

LAPORAN PRAKTIKUM

PEMROGRAMAN 1

MODUL 4



Oleh:

NAMA : AFAD FATH MUSYAROF HALIM

NIM : 2211104030

KELAS : S1SE-06-A

PRODI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2023

I. DASAR TEORI.

Dalam pembuatan program, terkadang diperlukan untuk melakukan pengulangan suatu aksi misalnya untuk melakukan perhitungan berulang dengan menggunakan formula yang sama. Contohnya saat ingin menulis angka dari 1 sampai 10, maka tidak perlu menulis kode sebanyak 10 kali namun hanya perlu menempatkannya dalam satu statement pengulangan sehingga akan menjadi lebih efisien.

Contoh apabila menggunakan metode konvensional :

```
1 print("Ini angka ke 1")
2 print("Ini angka ke 2")
3 print("Ini angka ke 3")
4 print("Ini angka ke 4")
5 print("Ini angka ke 5")
6 print("Ini angka ke 6")
7 print("Ini angka ke 7")
8 print("Ini angka ke 8")
9 print("Ini angka ke 9")
10 print("Ini angka ke 10")
```

Output



```
Ini angka ke 1
Ini angka ke 2
Ini angka ke 3
Ini angka ke 4
Ini angka ke 5
Ini angka ke 6
Ini angka ke 7
Ini angka ke 8
Ini angka ke 9
Ini angka ke 10
```

Pada Kode di atas kurang efisien dikarenakan perlu menulis ulang terus menerus sintak yang diperlukan.

Untuk membuatnya lebih efisien diperlukan metode pengulangan, pengulangan adalah suatu kondisi dimana suatu instruksi akan terus dijalankan secara berulang sampai kondisi tertentu terpenuhi. Perbedaan pengulangan dengan percabangan yaitu pengulangan akan terus mengulang baris kode sampai kondisi terpenuhi sedangkan percabangan akan menjalankan kode selama kondisi terpenuhi.

A. For Loop

For Dalam Python lebih dikenal dengan foreach. Digunakan untuk menjalankan kode yang banyaknya sudah pasti atau telah diketahui sebelum kode dijalankan, dalam foreach diperlukan nilai awal pengulangan dan kondisi untuk menghentikan pengulangan.

Bentuk umumnya seperti di bawah ini :



Dalam python, perulangan For memiliki 3 bentuk dasar yaitu :

1. Range (max)

Contoh :

```
1 for i in range(11):
2     print("Ini angka ke", i)
```

Untuk for jenis ini, nilai variable `i` akan menghitung dimulai dari 0 sampai pada yang tertera pada '`range()`', dimana setiap perulangan akan bertambah 1.

Output :

```
Ini angka ke 0
Ini angka ke 1
Ini angka ke 2
Ini angka ke 3
Ini angka ke 4
Ini angka ke 5
Ini angka ke 6
Ini angka ke 7
Ini angka ke 8
Ini angka ke 9
Ini angka ke 10
```

2. Range(Min, Max)

Contoh :

```
1 for i in range(1, 11):
2     print("Ini angka ke", i)
```

Pada perulangan ini nilai awal dari `i` adalah 1 dengan nilai akhir adalah 11.

Nilai akhirnya selalu dikurangi 1 sehingga dalam output akan menghasilkan nilai dari 1 sampai 10.

Output :

```
Ini angka ke 1
Ini angka ke 2
Ini angka ke 3
Ini angka ke 4
Ini angka ke 5
Ini angka ke 6
Ini angka ke 7
Ini angka ke 8
Ini angka ke 9
Ini angka ke 10
```

3. Range (Min, Max, Step)

Contoh :

```
1 for i in range(1, 11, 2):
2     print("Ini angka ke", i)
```

Pada perulangan ini nilai awal adalah 1 dan nilai akhirnya 11. Dalam kode tersebut ditambahkan step 2 yang artinya program perulangan for akan menghitung dari nilai awal dengan pertambahan 2 untuk setiap kode yang selesai dijalankan.

Output :

```
Ini angka ke 1
Ini angka ke 3
Ini angka ke 5
Ini angka ke 7
Ini angka ke 9
```

4. Range Terbalik

Perulangan for dapat di balik untuk iterasinya dengan menukar posisi minimal dan maksimal nilai.

Contoh :

```
1 for i in range(5, 0, -1):
2     print(f"Ini angka ke-{i}")
```

Output :

```
Ini angka ke-5
Ini angka ke-4
Ini angka ke-3
Ini angka ke-2
Ini angka ke-1
```

B. WHILE LOOP

While Loop akan memeriksa kondisi di bagian awal dan baris kode akan dijalankan apabila kondisi bernilai true.

Bentuk umumnya seperti di bawah ini :



Sebagai perbandingan apabila menggunakan program dari For Loop :

```
1 a = 0
2 b = 5
3 while a < b:
4     print(f"ini angka ke-{a}")
5     a += 1
```

Output :

```
ini angka ke-0
ini angka ke-1
ini angka ke-2
ini angka ke-3
ini angka ke-4
```

Apabila iterasinya dibalik :

```
1 a = 10
2 b = 0
3 while a > b:
4     print(f"ini angka ke-{a}")
5     a -= 1
```

Output :

```
ini angka ke-5
ini angka ke-4
ini angka ke-3
ini angka ke-2
ini angka ke-1
```

Terdapat 2 fungsi dalam perulangan python, yaitu **break** dan **continue**

1. Break

Digunakan untuk menghentikan paksa suatu perulangan dan mengakhiri perulangan tersebut

Contoh :

```
1 for i in range(1, 6):
2     if i == 3:
3         print("Bilangan ke-3 ditemukan")
4         print("Program dihentikan")
5         break
6
7     print(f"Bilangan ke-{i}")
```

Output :

```
Bilangan ke-1
Bilangan ke-2
Bilangan ke-3 ditemukan
Program dihentikan
```

2. Continue

Digunakan untuk melewati perulangan dengan iterasi saat ini dan dilanjutkan ke iterasi berikutnya.

