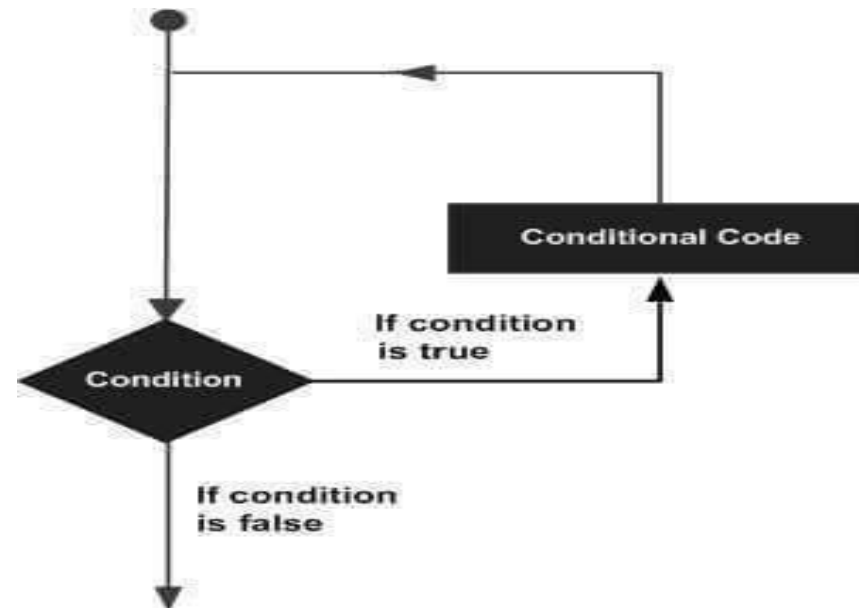


Pertemuan 5

Perulangan

Perulangan

Secara umum, Python mengeksekusi program baris perbaris. Mulai dari baris satu, dua, dan seterusnya. Ada kalanya, kita perlu mengeksekusi satu baris atau satu blok kode program beberapa kali. Hal ini disebut dengan perulangan atau biasa disebut looping atau iterasi. Untuk lebih jelasnya perhatikan gambar berikut:



Pada gambar bisa dilihat bahwa perulangan juga memerlukan tes kondisi. Bila hasil tes kondisi True, maka blok kode kembali dieksekusi. Tapi jika False, maka keluar dari perulangan. Pada python, perulangan bisa dilakukan dengan dua cara atau metode, yaitu:

