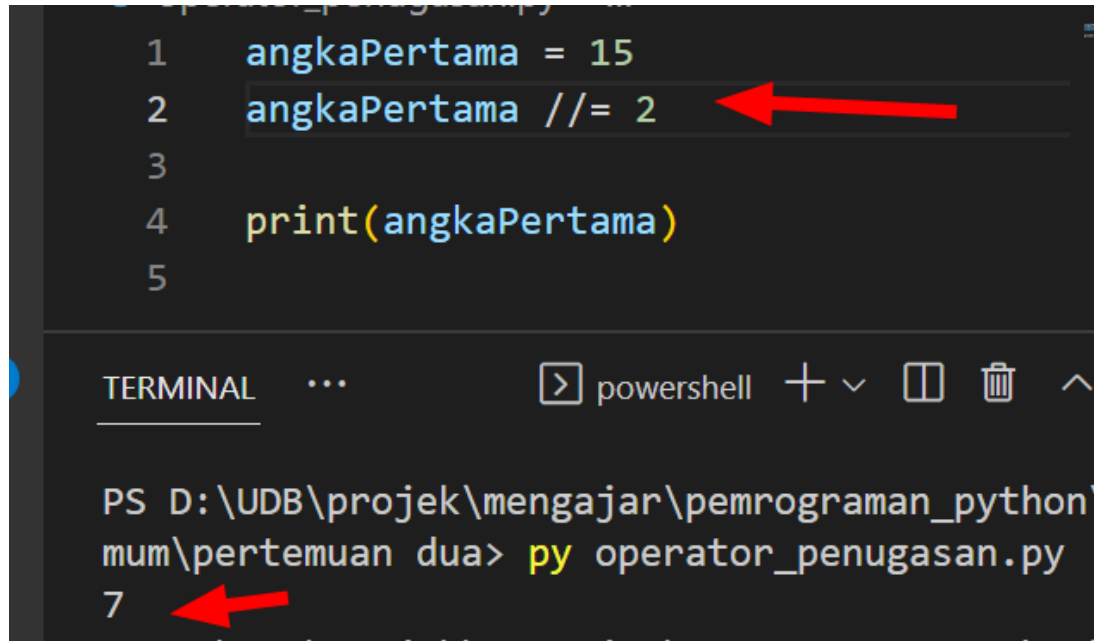
operator penugasan / assignment

```
1 angkaPertama = 15
2 angkaPertama //= 2
3
4 print(angkaPertama)
5
```

TERMINAL ... powershell + v [] [] ^

PS D:\UDB\projek\mengajar\pemrograman_python\mum\pertemuan dua> py operator_penugasan.py

