Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71231005**
Nama Lengkap	**Josephine Marcelia**
Minggu ke / Materi	**06 / Percabangan dan Perulangan Kompleks**

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

Percabangan Kompleks

Struktur percabangan kompleks digunakan untuk mengontrol alur eksekusi kode berdasarkan kondisi yang diberikan. Percabangan ini biasanya terjadi karena ada banyak syarat yang harus dipenuhi sehingga kondisi pemilihan tidak hanya satu tetapi bisa terdiri atas banyak alternatif.

Contoh:

Program untuk menentukan harga tiket masuk ke museum

```
umur = int(input("Masukkan umur anda: "))
status = input("Apakah anda memiliki kartu member? (ya/tidak): ")


if umur < 3:
    harga_tiket = 0
elif umur < 12:
    if status == "ya":
        harga_tiket = 3000
    else:
        harga_tiket = 5000
elif umur < 60:
    if status == "ya":
        harga_tiket = 7000
    else:
        harga_tiket = 10000
else:
    if status == "ya":
        harga_tiket = 5000
    else:
        harga_tiket = 7500

print(f"Harga tiket masuk adalah Rp. {harga_tiket}")
```

Pada contoh di atas kita bisa melihat bahwa program akan mengambil input umur dan status member dari pengguna dan menentukan harga tiket masuk ke museum berdasarkan umur dan status member tersebut. Struktur percabangan kompleks "if-elif-else" digunakan untuk menentukan harga tiket.

Dimana ketentuannya adalah :

- Jika umur kurang dari 3, maka harga_tiket diberikan nilai 0.
- Jika umur lebih besar atau sama dengan 3, maka kode akan melanjutkan ke kondisi berikutnya(elif)
- Jika umur kurang dari 12, maka:
  - Jika status member adalah "ya", maka harga tiket diberikan nilai 3000.
  - Jika status member adalah "tidak", maka harga tiket diberikan nilai 5000.
- Jika umur lebih besar atau sama dengan 12 dan kurang dari 60, maka:
  - Jika status member adalah "ya", maka harga tiket diberikan nilai 7000.
  - Jika status member adalah "tidak", maka harga tiket diberikan nilai 10000.
- Jika umur lebih besar atau sama dengan 60, maka:
  - Jika status member adalah "ya", maka harga tiket diberikan nilai 5000.
  - Jika status member adalah "tidak", maka harga tiket diberikan nilai 7500.

Output :

```
Masukkan umur anda: 5
Apakah anda memiliki kartu member? (ya/tidak): ya
Harga tiket masuk adalah Rp. 3000
```

```
Masukkan umur anda: 15
Apakah anda memiliki kartu member? (ya/tidak): ya
Harga tiket masuk adalah Rp. 7000
```

```
Masukkan umur anda: 35
Apakah anda memiliki kartu member? (ya/tidak): tidak
Harga tiket masuk adalah Rp. 10000
```

```
Masukkan umur anda: 60
Apakah anda memiliki kartu member? (ya/tidak): tidak
Harga tiket masuk adalah Rp. 7500
```

```
Masukkan umur anda: 70
Apakah anda memiliki kartu member? (ya/tidak): ya
Harga tiket masuk adalah Rp. 5000
```

Contoh lainnya menggunakan while

```
i = 1
j = 1

while True:
    while True:
        if i == 4 and j == 4:
            break
        if j > i:
            break

        print(f"{i} x {j} = {i * j}")
        j += 1

    i += 1
    j = 1
```

Pada contoh di atas, dapat dilihat bahwa kita memiliki dua perulangan, yaitu perulangan luar dan perulangan dalam. Perulangan luar akan selalu benar, sama halnya dengan perulangan dalam.

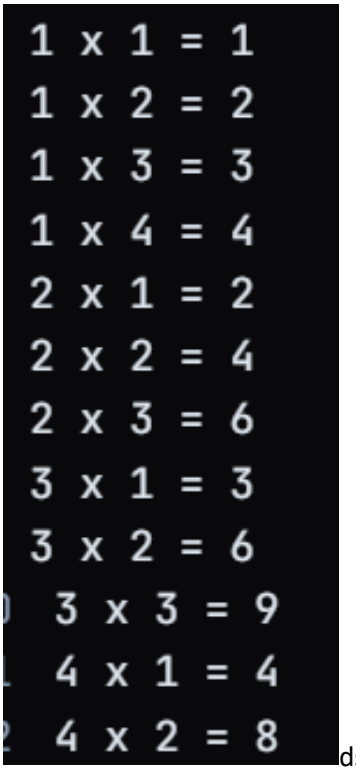
Dalam perulangan luar, kita menginisialisasi variabel i dengan nilai 1 dan variabel j dengan nilai 1. Kemudian, kita memulai perulangan dengan while True.

