

**LAPORAN PRAKTIKUM**

**PEMROGRAMAN 1**

**MODUL 04**



Oleh:

Nama : Alfian Mutakim

Nim : 2211104024

Kelas : SE06A

**PRODI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2023**

## I. DASAR TEORI

Dalam pembuatan program, terkadang kita harus melakukan pengulangan suatu aksi misalnya untuk melakukan perhitungan berulang dengan menggunakan formula yang sama. Sebagai contoh, misalnya kita ingin membuat program yang dapat menampilkan sebuah teks “Saya sedang belajar python” sebanyak 5 kali, maka kita tidak perlu untuk menuliskan 5 buah statement melainkan kita hanya tinggal menempatkan satu buah statement ke dalam suatu struktur pengulangan. Dengan demikian program kita akan lebih efisien.

### 1. STRUKTUR FOR.

For pada python lebih dikenal sebagai foreach. Struktur for ini digunakan untuk menuliskan jenis perulangan yang banyaknya sudah pasti atau telah diketahui sebelumnya. Oleh karena itu, disini kita harus melakukan inisialisasi nilai untuk kondisi awal pengulangan dan juga harus menuliskan kondisi untuk mengentikan proses pengulangan.

Adapun bentuk umum dari pendefinisian struktur for untuk satu statement adalah sebagai berikut :



Pada praktikum ini kita akan belajar tiga jenis perulangan for, yaitu:

#### a. Range (max)

```
# for range max  
for i in range(10):  
    print("why")
```

Output:

```
why
why
why
why
why
why
why
why
why
why
why
```

Untuk `for` jenis pertama ini, kita masukan banyaknya perulangan yang ingin dilakukan ke dalam `range()`. Nilai variable `i` nantinya akan berubah, dimulai dari 0 hingga bilangan yang dimasukan ke `range()`, dan setiap perulangan bilangan tersebut akan dikurangi satu.

b. Range (min,max)

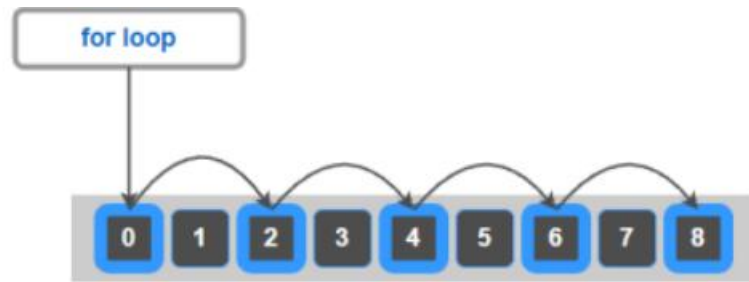
```
# for range min max
for i in range (1, 10):
    print (f"perlungan ke- {i}")
```

Output:

```
perlungan ke- 1
perlungan ke- 2
perlungan ke- 3
perlungan ke- 4
perlungan ke- 5
perlungan ke- 6
perlungan ke- 7
perlungan ke- 8
perlungan ke- 9
```

Pada perulangan ini counter i menyimpan nilai pada range min (nilai awal) adalah 1 dan max (batas akhir) adalah 10. Perlu diingat bahwa batas akhir selalu dikurang 1. Sehingga, hasil perulangan yang dihasilkan adalah 1 sampai 9.

Contoh diatas adalah perulangan for tanpa menggunakan STEP, berikut adalah ilustrasi jika perulangan for menggunakan STEP:



Dengan menggunakan STEP kita dapat mengatur berapa pertambahan di setiap perulangan yang dijalankan.

c. Range (min, max, step)

```
# for range min, max, step
for i in range (0,15,2):
    print(f"step ke-{i}")
```

Output:

```
step ke-0
step ke-2
step ke-4
step ke-6
step ke-8
step ke-10
step ke-12
step ke-14
```

Pada struktur perulangan for ini counter i menyimpan nilai pada range min (nilai awal) adalah 0 dan max (nilai akhir) adalah 20, lalu pada code tersebut kita menambahkan step 2 untuk setiap perulangan.