7



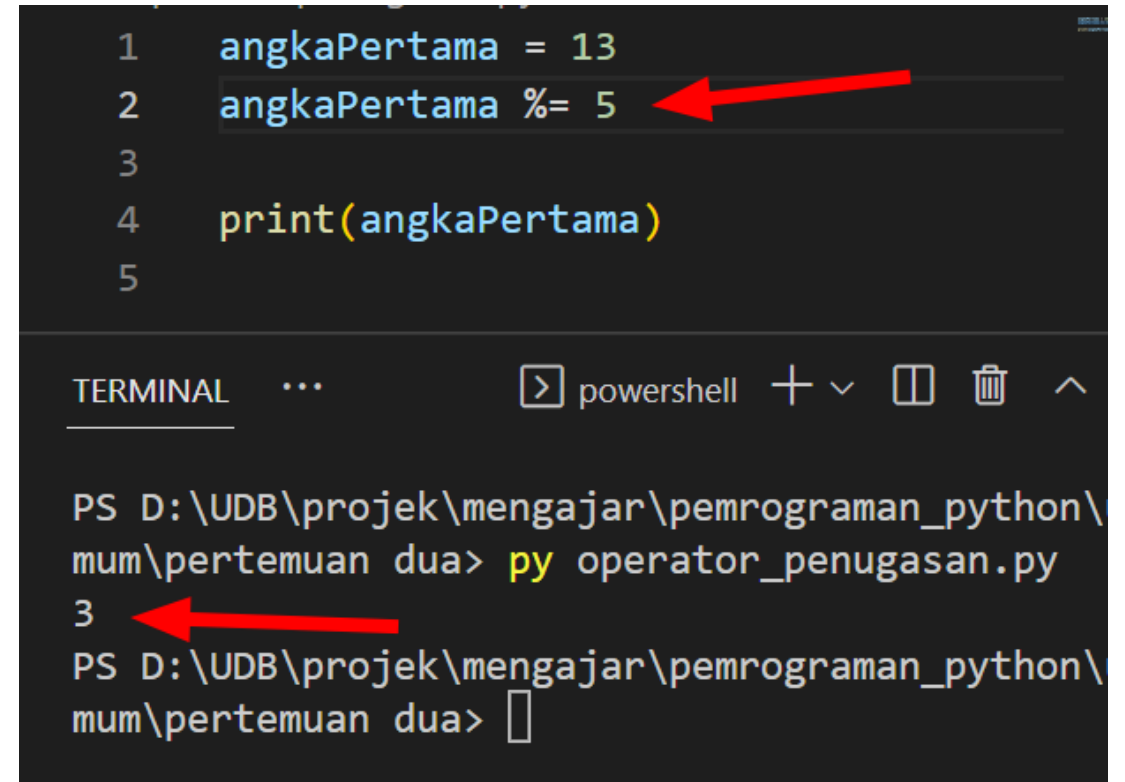
```
1 angkaPertama = 13
2 angkaPertama %= 5
3
4 print(angkaPertama)
5
```

TERMINAL ... powershell + v [] [] ^

PS D:\UDB\projek\mengajar\pemrograman_python\mum\pertemuan dua> py operator_penugasan.py

3

PS D:\UDB\projek\mengajar\pemrograman_python\mum\pertemuan dua> []



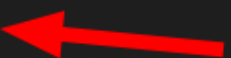
String

untuk mengetahui panjang karakter gunakan fungsi **len()**

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU)
3
```

TERMINAL ... powershell + v [] [] ^



```
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> py string.py
duta bangsa
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> [ ]
```



```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(len(kampusKU))
3
```

TERMINAL ... powershell + v [] [] ^ >

```
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> py string.py
11
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> [ ]
```



index dan irisan

- index di mulai dari 0

d	u	t	a		b	a	n	g	s	a
0	1	2	3	4	5	6	7	8	9	10

```
string.py >
1  kampusKU = 'duta bangsa'
2  print(kampusKU[0])
3
```

menampilkan index ke 0

TERMINAL ... powershell + v

```
PS D:\UDB\projek\mengajar\pemrograman_mum\pertemuan dua> py string.py
d
PS D:\UDB\projek\mengajar\pemrograman_mum\pertemuan dua>
```

mencetak index ke 7

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU[7])
3
```

mencetak index ke 7

TERMINAL ... powershell + v

```
PS D:\UDB\projek\mengajar\pemrograman_pythor_mum\pertemuan dua> py string.py
n
PS D:\UDB\projek\mengajar\pemrograman_pythor_mum\pertemuan dua>
```

index negatif

```
string.py M X
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU[-3])
3


TERMINAL ... powershell + v [ ] [ ] ^
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> py string.py
g
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> [ ]
```

d	u	t	a		b	a	n	g	s	a
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

pada irisan, index terakhir tidak di ambil


mengambil irisan antara index 3 sampai 7

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU[3:7])
3
```



TERMINAL ... powershell + v [] ^

```
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> py string.py
a ba
```

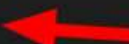


```
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua>
```

karakter yang diambil adalah index 3 sampai index 6


mengambil irisan antara index 6 sampai 10

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU[6:10])
3
```



TERMINAL ... powershell + v []

