

Pertemuan 6

List & Tuple

List

Python menyediakan sejumlah tipe data yang dikenal dengan tipe data berurut (sequence).

List adalah tipe data yang berisi satu atau beberapa nilai di dalamnya. Nilai – nilai ini sering juga disebut item, elemen, atau anggota list. List dibuat dengan menempatkan semua item di dalam tanda kurung [], dipisahkan oleh tanda koma. Anggota list bisa berisi satu tipe data, atau campuran.

```
# list kosong

my_list = []

# list berisi integer  my_list = [1,2,3,4,5]
# list berisi tipe campuran  my_list = [1, 3.5, "Hello"]
```

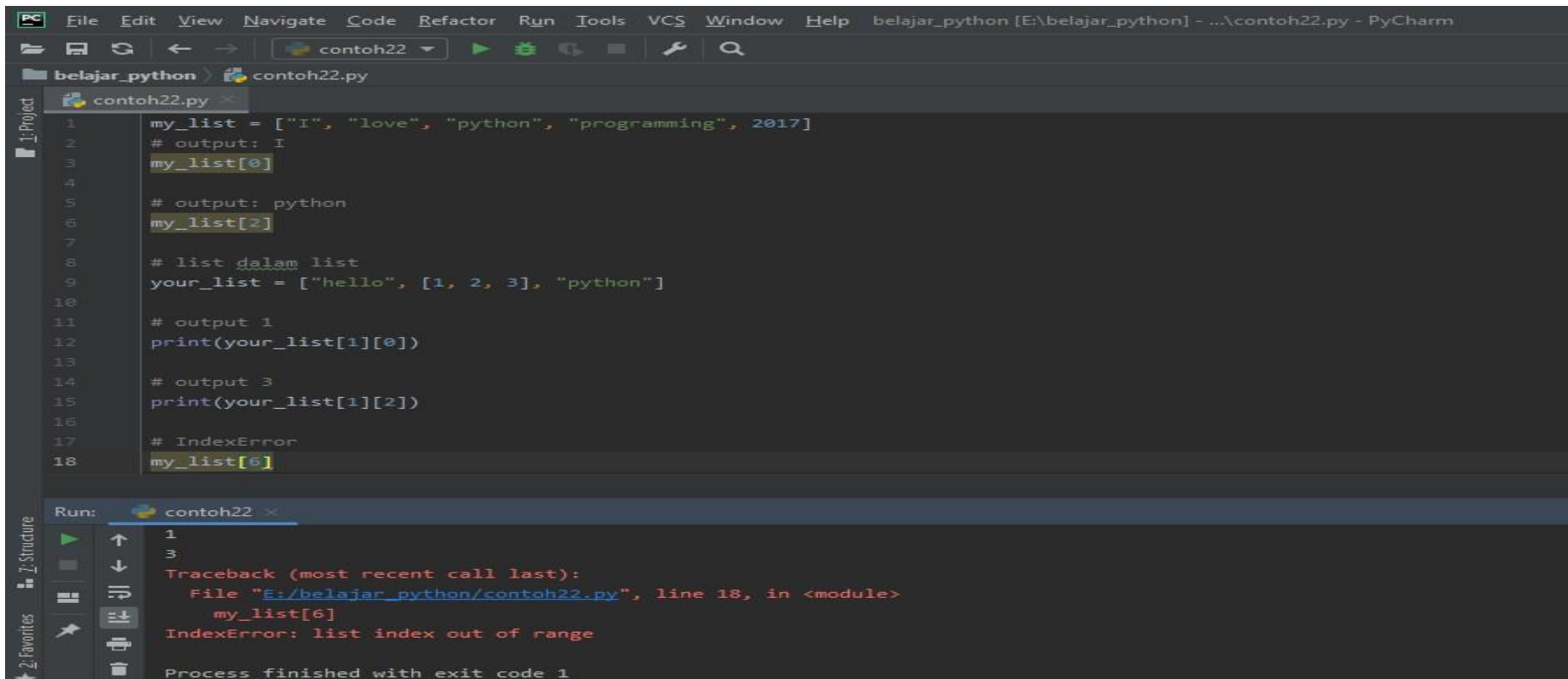
List juga bisa berisi list lain. Ini disebut list bersarang

```
# list bersarang

my_list = ["hello", [2,4,6], ['a','b']]
```

Mengkases Anggota List

Kita bisa mengakses anggota list dengan menggunakan indeksnya dengan format `namalist[indeks]`. Indeks list dimulai dari 0. List yang memiliki 5 anggota akan memiliki indeks mulai dari 0 s/d 4. Mencoba mengakses anggota list di luar itu akan menyebabkan error `IndexError`.



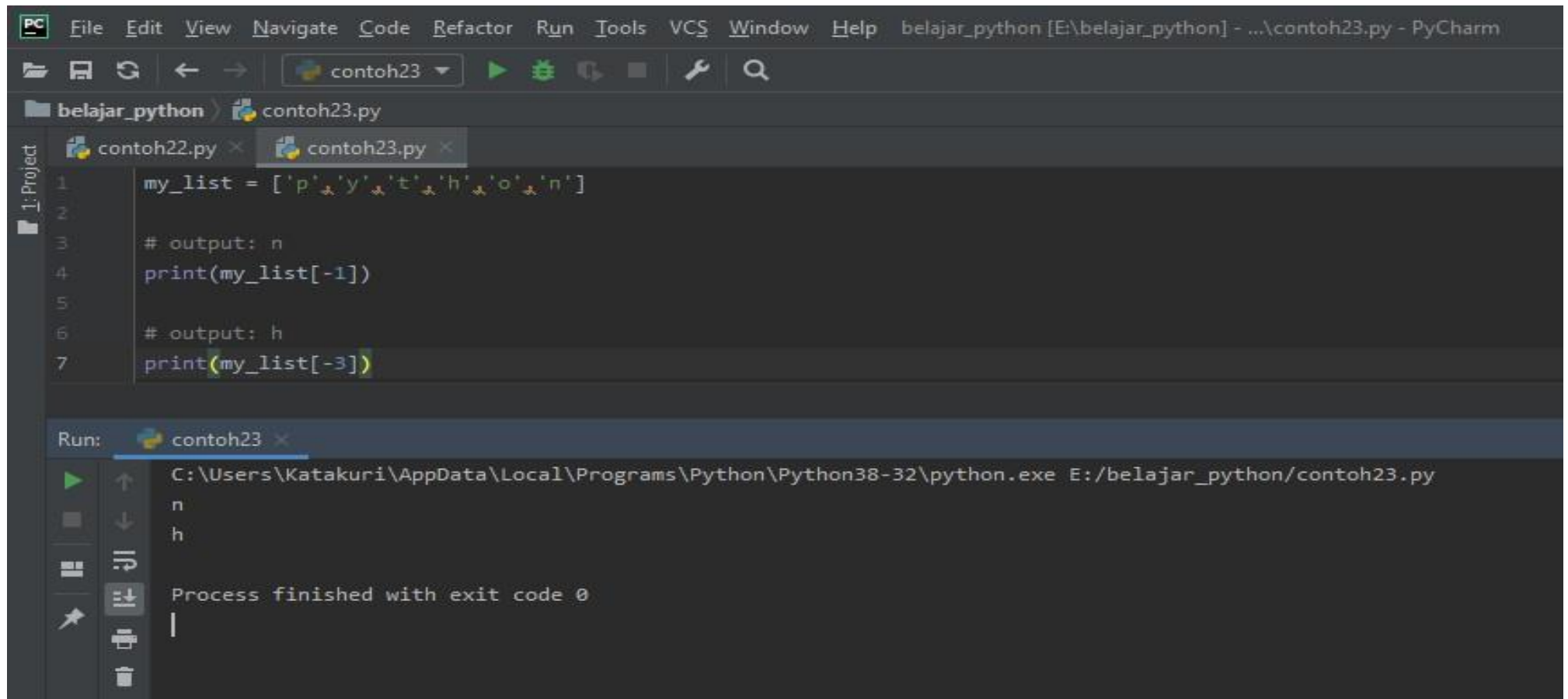
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh22.py - PyCharm
contoh22
1 my_list = ["I", "love", "python", "programming", 2017]
2 # output: I
3 my_list[0]
4
5 # output: python
6 my_list[2]
7
8 # list dalam list
9 your_list = ["hello", [1, 2, 3], "python"]
10
11 # output 1
12 print(your_list[1][0])
13
14 # output 3
15 print(your_list[1][2])
16
17 # IndexError
18 my_list[6]
```

Run: contoh22 x

