



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231052
Nama Lengkap	GABRIEL SACHIO ATMADJAJA
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%)

Sumber Materi: Modul Pratikum Algoritma dan Pemrograman, kegiatan pratikum

MATERI 5

Struktur kontrol adalah pengaturan jalannya suatu program (cth: sekuensial, percabangan, perulangan, atau ketiganya). Perulangan digunakan untuk mengulangi hal yang sama beberapa kali, mengakses sekumpulan data, dll.

Perulangan **for** digunakan untuk jumlah perulangan yang sudah diketahui. Pada perulangan for, terdapat operator in, serta terdapat fungsi range(). Berikut kegunaannya:

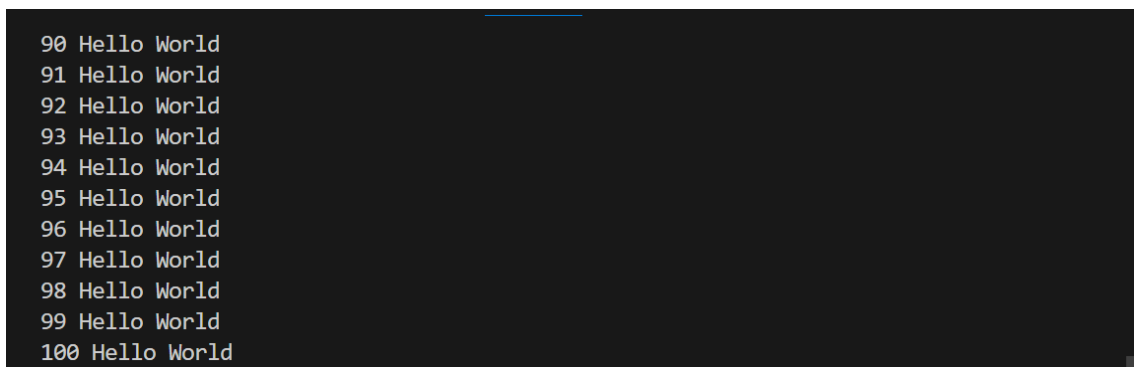
- range(stop) -> menghasilkan rentang dari **0** sampai **stop - 1**
- range(start, stop, [step]) -> menghasilkan rentang dari start sampai stop -1 dengan peningkatan jumlah step. Jika step tidak diisi, defaultnya + 1

Berikut contoh kode program perulangan for:



```
perulangan.py > ...
1  for i in range(1, 101):
2      print(i, "Hello World")
```

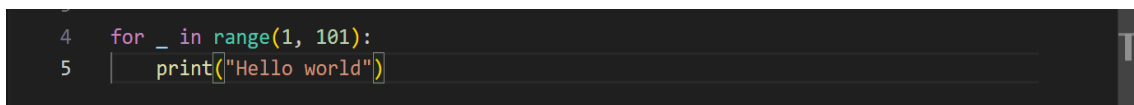
Gambar 1.1: cth kegunaan fungsi range



```
90 Hello World
91 Hello World
92 Hello World
93 Hello World
94 Hello World
95 Hello World
96 Hello World
97 Hello World
98 Hello World
99 Hello World
100 Hello World
```

Gambar 1.2: Output dari kode di gambar 1.1

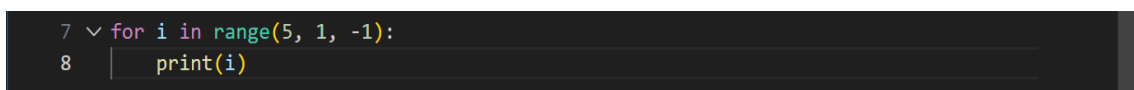
Variable i digunakan sebagai counter, di mana nilai i akan naik berurutan sesuai dengan nilai yang dihasilkan. Bisa juga tidak menggunakan variable i:



```
4  for _ in range(1, 101):
5      print("Hello world")
```

Gambar 1.3: Contoh bila tidak menggunakan variable i

Lalu, step dalam perulangan bisa negatif. Berikut contohnya:



```
7  for i in range(5, 1, -1):
8      print(i)
```

Gambar 1.4: Cth perulangan dengan step negatif

```
PS D:\Sachio\Kampus\PrakAlPro\Pert5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python
.exe d:/Sachio/Kampus/PrakAlPro/Pert5/perulangan.py
5
4
3
2
```

Gambar 1.5: Output dari kode di gambar 1.4

Perulangan **while** digunakan pada kondisi di mana jumlah perulangan belum diketahui. Berikut contoh kode program perulangan while:

```
while.py > ...
1  bilangan = 0
2  genap = False
3  while genap == False:
4      bilangan = int(input("Masukkan bilangan genap: "))
5      if bilangan % 2 == 0:
6          genap = True
7  print(bilangan, "yang Anda masukkan adalah bilangan genap")
```