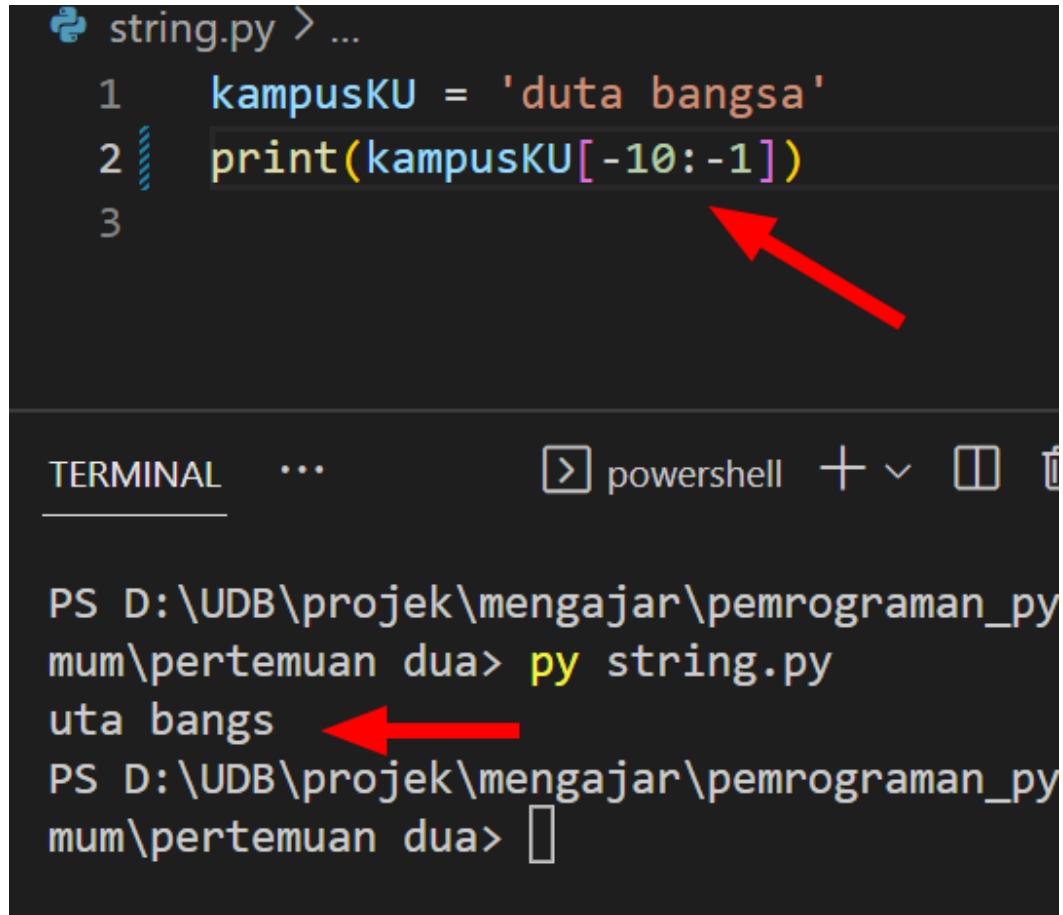
```
PS D:\UDB\projek\mengajar\pemrograman_py
mum\pertemuan dua> py string.py
angs
```