1. Menggunakan for
2. Menggunakan while

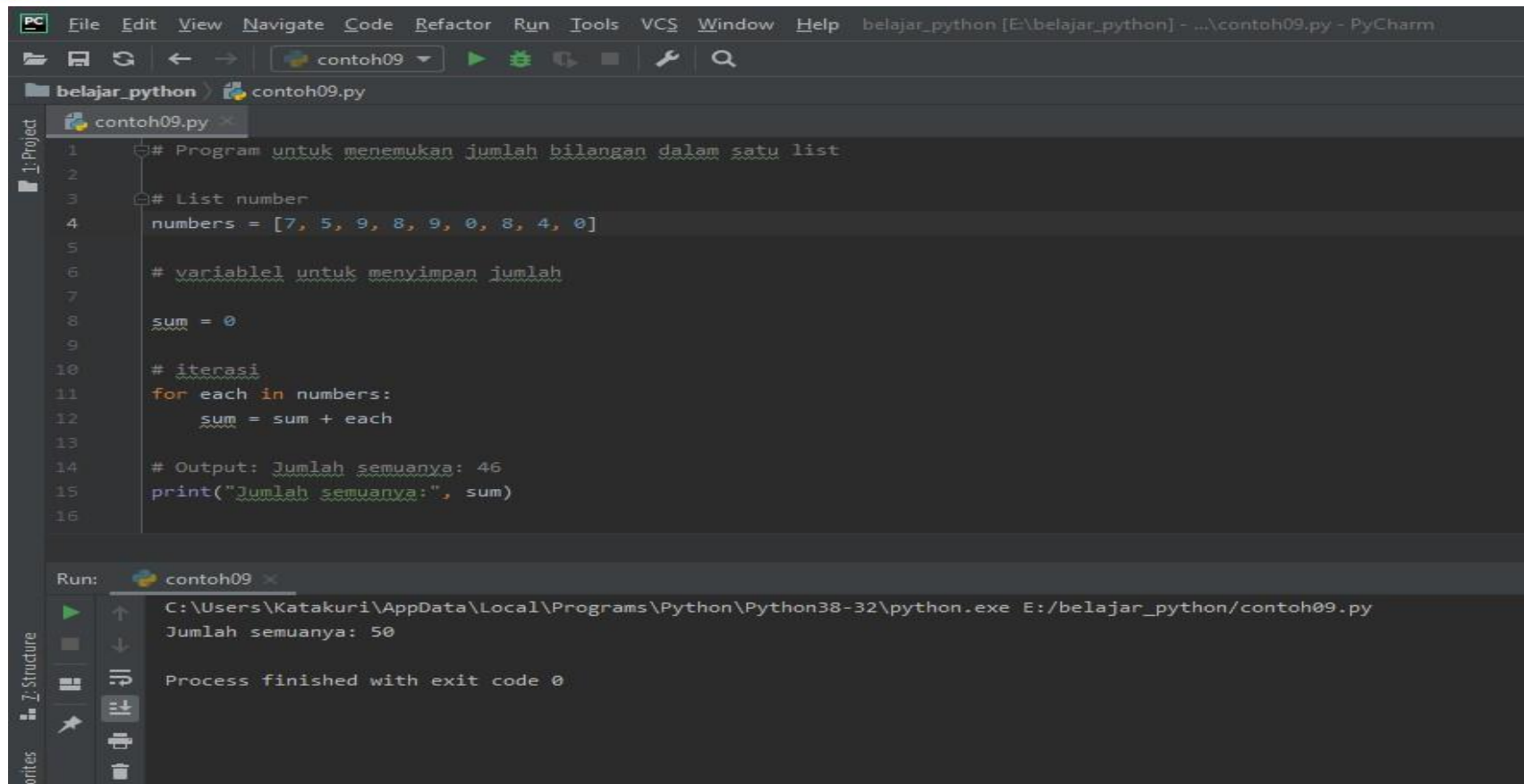
Perulangan dengan Menggunakan For

Perulangan dengan menggunakan for memiliki sintaks seperti berikut:

```
for var in sequence:  
    body of for
```

var adalah variabel yang digunakan untuk penampung sementara nilai dari sequence pada saat terjadi perulangan. Sequence adalah tipe data berurut seperti string, list, dan tuple.

Perulangan terjadi sampai looping mencapai elemen atau anggota terakhir dari sequence. Bila loop sudah sampai ke elemen terakhir dari sequence, maka program akan keluar dari looping.



```
1  # Program untuk menemukan jumlah bilangan dalam satu list
2
3  # List number
4  numbers = [7, 5, 9, 8, 9, 0, 8, 4, 0]
5
6  # variabel untuk menyimpan jumlah
7
8  sum = 0
9
10 # iterasi
11 for each in numbers:
12     sum = sum + each
13
14 # Output: jumlah semuanya: 46
15 print("Jumlah semuanya:", sum)
16
```

Run: contoh09 x

C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh09.py

Jumlah semuanya: 50

Process finished with exit code 0

Fungsi Range

Fungsi `range()` dapat digunakan untuk menghasilkan deret bilangan. `range(10)` akan menghasilkan bilangan dari 0 sampai dengan 9 (10 bilangan). Kita juga bisa menentukan batas bawah, batas atas, dan interval dengan format `range(batas bawah, batas atas, interval)`. Bila interval dikosongkan, maka nilai default 1 yang akan digunakan.