Gambar 1.6: Contoh penggunaan perulangan while

Keterangan gambar 1.6:

- baris 1 -> deklarasi variable saja supaya bisa digunakan di dalam while
- baris 2 -> kondisi awal untuk digunakan di baris 3
- baris 3 -> akan melakukan perulangan terus menerus jika kondisi genap == False
- baris 4 -> meminta input pengguna yang nantinya akan di cek apakah itu genap atau tidak
- baris 5 -> percabangan yang digunakan untuk mengecek apakah suatu bilangan termasuk genap atau tidak
- baris 6 -> jika kondisi percabangan di baris 5 terpenuhi, maka ubah genap == True supaya perulangan berhenti
- baris 7 -> menampilkan bilangan yang genap

Berikut outputnya:

```
PS D:\Sachio\Kampus\PrakAlPro\Pert5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAl
Pro/Pert5/while.py
Masukkan bilangan genap: 21
Masukkan bilangan genap: 22
22 yang Anda masukkan adalah bilangan genap
```

Gambar 1.7: Output dari kode di gambar 1.6

Perulangan juga dapat di kontrol dengan **continue** atau **break**. Break digunakan untuk menghentikan perulangan, sedangkan continue untuk melanjutkan perulangan ke iterasi berikutnya. Berikut contohnya:

```

break&continue.py > ...
1 # break = menghentikan perulangan
2 for i in range(1, 11):
3     if i == 5:
4         break
5     else:
6         print(i)
7 print("selesai")

```

Gambar 1.8: Contoh penggunaan break

Maksud dari gambar 1.8 adalah ketika mencapai perulangan yang ke 5, maka perulangan akan dihentikan atau keluar dari perulangan tersebut dan menampilkan selesai.

```

PS D:\Sachio\Kampus\PrakAIPro\Pert5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe "d:/Sachio/Kampus/PrakAIPro/Pert5/break&continue.py"
1
2
3
4
selesai

```

Gambar 1.9: Output dari kode program break

```

9 # continue = melanjutkan iterasi perulangan berikutnya
10 for i in range(1, 11):
11     if i == 4:
12         continue
13     else:
14         print(i)
15 print("selesai")

```

Gambar 1.10: Contoh penggunaan continue

```

PS D:\Sachio\Kampus\PrakAIPro\Pert5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe "d:/Sachio/Kampus/PrakAIPro/Pert5/break&continue.py"
1
2
3
5
6
7
8
9
10
selesai


```

Gambar 1.11: Output dari kode program continue

Hal penting lainnya yaitu sebagian besar perulangan for dapat dikonversi menjadi perulangan while. Syaratnya:

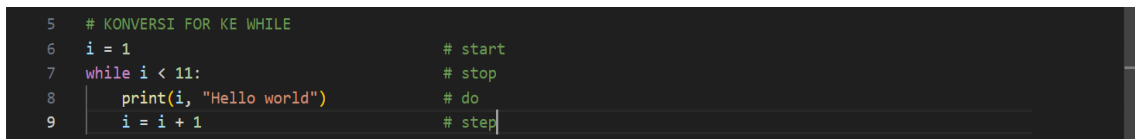
- Harus ada nilai awal
- Harus ada nilai akhir
- Harus ada Langkah

Berikut contohnya:

A screenshot of a code editor showing a Python script. The script is titled 'konversi.py' and contains a for loop that iterates from 1 to 10, printing 'Hello world' for each iteration. The code is as follows:

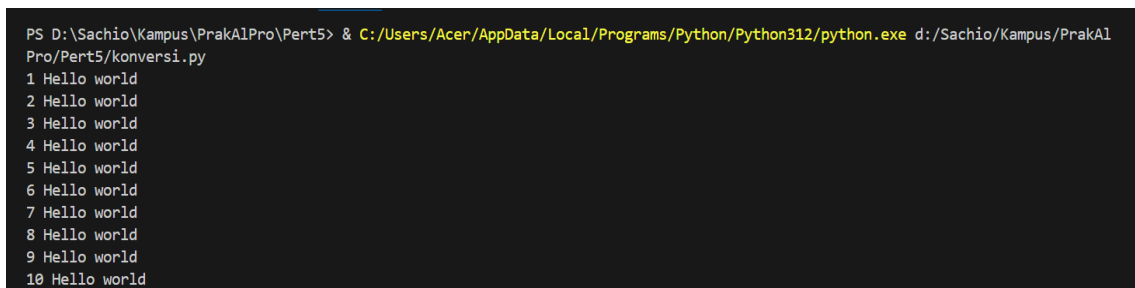
```
1 # FOR
2 for i in range(1, 11):
3     print(i, "Hello world")
```