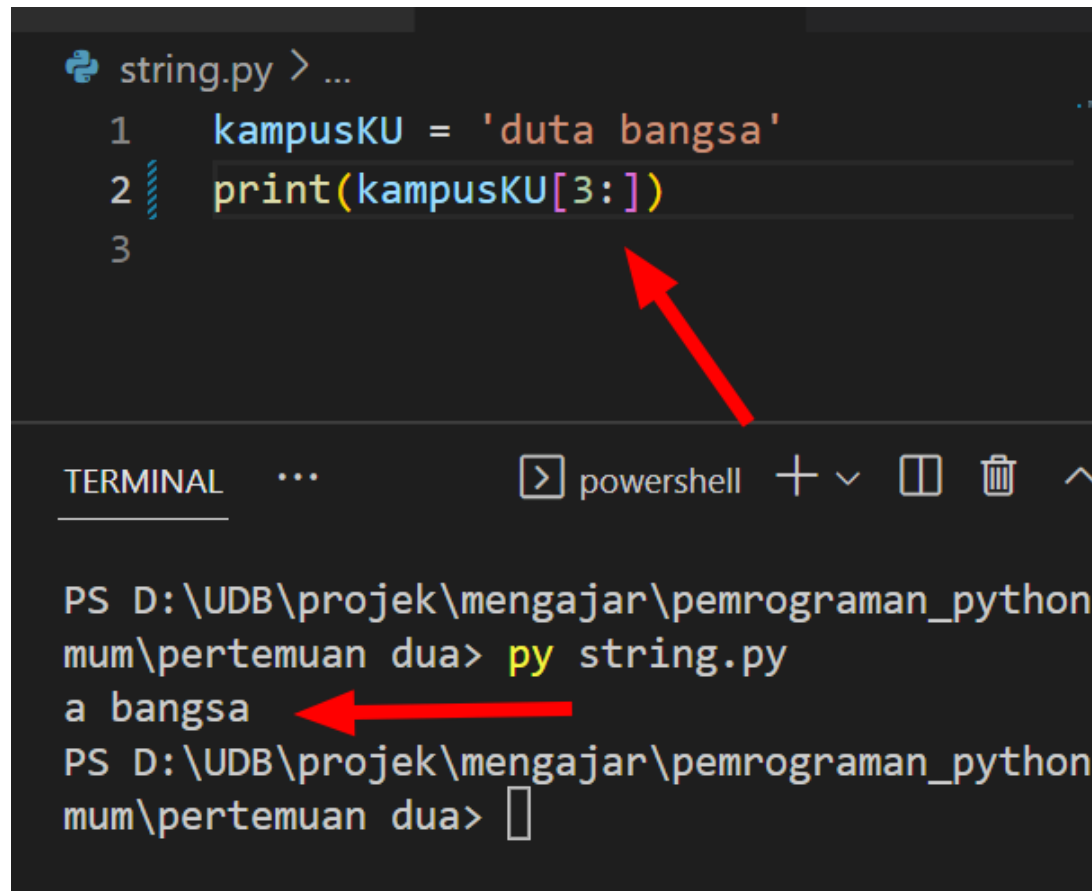
```
PS D:\UDB\projek\mengajar\pemrograman_py
mum\pertemuan dua>
```

irisan dengan negatif

```
string.py > ...  
1   kampusKU = 'duta bangsa'  
2   print(kampusKU[-10:-1])  
3  
  
TERMINAL    ...    powershell + v [ ] [ ]  
  
PS D:\UDB\projek\mengajar\pemrograman_py  
mum\pertemuan dua> py string.py  
uta bangs  
PS D:\UDB\projek\mengajar\pemrograman_py  
mum\pertemuan dua> [ ]
```