```
1
3
Traceback (most recent call last):
  File "E:/belajar_python/contoh22.py", line 18, in <module>
    my_list[6]
IndexError: list index out of range
Process finished with exit code 1
```

List dengan Indeks Negatif

Python mendukung indeks negatif, yaitu urutan dimulai dari anggota terakhir. Indeks anggota paling belakang adalah -1, kemudian -2, dan seterusnya.



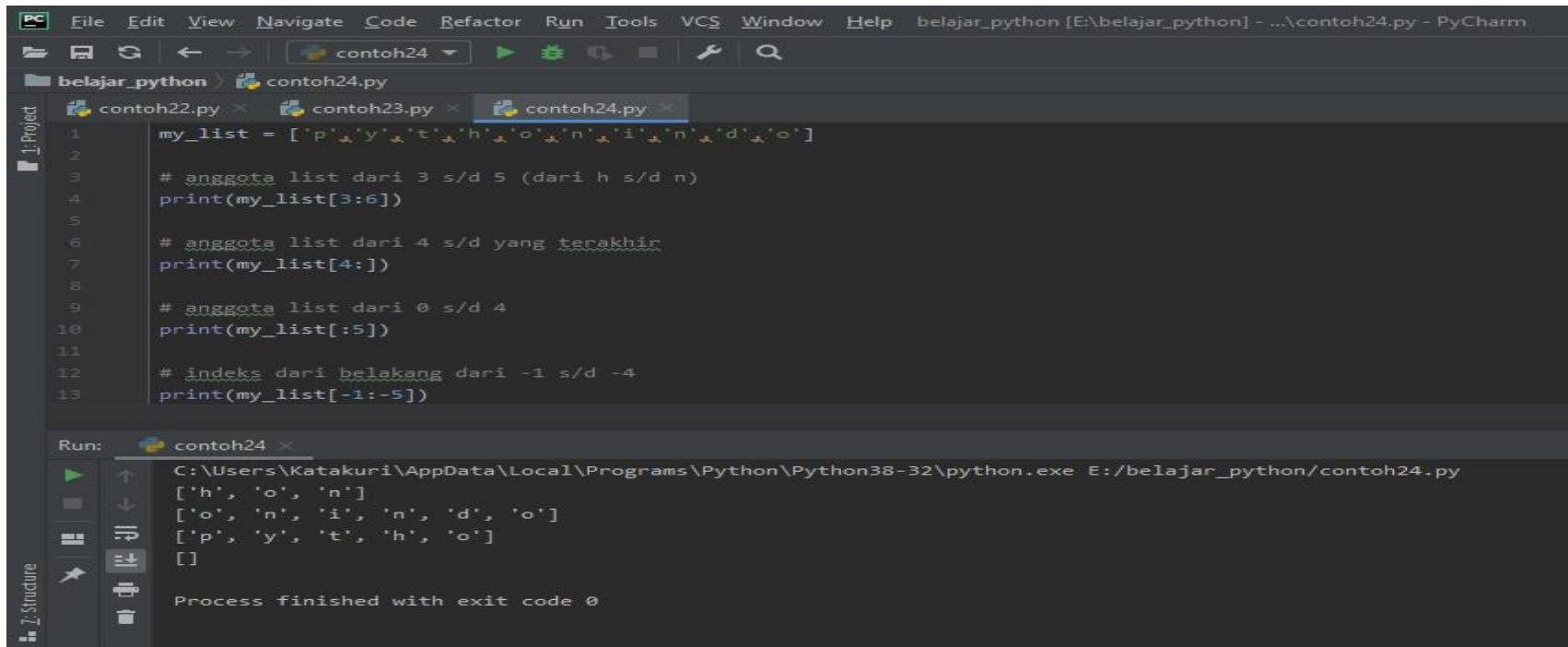
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh23.py - PyCharm
contoh23
belajar_python > contoh23.py
1: Project
  contoh22.py x  contoh23.py x
1  my_list = ['p','y','t','h','o','n']
2
3  # output: n
4  print(my_list[-1])
5
6  # output: h
7  print(my_list[-3])

Run: contoh23 x
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh23.py
n
h
Process finished with exit code 0
```

Memotong (Slicing) List

Kita bisa mengakses anggota list dari range tertentu dengan menggunakan operator slicing titik dua (:). Slicing akan lebih mudah bila kita memahami indeks dengan baik. Perhatikan gambar berikut:

0	1	2	3	4	5	6	7	8	9
P	Y	T	H	O	N	I	N	D	O
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1



```
1 my_list = ['p','y','t','h','o','n','i','n','d','o']
2
3 # anggota list dari 3 s/d 5 (dari h s/d n)
4 print(my_list[3:6])
5
6 # anggota list dari 4 s/d yang terakhir
7 print(my_list[4:])
8
9 # anggota list dari 0 s/d 4
10 print(my_list[:5])
11
12 # indeks dari belakang dari -1 s/d -4
13 print(my_list[-1:-5])
```

