



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230981
Nama Lengkap	Deo Dewanto
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



Edit dengan WPS Office

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

Struktur Percabangan Kompleks

Struktur percabangan kompleks dalam Python memberikan kemampuan untuk mengevaluasi beberapa kondisi secara berurutan dan menjalankan blok kode yang sesuai dengan kondisi yang terpenuhi.

- If

Digunakan untuk mengevaluasi kondisi pertama. Jika kondisi pertama terpenuhi (evaluasi menghasilkan nilai True), maka blok kode di dalam if akan dieksekusi,

- Contoh penggunaan if



```
contoh_if.py > ...
1 X = 15
2 if X > 5:
3     print("X lebih besar dari 5")
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python + - [ ] ... ^ X
/usr/local/bin/python3 /Users/macbook/Documents/LAPORAN6/contoh_if.py
macbook@MacBook-Pro-2 LAPORAN6 % /usr/local/bin/python3 /Users/macbook/Documents/LAPORAN6/contoh_if.py
X lebih besar dari 5
macbook@MacBook-Pro-2 LAPORAN6 %
```

Pada contoh di atas, if digunakan untuk mengevaluasi apakah nilai x lebih besar dari 5.

- Elif

Digunakan untuk mengevaluasi kondisi kedua (dan seterusnya) jika kondisi sebelumnya tidak terpenuhi. Contohnya pada code di bawah ini:



```
1 x = 10
2 if x > 10:
3     print("x lebih besar dari 10")
4 elif x == 5:
5     print("x sama dengan 10")
6 else:
7     print("x kurang dari 10")
```

macbook@MacBook-Pro-2 LAPORAN6 % /usr/local/bin/python3 /Users/macbook/Documents/LAPORAN6/contoh_elif.py
x kurang dari 10
macbook@MacBook-Pro-2 LAPORAN6 %

Pada contoh di atas, jika nilai x tidak lebih besar dari 5, maka Python akan mengecek apakah nilai x sama dengan 5. Jika ya, maka blok kode di dalam elif akan dieksekusi.

- Else

Digunakan sebagai kondisi terakhir yang akan dievaluasi jika semua kondisi sebelumnya tidak terpenuhi. Contohnya pada code di bawah ini:

```
1 x = 5
2 if x > 10:
3     print("x lebih besar dari 10")
4 else:
5     print("x kurang dari atau sama dengan 10")
```

macbook@MacBook-Pro-2 LAPORAN6 % /usr/local/bin/python3 /Users/macbook/Documents/LAPORAN6/contoh_else.py
x kurang dari atau sama dengan 10
macbook@MacBook-Pro-2 LAPORAN6 %

Pada contoh di atas, else digunakan untuk mengevaluasi kondisi yang berbeda dari kondisi if. Jika kondisi pada if tidak terpenuhi, maka blok kode di dalam else akan dieksekusi.



Struktur Perulangan Kompleks

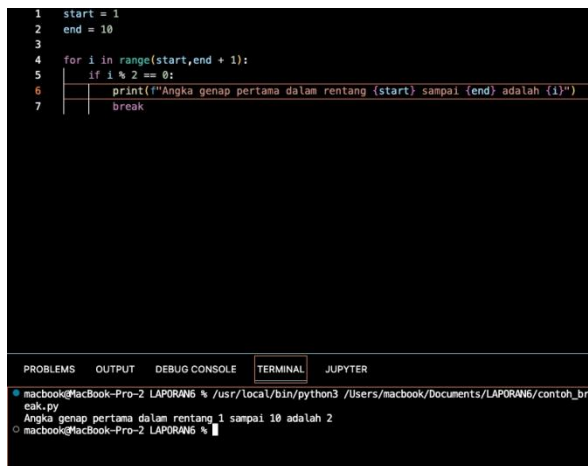
Struktur perulangan kompleks dalam Python memungkinkan pengontrolan yang lebih fleksibel dan adaptif terhadap alur eksekusi program, terutama saat kita perlu menangani kondisi-kondisi yang kompleks.

- Break :

Pernyataan break dalam Python digunakan untuk menghentikan iterasi secara paksa dalam suatu perulangan saat kondisi tertentu terpenuhi.