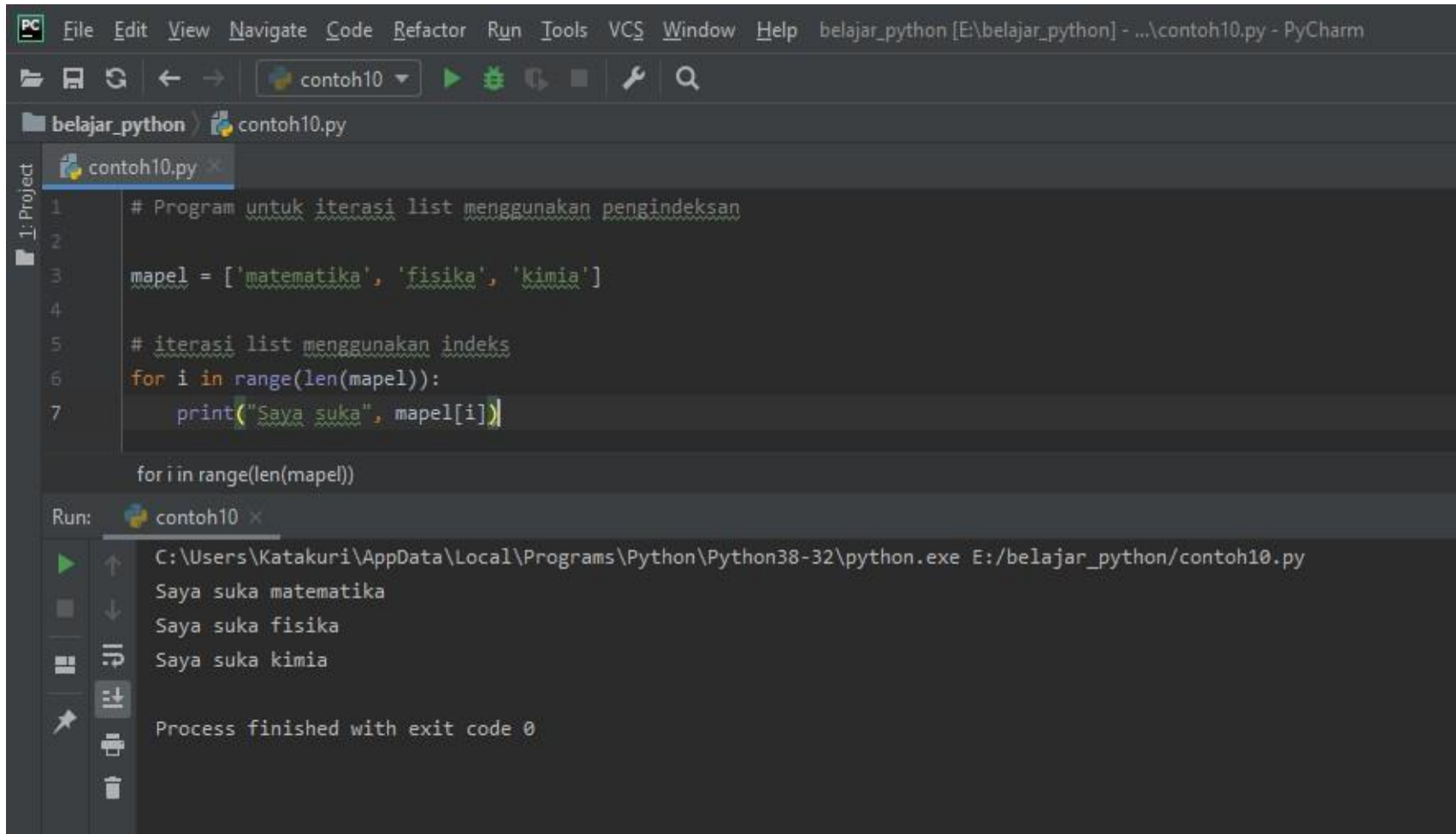
Fungsi range tidak menyimpan semua nilai dalam memori secara langsung. Ia hanya akan mengingat batas bawah, batas atas, dan interval dan membangkitkan hasilnya satu persatu hanya bila dipanggil. Untuk membuat fungsi ini langsung menampilkan semua item, kita bisa menggunakan fungsi list(). Untuk jelasnya perhatikan contoh berikut:

```
# Output: range(0,10)
print(range(10))
# Output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(list(range(10)))

# Output: [2, 3, 4, 5, 6, 7]
print(list(range(2,8)))

# Output: [2, 5, 8, 11, 14, 17]
print(list(range(2, 20, 3)))
```

Kita bisa menggunakan fungsi `range()` dalam perulangan menggunakan `for` untuk iterasi bilangan berurut. Hal ini dengan cara mengkombinasikan fungsi `range()` dengan fungsi `len()`. Fungsi `len()` berfungsi untuk mendapatkan panjang atau jumlah elemen suatu data sekuensial atau berurut.



```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh10.py - PyCharm
contoh10
belajar_python > contoh10.py
contoh10.py x
1 # Program untuk iterasi list menggunakan pengindeksan
2
3 mapel = ['matematika', 'fisika', 'kimia']
4
5 # iterasi list menggunakan indeks
6 for i in range(len(mapel)):
7     print("Saya suka", mapel[i])
8
9 for i in range(len(mapel))
Run: contoh10 x
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh10.py
Saya suka matematika
Saya suka fisika
Saya suka kimia
Process finished with exit code 0
```