batas default dalam irisan

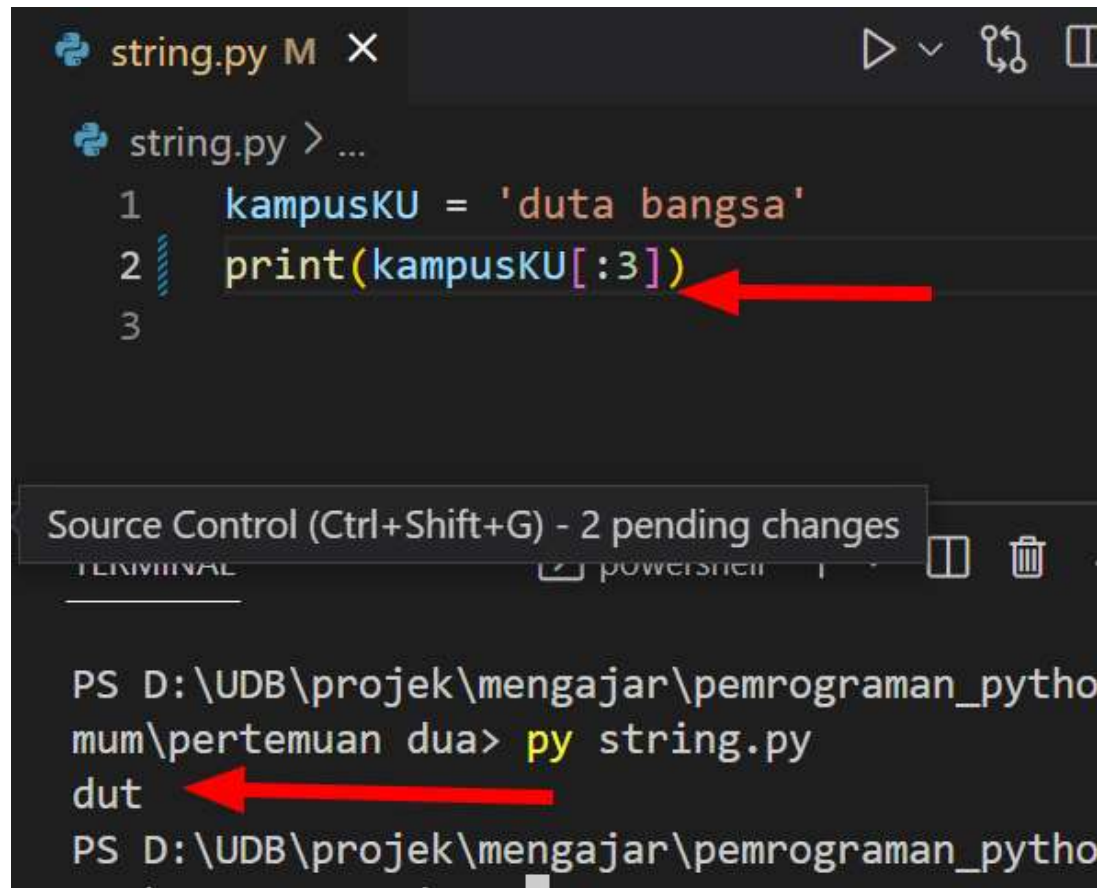


The image shows a code editor window with a file named `string.py`. The code contains two lines: `kampusKU = 'duta bangsa'` on line 1 and `print(kampusKU[3:])` on line 2. A red arrow points from the `3:` in the slice notation to the terminal output. The terminal window below shows the command `py string.py` being executed, resulting in the output `a bangsa`. Another red arrow points from the output `a bangsa` back to the slice notation in the code editor.

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU[3:])
3

TERMINAL  ...  powershell + v [ ] [ ] ^
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> py string.py
a bangsa
PS D:\UDB\projek\mengajar\pemrograman_python\
mum\pertemuan dua> [ ]
```

- `kampusKu[3:]`
- artinya yang di ambil adalah data mulai index ke 3 sampai terakhir



The image shows a code editor window with a file named `string.py`. The code inside is:

```
1 kampusKU = 'duta bangsa'
2 print(kampusKU[:3])
3
```

A red arrow points to the `[:3]` slice in line 2. Below the code editor, there is a terminal window showing the command `py string.py` being executed. The output of the command is `dut`, with a red arrow pointing to it. A source control notification bar is visible above the terminal, indicating 2 pending changes.

- `kampusKu[:3]`
- artinya yang di ambil adalah data mulai index pertama sampai index 2

mencari string