Berikut contoh penggunaan break:

```
1 start = 1
2 end = 10
3
4 for i in range(start, end + 1):
5     if i % 2 == 0:
6         print(f"Angka genap pertama dalam rentang {start} sampai {end} adalah {i}")
7         break
```



The screenshot shows a Jupyter Notebook interface with a terminal window. The terminal output displays the execution of a Python script. The script defines a range from 1 to 10 and iterates through it. When it encounters the first even number, 2, it prints a message and then breaks the loop. The terminal output shows the command being run and the resulting output.

- Continue

Pernyataan continue digunakan dalam Python untuk melanjutkan ke iterasi berikutnya dalam perulangan tanpa menjalankan kode di bawahnya dalam iterasi saat ini.

Berikut adalah contoh penggunaan continue:



```
1 for i in range(1,10):
2     if i == 3:
3         continue
4     print(i)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER

ntinu.py
1
2
4
5
6
7
8
9

Pada contoh di atas, kita menggunakan perulangan for untuk mengiterasi nilai dari 1 hingga 5. Saat nilai i sama dengan 3, pernyataan continue dieksekusi, dan kode di bawahnya dalam iterasi saat ini tidak dijalankan. Sehingga, angka 3 tidak dicetak, dan iterasi langsung melanjutkan ke angka berikutnya.

- Perulangan bertingkat

Perulangan bertingkat dalam Python adalah konsep di mana satu atau lebih perulangan ditempatkan di dalam perulangan lain.

Berikut adalah contoh penggunaan perulangan bertingkat :

```
perulangan.py > ...
1 for i in range(1,5):
2     for j in range(1,5):
3         print(f'({i}, {j})')
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYT

(2, 1)
(2, 2)
(2, 3)
(2, 4)
(3, 1)
(3, 2)
(3, 3)
(3, 4)
(4, 1)
(4, 2)



Edit dengan WPS Office



Edit dengan WPS Office

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

SOURCE CODE DAN OUTPUT

```
#latihan 1
def is_prima(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def cari_prima_terdekat(num):
    for i in range(num - 1, 1, -1):
        if is_prima(i):
            return i
    return None

n = int(input("Masukkan bilangan n: "))
prima_terdekat = cari_prima_terdekat(n)
if prima_terdekat:
    print(f"Bilangan prima terdekat < {n} adalah {prima_terdekat}")
else:
    print(f"Tidak ditemukan bilangan prima terdekat < {n}")
```

✓ 1.7s Python

... Bilangan prima terdekat < 12 adalah 11

PENJELASAN

Program di atas berfungsi untuk mencari dan mencetak bilangan prima terdekat yang lebih kecil dari bilangan yang dimasukkan, dengan menghasilkan dan memeriksa bilangan prima secara berurutan dari bilangan. Di sini saya memasukkan 12 dan akan muncul <12 adalah 11

SOAL 2

SOURCE CODE DAN OUTPUT



Edit dengan WPS Office

```
#Latihan 2
def deret(n):
    for i in range(n, 0, -1):
        print(str(faktorial(i)) + " ", end="")
        for j in range(i, 0, -1):
            print(str(j) + " ", end="")
        print()

    def faktorial(num):
        if num == 1 or num == 0:
            return 1
        return num * faktorial(num - 1)

    n = int(input("Masukkan jumlah baris pattern: "))
    deret(n)

[1] ✓ 33.4s Python
```

```
... 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

Program ini bertujuan untuk mencetak pola segitiga dengan baris-baris yang menurun, di mana setiap barisnya terdiri dari nilai faktorial dari nomor baris tersebut, diikuti dengan deret angka yang terus berkurang dari nomor baris tersebut hingga 1.

SOAL 3

SOURCE CODE DAN OUTPUT

```
#Latihan 3
Tinggi = int(input("tinggi = "))
Lebar = int(input("lebar = "))
Angka = 1
for i in range(Tinggi):
    for j in range(Lebar):
        print(Angka, end=" ")
        Angka += 1
    print()

[5] ✓ 2.8s Python
```

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

PENJELASAN

Program ini bertujuan untuk mencetak pola persegi panjang dengan tinggi dan lebar yang ditentukan oleh user. Program dimulai dengan meminta user untuk memasukkan nilai tinggi dan lebar persegi panjang dengan menggunakan fungsi input.

LINK GITHUB: https://github.com/Deodewanto07/LAPRAK_6.git



Edit dengan WPS Office