Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230992**
Nama Lengkap	**Andriano Kurniawan Ladjeba**
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%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Definisi Perulangan

Alur program dapat disusun secara berurutan, bercabang, perulangan, atau kombinasi ketiganya. Susunan ini biasa disebut dengan struktur kendali. Loop digunakan ketika suatu program perlu melakukan hal berikut:

- 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.

Dengan Python, Anda dapat menggunakan **for, while**, atau untuk mengeksekusi loop secara **rekursif**. Pada pertemuan kali ini kita telah membahas perulangan for dan perulangan while.

Bentuk Perulangan for

Pada Python, perulangan dapat dinyatakan dalam bentuk seperti **for** dan **while.** Perulangan **for** biasanya digunakan pada kondisi:

- **Jumlah perulangan sudah diketahui sejak awal**. Misalnya pada saat akan dilakukan pembacaan data dari 10 file teks. Meskipun konten setiap file teks berbeda, memuat file teks hampir sama. Pembacaan dimulai dari file ke-1, ke-2, ke-3, ke- hingga ke file ke-10.
- **Pengulangan ini disebabkan oleh operasi yang sama pada rentang data atau rentang nilai**. Misalnya, untuk mencari jumlah 100 bilangan pertama, tambahkan dengan urutan 1 + 2 + 3 + ... + 100. Artinya ini dilakukan pada rentang 1 hingga 100 dari.

Perulangan for pada saat tertentu lebih mudah dilakukan menggunakan bantuan fungsi range(), yang bentuknya sebagai berikut:

range(stop). Digunakan untuk menghasilkan rentang dari 0 sampai stop-1. Misalnya range(6), berarti menghasilkan rentang 0-5.

- range(start, stop, [step]). Digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step.
- range(start, stop, [step]). Digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step.

Berikut adalah contoh program untuk menampilkan bilangan dari 1 hingga 100 dengan menggunakan for dan range():

```
for i in range(1, 101):
    print(i)
```

Pada program tersebut, ada perulangan for dengan menggunakan range(), dimulai dari 1 (start) sampai 101 (stop-1) dengan langkah 1 (default step adalah 1). Kemudian variabel i digunakan sebagai bentuk counter, di mana nilai i akan naik secara berurutan sesuai dengan nilai hasil dari fungsi range() tersebut. Untuk kasus yang dimana tidak diperlukan counter, maka bentuk perulangan bisa seperti berikut ini:

```
for _ in range(2, 101, 2):
    print('hEllo World')
```

Program ini akan menampilkan tulisan Hello World sebanyak 100 kali, yang tidak membutuhkan nilai dari suatu counter.

Step negatif

Perhatikan perulangan for berikut ini yang akan menampilkan seluruh bilangan genap dari 2 sampai 100:

```
for I in range(2, 101, 2):
    print(i)
```

Perulangan dilakukan pada rentang 2-100, dengan langkah 2. Maka rentang yang dipakai adalah 2,4,6,8,10,12,…,100. Bagaimana jika diperlukan untuk menampilkan bilangan genap dari 100 sampai 2? Fungsi range() bisa menerima step negatif, seperti pada contoh berikut:

```
for i in range(100, 1, -2):
    print(i)
```

Program ini akan menampilkan bilangan genap mulai dari 100,98,96,94, ….., sampai 2.

Bentuk Perulangan While

Bentuknya while biasanya digunakan pada saat kondisi di mana suatu jumlah perulangan masih belum diketahui sebelumnya. Bentuk perulangan while secara umum adalah sebagai berikut:

```
<start>
while <stop>:
    operation
    operation
    <step (opsional)>
```

Berikut ini Contoh perulangan dengan while digunakan di sini untuk memastikan bahwa input pengguna adalah bilangan genap:

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

Gambar berikut menunjukkan output yang dihasilkan oleh program tersebut. Meskipun program meminta pengguna memasukkan angka ganjil, itu akhirnya berhenti setelah pengguna memasukkan angka 88, yang merupakan angka genap. Karena tidak diketahui sampai berapa kali pengguna memasukkan bilangan ganjil yang tidak sesuai dengan permintaan, kasus ini sangat sesuai jika menggunakan perulangan while.

```
an\Tugas 5\latihan> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/KAMPUS/Semester 2/Prakkrikum Algoritma dan Pemograman/Tugas 5/latihan/sc.py"
Masukkan bilangan genap: 99
Masukkan bilangan genap: 23
Masukkan bilangan genap: 11
Masukkan bilangan genap: 37
Masukkan bilangan genap: 87651
Masukkan bilangan genap: 88
88 yang anda masukkan adalah blangan genap
PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\Tugas 5\latihan>
```

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. Periksa program berikut ini, yang akan menampilkan angka 1–10.