```
string.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU.find('a'))
3

TERMINAL  ...  powershell  + v  [ ]  [ ]  ^  x

PS D:\UDB\projek\mengajar\pemrograman_python\umum\pertemuan dua> py string.py
3
PS D:\UDB\projek\mengajar\pemrograman_python\umum\pertemuan dua> [ ]
```

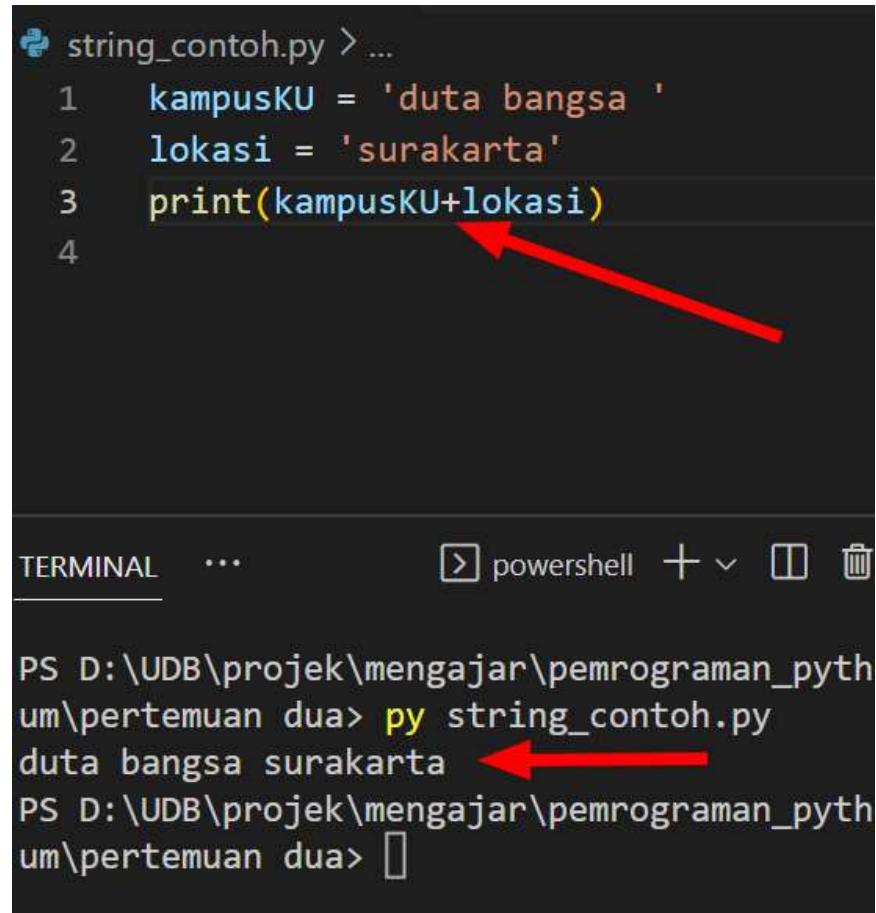
- yang dicetak adalah index terkecil

penggabungan dan pengulangan string

```
string_contoh.py > ...
1  kampusKU = 'duta bangsa '
2  lokasi = 'surakarta'
3  print(kampusKU+lokasi)
4
```

TERMINAL ... powershell + v [] []

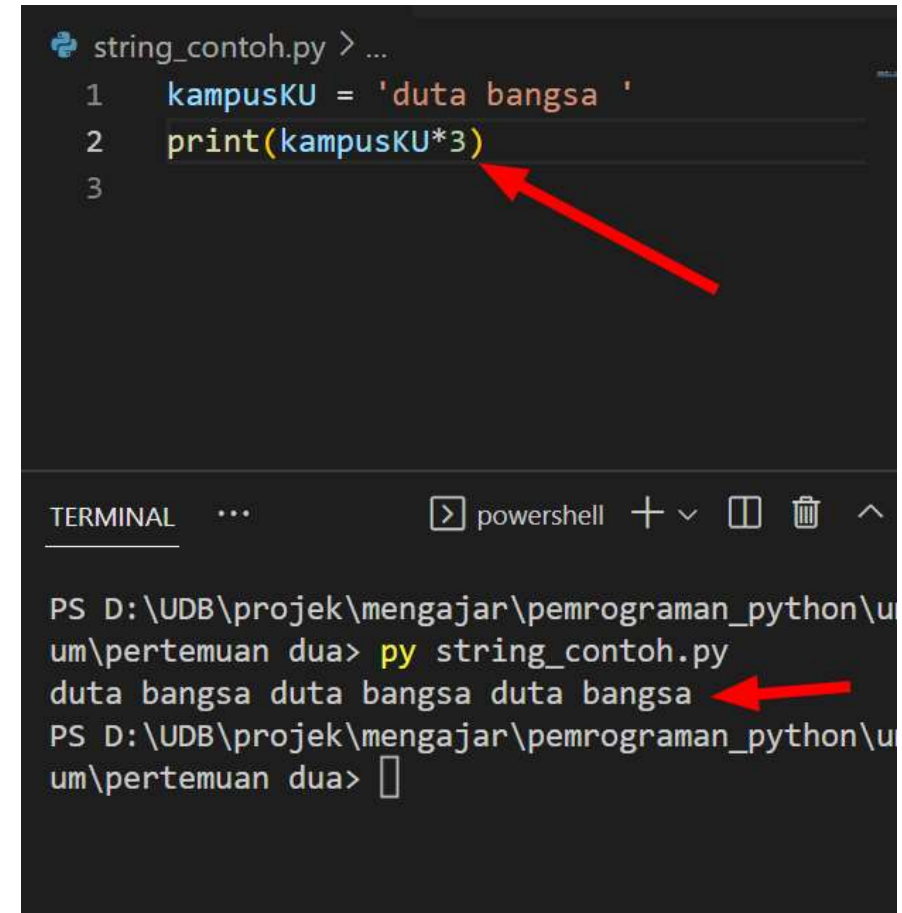
```
PS D:\UDB\projek\mengajar\pemrograman_python\um\pertemuan dua> py string_contoh.py
duta bangsa surakarta
PS D:\UDB\projek\mengajar\pemrograman_python\um\pertemuan dua> [ ]
```



