



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230978
Nama Lengkap	Jonathan Satriani Gracio Andrianto
Minggu ke / Materi	05 / Struktur Kontrol 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

Perulangan dalam pemrograman singkatnya adalah untuk menyuruh computer melakukan suatu hal berulang-ulang. Perulangan digunakan apabila dalam program diperlukan untuk :

- Melakukan suatu hal yang sama beberapa kali.
- Melakukan suatu hal secara bertahap, di mana setiap tahap sebenarnya memiliki langkah yang sama.
- Mengakses sekumpulan data dalam suatu struktur data, misalnya: List, Tuple, Queue, Stack dan beberapa struktur data lainnya

Pada python terdapat jenis perulangan **for**, **while** dan **rekursif**. Tetapi kali ini kita hanya akan membahas for dan while.

Perulangan For

Perulangan for sering disebut counted loop (perulangan yang terhitung). Perulangan for biasanya digunakan dalam kondisi :

- Jumlah perulangan atau range perulangan sudah diketahui sejak awal
- Perulangan terjadi karena operasi yang sama pada suatu rentang data atau rentang nilai. Misalnya dalam mencari jumlah dari 100 bilangan pertama, maka secara berturut-turut dilakukan penjumlahan $1 + 2 + 3 + \dots + 100$. Berarti dilakukan dalam rentang mulai dari 1 sampai 100.

Perulangan for lebih mudah karena menggunakan fungsi range(). Bentuk fungsi range() adalah berikut :

- range(stop). Digunakan untuk menghasilkan rentang dari **0** sampai **stop-1**. Misalnya range(8), berarti menghasilkan rentang 0-7.
- range(start, stop, [step]). Digunakan untuk menghasilkan rentang dari **start**, sampai **stop** dengan peningkatan sejumlah **step**. Pada bentuk ini **step**nya opsional.

Contoh program :

Program menampilkan bilangan dari 1 – 5 dengan for dan range()

```
for i in range(1, 6) :  
    print(i)
```

Program perulangan for

Output

```
1  
2  
3  
4  
5
```

Program menampilkan "hello ges" sebanyak 5 kali dengan for dan range()

```
for _ in range(1, 6) :  
    print("hello ges")
```

Program "hello ges"

```
hello ges  
hello ges  
hello ges  
hello ges  
hello ges
```

Output

Perulangan While

Perulangan while sering disebut uncounted loop atau perulangan yang tak terhitung karena biasanya digunakan pada kondisi di mana jumlah perulangan belum diketahui sebelumnya. Secara umum bentuk perulangan while adalah :

```
while <stop>:  
    operation  
    operation  
    <step (optional)>
```

Bentuk umum perulangan while

Berbeda dengan perulangan for yang menggunakan range, perulangan while menggunakan kondisional. Berikut ini adalah contoh perulangan dengan menggunakan while yang digunakan untuk memastikan input yang dimasukkan oleh pengguna adalah bilangan ganjil :

```
bilangan = 0  
ganjil = False  
while ganjil == False:  
    bilangan = int(input('Masukkan bilangan ganjil: '))  
    if bilangan % 2 == 1 :  
        ganjil = True  
  
print(bilangan, 'yang anda masukkan adalah bilangan ganjil')
```

Program pengecekan bilangan ganjil

Penggunaan Break dan Continue

Perulangan dapat dikontrol dengan menggunakan break dan continue. Secara umum break digunakan untuk menghentikan perulangan, sedangkan continue digunakan untuk melanjutkan perulangan ke iterasi berikutnya. Contoh penggunaan break dan continue :

Break

```
for i in range(1, 11) :  
    if i == 5:  
        break  
    else:  
        print(i)  
print('Selesai')
```

```
1  
2  
3  
4  
Selesai
```

Output

Contoh penggunaan break dalam perulangan

Pada saat nilai i = 5 maka perulangan berhenti dan menampilkan tulisan 'selesai'

Continue

```
for i in range(1, 11) :  
    if i == 5:  
        continue  
    else:  
        print(i)  
print('Selesai')
```

```
1  
2  
3  
4  
6  
7  
8  
9  
10  
Selesai
```

Output

Contoh penggunaan continue dalam perulangan

Program menampilkan angka 1 sampai 10 tapi melewati angka 5.

Konversi dari Bentuk for Menjadi Bentuk while

Bentuk perulangan for sebagian besar dapat dikonversi menjadi bentuk while. Beberapa hal yang ada di bentuk for dan while adalah sebagai berikut :

- Harus ada nilai awal, untuk memulai perulangan.
- Harus ada nilai akhir, untuk mengakhiri perulangan.
- Harus ada langkah, agar iterasi dari nilai awal bisa terus berjalan sampai mencapai nilai akhir.

