



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231050
Nama Lengkap	Samuel Natanael
Minggu ke / Materi	05 / Perulangan

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

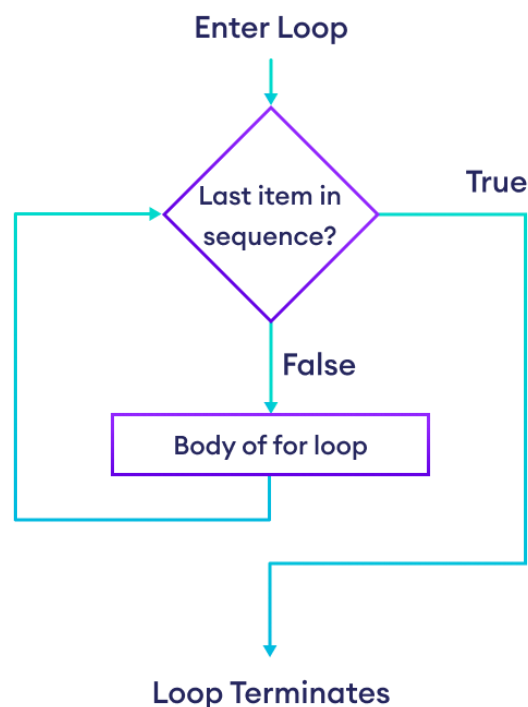
BAGIAN 1: MATERI MINGGU INI (40%)

Definisi Perulangan

Suatu program dapat diatur secara sekuensial, percabangan, perulangan, atau kombinasi dari ketiganya. Metode pengaturan ini dikenal sebagai struktur kontrol. Perulangan digunakan jika diperlukan oleh program untuk:

- Melakukan hal yang sama lebih dari 1 kali
- Melakukan sesuatu dengan langkah demi langkah, di mana setiap langkah sebenarnya memiliki langkah yang sama.
- mengakses sekumpulan data dalam struktur data seperti Daftar, Tuple, Queue, Stack dan beberapa struktur data lainnya.

Perulangan dapat digunakan dengan menggunakan **for**, **while** atau dengan cara **rekursif**. Pada pembelajaran ini, perulangan **for** dan **while** akan dibahas terlebih dahulu.



Gambar 5.1 Flowchart of Python for Loop (diambil dari <https://www.programiz.com/python-programming/for-loop>)

Bentuk Perulangan for

Bentuk perulangan dapat dinyatakan dalam bentuk **for** dan **while**. Perulangan **for** biasa digunakan pada kondisi-kondisi tertentu yaitu:

- Jumlah perulangan sudah diketahui sejak awal. Misalnya, sepuluh file teks akan dibaca datanya. Meskipun isi dalam setiap file teks berbeda, cara membaca file teks umumnya sama. Pembacaan dilakukan dari file pertama, kedua, dan seterusnya sampai file yang terakhir.
- Perulangan terjadi ketika operasi yang sama dilakukan pada rentang data atau nilai yang sama. Misalnya, untuk menemukan jumlah dari 100 bilangan pertama, Anda harus melakukan penjumlahan $1 + 2 + 3 + \dots + 100$. Ini menunjukkan bahwa itu dilakukan dalam rentang mulai dari 1 sampai 100.

Perulangan for yang menggunakan rentang tertentu akan lebih mudah dilakukan dengan menggunakan fungsi `range()`, cara penggunaannya adalah sebagai berikut:

- `Range(stop)`. Berarti perulangan akan stop tergantung limit yang diberikan. Misalnya, untuk menghasilkan rentang 0-5, digunakan `range(6)`. Karena dalam Bahasa percodingan angka dimulai dari 0.
- `Range(start, stop)`. Sama seperti `range(stop)`, tetapi ditambah start untuk menghasilkan rentang dari start sampai stop.
- `Range(start, stop, step)`. Digunakan untuk rentang start sampai stop dengan peningkatan step.