Gambar 1.12: Bentuk for

A screenshot of a code editor showing a Python script that converts a for loop into a while loop. The script is titled 'KONVERSI FOR KE WHILE' and contains a while loop that iterates from 1 to 10, printing 'Hello world' for each iteration. The code is as follows:

```
5 # KONVERSI FOR KE WHILE
6 i = 1 # start
7 while i < 11: # stop
8     print(i, "Hello world") # do
9     i = i + 1 # step
```

Gambar 1.13: Konversi ke bentuk while

A screenshot of a terminal window showing the output of the Python scripts. The terminal prompt is 'PS D:\Sachio\Kampus\PrakAlPro\Pert5>'. The command executed is 'C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert5/konversi.py'. The output is ten lines of 'Hello world', each preceded by a number from 1 to 10.

```
PS D:\Sachio\Kampus\PrakAlPro\Pert5> C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert5/konversi.py
1 Hello world
2 Hello world
3 Hello world
4 Hello world
5 Hello world
6 Hello world
7 Hello world
8 Hello world
9 Hello world
10 Hello world
```

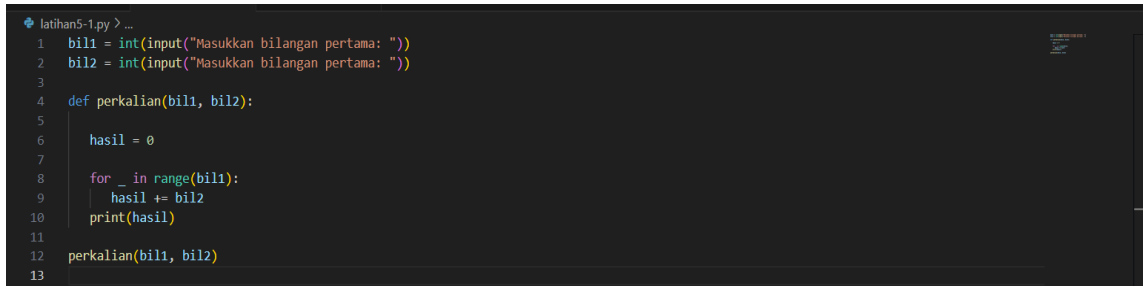
Gambar 1.14: Output dari gambar 1.12 dan gambar 1.13

BAGIAN 2: LATIHAN MANDIRI (60%)

Source Code: <https://github.com/gabrielsachioa/PrakAIPro5.git>

SOAL 1

Berikut kode programnya:



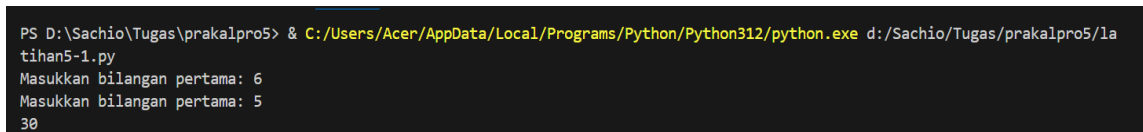
```
latihan5-1.py > ...
1  bil1 = int(input("Masukkan bilangan pertama: "))
2  bil2 = int(input("Masukkan bilangan pertama: "))
3
4  def perkalian(bil1, bil2):
5
6      hasil = 0
7
8      for _ in range(bil1):
9          hasil += bil2
10         print(hasil)
11
12     perkalian(bil1, bil2)
13
```

Gambar 2.1: Kode program perkalian dengan pertambahan

Keterangan gambar 2.1:

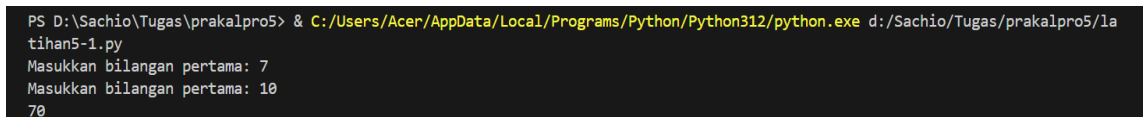
- Baris 1, 2 -> membuat variable yang meminta input bilangan dari user
- Baris 4 -> deklarasi fungsi perkalian dengan parameter bil1, bil2
- Baris 6 -> membuat variable hasil untuk menampung perkalian dengan pertambahan nantinya
- Baris 8 -> membuat perulangan for tanpa variable i dengan range bil1. Jadi, akan mengulangi perulangan sebanyak inputan pada bil1
- Baris 9 -> akan menghitung hasil + bil2, lalu kembali mengulangi perulangan sebanyak bil1
- Baris 10 -> setelah perulangan selesai, maka akan menampilkan hasilnya
- Baris 12 -> memanggil fungsi perkalian