Misalkan ada perulangan :

```
for i in range(1, 11):  
    print(i)
```

Perulangan for

Pada perulangan tersebut, start = 1, stop = 11, step = 1 (default). Konversi ke bentuk while dapat dilakukan dengan mudah, menghasilkan perulangan while berikut ini yang menghasilkan output yang sama :

```
i = 1 // start  
while i <= 10: // stop  
    print(i)  
    i = i + 1 // step
```

Perulangan while

Deret Bilangan

- Deret fibonacci

Deret bilangan fibonacci adalah deret bilangan yang tersusun dari penjumlahan dua suku sebelumnya dari deret bilangan tersebut. Biasanya deret bilangan fibonacci dimulai dari 1, 1, 2, 3, ... dan seterusnya. Programnya dapat dilihat dibawah ini :

```
def fibo(batas):  
    bil1 = 1  
    bil2 = 1  
    if bil1 < batas:  
        print(bil1, end='\t')  
        print(bil2, end='\t')  
    suku_baru = bil1 + bil2  
    while suku_baru < batas:  
        print(suku_baru, end='\t')  
        bil1 = bil2  
        bil2 = suku_baru  
        suku_baru = bil1 + bil2  
  
batas = int(input('Masukkan batas dari deret fibonacci: '))  
fibo(batas)
```

Program Fibonacci

Jika dijalankan dengan batas 100

```
Masukkan batas dari deret fibonacci: 100
1  1  2  3  5  8 13 21 34 55 89
```

Output

- Deret Konvergen

Buatlah program yang dapat menampilkan deret bilangan konvergen, yang diawali dari input bilangan dari pengguna. Suku-suku dari deret tersebut berikutnya didapatkan dengan cara berikut ini :

- Jika ganjil, maka kalikan dengan tiga, lalu tambah 1.
- Jika genap, bagi dengan 2.

Tampilkan suku-suku dari deret bilangan konvergen tersebut yang berakhir jika mencapai nilai 1. Programnya dapat dilihat dibawah ini :

```
def konvergen(start):
    suku = start
    while suku != 1:
        print(suku)
        if suku % 2 == 0:
            suku = suku // 2
        else:
            suku = suku * 3 + 1

start = int(input('Masukkan suku pertama dari deret konvergen : '))
konvergen(start)
```

Program Konvergen

Jika dijalankan dengan suku = 12 :

```
Masukkan suku pertama dari deret konvergen : 12
12
6
3
10
5
16
8
4
2
```

Output

Penggunaan Break

Buatlah program yang dapat menghitung rata-rata dari sejumlah input yang diberikan oleh pengguna. Program akan terus meminta input dari pengguna, sampai pengguna memasukkan bilangan negatif atau nol. Program kemudian menampilkan rata-rata dari keseluruhan input (abaikan input negatif atau nol). Programnya dapat dilihat dibawah :

```
def average():
    total = 0
    count = 0
    while True:
        input_user = int(input('Masukkan nilai (nol atau negatif
                                untuk berhenti): '))
        if input_user < 1:
            break
        else:
            total = total + input_user
            count = count + 1
    if count > 0:
        return total / count
    else:
        return 0

hasil = average()
print('Rata-rata: ', hasil)
```

Program average()

```
Masukkan nilai (nol atau negatif untuk berhenti): 5
Masukkan nilai (nol atau negatif untuk berhenti): 4
Masukkan nilai (nol atau negatif untuk berhenti): 3
Masukkan nilai (nol atau negatif untuk berhenti): 2
Masukkan nilai (nol atau negatif untuk berhenti): 1
Masukkan nilai (nol atau negatif untuk berhenti): 0
Rata-rata: 3.0
```

Output

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

Source Code

```
def perkalian(bil1, bil2) :  
    hasil_kali = bil1 * bil2  
    print(f"{bil1} x {bil2} = ", end = '')  
    for i in range(bil1) :  
        print(bil2, end = '')  
        if i < bil2 :  
            print(" + ", end = '')  
        else :  
            print(f" = {hasil_kali}")  
  
bil1 = int(input("masukkan bilangan pertama = "))  
bil2 = int(input("masukkan bilangan kedua = "))  
perkalian(bil1, bil2)
```

Penjelasan

def perkalian(bil1, bil2) : → mendefinisikan fungsi

hasil_kali = bil1 * bil2 → hasil kedua dua bilangan

print(f"{bil1} x {bil2} = ", end = '') → menampilkan bilangan pertama x bilangan kedua

for i in range(bil1) : → perulangan dengan range bilangan 1

print(bil2, end = '') → untuk menampilkan bilangan 2 sebanyak bilangan 1

if i < bil2 : → kondisi jika jumlah perulangan kurang dari bilangan 2

print(" + ", end = '') → menampilkan tanda '+' jika kondisi terpenuhi

else :

print(f" = {hasil_kali}") → menampilkan hasil kali kedua bilangan

bil1 = int(input("masukkan bilangan pertama = ")) → meminta user menginput bilangan 1

bil2 = int(input("masukkan bilangan kedua = ")) → meminta user menginput bilangan 2

perkalian(bil1, bil2) → memanggil fungsi perkalian

Output

```
masukkan bilangan pertama = 6
masukkan bilangan kedua = 5
6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
```