Run: contoh24 x

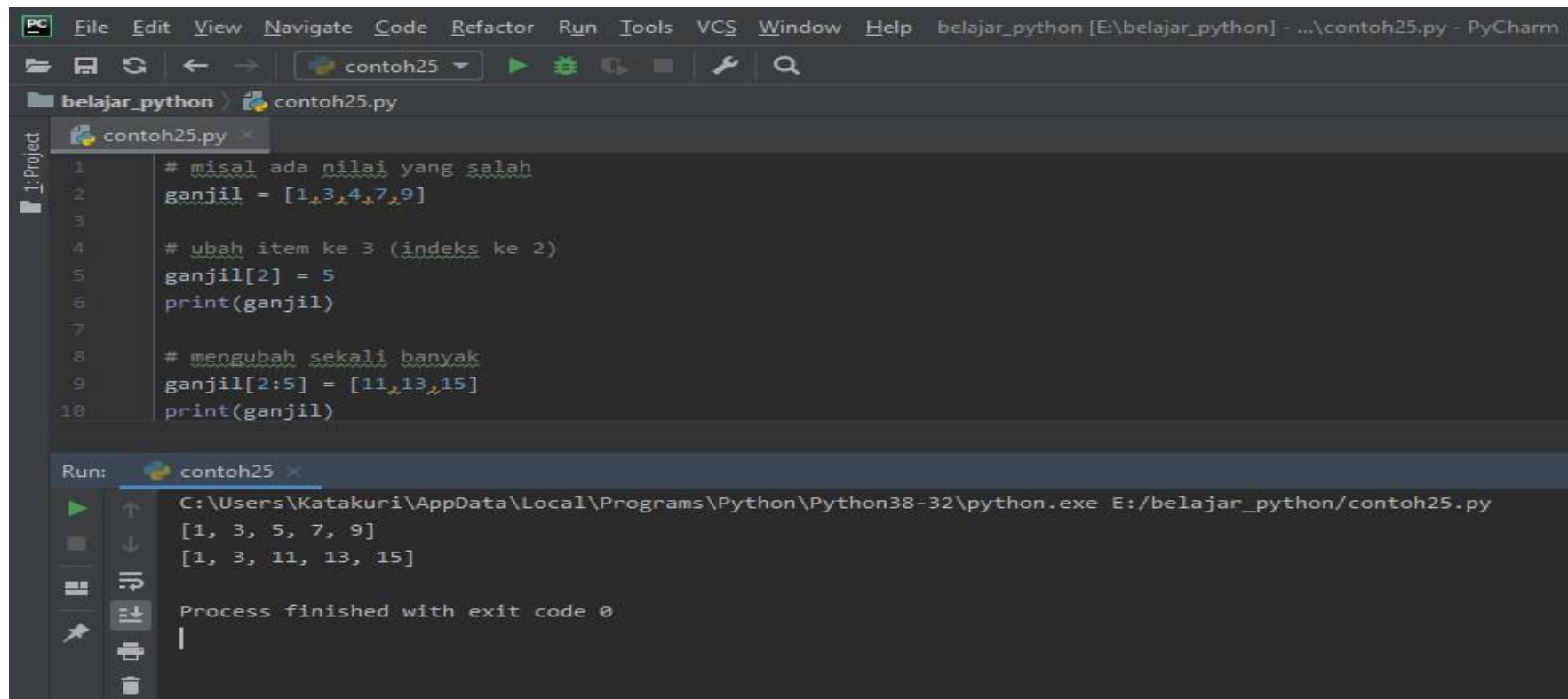
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh24.py

```
['h', 'o', 'n']
['o', 'n', 'i', 'n', 'd', 'o']
['p', 'y', 't', 'h', 'o']
[]
```

Process finished with exit code 0

Mengubah Anggota List

List adalah tipe data yang bersifat mutable, artinya anggotanya bisa diubah. Ini berbeda dengan string dan tuple yang bersifat immutable.



```
1 # misal ada nilai yang salah
2 ganjil = [1, 3, 4, 7, 9]
3
4 # ubah item ke 3. (indeks ke 2)
5 ganjil[2] = 5
6 print(ganjil)
7
8 # mengubah sekali banyak
9 ganjil[2:5] = [11, 13, 15]
10 print(ganjil)
```

Run: contoh25 x

```
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh25.py
[1, 3, 5, 7, 9]
[1, 3, 11, 13, 15]
Process finished with exit code 0
```


Metode List

List memiliki banyak metode untuk operasi seperti menambahkan anggota, menghapus, menyisipkan, menyortir, dan lain sebagainya. Mereka bisa diakses menggunakan format `list.metode()`.

Menambahkan Anggota List

Fungsi `append()` berguna untuk menambahkan anggota ke dalam list. Selain itu, ada metode `extend()` untuk menambahkan anggota list ke dalam list.

```
>>> ganjil = [1,3,5,7]

>>> ganjil.append(9)
>>> print(ganjil)    [1,3,5,7,9]

>>> ganjil.extend([11,13,15])

>>> print(ganjil)    [1,3,5,7,9,11,13,15]
```

Kita juga bisa menggunakan operator + untuk menggabungkan dua list, dan operator * untuk melipatgandakan list.

```
>>> genap = [2, 4, 6]

>>> print(genap + [8, 10, 12])
[2, 4, 6, 8, 10, 12]

>>> print(['p', 'y'] * 2)
['p', 'y', 'p', 'y']
```

Menyisipkan Anggota List

Fungsi insert() berfungsi untuk menyisipkan anggota list pada indeks tertentu.

```
>>> ganjil = [5, 7, 11, 13, 15]

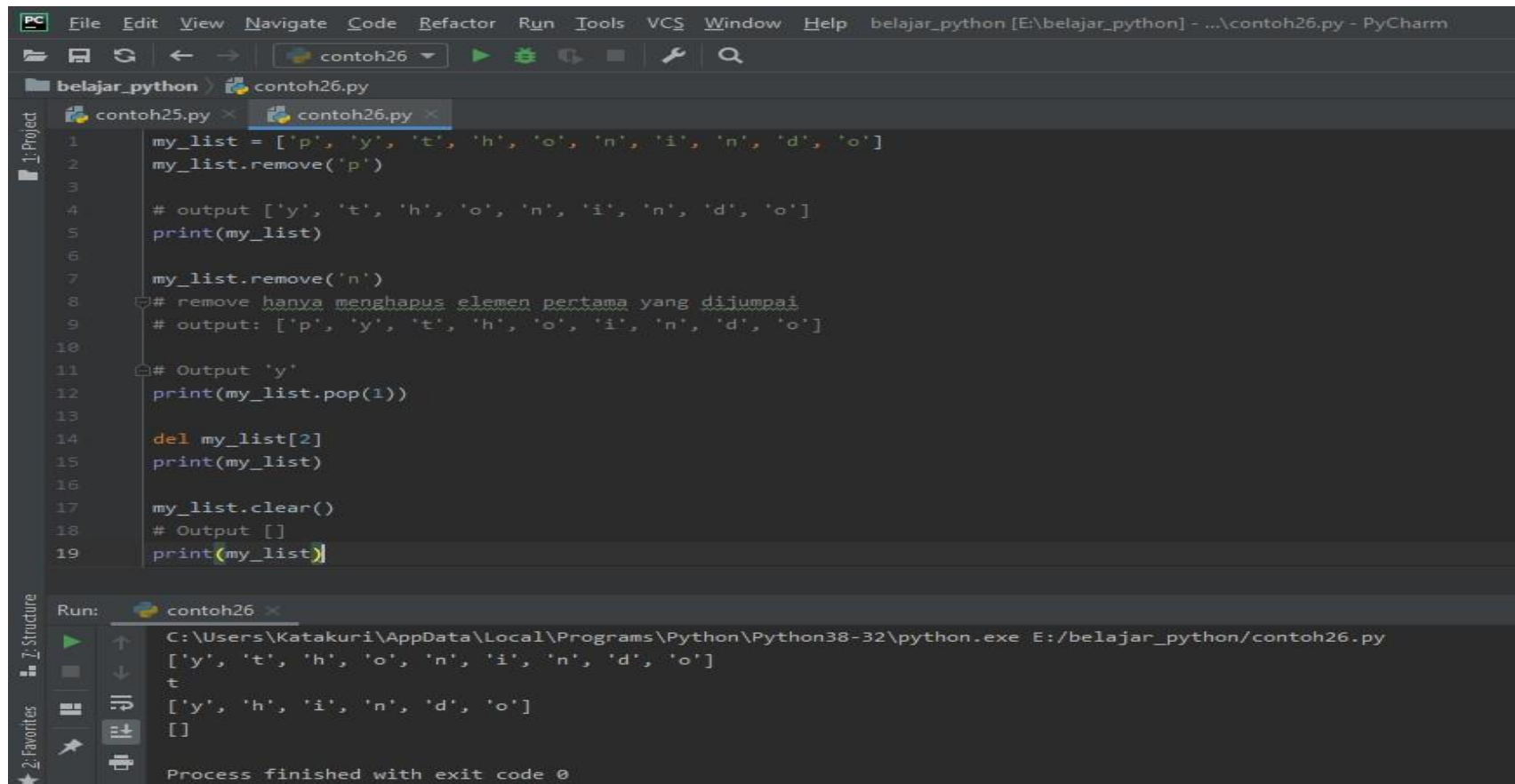
>>> # kita akan menyisipkan 9 setelah angka 7

>>> ganjil.insert(2, 9)
>>> print(ganjil)    [5, 7, 9, 11, 13, 15]
```