Dalam perulangan dalam, kita juga menginisialisasi variabel j dengan nilai 1. Kemudian, kita memulai perulangan dengan while True.Kemudian, kita menambahkan kondisi if untuk menghentikan perulangan jika kondisi tertentu terpenuhi. Kondisi pertama adalah jika nilai variabel i dan j sama dengan 4, maka kita menggunakan break untuk menghentikan perulangan dalam. Kondisi kedua adalah jika nilai variabel j lebih besar dari nilai variabel i, maka kita menggunakan break untuk menghentikan perulangan dalam dan increment variabel i.

Lalu, kita menampilkan hasil perkalian dari nilai variabel i dan j dengan print. Kemudian, kita increment variabel j dengan j += 1.

Setelah perulangan dalam selesai, kita increment variabel i dengan i += 1 dan setel ulang variabel j dengan j = 1.

Output yang dihasilkan adalah :



dst

Perulangan Bertingkat

Struktur perulangan kompleks dalam python merupakan bentuk per-ulangan di mana di dalam suatu perulangan terdapat perulangan lain sdehingga terjadilah perulangan bertingkat. . Intinya, masalah yang dapat diselesaikan menggunakan perulangan kompleks biasanya memiliki pola grid (kotak-kotak) yang memiliki lebar dan panjang.

```python
kota = ["Jakarta", "Bandung", "Surabaya", "Yogyakarta", "Bali"]
hotel = ["Hotel A", "Hotel B", "Hotel C"]

for i in range(len(kota)):
    print(f"Kota: {kota[i]}")

    for j in range(len(hotel)):
        print(f"    Hotel: {hotel[j]}")
```

Seperti yang dapat kita lihat pada contoh program di atas, kita memiliki dua perulangan. Perulangan luar akan melakukan iterasi dari 0 sampai 4, yang menunjukkan indeks dari setiap elemen di daftar kota. Perulangan dalam akan melakukan iterasi dari 0 sampai 2, yang menunjukkan indeks dari setiap elemen di daftar hotel. Untuk setiap iterasi dari perulangan luar, kita akan melakukan iterasi dari perulangan dalam.

Dalam perulangan luar, kita menampilkan nama kota yang sesuai dengan indeks yang sedang diiterasi. Dalam perulangan dalam, kita menampilkan nama hotel yang sesuai dengan indeks yang sedang diiterasi.

Output yang dihasilkan adalah :

```
Kota: Jakarta
    Hotel: Hotel A
    Hotel: Hotel B
    Hotel: Hotel C
Kota: Bandung
    Hotel: Hotel A
    Hotel: Hotel B
    Hotel: Hotel C
Kota: Surabaya
    Hotel: Hotel A
    Hotel: Hotel B
    Hotel: Hotel C
Kota: Yogyakarta
    Hotel: Hotel A
    Hotel: Hotel B
    Hotel: Hotel C
Kota: Bali
    Hotel: Hotel A
    Hotel: Hotel B
    Hotel: Hotel C
```

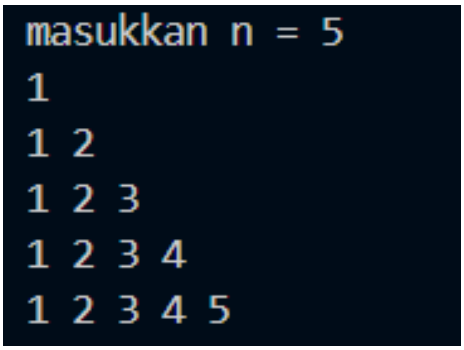
Contoh Perulangan lainnya menggunakan while

```python
i = 1
j = 1

while i <= 5:
    while j <= i:
        print(j, end=" ")
        j += 1

    i += 1
    print()
```

Pada contoh program perulangan bertingkat di atas, kita memiliki dua perulangan. Perulangan luar menggunakan while dan akan melakukan iterasi sampai nilai variabel i lebih kecil atau sama dengan 5. Perulangan dalam juga menggunakan while dan akan melakukan iterasi sampai nilai variable j lebih kecil atau sama dengan nilai variable i.