Berikut outputnya:



```
PS D:\Sachio\Tugas\prakalpro5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Tugas/prakalpro5/latihan5-1.py
Masukkan bilangan pertama: 6
Masukkan bilangan pertama: 5
30
```

Gambar 2.2: Contoh output 1



```
PS D:\Sachio\Tugas\prakalpro5> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Tugas/prakalpro5/latihan5-1.py
Masukkan bilangan pertama: 7
Masukkan bilangan pertama: 10
70
```

Gambar 2.3: Contoh output 2

SOAL 2

Berikut kode programnya:

```
latihan5-2.py > ...
1 # CARA 1
2 batas_bawah = int(input("Masukkan batas bawah: "))
3 batas_atas = int(input("Masukkan batas atas: "))
4
5 def ganjil(batas_bawah, batas_atas):
6
7     langkah = 1 if batas_bawah < batas_atas else -1
8
9     for i in range(batas_bawah, batas_atas, langkah):
10         if i % 2 == 1:
11             print(i)
12
13 ganjil(batas_bawah, batas_atas)
14
```

Gambar 2.4: Kode program bilangan ganjil

Keterangan gambar 2.4:

- Baris 1 -> comment
- Baris 2,3 -> membuat variable batas_bawah dan batas_atas yang masing-masing meminta input bilangan
- Baris 5 -> membuat fungsi ganjil dengan parameter batas_bawah dan batas_atas
- Baris 7 -> ternary operator, membuat variable langkah = 1 jika batas_bawah kurang dari batas_atas, selain dari itu langkahnya -1
- Baris 9 -> membuat perulangan for beserta variable i dengan range. Dimulai dari batas_bawah, lalu sampai batas_atas, dan stepnya sesuai dengan kondisi di variable langkah tadi.
- Baris 10 -> percabangan yang memeriksa apakah bilangan tersebut ganjil atau tidak
- Baris 11 -> jika kondisi percabangan True, maka akan print(i), yang akan menampilkan bilangan ganjil
- Baris 13 -> memanggil fungsi ganjil

Output:

```
Masukkan batas bawah: 10
Masukkan batas atas: 30
11
13
15
17
19
21
23
25
27
29
```

Gambar 2.5: Contoh output 1

```
Masukkan batas bawah: 97
Masukkan batas atas: 82
97
95
93
91
89
87
85
83
```

Gambar 2.6: Contoh Output 2

SOAL 3

Berikut kode programnya:

```
latihan5-3.py > ...
1  jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))
2
3  def program_ips(jumlah_matkul):
4      jumlah_sks = 3
5      total_poin = 0
6
7      for i in range(jumlah_matkul):
8          huruf_nilai = input(f"Nilai MK {i + 1}: ")
9
10         if huruf_nilai == "A":
11             poin = 4
12         elif huruf_nilai == "B":
13             poin = 3
14         elif huruf_nilai == "C":
15             poin = 2
16         elif huruf_nilai == "D":
17             poin = 1
18         else:
19             poin = 0
20
21         total_poin += poin
22
23     hasil_ips = (jumlah_sks * total_poin) / (jumlah_sks * jumlah_matkul)
24     return ('%.2f' % hasil_ips)
25
26
27 nilai_ips = program_ips(jumlah_matkul)
28 print(f"Nilai IPS Anda semester ini {nilai_ips}")
29
```

Gambar 2.7: Kode program nilai IPS

Keterangan gambar 2.7:

- Baris 1 -> variable jumlah_matkul meminta input integer
- Baris 3 -> membuat fungsi program_ips dengan parameter jumlah_matkul
- Baris 4 -> membuat variable jumlah_sks untuk jumlah sks
- Baris 5 -> membuat variable total_poin untuk menghitung total poin nantinya
- Baris 7 -> membuat perulangan for menggunakan variable i dengan range jumlah_matkul (akan mengulang sebanyak input dari jumlah_matkul)
- Baris 8 -> membuat variable huruf_nilai yang meminta input nilainya
- Baris 10 sampai 19 -> Membuat percabangan yang mengecek apakah input dari variable huruf_nilai sesuai dengan kondisi di percabangan. Jika, ya maka poin sesuai dengan kondisi tersebut. Alasan menggunakan else karena ada kemungkinan user meng-inputkan huruf nilainya dengan huruf kecil atau input-an lain
- Baris 21 -> setelah mengecek, variable total_poin akan menyimpan hasil perhitungan dari akumulasi poin
- Baris 23 -> membuat variable hasil_ips untuk menghitung IPS
- Baris 24 -> mengembalikan nilai dari variable hasil_ips, dengan 2 digit di belakang koma
- Baris 27 -> memanggil fungsi program_ips di dalam variable nilai_ips
- Baris 28 -> menampilkan output nilai_ips

Output:

```
Masukkan jumlah mata 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
```

Gambar 2.8: Contoh output dari program IPS