Menghapus Anggota List

Kita bisa menggunakan metode `remove()`, `pop()`, atau kata kunci `del` untuk menghapus anggota list. Selain itu kita bisa menggunakan `clear()` untuk mengosongkan list

Fungsi `pop()` selain menghapus anggota list, juga mengembalikan nilai indeks anggota tersebut. Hal ini berguna bila kita ingin memanfaatkan indeks dari anggota yang terhapus untuk digunakan kemudian.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh26.py - PyCharm
contoh26
belajar_python > contoh26.py
contoh25.py x contoh26.py x
1 my_list = ['p', 'y', 't', 'h', 'o', 'n', 'i', 'n', 'd', 'o']
2 my_list.remove('p')
3
4 # output ['y', 't', 'h', 'o', 'n', 'i', 'n', 'd', 'o']
5 print(my_list)
6
7 my_list.remove('n')
8 # remove hanya menghapus elemen pertama yang dijumpai
9 # output: ['p', 'y', 't', 'h', 'o', 'i', 'n', 'd', 'o']
10
11 # Output 'y'
12 print(my_list.pop(1))
13
14 del my_list[2]
15 print(my_list)
16
17 my_list.clear()
18 # Output []
19 print(my_list)
```

Run: contoh26 x

```
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh26.py
['y', 't', 'h', 'o', 'n', 'i', 'n', 'd', 'o']
t
['y', 'h', 'i', 'n', 'd', 'o']
[]
Process finished with exit code 0
```

Mengurutkan Anggota List

Pada saat kita perlu mengurutkan atau menyortir anggota list, kita bisa menggunakan metode `sort()`. Untuk membalik dengan urutan sebaliknya bisa dengan menggunakan argumen `reverse=True`.

```
>>> alfabet = ['a','b','d','f','e','c','h','g','j','i']
>>> alfabet.sort()

>>> print(alfabet)
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

>>> alfabet.sort(reverse=True)
>>> print(alfabet)
['j', 'i', 'h', 'g', 'f', 'e', 'd', 'c', 'b', 'a']
```

Membalik Urutan List

Selain mengurutkan, kita juga bisa membalikkan urutan list dengan menggunakan metode `reverse()`.

```
>>> alfabet = ['a','c','d','e','b']  
  
>>> alfabet.reverse()  
>>> print(alfabet)  ['b','e','d','c','b','a']
```

Tuple

Tuple mirip dengan list. Bedanya, tuple bersifat immutable, sehingga anggotanya tidak bisa diubah. Kalau mirip, mengapa harus menggunakan tuple?

Kita menggunakan tuple tergantung kebutuhan. Untuk beberapa hal, tuple memiliki kelebihan sebagai berikut:

- Karena tuple adalah immutable, maka iterasi pada tuple lebih cepat dibandingkan list.
- Tuple bisa berisi anggota yang immutable yang dapat digunakan sebagai key untuk dictionary. List tidak bisa dipakai untuk itu.
- Kalau kita memerlukan data yang memang tidak untuk diubah, maka menggunakan tuple bisa menjamin bahwa data tersebut akan write-protected.

Membuat Tuple

Tuple dibuat dengan meletakkan semua anggota di dalam tanda kurung (), masing-masing dipisahkan oleh tanda koma. Menggunakan tanda kurung sebenarnya hanya opsional, tapi kita sebaiknya tetap menggunakannya untuk kemudahan pembacaan kode.

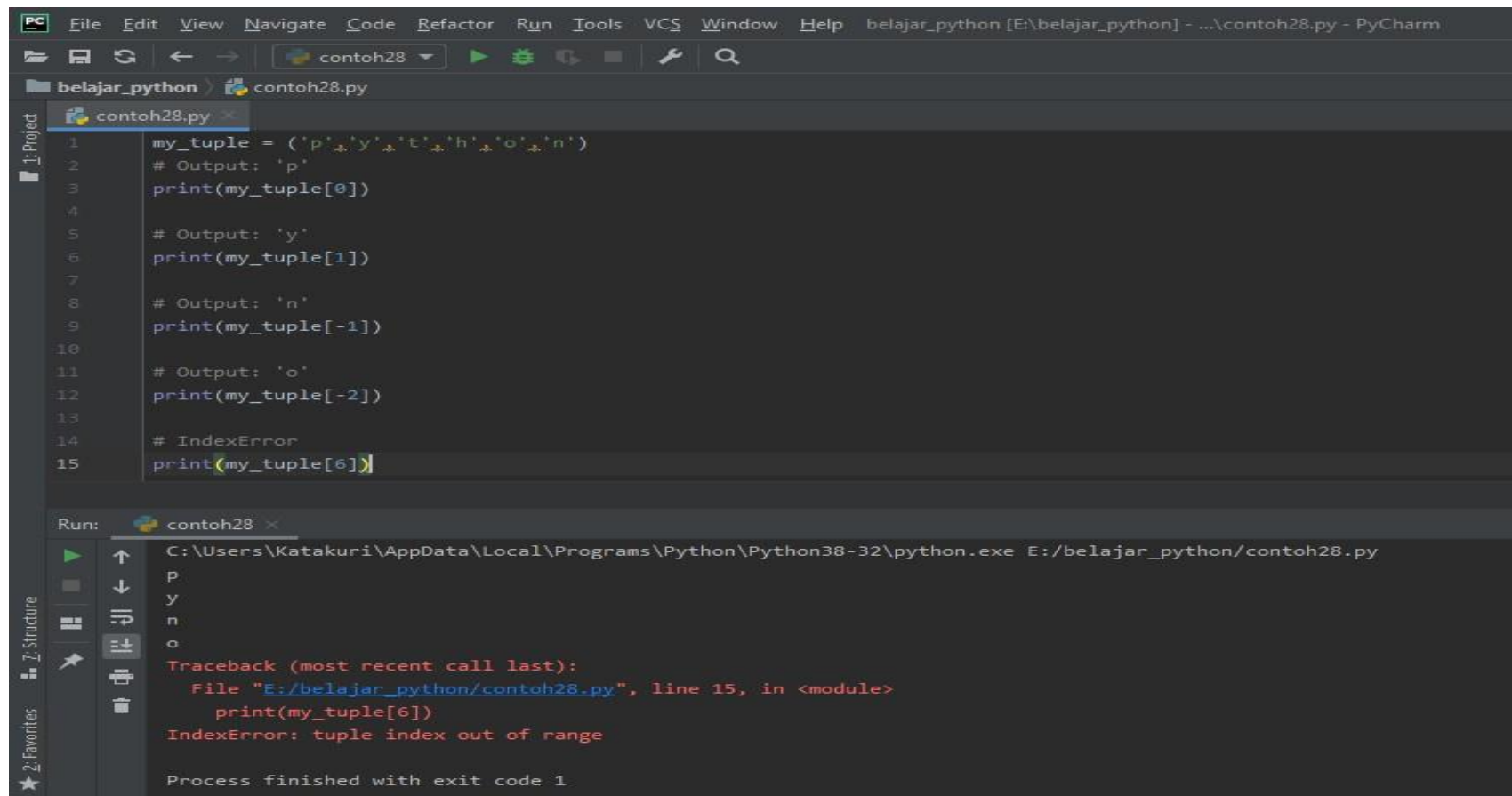
Tuple dapat berisi tipe data yang sama maupun campuran.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh27.py - PyCharm
contoh27
belajar_python > contoh27.py
contoh25.py x contoh26.py x contoh27.py x
1 # membuat tuple kosong
2 my_tuple = ()
3 print(my_tuple)
4
5 # tuple dengan 1 elemen
6 # Output: (1,)
7 my_tuple = (1,)
8 print(my_tuple)
9
10 # tuple berisi integer
11 # output = (1, 2, 3)
12 my_tuple = (1, 2, 3)
13 print(my_tuple)
14
15 # tuple bersarang
16 # Output: ("hello", [1, 2, 3], (4, 5, 6))
17 my_tuple = ("hello", [1, 2, 3], (4, 5, 6))
18 print(my_tuple)
19
20 # Tuple bisa tidak menggunakan tanda ()
21 # Output (1, 2, 3)
22 my_tuple = 1, 2, 3
23
24 # memasukkan anggota tuple ke variabel yang bersesuaian
25 # a akan berisi 1, b berisi 2, dan c berisi 3
26 # output 1 2 3
27 a, b, c = my_tuple
28 print(a, b, c)
```

```
Run: contoh27 x
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh27.py
()
(1,)
(1, 2, 3)
('hello', [1, 2, 3], (4, 5, 6))
1 2 3
Process finished with exit code 0
```