Contoh lainnya kita akan membuat sebuah perulangan menurun/decrement. Source code:

```
for i in range (20,0,-2):
    print(i)
```

20  
18  
16  
14  
12  
10  
8  
6  
4  
2

Pada struktur pengulangan while kondisi akan diperiksa di bagian awal. Hal ini tentu menyebabkan kemungkinan bahwa apabila ternyata kondisi yang kita definisikan tidak terpenuhi (bernilai salah), maka proses pengulangan pun tidak akan pernah dilakukan. Bentuk umum dari struktur while :



```
i = 0
while i < 7:
    print("Hallo Guys Dafid Di Sini")
    i += 1
```

[illegible]

Perulangan increment:

```
a = 1
b = 11
while a < b:
    print("step ke-", a)
    a += 1
```

Output:

```
step ke- 1
step ke- 2
step ke- 3
step ke- 4
step ke- 5
step ke- 6
step ke- 7
step ke- 8
step ke- 9
```

Perulangan decrement:

```
a = 10
b = 0
while a > b:
    print("step ke-", a)
    a -= 1
```

Output:

```
step ke- 10
step ke- 9
step ke- 8
step ke- 7
step ke- 6
step ke- 5
step ke- 4
step ke- 3
step ke- 2
step ke- 1
```

### 3. FUNGSI BREAK & CONTINUE

Pada python, kita bisa menginterupsi dan juga men-skip suatu iterasi pada perulangan. Terdapat 2 perintah yang bisa kita gunakan, yaitu: break untuk interupsi (memberhentikan paksa) sebuah perulangan dan continue untuk menskip ke iterasi selanjutnya.

Contoh break pada perulangan for:

```
#for range min max
for i in range (1, 10):
    print (f"perlungan ke- {i}")
    if i == 5:
        print("perlungan ke- ", i,"dihentikan")
        break
```

Output:

```
perlungan ke- 1
perlungan ke- 2
perlungan ke- 3
perlungan ke- 4
perlungan ke- 5
perlungan ke- 5 dihentikan
```

Contoh continue pada perulangan for:

```
for i in range (0,10):
    if (i == 5):
        continue
    print(i)
```

Output:

```
0
1
2
3
4
6
7
8
9
```

Contoh break pada perulangan while:

```
a = 0
while True:
    print("step ke-",a)
    a += 1
    if a == 5:
        print("step ke-",a ,"dihentikan")
        break
```

Output:

```
step ke- 0
step ke- 1
step ke- 2
step ke- 3
step ke- 4
step ke- 5 dihentikan
```

Contoh continue pada perulangan while:

```
angka=['1','2','3','4','5']
i = -1
while i < len(angka):
    i += 1
    if i % 2==0 and i>0:
        print("skip")
        continue
    print(angka[i])
```

Output :

```
1
2
skip
4
skip
```



## II. GUIDED

1. Buatlah sebuah sistem login sederhana, dengan ketentuan jika user memasukan password yang salah dalam waktu 3x maka tidak dapat melakukan login kembali.

```
print("=====Sistem Login Sederhana=====")
# inisialisasi variabel
password = "1234"
username = "alfian"
percobaan = 3

# pengguna diminta untuk memasukkan password
while percobaan > 0:
    input_username = input("Masukkan username: ")
    input_password = input("Masukkan password: ")
    if input_password and input_username == password and username:
        print("Selamat datang", username, "!")
        break
    else:
        percobaan -= 1
        print("Password salah. Anda memiliki", percobaan,
"percobaan lagi.")
else:
    print("Anda telah mencoba 3 kali. Mohon tunggu beberapa saat
untuk mencoba lagi.")
```

Berikut output yang akan ditampilkan ketika memasukkan username dan password dengan benar:

```
on3.11.exe "c:/Users/User/Documents/Prak
=====Sistem Login Sederhana=====
Masukkan username: alfian
Masukkan password: 1234
Selamat datang alfian !
```

Berikut output yang akan ditampilkan ketika memasukkan username dan password salah, dan kesalahan dilakukan sampai 3 kali:

```

=====Sistem Login Sederhana=====
Masukkan username: qalfian
Masukkan password: 1
Password salah. Anda memiliki 2 percobaan lagi.
Masukkan username: 1
Masukkan password: alf ian
Password salah. Anda memiliki 1 percobaan lagi.
Masukkan username: wkwkwk
Masukkan password: huhu
Password salah. Anda memiliki 0 percobaan lagi.
Anda telah mencoba 3 kali. Mohon tunggu beberapa saat untuk mencoba lagi.
PS C:\Users\User\Documents\Praktikum Pemograman1\LapModul4 SE06A 2211104024 Alfian Mutakim>