SOAL 2

Source Code

```
def ganjil(bawah, atas) :
    i = 0
    if bawah < atas :
        while bawah <= atas :
            if bawah % 2 == 1 :
                print(bawah, end=' ')
            bawah += 1
    else :
        while atas <= bawah :
            if bawah % 2 == 1 :
                print(bawah, end=' ')
            bawah -= 1

bawah = int(input("Masukkan batas bawah = "))
atas = int(input("Masukkan batas atas = "))

print("Deret bilangan ganjil = ", end = '')
ganjil(bawah, atas)
```

Penjelasan

def ganjil(bawah, atas) : → mendefinisikan

if bawah < atas : → kondisional batas bawah lebih kecil dari batas atas

while bawah <= atas : → perulangan ketika batas bawah kurang dari sama dengan batas atas

if bawah % 2 == 1 : → kondisional bilangan ganjil

print(bawah, end=' ') → menampilkan bilangan yang memenuhi kondisi atau bilangan ganjil

bawah += 1 → step dari while

else : → kondisional batas atas lebih kecil dari batas bawah

while atas <= bawah : → perulangan ketika batas atas kurang dari sama dengan batas bawah

if bawah % 2 == 1 : → kondisional bilangan ganjil

print(bawah, end=' ') → menampilkan bilangan yang memenuhi kondisi atau bilangan ganjil

bawah -= 1 → step dari while

bawah = int(input("Masukkan batas bawah = ")) → meminta user menginput batas bawah

atas = int(input("Masukkan batas atas = ")) → meminta user menginput batas bawah

print("Deret bilangan ganjil = ", end = "") → menampilkan output Deret bilangan ganjil

ganjil(bawah, atas) → memanggil fungsi

Output

```
Masukkan batas bawah = 1
Masukkan batas atas = 10
Deret bilangan ganjil = 1 3 5 7 9
```

SOAL 3

Source Code

```
def mata_kuliah(jumlah) :
    nilai_ = 0
    total = 0
    for i in range(1, jumlah + 1) :
        nilai = input(f"Nilai MK {i} = ")
        if nilai == "A" :
            nilai_ += 4
        elif nilai == "B" :
            nilai_ += 3
        elif nilai == "C" :
            nilai_ += 2
        elif nilai == "D" :
            nilai_ += 1
        else :
            print("Nilai invalid")
    total = nilai_ * 3
    jumlah_sks = jumlah * 3
    hasil = total / jumlah_sks
    print(f"IPS Anda semester ini = {hasil:.2f}")

jumlah = int(input("Masukkan jumlah mata kuliah anda = "))
mata_kuliah(jumlah)
```

Penjelasan

def mata_kuliah(jumlah) : ➔ mendefinisikan fungsi

nilai_ = 0 ➔ nilai bobot awal

total = 0 ➔ nilai awal total nilai semester

for i in range(1, jumlah + 1) : ➔ perulangan dengan start = 1, dan stop = jumlah sks + 1

nilai = input(f"Nilai MK {i} = ") ➔ meminta user menginput nilai huruf. User akan diminta menginput sebanyak jumlah mata kuliah, dan nilai huruf akan di ubah menjadi nilai bobot

if nilai == "A" :

 nilai_ += 4 ➔ nilai bobot jika nilai huruf = A

elif nilai == "B" :

 nilai_ += 3 ➔ nilai bobot jika nilai huruf = B

elif nilai == "C" :

 nilai_ += 2 ➔ nilai bobot jika nilai huruf = C

elif nilai == "D" :

 nilai_ += 1 ➔ nilai bobot jika nilai huruf = D

else :

 print("Nilai invalid") ➔ output jika inputan user berbeda dengan yang diharapkan

total = nilai_ * 3 ➔ total nilai semester

jumlah_sks = jumlah * 3 ➔ jumlah sks

hasil = total / jumlah_sks ➔ nilai IPS akhir

print(f"IPS Anda semester ini = {hasil:.2f}") ➔ menampilkan output nilai IPS dengan 2 angka dibelakang koma

jumlah = int(input("Masukkan jumlah mata kuliah anda = ")) ➔ meminta user menginputkan jumlah mata kuliah

mata_kuliah(jumlah) ➔ memanggil fungsi

LINK GITHUB

https://github.com/JonathanAndrianto123/Laporan_AIPro_5_71230978.git