Mengakses Anggota Tuple

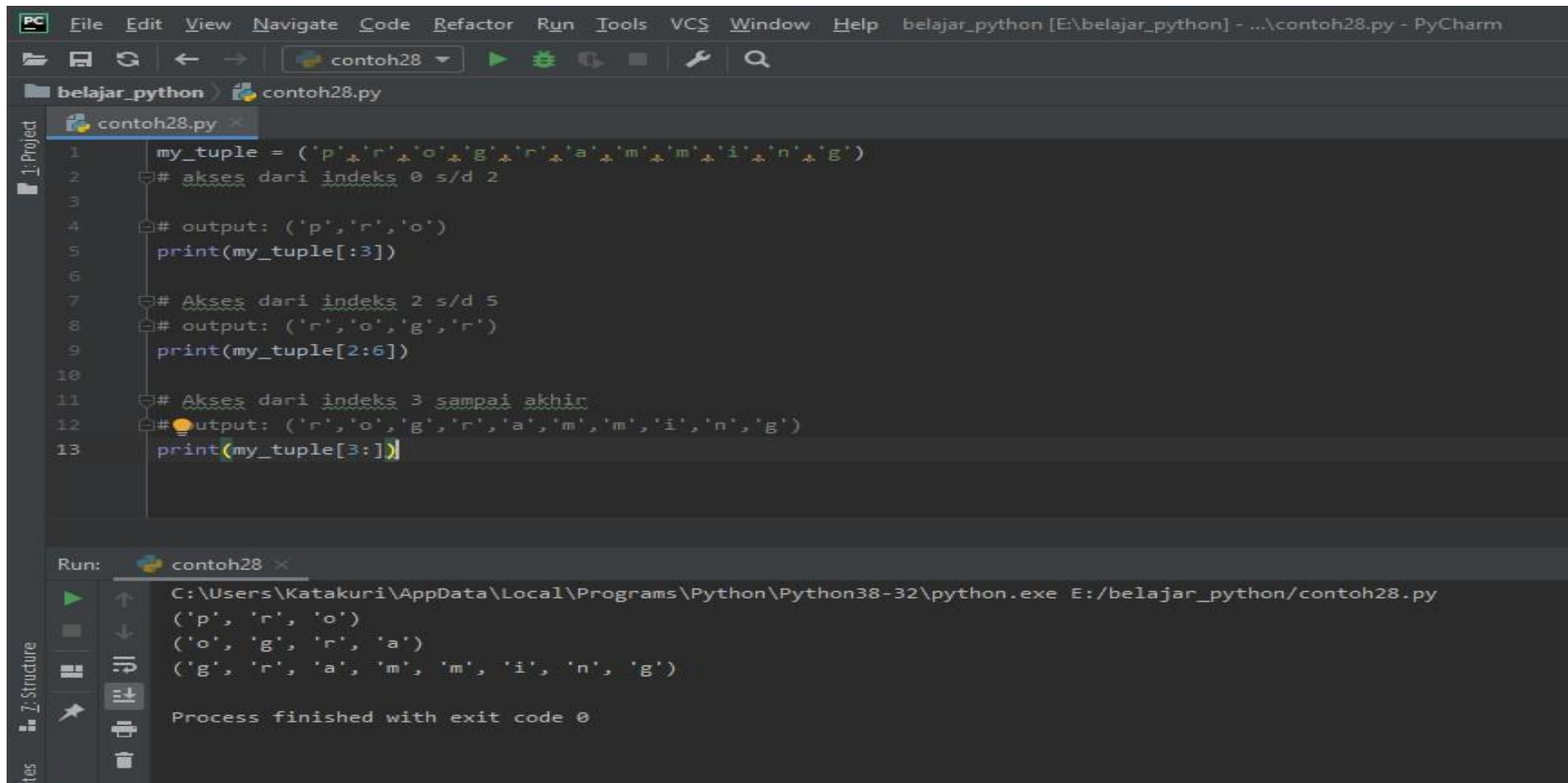
Seperti halnya list, kita bisa mengakses anggota tuple lewat indeksnya menggunakan format `namatuple[indeks]`. Indeks dimulai dari 0 untuk anggota pertama. Selain itu, indeks negatif juga bisa dipakai mulai dari -1 untuk anggota terakhir tuple.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh28.py - PyCharm
contoh28
1 my_tuple = ('p','y','t','h','o','n')
2 # Output: 'p'
3 print(my_tuple[0])
4
5 # Output: 'y'
6 print(my_tuple[1])
7
8 # Output: 'n'
9 print(my_tuple[-1])
10
11 # Output: 'o'
12 print(my_tuple[-2])
13
14 # IndexError
15 print(my_tuple[6])

Run: contoh28 x
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh28.py
p
y
n
o
Traceback (most recent call last):
  File "E:/belajar_python/contoh28.py", line 15, in <module>
    print(my_tuple[6])
IndexError: tuple index out of range
Process finished with exit code 1
```


Sama seperti list, kita bisa mengakses satu range anggota tuple dengan menggunakan operator titik dua (:).