```

2. Buatlah sebuah program mencari bilangan genap, dimana inputan berasal dari user adalah berupa range maksimal.

```

# cari bilangan genap
print("Mencari Bilangan Genap")
# Meminta input range maksimal dari user
max_range = int(input("Masukkan bilangan maksimal: "))

# Menampilkan semua bilangan genap dari range 2 hingga max_range
for i in range(2, max_range+1, 2):
    print(i)

```

Maka output yang akan dihasilkan sebagai berikut:

```

Mencari Bilangan Genap
Masukkan bilangan maksimal: 15
2
4
6
8
10
12
14
PS C:\Users\User\Documents\Praktikum

```

3. Membuat program dengan menggunakan for untuk menentukan nilai factor dari persekutuan terbesar dari dua buah bilangan bulat. Sebagai contoh kita memasukkan dua buah bilangan bulat yaitu 12 dan 72, maka FPB dari kedua bilangan tersebut adalah 12. Sehingga, output yang dihasilkan adalah:

Berikut kode program:

```

print("====PROGRAM MENCARI FPB====")
#mendefinisikan fungsi
def hitung_fpb(x, y):
    #memilih bilangan yang paling kecil
    if x > y:
        terkecil = y

```

```

    else:
        terkecil = x
    for i in range(1, terkecil+1):
        if((x % i == 0) and (y % i ==0)):
            fpb = i
    return fpb

nilai = int(input("Masukkan bilangan pertama: "))
nilai2 = int(input("Masukkan bilangan kedua: "))
print("FPB= ",hitung_fpb(nilai, nilai2))

```

Berikut output yang dihasilkan:

```

====PROGRAM MENCARI FPB====
Masukkan bilangan pertama: 12
Masukkan bilangan kedua: 72
hasil FPB= 12
PS C:\Users\User\Documents\Praktikum Pe

```

### III. UNGUIDED

1. Buatlah sebuah program dengan statement perulangan dimana dapat menghitung total nilai dari suatu bilangan yang diinputkan.

Kode program yang digunakan sebagai berikut:

```
print("==== PROGRAM SEDERHANA MENGHITUNG JUMLAH TOTAL  
BILANGAN ====")  
a = int(input("Masukkan angka: "))  
total = 0  
print("Total Nilai = ", end="")  
for i in range(1, a+1):  
    total += i  
    if i == 1:  
        print(str(i), end="")  
    else:  
        print(" + "+str(i), end="")  
print(" = "+str(total))
```

Dari kode program di atas, maka output yang akan dihasilkan sebagai berikut:

```
==== PROGRAM SEDERHANA MENGHITUNG JUMLAH TOTAL BILANGAN ====  
Masukkan angka: 24  
Total Nilai = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20 + 21 + 22 + 23 + 24 = 300  
PS C:\Users\User\Documents\Praktikum Pemrograman1\LapModul4_SE06A_2211104024 Alfian Mutakim>
```

Dari kode program diatas dapat diartikan program tersebut merupakan sebuah program sederhana untuk menghitung jumlah total bilangan dari 1 hingga angka yang dimasukkan oleh pengguna.

Pada awal program, terdapat sebuah input untuk menginputkan nilai angka dari pengguna. Kemudian, program akan menginisialisasi variabel total dengan nilai 0.

Selanjutnya, program akan menggunakan perulangan while atau for untuk menjumlahkan semua bilangan dari 1 hingga angka yang dimasukkan oleh pengguna, dengan cara melakukan iterasi dari angka tersebut hingga 1. Pada setiap iterasi, program akan menambahkan nilai dari iterasi tersebut ke variabel total.

Selain itu, program juga akan menampilkan setiap nilai yang ditambahkan ke dalam variabel total pada setiap iterasi, sehingga output yang dihasilkan berupa deret bilangan yang ditambahkan.

Setelah perulangan selesai dilakukan, program akan menampilkan hasil total dari seluruh bilangan yang telah dijumlahkan. Output akhir dari program berupa sebuah kalimat yang menampilkan hasil total dari seluruh bilangan yang telah dijumlahkan, dan deret bilangan yang ditambahkan.