Contoh :

```
1 for i in range(1, 6):
2     if i == 3:
3         print("Bilangan ke-3 ditemukan")
4         print("Program dilanjutkan")
5         continue
6
7     print(f"Bilangan ke-{i}")
```

Output :

```
Bilangan ke-1
Bilangan ke-2
Bilangan ke-3 ditemukan
Program dilanjutkan
Bilangan ke-4
Bilangan ke-5
```

II. GUIDED

A. Login Sederhana

1. Source Code

```
1  counter = 0
2  User = "afad"
3  Pass = "1234"
4
5  while counter <= 3:
6      username = input("Masukkan username: ")
7      password = input("Masukkan password: ")
8
9      if username == User and password == Pass:
10         print("Login Berhasil")
11         break
12     else:
13         print("Login Gagal")
14         counter += 1
15
16     if counter == 3:
17         print("Akun anda diblokir")
18         break
```

2. Output

```
Masukkan username: afad
Masukkan password: 1
Login Gagal
Masukkan username: afad
Masukkan password: 12
Login Gagal
Masukkan username: afad
Masukkan password: 123
Login Gagal
Akun anda diblokir
```

```
Masukkan username: afad
Masukkan password: 1234
Login Berhasil
```

Dalam program ini, jika username dan password tidak sesuai sebanyak 3 kali maka login gagal dan akun akan diblokir. Apabila username dan password sesuai maka login berhasil.

B. Mencari Faktor Persekutuan Terbesar 2 Bilangan

1. Source Code

```
1  def hitung_FPB (x, y):
2      if x > y:
3          Kecil = y
4      else:
5          Kecil = x
6
7      for i in range (1, Kecil+1):
8          if ((x % i == 0) and (y % i == 0)):
9              fpb = i
10
11     return fpb
12
13     num1 = int(input("Masukkan angka pertama: "))
14     num2 = int(input("Masukkan angka kedua: "))
15
16     print(f"FPB dari {num1} dan {num2} adalah {hitung_FPB(num1, num2)}")
```

2. Output

```
Masukkan angka pertama: 25
Masukkan angka kedua: 45
FPB dari 25 dan 45 adalah 5
```

C. Mencari Bilangan Genap

1. Source Code :

```
1  bil = int(input("Masukkan bilangan maksimal: "))
2
3  for i in range(0, bil, 2):
4      print(f"bilangan genap: {i}")
```

2. Output :

```
Masukkan bilangan maksimal: 20
bilangan genap: 0
bilangan genap: 2
bilangan genap: 4
bilangan genap: 6
bilangan genap: 8
bilangan genap: 10
bilangan genap: 12
bilangan genap: 14
bilangan genap: 16
bilangan genap: 18
```

Program ini akan membuat daftar bilangan genap dengan nilai maksimal yang telah di input

III. UNGUIDED

A. Task 1 – Menghitung total nilai yang telah di input

1. Source Code

```
1  # Menghitung total nilai yang telah di input
2  bilangan = int(input("Masukkan bilangan: "))
3  total = 0
4
5  print(f"Total Bilangan = ", end="")
6
7  for i in range(bilangan, 0, -1):
8      total = total + i
9      if i == 1:
10         print(f"{i} = {total}")
11         break
12     print(f"{i} + ", end="")
```

2. Output

```
Meet 04 - Perulangan\Task>py Task1.py
Masukkan bilangan: 30
Total Bilangan = 30 + 29 + 28 + 27 + 26 + 25 + 24
+ 23 + 22 + 21 + 20 + 19 + 18 + 17 + 16 + 15 + 14
+ 13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 +
2 + 1 = 465
```

B. Task 2 – Menghitung Hasil Pangkat Bilangan

1. Source Code

```
1  # Menghitung hasil pangkat bilangan
2  bilangan = int(input("Masukkan bilangan: "))
3  pangkat  = int(input("Masukkan pencacah: "))
4
5  hasil = 1
6
7  for i in range(pangkat):
8      hasil *= bilangan
9
10 print(f"Hasil {bilangan} pangkat {pangkat} = {hasil}")
```

2. Output

```
Meet 04 - Perulangan\Task>py Task2.py
Masukkan bilangan: 52
Masukkan pencacah: 2
Hasil 52 pangkat 2 = 2704
```

C. Task 3 – Mencari Kelipatan Persekutuan Terkecil

1. Source Code

```
1  # Mencari KPK dari 2 bilangan
2  bilangan1 = int(input("Masukkan bilangan pertama : "))
3  bilangan2 = int(input("Masukkan bilangan kedua   : "))
4
5  a = bilangan1
6  b = bilangan2
7
8  while a != b:
9      if a < b:
10         a += bilangan1
11     else:
12         b += bilangan2
13
14 print(f"KPK dari {bilangan1} dan {bilangan2} adalah {a}")
```

2. Output

```
Meet 04 - Perulangan\Task>py Task3.py
Masukkan bilangan pertama : 22
Masukkan bilangan kedua   : 30
KPK dari 22 dan 30 adalah 330
```