```
1 my_tuple = ('p','r','o','g','r','a','m','m','i','n','g')
2 # akses dari indeks 0 s/d 2
3
4 # output: ('p','r','o')
5 print(my_tuple[:3])
6
7 # Akses dari indeks 2 s/d 5
8 # output: ('r','o','g','r')
9 print(my_tuple[2:6])
10
11 # Akses dari indeks 3 sampai akhir
12 # output: ('r','o','g','r','a','m','m','i','n','g')
13 print(my_tuple[3:])
```

Run: contoh28 x

C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh28.py

('p', 'r', 'o')

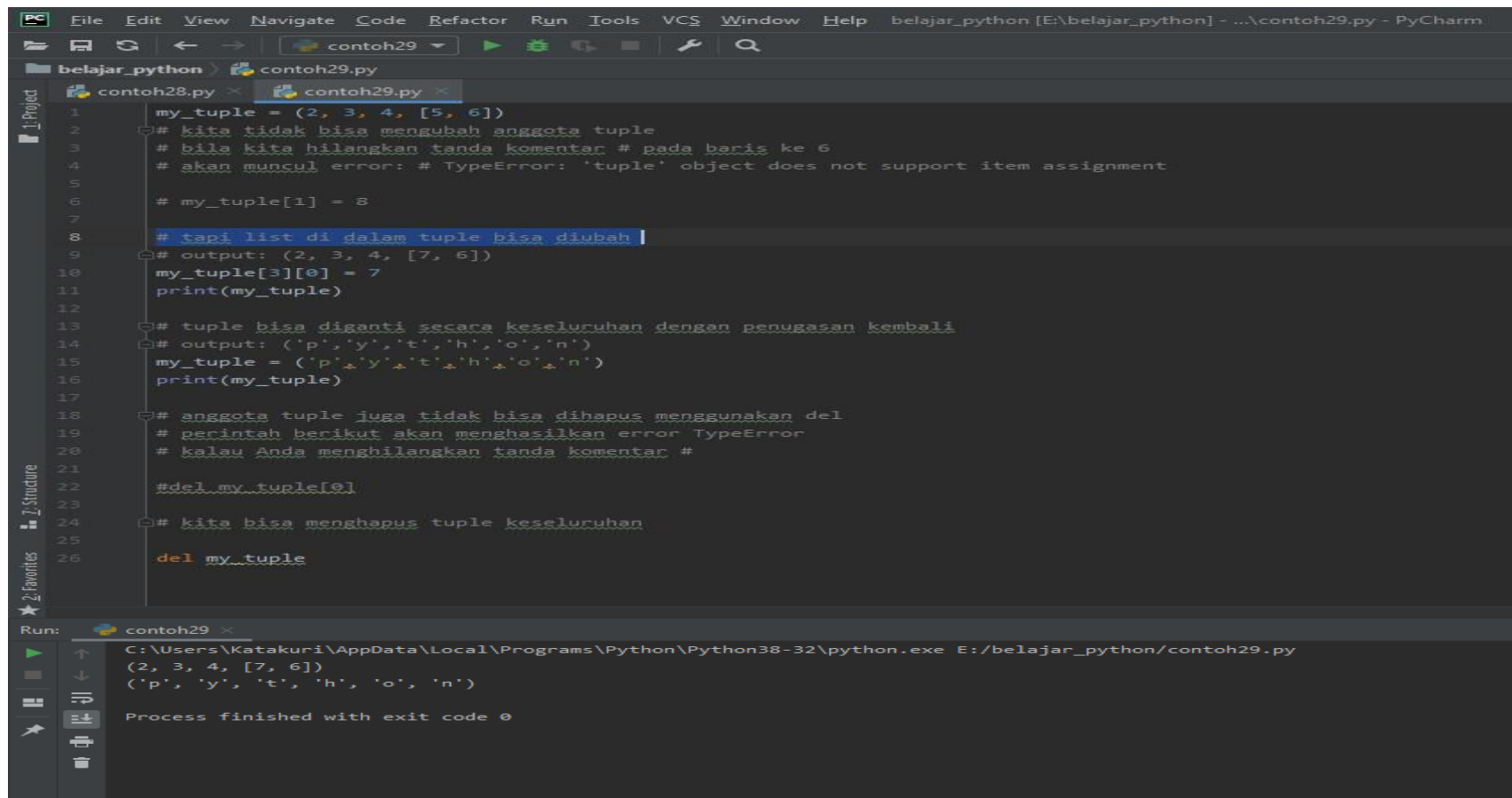
('o', 'g', 'r', 'a')

('g', 'r', 'a', 'm', 'm', 'i', 'n', 'g')

Process finished with exit code 0

Mengubah Anggota Tuple

Setelah tuple dibuat, maka anggota tuple tidak bisa lagi diubah atau dihapus. Akan tetapi, bila anggota tuple-nya adalah tuple bersarang dengan anggota seperti list, maka item pada list tersebut dapat diubah. Jelasnya ada pada contoh berikut:



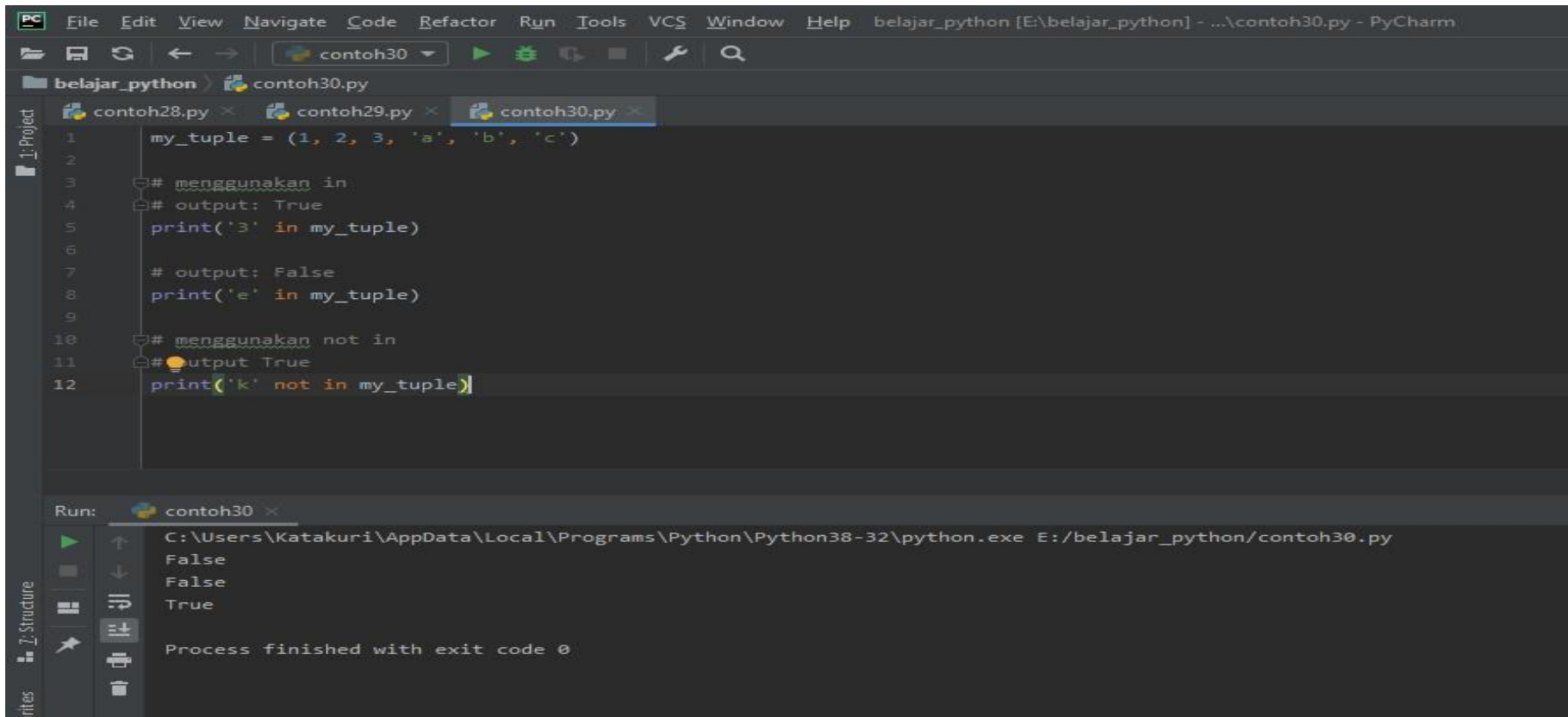
```
1 my_tuple = (2, 3, 4, [5, 6])
2 # kita tidak bisa mengubah anggota tuple
3 # bila kita hilangkan tanda komentar # pada baris ke 6
4 # akan muncul error: # TypeError: 'tuple' object does not support item assignment
5
6 # my_tuple[1] = 8
7
8 # tapi list di dalam tuple bisa diubah
9 # output: (2, 3, 4, [7, 6])
10 my_tuple[3][0] = 7
11 print(my_tuple)
12
13 # tuple bisa diganti secara keseluruhan dengan penugasan kembali
14 # output: ('p', 'y', 't', 'h', 'o', 'n')
15 my_tuple = ('p', 'y', 't', 'h', 'o', 'n')
16 print(my_tuple)
17
18 # anggota tuple juga tidak bisa dihapus menggunakan del
19 # perintah berikut akan menghasilkan error TypeError
20 # kalau Anda menghilangkan tanda komentar #
21
22 #del my_tuple[0]
23
24 # kita bisa menghapus tuple keseluruhan
25
26 del my_tuple
```

Run: contoh29 x

C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh29.py
(2, 3, 4, [7, 6])
('p', 'y', 't', 'h', 'o', 'n')
Process finished with exit code 0

Menguji Keanggotaan Tuple

Seperti halnya string dan list, kita bisa menguji apakah sebuah objek adalah anggota dari tuple atau tidak, yaitu dengan menggunakan operator `in` atau `not in` untuk kebalikannya.