```
string_contoh.py > ...
1  kampusKU = 'duta bangsa '
2  print(kampusKU*3)
3
```

TERMINAL ... powershell + v [] []

```
PS D:\UDB\projek\mengajar\pemrograman_python\um\pertemuan dua> py string_contoh.py
duta bangsa duta bangsa duta bangsa
PS D:\UDB\projek\mengajar\pemrograman_python\um\pertemuan dua> [ ]
```



operasi string

Tabel 3.1. String operations (str1 = "Python").

Fungsi	Contoh	Output	Deskripsi
Len	len(str1)	6	Jumlah karakter dalam string
Upper	str1.upper()	"PYTHON"	Huruf besar setiap karakter alphabet
Lower	str1.lower()	"python"	Huruf kecil setiap karakter alphabet
Count	str1.count('h')	1	Jumlah kemunculan substring yang tidak tumpang tindih
capitalize	"coDE".capitalize()	"Code"	Mengkapitalisasi huruf pertama dari string dan menurunkan sisanya
Title	"staTistika bisnis".title()	"Statistika Bisnis"	Menggunakan huruf kapital dari huruf pertama dari setiap kata dalam string dan menurunkan sisanya
Tstrip	"ab ".rstrip()	"ab"	Menghilangkan spasi dari sisi kanan string

```
1 kampusKU = 'duta bangsa'  
2 print(len(kampusKU))  
3
```

TERMINAL

powershell