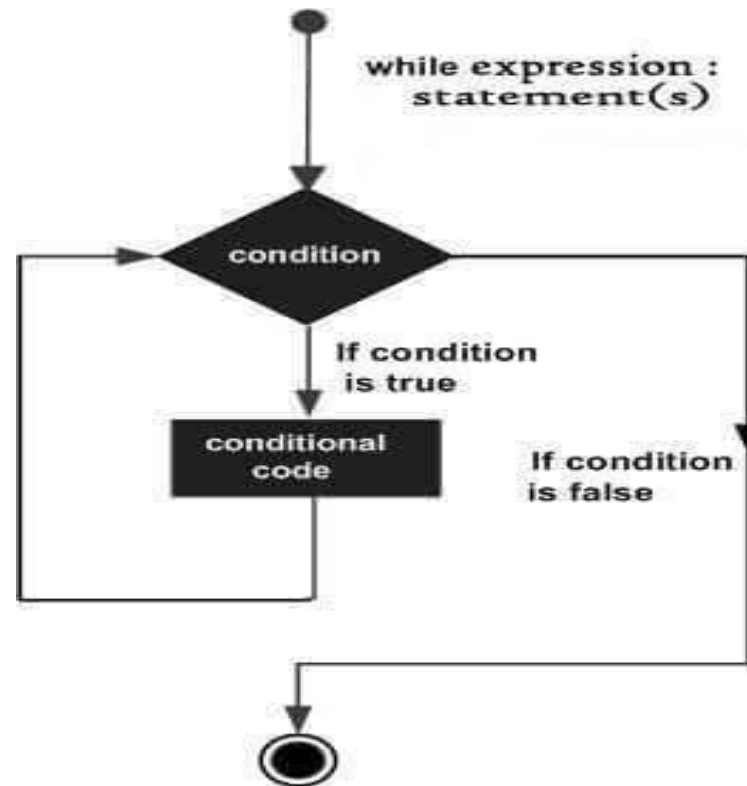

Perulangan Menggunakan While

Perulangan menggunakan while akan menjalankan blok pernyataan terus menerus selama kondisi bernilai benar. Adapun sintaks dari perulangan menggunakan while adalah:

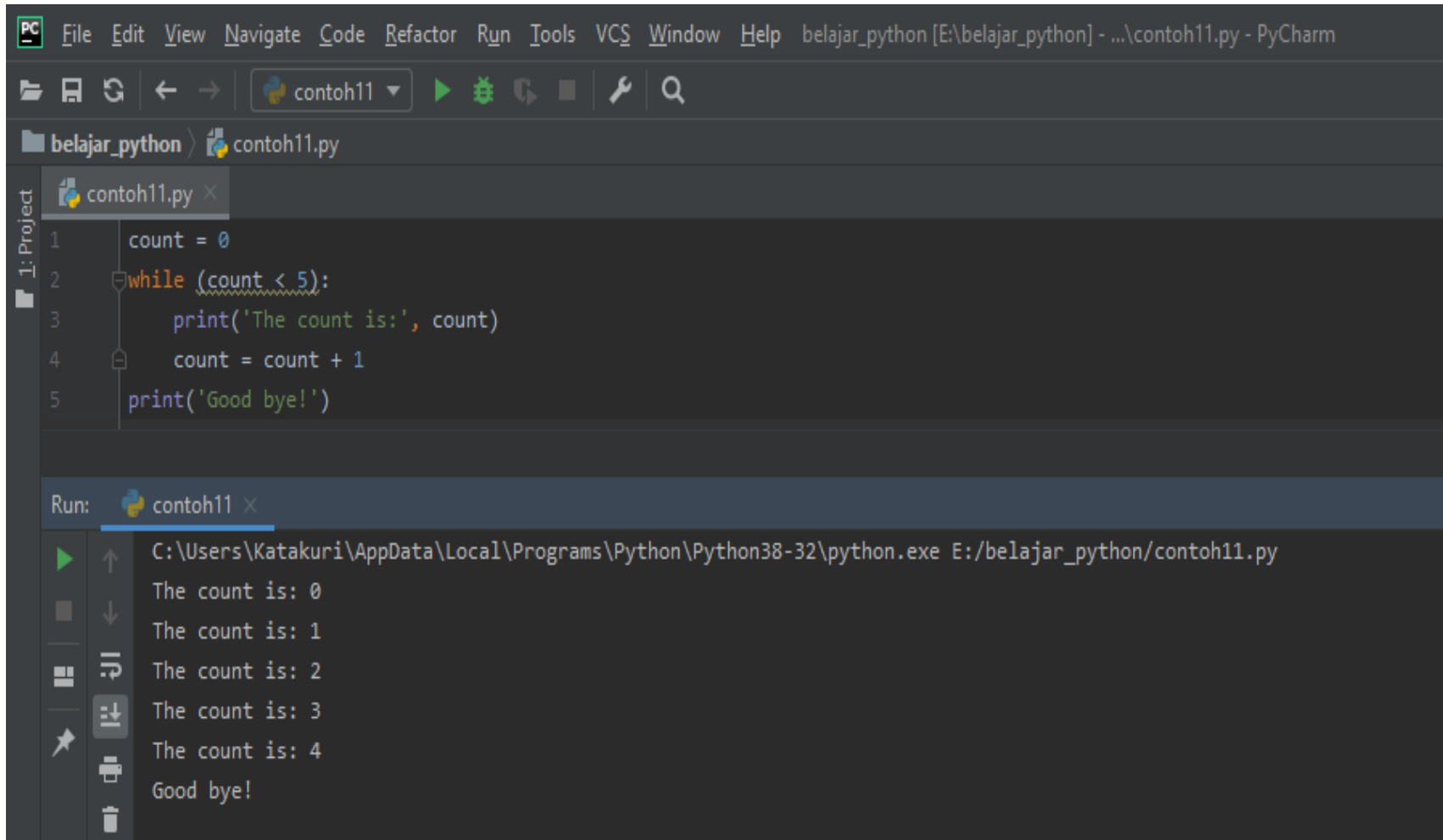
```
while expression:  
    statement (s)
```

Di sini, statement (s) bisa terdiri dari satu baris atau satu blok pernyataan. Expression merupakan ekspresi atau kondisi apa saja, dan untuk nilai selain nol dianggap True. Iterasi akan terus berlanjut selama kondisi benar. Bila kondisi salah, maka program akan keluar dari

while dan lanjut ke baris pernyataan di luar while. Adapun diagram alir while adalah seperti gambar berikut:



Perhatikan bahwa bila kondisi yang diuji bernilai salah, maka loop tidak akan pernah dieksekusi.