Berikut adalah contoh penerapan pengulangan for dan range untuk menuliskan angka 1-50

```
for i in range(1, 51):
```

```
    print(i)
```

Outputnya akan menghasilkan bilangan 1 sampai 50 menggunakan `range()`, dimulai dari 1(start) dan 51(stop), dengan step 1 (hal ini dikarenakan jika step tidak ditulis maka defaultnya menjadi 1). Kemudian nilai `i` akan naik secara berurutan sesuai dengan nilai yang dihasilkan dari fungsi `range()` saat variabel `i` digunakan sebagai counter. Jika counter tidak diperlukan, bentuk perulangan seperti ini dapat digunakan:

```
for _ in range(1, 51):
```

```
    print('Hallo')
```

Outputnya akan menuliskan “hallo” sebanyak 50 kali. Program tersebut tidak menggunakan nilai dari suatu counter

Step negative

Berikut adalah contoh penggunaan perulangan for menggunakan `range(start, stop, step)`

```
for i in range(50, 60, 3):  
    print(i)
```

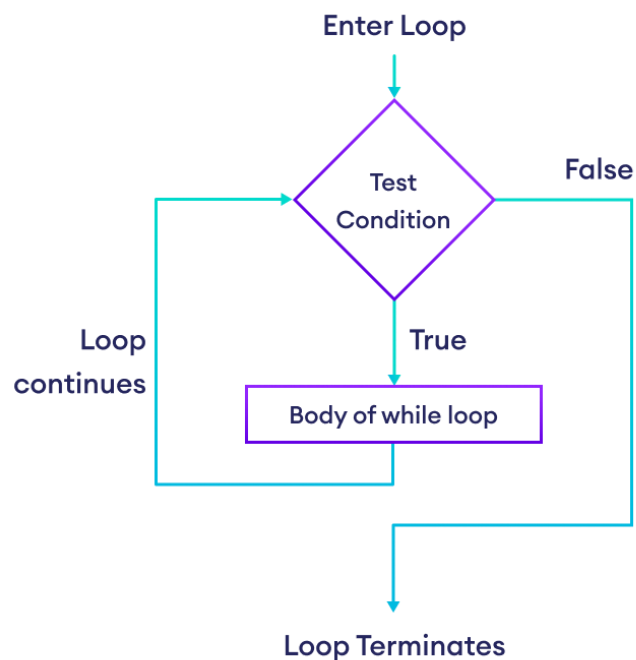
Perulangan tersebut berarti memiliki rentang 50-60, dengan Langkah 3. Maka outputnya adalah 50-53-56-59. Tetapi, terkadang ada beberapa kasus dimana kita harus mencari rentang yang sebaliknya seperti 60-50. Untuk menyelesaikan masalah tersebut, digunakanlah step negative. Fungsi range() dapat menggunakan step negative seperti contoh dibawah ini

```
for i in range(60, 50, -3):  
    print(i)
```

Output program tersebut sama seperti program yang sebelumnya tetapi akan mulai dari yang besar terlebih dahulu, 60-57-54-51.

Bentuk Perulangan while

Bentuk while hampir mirip dengan for, tetapi hal yang membedakan mereka berdua adalah penggunaannya. Perulangan for digunakan ketika jumlah perulangan diketahui, sementara perulangan while digunakan ketika jumlah perulangan belum diketahui.



Gambar 5.2 Flowchart of Python while Loop (diambil dari <https://www.programiz.com/python-programming/while-loop>)

```
14 x = 0
15 while x <= 5:
16     print("hai")
17     x += 1
18
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
PS D:\Ngampus\Semester 2\PrakAlPro> python			
hai			
hai			
hai			
hai			
hai			
hai			

Gambar 5.3 Bentuk Umum While Dalam Python

Pada gambar diatas, perulangan while terdiri dari kondisi dan operation. X adalah 0, kondisi “while x <= 5” yang berarti perulangan akan berjalan jika x masih kurang atau sama dengan 5, dan operation “print “hai”” akan menuliskan hai setiap kali kondisi itu berlaku.

Berikut adalah contoh penggunaan perulangan while dalam memeriksa apakah harga suatu barang melebihi budget:

```
harga = 0
melebihi_budget = False
while melebihi_budget == False:
    harga = int(input("Masukkan harga barang: "))
    if harga >= 100000:
        melebihi_budget = True
print(harga, "melebihi budget")
```

output:

```
Masukkan harga barang: 90000
Masukkan harga barang: 80000
Masukkan harga barang: 100000
100000 melebihi budget
```

Dalam kode diatas, pengguna akan diminta untuk memasukan harga barang tertentu yang akan diperiksa oleh kode apakah harga tersebut melebihi budget atau tidak. Pada input pertama pengguna memasukan 90000 yang kurang dari 100000, jadi loopnya terus berjalan sehingga mendapatkan input yang sama atau lebih dari 100000.

Penggunaan Break dan Continue

Ada dua alat yang dapat digunakan untuk mengontrol perulangan: break, yang biasanya digunakan untuk menghentikan perulangan, dan continue yang biasanya digunakan untuk melanjutkan perulangan ke iterasi berikutnya.

Kode dibawah ini menggunakan perulangan for untuk mencetak angka tergantung batas bawah dan atas.