```
PS D:\UDB\projek\mengajar\pemrograman_py  
um\pertemuan dua> py string_contoh.py  
11  
PS D:\UDB\projek\mengajar\pemrograman_py  
um\pertemuan dua>
```

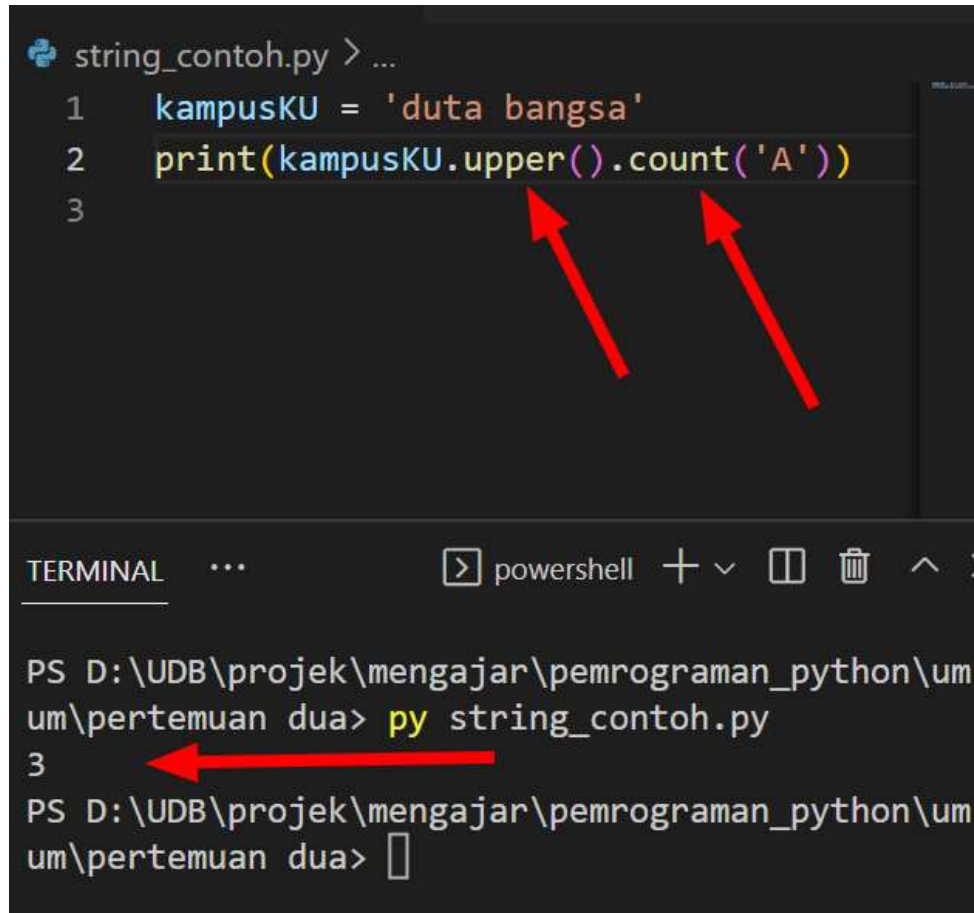
```
1 kampusKU = 'duta bangsa'  
2 print(kampusKU.upper())  
3
```

TERMINAL

powershell

```
PS D:\UDB\projek\mengajar\pemrograman_pytho  
um\pertemuan dua> py string_contoh.py  
DUTA BANGSA  
PS D:\UDB\projek\mengajar\pemrograman_pytho  
um\pertemuan dua>
```

metode berantai/ methode chaining



The image shows a code editor window with a file named `string_contoh.py`. The code contains two lines: `kampusKU = 'duta bangsa'` and `print(kampusKU.upper().count('A'))`. Two red arrows point from the `upper()` and `count('A')` methods to the terminal output. The terminal shows the command `py string_contoh.py` being executed, and the output is `3`. A red arrow points from the output `3` back to the `count('A')` method in the code.

```
string_contoh.py > ...
1  kampusKU = 'duta bangsa'
2  print(kampusKU.upper().count('A'))
3

TERMINAL  ...  powershell + v [ ] [ ] ^
PS D:\UDB\projek\mengajar\pemrograman_python\um
um\pertemuan dua> py string_contoh.py
3
PS D:\UDB\projek\mengajar\pemrograman_python\um
um\pertemuan dua> [ ]
```

- kampusKu diubah ke Uppercase lalu dilakukan penghitungan pada huruf kapital A

latihan praktek

1. buat program yang meminta nama klub bola atau atlit, jumlah pertandingan yang dimenangkan dan jumlah pertandingan yang kalah, lalu tampilkan persentase kemenangan
2. buat program
 - buat variabel namaDepan = "Thomas"
 - buat variabel namaTengah = "Alva"
 - buat variabel namaBelakang = "Edison"
 - buat variabel tahunLahir = 1847
 - tampilkan kalimat " tahun kelahiran " diikuti nama lengkap penemu, diikuti oleh "adalah" dan tahun lahir

List dan Tuple