Dalam perulangan luar, kita menampilkan setiap baris pola angka. Dalam perulangan dalam, kita menampilkan setiap bilangan sampai nilai variable j lebih kecil atau sama dengan nilai variable i.

Output yang dihasilkan adalah :



.

BAGIAN 2: LATIHAN MANDIRI (60%)

Latihan 6.1

```
def prim(angka):
    if angka < 2:
        return False
    for j in range(2, int(angka**0.5) + 1):
        if angka % j == 0:
            return False
    return True

def prima_terdekat(n):
    for k in range(n-1, 1, -1):
        if prim(k):
            return k
    return 1

n = int(input("Masukkan angka: "))
prim_terdekat = prima_terdekat(n)

print(f"bilangan prima terdekat kurang dari {n} adalah {prim_terdekat}.")
```

```
Masukkan angka: 12
bilangan prima terdekat kurang dari 12 adalah 11.
```

Penjelasan :

- Fungsi prim(angka) saya gunakan untuk menentukan bilangan prima atau bukan.Jika prima fungsi akan mengembalikan nilai true, sedangkan kalau bukan prima fungsi akan mengembalikan nilai false.
- Fungsi prima_terdekat(n) saya gunakan untuk menemukan bilangan prima terdekat yang kurang dari bilangan yang diinputkan. Lalu, fungsi ini akan melakukan perulangan dari n-1 sampai 1 dengan langkah -1.
- Jika bilangan tersebut merupakan bilangan prima, maka fungsi ini akan mengembalikan bilangan tersebut.
- Variabel prim_terdekat saya gunakan untuk menyimpan hasil dari pemanggilan fungsi prima_terdekat(n).
- Pada bagian akhir program, saya meminta puser untuk memasukkan bilangan. Kemudian program akan mencetak hasil dari pemanggilan fungsi prima_terdekat(n) ke layar.

Latihan 6.2

```
def faktorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * faktorial(n-1)

def deret_faktorial(n):
    for k in range(n, 0, -1):
        print(faktorial(k), end=" ")
        for l in range(k, 0, -1):
            print(l, end=" ")
        print()

nilai = int(input("Masukkan nilai n: "))
deret_faktorial(nilai)
```

```
Masukkan nilai n: 6
720 6 5 4 3 2 1
120 5 4 3 2 1
24 4 3 2 1
6 3 2 1
2 2 1
1 1
```

Penjelasan:

- Fungsi faktorial(n) saya gunakan untuk menghitung faktorial dari sebuah bilangan. Lalu, fungsi ini akan melakukan perulangan sampai n sama dengan 0 atau 1, dan pada setiap perulangan akan mengalikan n dengan hasil dari pemanggilan fungsi faktorial dengan argumen n - 1.

- Fungsi deret_faktorial(n) saya gunakan untuk menampilkan deret factorial. Lalu, fungsi ini akan melakukan perulangan sampai n sama dengan 0, dan pada setiap perulangan akan mencetak hasil dari pemanggilan fungsi faktorial dengan argumen k dan deret angka dari k sampai 1.

-Variabel nilai digunakan untuk menyimpan inputan nilai n dari pengguna.

-Pada bagian akhir program, saya meminta user untuk memasukkan nilai n. Lalu program akan mencetak hasil dari pemanggilan fungsi deret_faktorial(nilai).

Latihan 6.3

```
def deret_angka(t, l):
    awal = 1
    for j in range(t):
        for k in range(l):
            print(awal, end=' ')
            awal += 1
        print()

tinggi = int(input("Masukkan tinggi: "))
lebar = int(input("Masukkan lebar: "))

deret_angka(tinggi, lebar)
```

```
Masukkan tinggi: 5
Masukkan lebar: 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
```

Penjelasan :

- Fungsi deret_angka(t, l) saya gunakan untuk menampilkan deret angka sesuai dengan tinggi dan lebar yang diinputkan. Pada setiap perulangan, fungsi ini akan mencetak deret angka dari awal sampai awal + lebar - 1.
- Variabel awal saya gunakan untuk menyimpan nilai awal dari deret angka yang akan dicetak. Lalu Variabel tinggi dan lebar saya gunakan untuk menyimpan inputan tinggi dan lebar dari user.
- Pada bagian akhir program, saya meminta user untuk memasukkan tinggi dan lebar. Lalu program akan mencetak hasil dari pemanggilan fungsi deret_angka(tinggi, lebar).

Link github :
https://github.com/Josephinemrc/Tugas6-PrakAlpro.git