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

Program akan menampilkan bilangan 1 hingga 10, dengan tulisan "Selesai" di baris terakhir. Jika diinginkan untuk menghentikan perulangan yang seharusnya sampai 10, tetapi dihentikan saat mencapai 5, maka diperlukan break dengan kondisi seperti program berikut:

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

Jika nilai i=5 dan ekspresi boolean dari i==5 menghasilkan nilai benar, program akan menjalankan instruksi break. Kemudian perulangan dihentikan, dan program berlanjut ke baris setelah perulangan, menampilkan tulisan "Selesai".
Apa yang membedakan break dari continue? Selama diperlukan, gunakan terus untuk melewati tahap perulangan saat ini. Sebagai contoh, program ini menampilkan angka dari 1 hingga 10 tetapi tidak menampilkan angka 6:

```
for i in range(1, 11):
    if i == 6:
        continue
    else:
        print(i)
print('Selesai')
```

Outputnya yang dihasilkan dari contoh pengunaan continue dapat dilihat pada contoh dibawah. Program menampilkan angka 1 sampai dengan 10, tetapi melewati 6.

```
PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\Tugas 5\latihan> & C:/Users/Acer/A
ppData/Local/Programs/Python/Python310/python.exe "d:/KAMPUS/Semester 2/Prakkrikum Algoritma d
an Pemograman/Tugas 5/latihan/sc.py"
1
2
3
4
5
7
8
9
10
Selesai
PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\Tugas 5\latihan>
```

Konversi dari Bentuk for Menjadi Bentuk while

Sebagian besar, bentuk perulangan for dapat diubah menjadi bentuk while. Berikut adalah beberapa hal yang tersedia dalam bentuk for dan while:

- 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.

Misalnya ada perulangan for seperti berikut ini:

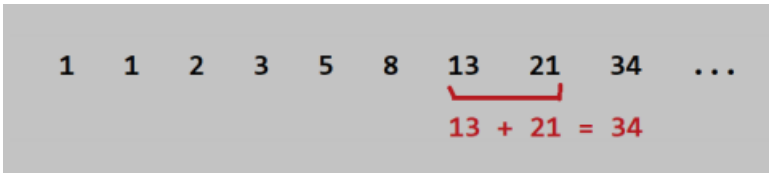
```
for i in range(1, 11):
        print(i)
```

Dimungkinkan untuk mengetahui bahwa while tersebut dimulai dari 1, berakhir di 10, dan stepnya adalah 1. Dengan cara ini, Anda dapat dengan mudah mengubahnya ke bentuk while, yang menghasilkan while berikut ini, yang menghasilkan output yang sama:

```
i = 1                //awal
while i <= 10:       //kondisi akhir
    print(i)
    i = i + 1        //step
```

Deret Bilangan

Barisan bilangan Fibonacci adalah barisan bilangan yang terdiri dari penjumlahan suku-suku dari dua barisan bilangan sebelumnya. Biasanya deret Fibonacci diawali dengan angka 1, 1, 2, 3,...dan seterusnya. Ilustrasinya dapat dilihat pada Gambar dibawah.



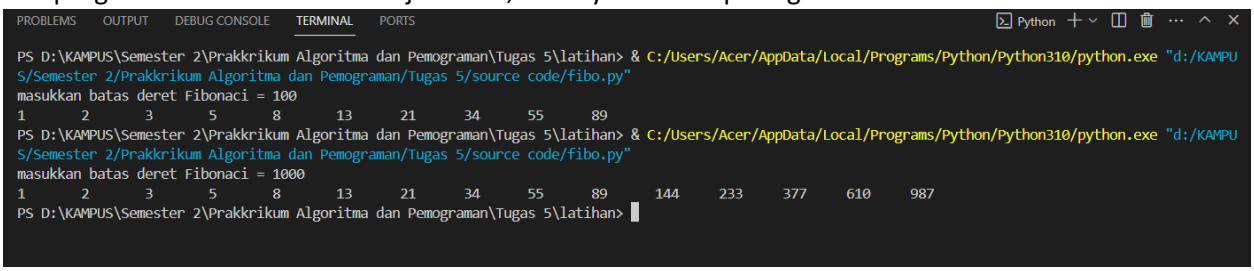
Deret fibonacci akan bekerja lebih baik jika diimplementasikan sebagai fungsi fibo() yang menerima parameter batas. Proses pelaksanaannya adalah sebagai berikut:

1. Minta nilai batas dari pengguna.
2. Panggil fungsi fibo() untuk menampilkan deret fibonacci dengan menggunakan perulangan while.

Program untuk menjawab permasalahan tersebut adalah sebagai berikut:

```python
def fibo(batas):
    bil1 = 1
    bil2 = 2

    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 deret Fibonaci = "))
fibo(batas)
```

Jika program fibonacci tersebut dijalankan, hasilnya akan seperti gambar dibawah ini.