The screenshot displays the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The toolbar shows icons for file operations and running code. The project explorer on the left shows a folder named 'belajar_python' containing a file 'contoh11.py'. The editor window shows the following Python code:

```
1 count = 0
2 while (count < 5):
3     print('The count is:', count)
4     count = count + 1
5 print('Good bye!')
```

Below the editor, the 'Run' tab is active, showing the command used to execute the script: `C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh11.py`. The output of the script is displayed in the console:

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
Good bye!
```

Di sini, blok pernyataan `print('The count is:', count)`, dijalankan terus selama `count` masih lebih kecil dari 5. `Count` ditambah 1 setiap kali iterasi. Pada saat nilai `count` mencapai 5, maka kondisi menjadi `False` dan program keluar dari looping `while` dan melanjutkan baris selanjutnya yaitu `print("Good bye")`.

Infinite Loop

Sebuah kondisi dimana loop selalu benar dan tidak pernah salah disebut loop tidak terbatas (infinite loop). Terkadang hal ini menjadi masalah. Tapi sering juga infinite loop berguna, misalnya untuk program client/server dimana server perlu menjaga komunikasi tetap hidup dan tidak terputus. Pada contoh program while di atas, bila kita lupa menuliskan kode `count = count + 1`, maka akan jadi infinite loop. Hasilnya akan jadi seperti berikut:

```
The count is: 0
```

```
The count is: 0 The count is: 0
```

```
The count is: 0 The count is: 0
```

```
Traceback (most recent call last):
```

```
File "<pyshell#4>", line 2, in <module> print('The count is:',  
count)
```

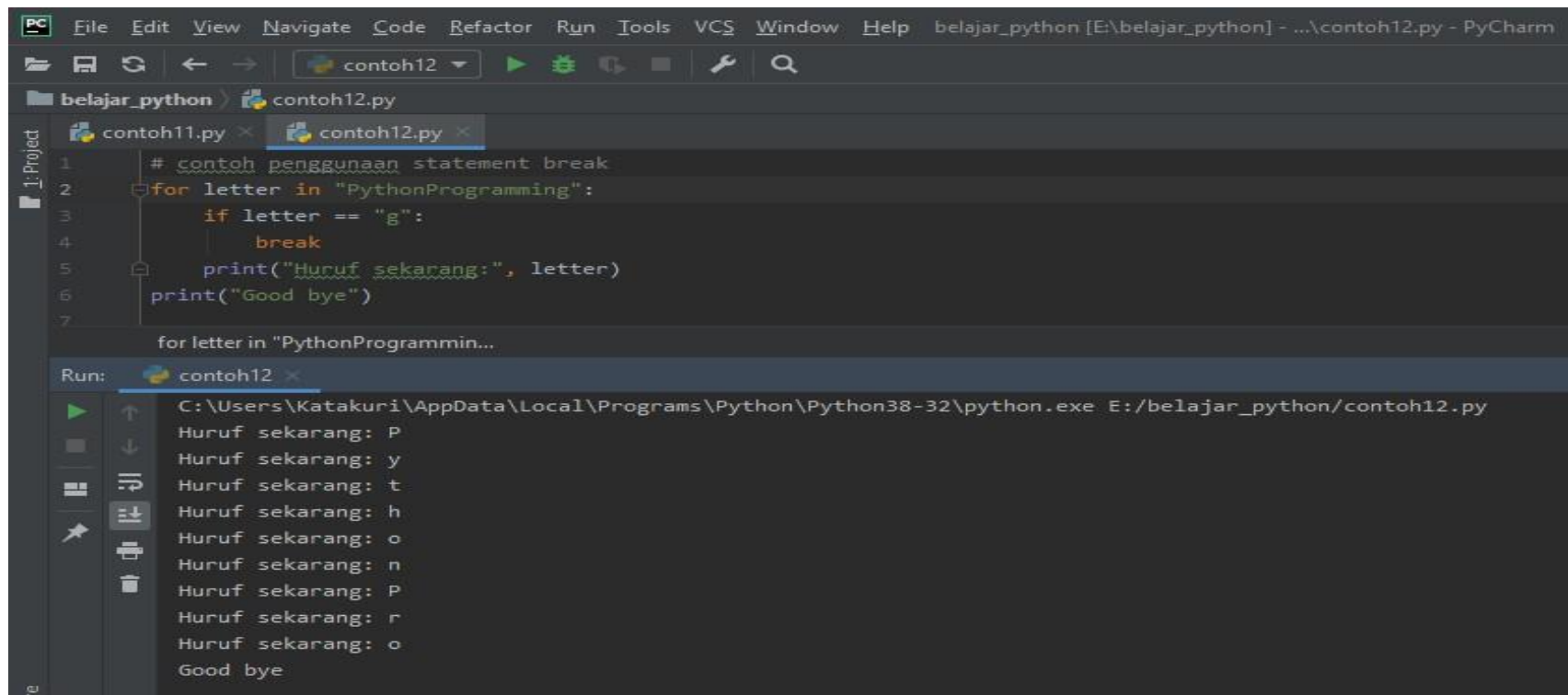
```
File "C:\Python34\lib\idlelib\PyShell.py", line 1344, in write  
return self.shell.write(s, self.tags) KeyboardInterrupt
```

Kita perlu menekan CTRL+C untuk menghentikan program.

Kendali Looping

Looping umumnya akan berhenti bila kondisi sudah bernilai salah. Akan tetapi, seringkali kita perlu keluar dari looping di tengah jalan tergantung keperluan. Hal ini bisa kita lakukan dengan menggunakan kata kunci **break** dan **continue**.

Statement **break** memaksa program keluar dari blok looping di tengah jalan. Sedangkan statement **continue** menyebabkan program langsung melanjut ke step / interval berikutnya dan mengabaikan (skip) baris kode di bawahnya (yang satu blok). Jelasnya perhatikan contoh berikut:



```
# contoh penggunaan statement break
for letter in "PythonProgramming":
    if letter == "g":
        break
    print("Huruf sekarang:", letter)
print("Good bye")

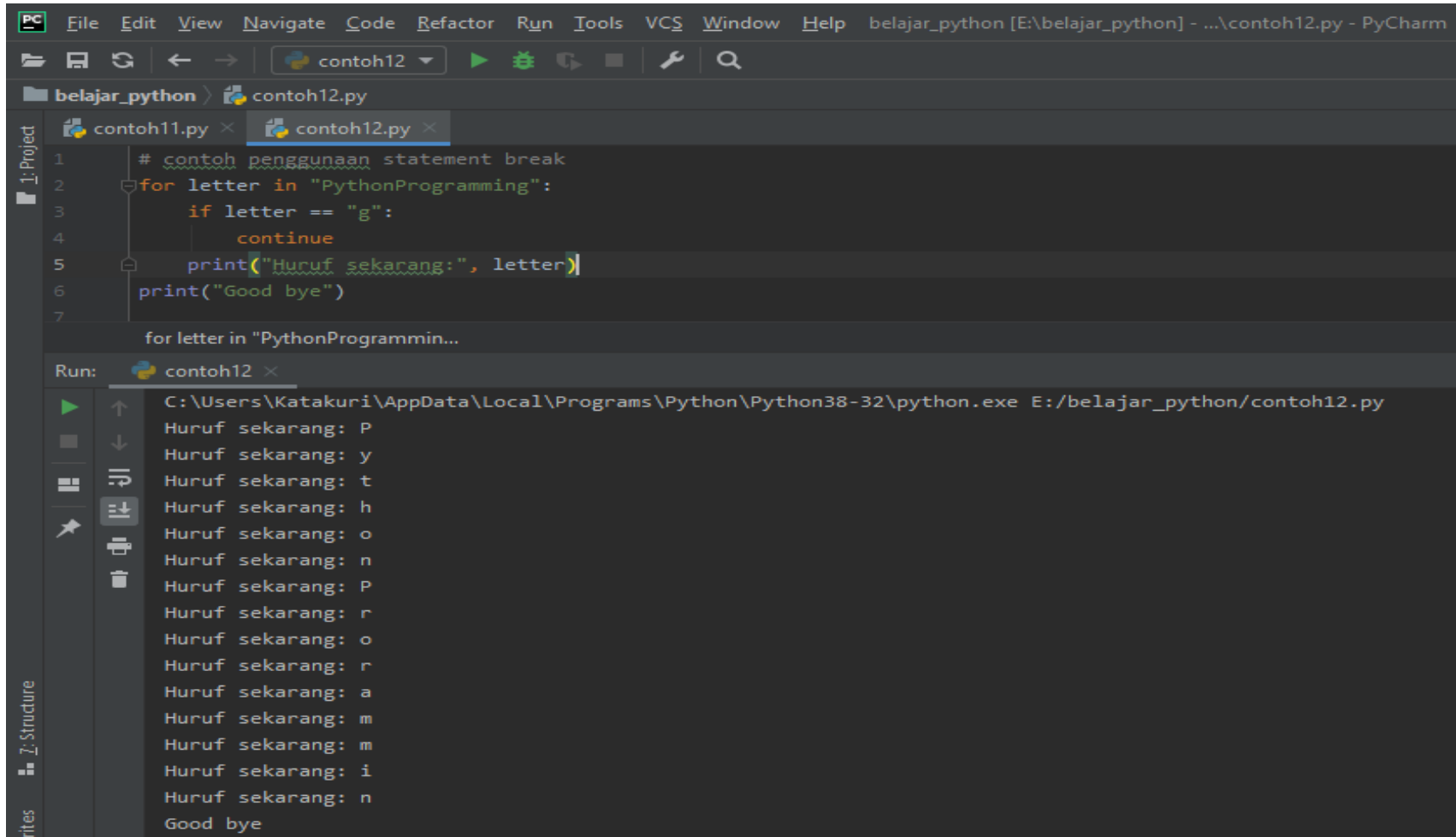
for letter in "PythonProgrammin..."
```

Run: contoh12 x

C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh12.py

Huruf sekarang: P
Huruf sekarang: y
Huruf sekarang: t
Huruf sekarang: h
Huruf sekarang: o
Huruf sekarang: n
Huruf sekarang: P
Huruf sekarang: r
Huruf sekarang: o
Good bye

Bila pada program di atas kita ganti kode **break** menjadi **continue**, maka hasilnya akan jadi seperti berikut:



```
# contoh penggunaan statement break
for letter in "PythonProgramming":
    if letter == "g":
        continue
    print("Huruf sekarang:", letter)
print("Good bye")

for letter in "PythonProgrammin..."
```

Run: contoh12 x

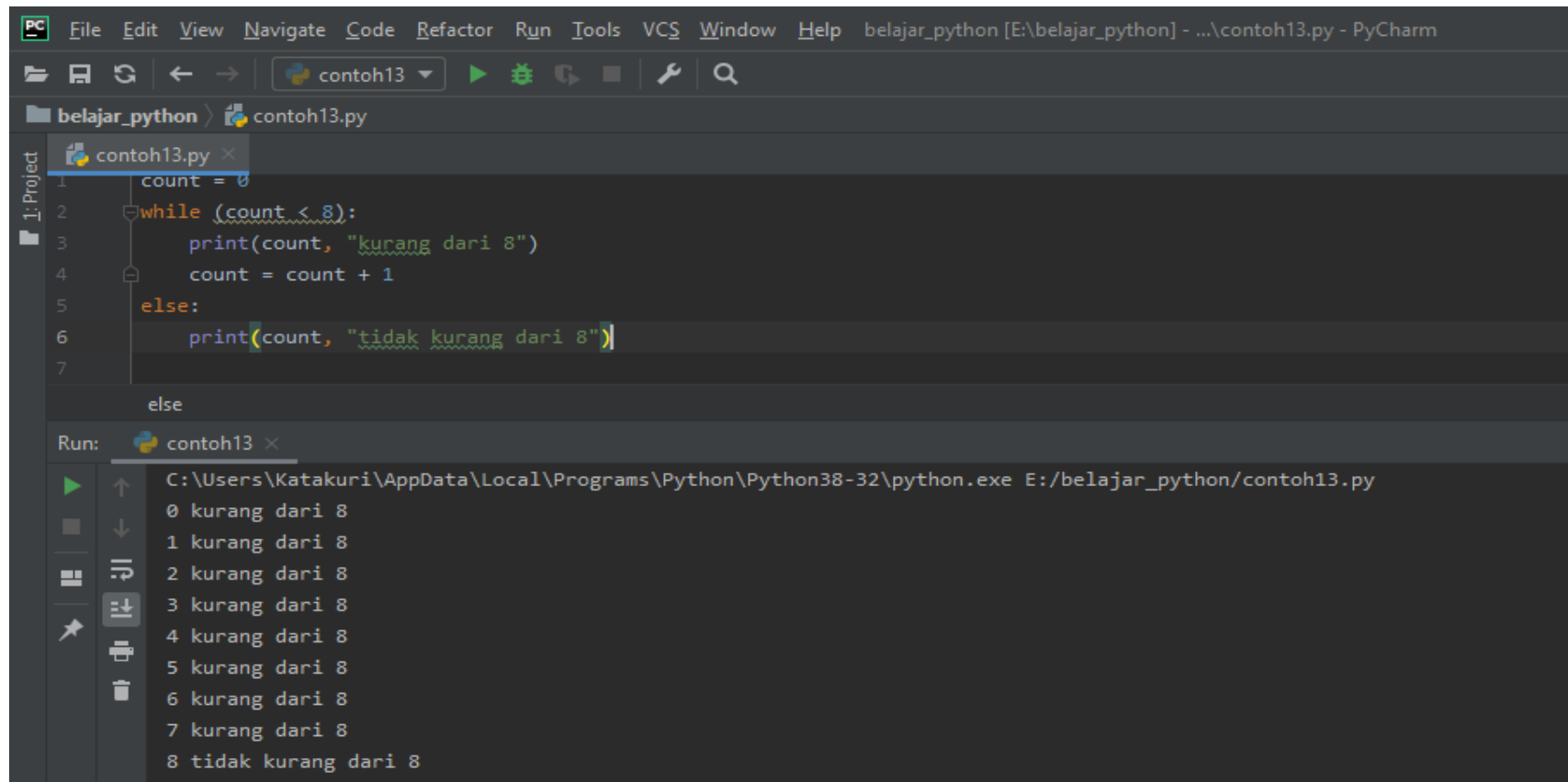
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh12.py

Huruf sekarang: P
Huruf sekarang: y
Huruf sekarang: t
Huruf sekarang: h
Huruf sekarang: o
Huruf sekarang: n
Huruf sekarang: P
Huruf sekarang: r
Huruf sekarang: o
Huruf sekarang: r
Huruf sekarang: a
Huruf sekarang: m
Huruf sekarang: m
Huruf sekarang: i
Huruf sekarang: n
Good bye

Perhatikan bahwa huruf **g** tidak pernah ditampilkan karena diabaikan karena kode **continue**.

While Else

Python mendukung penggunaan else sebagai pasangan dari while. Blok pernyataan else hanya akan dieksekusi bila kondisi while bernilai salah.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help belajar_python [E:\belajar_python] - ...\contoh13.py - PyCharm
contoh13
belajar_python > contoh13.py
1 count = 0
2 while (count < 8):
3     print(count, "kurang dari 8")
4     count = count + 1
5 else:
6     print(count, "tidak kurang dari 8")
7
else
Run: contoh13 x
C:\Users\Katakuri\AppData\Local\Programs\Python\Python38-32\python.exe E:/belajar_python/contoh13.py
0 kurang dari 8
1 kurang dari 8
2 kurang dari 8
3 kurang dari 8
4 kurang dari 8
5 kurang dari 8
6 kurang dari 8
7 kurang dari 8
8 tidak kurang dari 8
```