```
1 my_tuple = (1, 2, 3, 'a', 'b', 'c')
2
3 # menggunakan in
4 # output: True
5 print('3' in my_tuple)
6
7 # output: False
8 print('e' in my_tuple)
9
10 # menggunakan not in
11 # output True
12 print('k' not in my_tuple)
```

Run: contoh30 x

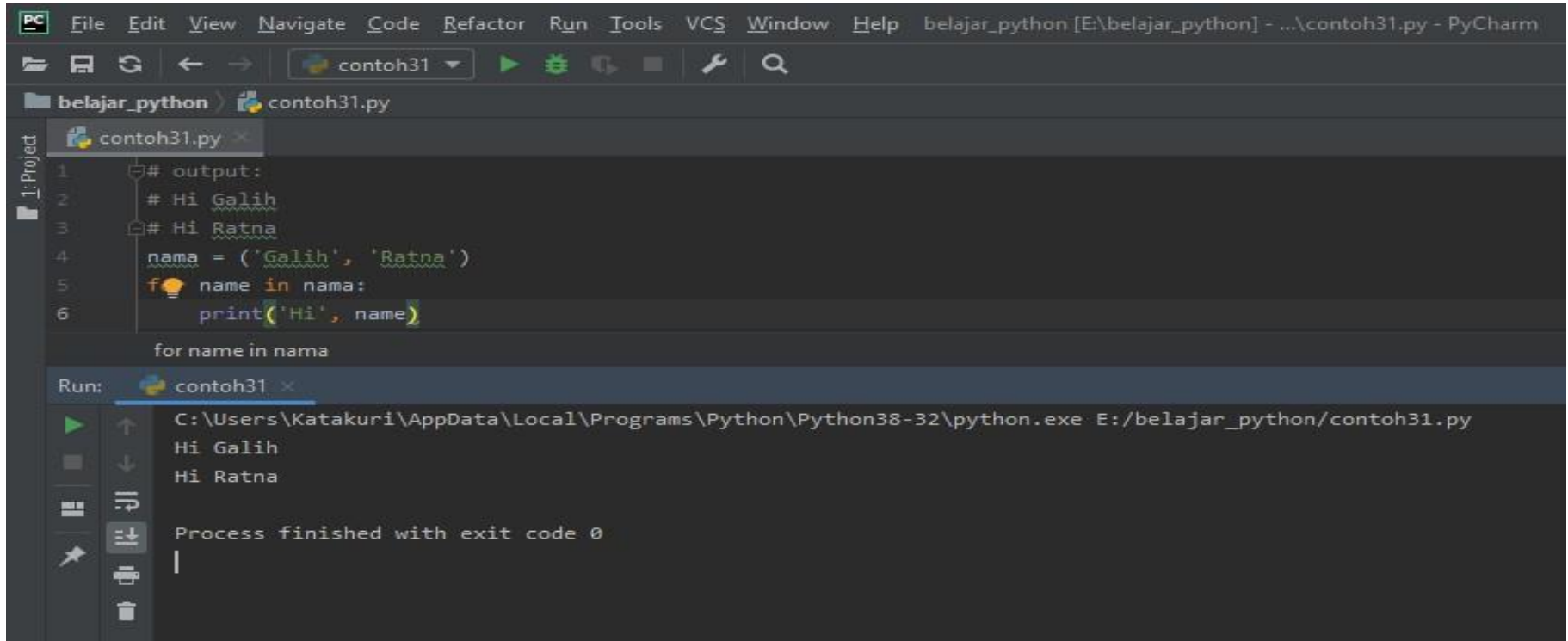
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh30.py

False
False
True

Process finished with exit code 0

Iterasi Pada Tuple

Kita bisa menggunakan for untuk melakukan iterasi pada tiap anggota dalam tuple.