```
x = int(input("Batas Bawah: "))
y = int(input("Batas Atas: "))
for i in range(x, y+1):
    print(i)
```

Jika ingin perulangan seperti diatas, tetapi dihentikan ketika ada yang bisa dibagi 9, maka digunakanlah break. Berikut adalah contoh penggunaan break dalam perulangan Python:

```
x = int(input("Batas Bawah: "))
y = int(input("Batas Atas: "))
for i in range(x, y+1):
    if i % 9 == 0:
        break
    else:
        print(i)
```

output:

```
Batas Bawah: 10
Batas Atas: 20
10
11
12
13
14
15
16
17
```

Program akan meminta input untuk dijadikan sebagai batasan bawah dan atas, lalu akan mencetak angka diantara rentang tersebut, tetapi akan berhenti ketika ada bilangan yang bisa dibagi 9. Dalam kasus ini, perulangan berhenti di angka 17 karena angka selanjutnya 18 yang bisa dibagi 9.

Perbedaan break dan continue adalah break digunakan untuk memberhentikan sebuah perulangan ketika kondisi tertentu terpenuhi, dan continue melakukan hal yang terbalik yaitu melanjutkan perulangan ketika suatu kondisi terpenuhi, tetapi akan langsung melompat ke perulangan berikutnya.

Berikut adalah contoh penggunaan continue:

```
x = int(input("Batas Bawah: "))
y = int(input("Batas Atas: "))
for i in range(x, y+1):
    if i % 2 == 0:
        continue
    else:
        print(i)
```

output:

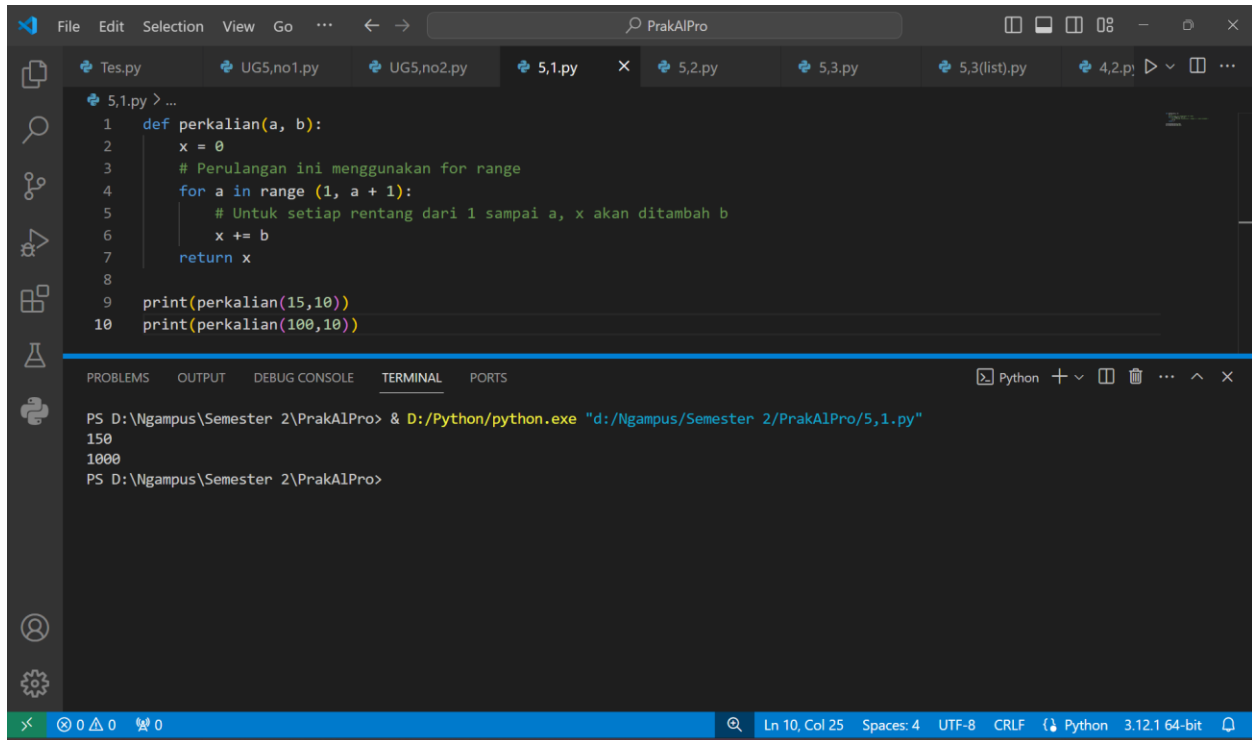
```
Batas Bawah: 1
Batas Atas: 20
1
3
5
7
9
11
13
15
17
19
```

Program diatas memiliki system yang sama seperti diatas yaitu meminta batasan bahwa dan atas, lalu akan memprint bilangan diantara batasan it, tetapi akan melompatkan bilangan yang genap.

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 5,1



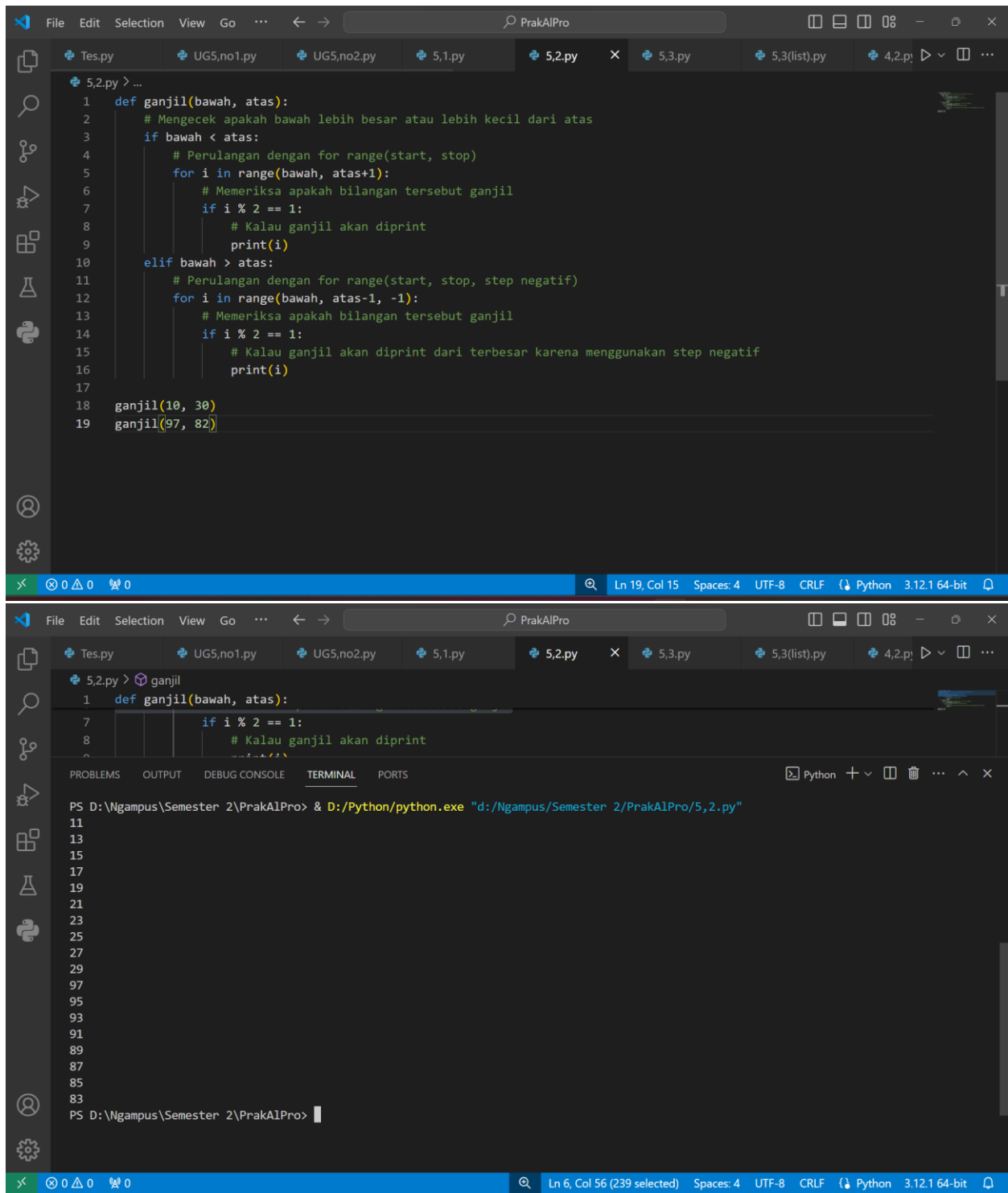
The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The editor is open to a file named `5,1.py`. The code defines a function `perkalian(a, b)` that calculates the product of `a` and `b` using a loop. The function is then called with `perkalian(15, 10)` and `perkalian(100, 10)`. The terminal shows the command to run the script and the resulting output: `150` and `1000`.