2. Buatlah sebuah program dengan statement perulangan, dimana dapat menghitung hasil pangkat suatu bilangan.

Untuk menghitung hasil pangkat dari suatu bilangan, maka berikut kode program yang digunakan:

```
print("==== PROGRAM SEDERHANA MENGHITUNG PANGKAT ====")
angka = int(input("Masukkan bilangan bulat: "))
pangkat = int(input("Masukkan pangkat: "))
x = 1
for i in range(pangkat):
    x *= angka
print("Hasil dari bilangan yang dimasukkan adalah: " + str(x))
```

Berikut output yang dihasilkan dari kode program diatas:

```
==== PROGRAM SEDERHANA MENGHITUNG PANGKAT ====
Masukkan bilangan bulat: 46
Masukkan pangkat: 2
Hasil dari bilangan yang dimasukkan adalah: 2116
PS C:\Users\User\Documents\Praktikum Pemrograman1\LapMod
```

Program di atas berfungsi untuk menghitung hasil pangkat dari sebuah bilangan yang dimasukkan oleh pengguna, dengan menggunakan pangkat yang juga dimasukkan oleh pengguna.

Program meminta input dari pengguna dengan menggunakan fungsi `input()`, dan mengonversi input tersebut menjadi bilangan bulat dengan menggunakan fungsi `int()`. Setelah itu, program menggunakan loop `for` untuk mengalikan bilangan tersebut dengan dirinya sendiri sebanyak pangkat yang dimasukkan, dan menyimpan hasilnya dalam variabel `x`. Kemudian, program mencetak hasil pangkat ke layar dengan menggunakan fungsi `print()`, dengan menambahkan string "Hasil dari bilangan yang dimasukkan adalah: " dan hasil pangkat yang telah dihitung.

Dengan demikian, fungsi program ini adalah untuk mempermudah pengguna dalam menghitung hasil pangkat bilangan dengan menggunakan metode iterasi sederhana, sehingga dapat membantu dalam pemecahan masalah matematis atau komputasi yang melibatkan perhitungan pangkat.

3. Buatlah sebuah program dengan statement perulangan untuk menentukan KPK dari dua buah bilangan bulat.

Untuk menghitung KPK suatu bilangan berikut kode program yang digunakan:

```
print("====PROGRAM MENCARI KPK====")
# mendefinisikan fungsi
def hitung_kpk(x, y):
    # mencari bilangan terbesar antara x dan y
    if x > y:
        terbesar = x
    else:
        terbesar = y
    while True:
        # mencari kelipatan dari bilangan terbesar
        if (terbesar % x == 0) and (terbesar % y == 0):
            kpk = terbesar
            break
        terbesar += 1
    return kpk

nilai1 = int(input("Masukkan bilangan pertama: "))
nilai2 = int(input("Masukkan bilangan kedua: "))
print("Hasil KPK= ", hitung_kpk(nilai1, nilai2))
```

Dan berikut Output yang dihasilkan dari kode program tersebut:

```
013.11.exe C:/Users/User/Documents/Praktikum Pemograman1
====PROGRAM MENCARI KPK====
Masukkan bilangan pertama: 4
Masukkan bilangan kedua: 15
Hasil KPK= 60
PS C:/Users/User/Documents/Praktikum Pemograman1> LapModu
```

Program di atas adalah program Python untuk menghitung KPK (Kelipatan Persekutuan Terkecil) dari dua bilangan yang dimasukkan oleh pengguna. KPK adalah bilangan bulat terkecil yang merupakan kelipatan dari kedua bilangan yang diinputkan.

Fungsi `hitung_kpk(x, y)` adalah fungsi yang mengimplementasikan algoritma untuk mencari KPK dari dua bilangan `x` dan `y`. Di dalam fungsi ini, kita mencari bilangan terbesar antara `x` dan `y`, kemudian mencari kelipatan dari bilangan terbesar tersebut yang juga merupakan kelipatan dari kedua bilangan `x` dan `y`. Setelah ditemukan, bilangan tersebut adalah KPK dari kedua bilangan `x` dan `y`. Fungsi ini mengembalikan nilai KPK.

Selanjutnya, program meminta pengguna untuk memasukkan dua bilangan (`nilai1` dan `nilai2`) dan menjalankan fungsi `hitung_kpk()` dengan memasukkan kedua bilangan tersebut sebagai argumen. Hasil KPK kemudian dicetak ke layar.

Dengan demikian, program ini berfungsi untuk menghitung KPK dari dua bilangan yang dimasukkan oleh pengguna.