Latihan

Lakukan pengulangan input data sebanyak 2 kali dengan data dibawah ini :

Data Ke- <berulang>

Masukkan NIM anda : <Input Data Ke 1>

Masukkan Nilai UTS : <Input Data Ke 1>

Masukkan Nilai UAS : <Input Data Ke 1>

Nim anda adalah <outputnim1> nilai UTS anda <outpututs1> nilai UTS anda <outputuas1>

Jawaban

```
ulang=2
for i in range(ulang):
    print ("data Ke - " + str(i+1))
    nama=input("Masukkan Nim anda : ")
    uts=int(input("Masukkan Nilai UTS anda :"))
    uas=int(input("Masukkan Nilai UAS : "))
    print("Nim anda adalah %s nilai UTS anda %i nilai UTS anda %i" % (nama,uts,uas))
    print("-----\n")
```


Output

```
perulangan x
"E:\OneDrive - Bina Sarana Informatika\0 - KAPRODI SI\Buku Ajar\Tah
data Ke - 1
Masukkan Nim anda : 12101707
Masukkan Nilai UTS anda :90
Masukkan Nilai UAS : 90
Nim anda adalah 12101707 nilai UTS anda 90 nilai UTS anda 90
-----

data Ke - 2
Masukkan Nim anda : 12181918
Masukkan Nilai UTS anda :80
Masukkan Nilai UAS : 80
Nim anda adalah 12181918 nilai UTS anda 80 nilai UTS anda 80
-----

Process finished with exit code 0
|
```


Tugas 2

Sebuah perusahaan ayam goreng dengan nama “**GEROBAK FRIED CHICKEN**” yang telah lumayan banyak pelanggannya, ingin dibantu dibuatkan program untuk membantu kelancaran usahanya.

“**GEROBAK FRIED CHICKEN**” mempunyai daftar harga ayam sebagai berikut :

Kode	JenisPotong	Harga

D	Dada	Rp. 2500
P	Paha	Rp. 2000
S	Sayap	Rp. 1500

Buatlah programnya dengan ketentuan:

- Setiap pembeli dikenakan pajak sebesar 10% dari pembayaran.
- Banyak Jenis, Jenis Potong dan Banyak Beli diinput.
- Tampilan yang diinginkan sebagai berikut:

Layar Masukkan

GEROBAK FRIED CHICKEN

Kode	JenisPotong	Harga
------	-------------	-------

D	Dada	Rp. 2500
P	Paha	Rp. 2000
S	Sayap	Rp. 1500

Banyak Jenis : ... <diinput>

Jenis Ke - ... <proses counter>

Kode Potong [D/P/S] : ... <diinput>

Banyak Potong : ... <diinput>

<<Terus berulang tergantung Banyak Jenis>>

Layar Keluaran

GEROBAK FIRED CHICHEN

No.	Jenis	Harga	Bayak	Jumlah
	Potong	Satuan	Beli	Harga

...	Rp
...	Rp

Jumlah Bayar Rp

Pajak 10% Rp

Total Bayar Rp