```
def perkalian(a, b):
    x = 0
    # Perulangan ini menggunakan for range
    for a in range(1, a + 1):
        # Untuk setiap rentang dari 1 sampai a, x akan ditambah b
        x += b
    return x

print(perkalian(15, 10))
print(perkalian(100, 10))
```

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/5,1.py"
150
1000
PS D:\Ngampus\Semester 2\PrakAlPro>
```


SOAL 5,2



The image shows a Python IDE with two windows. The top window displays a Python script named `5,2.py` with the following code:

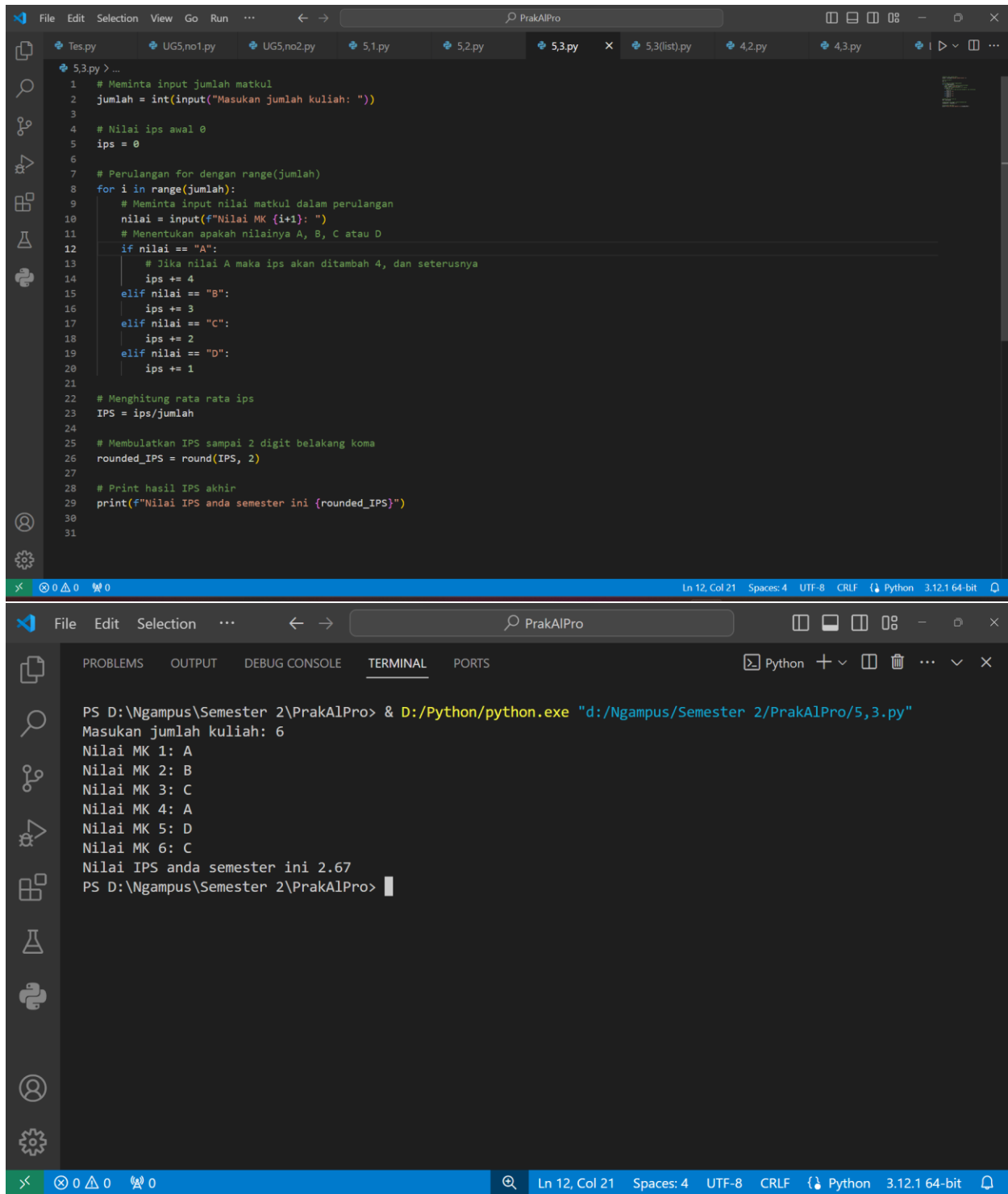
```
1 def ganjil(bawah, atas):
2     # Mengecek apakah bawah lebih besar atau lebih kecil dari atas
3     if bawah < atas:
4         # Perulangan dengan for range(start, stop)
5         for i in range(bawah, atas+1):
6             # Memeriksa apakah bilangan tersebut ganjil
7             if i % 2 == 1:
8                 # Kalau ganjil akan diprint
9                 print(i)
10    elif bawah > atas:
11        # Perulangan dengan for range(start, stop, step negatif)
12        for i in range(bawah, atas-1, -1):
13            # Memeriksa apakah bilangan tersebut ganjil
14            if i % 2 == 1:
15                # Kalau ganjil akan diprint dari terbesar karena menggunakan step negatif
16                print(i)
17
18    ganjil(10, 30)
19    ganjil(97, 82)
```