MATERI 2

Penjelasan materi 2, dst… sesuai format ini.

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.

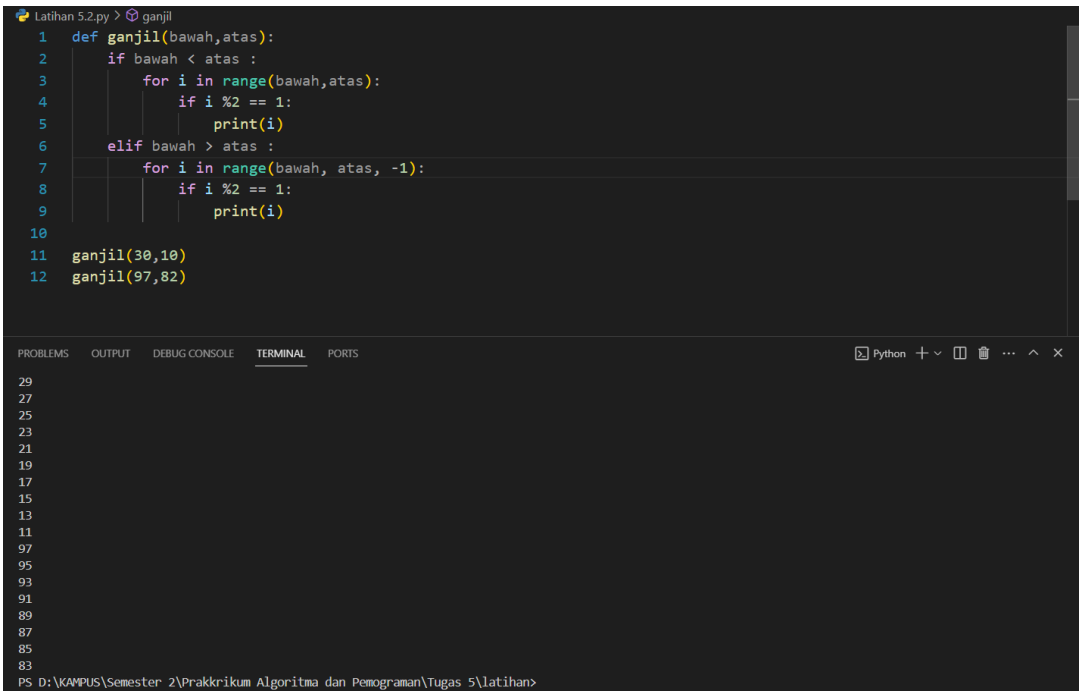
SOAL 1

Latihan 5.1

```python
import os

def perkalian (a,b):
    for i in range(b):
        if i == b-1:
            print(str(b) + " = ")
        else:
            print(str(b) + " + ")
    print(str(a*b))

run = True
while (run):
    print("Perkalian")
    try:
        a = int(input("Angka Pertama : "))
        b = int(input("Angka Kedua : "))
        print("==============")
        c = perkalian(a,b)
        d = str(c).replace("/n", "")
        print( str(c).strip())
        os.system('cls')
    except:
        print("==============")
        print ("Jenis format salah")
        os.system('cls')

```

SOAL 2

Latihan 5.2



SOAL 3

Latihan 5.3

```python
def hitung_ips(jumlah_matkul):
    total_sks = 0
    total_bobot = 0

    for i in range(jumlah_matkul):
        nilai_huruf = input("Masukkan nilai MK {} : ".format(i+1))
        sks = 3
        if nilai_huruf == 'A':
            bobot = 4
        elif nilai_huruf == 'B':
            bobot = 3
        elif nilai_huruf == 'C':
            bobot = 2
        elif nilai_huruf == 'D':
            bobot = 1
        else:
            print("Format anda salah silahkan gunakan huruf dan harus kapital")
            return

        total_sks = total_sks + sks
        total_bobot = total_bobot + sks * bobot

    ips = total_bobot / total_sks
    return ips

jumlah_matkul = int(input("Berapa jumlah mata kuliah: "))
hasil_ips = hitung_ips(jumlah_matkul)

print("Nilai IPS anda semester ini :", round(hasil_ips,2))
```

Hasil :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\Tugas 5\latihan> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/KA
S/Semester 2/Prakkrikum Algoritma dan Pemograman/Tugas 5/latihan/Latihan 5.3.py"
Berapa jumlah mata kuliah: 6
Masukkan nilai MK 1 : A
Masukkan nilai MK 2 : B
Masukkan nilai MK 3 : C
Masukkan nilai MK 4 : A
Masukkan nilai MK 5 : D
Masukkan nilai MK 6 : C
Nilai IPS anda semester ini : 2.67
PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\Tugas 5\latihan>
```

Github: https://github.com/EchoGinDev/tugasAlpro5.git