```
1  # output:
2  # Hi Galih
3  # Hi Ratna
4  nama = ('Galih', 'Ratna')
5  for name in nama:
6      print('Hi', name)
```

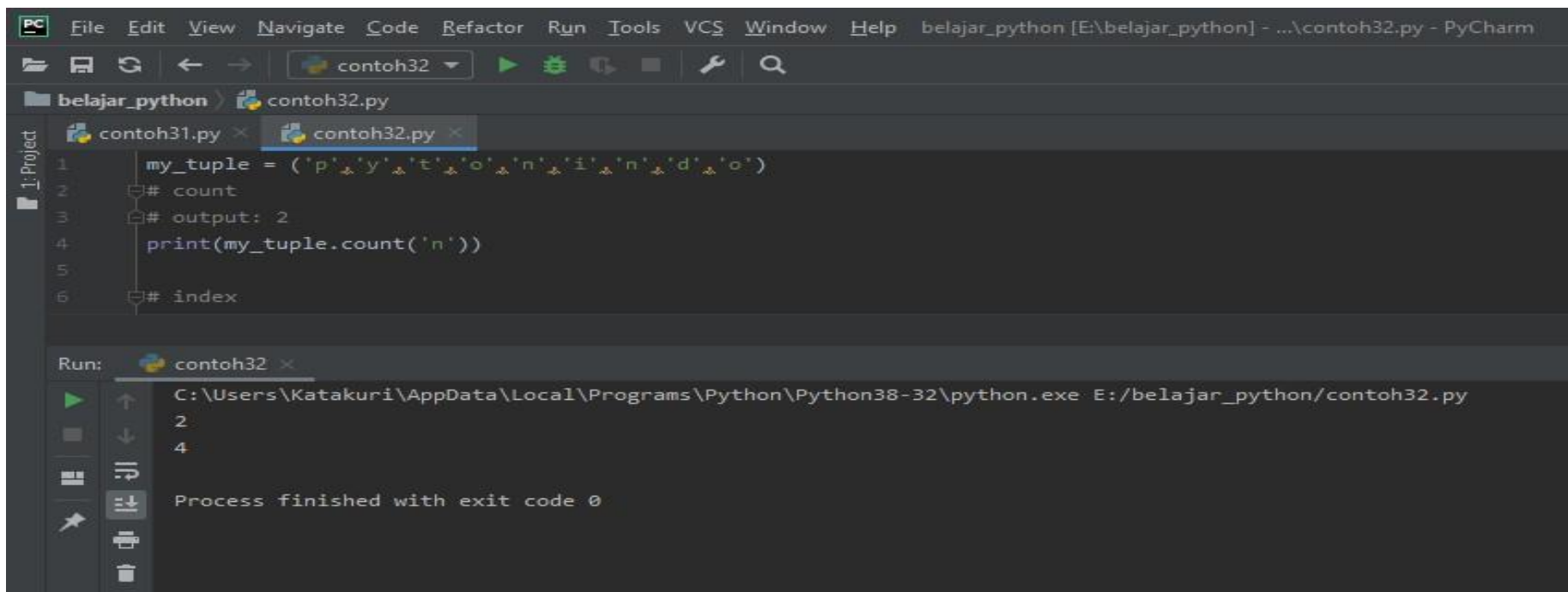
Run: contoh31

```
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh31.py
Hi Galih
Hi Ratna
Process finished with exit code 0
```

Metode dan Fungsi Bawaan Tuple

Tuple hanya memiliki dua buah metode yaitu `count()` dan `index()`.

- Metode `count(x)` berfungsi mengembalikan jumlah item yang sesuai dengan `x` pada tuple
- Metode `index(x)` berfungsi mengembalikan indeks dari item pertama yang sama dengan `x`.



```
1 my_tuple = ('p','y','t','o','n','i','n','d','o')
2 # count
3 # output: 2
4 print(my_tuple.count('n'))
5
6 # index
```

Run: contoh32 x

```
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh32.py
2
4

Process finished with exit code 0
```


Walaupun hanya memiliki dua metode, banyak fungsi bawaan python yang berfungsi untuk melakukan operasi pada tuple. Berikut adalah daftarnya:

Fungsi	Deskripsi
<code>all()</code>	Mengembalikan True jika semua anggota tuple adalah benar (tidak ada yang kosong)
<code>any()</code>	Mengembalikan True jika salah satu atau semua bernilai benar. Jika tuple kosong, maka akan mengembalikan False .
<code>enumerate()</code>	Mengembalikan objek enumerasi. Objek enumerasi adalah objek yang terdiri dari pasangan indeks dan nilai.
<code>len()</code>	Mengembalikan panjang (jumlah anggota) tuple
<code>max()</code>	Mengembalikan anggota terbesar di tuple
<code>min()</code>	Mengembalikan anggota terkecil di tuple
<code>sorted()</code>	Mengambil anggota tuple dan mengembalikan list baru yang sudah diurutkan
<code>sum()</code>	Mengembalikan jumlah dari semua anggota tuple
<code>tuple()</code>	Mengubah sequence (list, string, set, dictionary) menjadi tuple

Studi Kasus : Membuat List 2 Dimensi

Buatlah sebuah list (array) 2 dimensi dimana terdapat baris dan kolom dengan nilai sebagai berikut :

1	2	3
4	5	6
7	8	9
0		

Tampilkan lah :

- Baris Pertama, Kolom Pertama
- Baris Pertama, Kolom Kedua
- Baris Pertama, Kolom Ketiga
- Baris Ketiga, Kolom Ketiga
- Baris Ke Empat, Kolom Pertama

Hasil

```
Run: latihan x
D:\IMK\projek_python\venv\Scripts\python.exe D:/IMK/projek_python/latihan.py
Baris Pertama, Kolom Pertama Adalah :
1
Baris Pertama, Kolom Kedua Adalah :
2
Baris Pertama, Kolom Ketiga Adalah :
3
Baris Ketiga, Kolom Ketiga Adalah :
9
Baris Ke Empat, Kolom Pertama Adalah :
0

Process finished with exit code 0
```

Latihan

Buatlah input, proses dan output secara berulang dengan memanfaatkan fungsi matriks/list seperti pada koding dibawah ini :

#variable yg berulang menggunakan List/matriks

```
list_nim=[]
```

```
list_uts=[]
```

```
list_uas=[]
```

```
list_total=[]
```

```
ulang=2
```

```
for i in range(ulang):
```

```
    print ("data Ke - " + str(i+1))
```

```
    list_nim.append(input("Masukkan Nim anda : "))
```

```
    list_uts.append(int(input("Masukkan Nilai UTS anda :")))
```

```
    list_uas.append(int(input("Masukkan Nilai UAS :")))
```

#proses

```
for i in range(ulang):
```

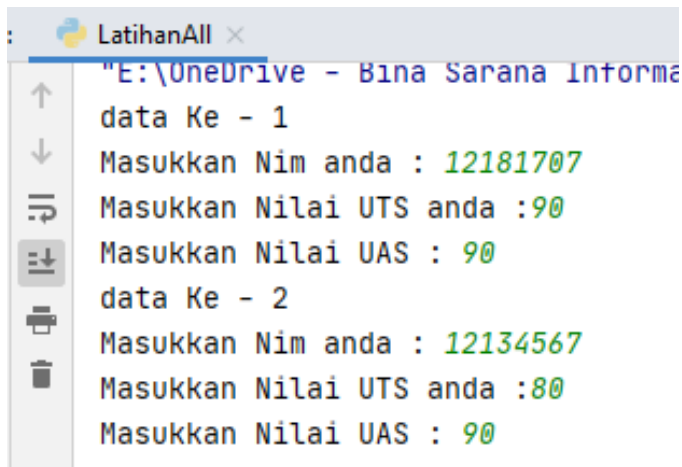
```
    list_total.append((list_uas[i] + list_uts[i]) / 2)
```

#Cetak

```
print("=====")
print("Nim    Nilai Uts    Nilai UAS    Total")
print("=====")
for i in range(ulang):
    print ("%s \t %i \t\t %i \t\t\t %i" % (list_nim[i],list_uts[i],list_uas[i],list_total[i]))

print("=====")
```

Tampilan Input



```
LatihanAll x
"E:\OneDrive - Bina Sarana Informa
data Ke - 1
Masukkan Nim anda : 12181707
Masukkan Nilai UTS anda : 90
Masukkan Nilai UAS : 90
data Ke - 2
Masukkan Nim anda : 12134567
Masukkan Nilai UTS anda : 80
Masukkan Nilai UAS : 90
```

Tampilan Output

```
=====
Nim        Nilai Uts        Nilai UAS        Total
=====
12181707    90                90                90
12134567    80                90                85
=====
```