The bottom window shows the execution of the script. The command prompt displays the command:

```
PS D:\Ngampus\Semester 2\PrakAIPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAIPro/5,2.py"
```

The output shows the numbers 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 97, 95, 93, 91, 89, 87, 85, and 83, which are the odd numbers in the ranges [10, 30] and [97, 82].

SOAL 5,3



The image shows a VS Code editor window with a Python script named `5,3.py` and a terminal window below it. The script calculates the average grade (IPS) based on six input grades (MK 1 to MK 6). The terminal shows the execution of the script, where the user enters 6 for the number of grades and then provides the grades: A, B, C, A, D, C. The final output is "Nilai IPS anda semester ini 2.67".

```
1 # Meminta input jumlah matkul
2 jumlah = int(input("Masukan jumlah kuliah: "))
3
4 # Nilai ips awal 0
5 ips = 0
6
7 # Perulangan for dengan range(jumlah)
8 for i in range(jumlah):
9     # Meminta input nilai matkul dalam perulangan
10    nilai = input(f"Nilai MK {i+1}: ")
11    # Menentukan apakah nilainya A, B, C atau D
12    if nilai == "A":
13        # Jika nilai A maka ips akan ditambah 4, dan seterusnya
14        ips += 4
15    elif nilai == "B":
16        ips += 3
17    elif nilai == "C":
18        ips += 2
19    elif nilai == "D":
20        ips += 1
21
22 # Menghitung rata rata ips
23 IPS = ips/jumlah
24
25 # Membulatkan IPS sampai 2 digit belakang koma
26 rounded_IPS = round(IPS, 2)
27
28 # Print hasil IPS akhir
29 print(f"Nilai IPS anda semester ini {rounded_IPS}")
30
31
```

```
PS D:\Ngampus\Semester 2\PrakAIPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAIPro/5,3.py"
Masukan jumlah kuliah: 6
Nilai MK 1: A
Nilai MK 2: B
Nilai MK 3: C
Nilai MK 4: A
Nilai MK 5: D
Nilai MK 6: C
Nilai IPS anda semester ini 2.67
PS D:\Ngampus\Semester 2\PrakAIPro>
```

Github : <https://github.com/SamuelN1508/PrakAIPro5.git>