



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	71220829
Nama Lengkap	TIMOTIUS EKANA TEJO KESUMA
Minggu ke / Materi	07 / Percabangan dan Perulangan Kompleks

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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 7.3.1 – Struktur Pecabangan Kompleks

Percabangan kompleks memiliki beberapa kombinasi, beberapa contohnya antara lain:



```
if(kondisi1):
    if(kondisi2):
        pass
    else:
        pass
```

```
if(kondisi1):
    if(kondisi2):
        pass
    else:
        pass
```

```
if(kondisi1):
    pass
    if(kondisi2):
        pass
else:
    if(kondisi3):
        pass
    else:
        pass
```

```
if(kondisi1):
    if(kondisi2):
        pass
    else:
        pass
```

```
if(kondisi1):
    pass
else:
    if(kondisi2):
        pass
    else:
        if(kondisi3):
            pass
        else:
            pass
```

Semua code tersebut adalah percabangan kompleks dimana percabangan dapat ditumpuk dan di dalam percabangan dapat memiliki percabangan lainnya

MATERI 7.3.2 – Struktur Perulangan Kompleks

- **Break**

Digunakan untuk menghentikan perulangan. Biasanya break dalam perulangan digunakan menggunakan if.

Contoh:

```
for i in range(1000):  
    print(i)  
    if i==5:  
        break
```

Output:

```
0  
1  
2  
3  
4  
5
```

- **Continue**

Digunakan untuk melewati suatu kondisi. Sama dengan break, continue digunakan dalam perulangan menggunakan if.

Contoh:

```
for i in range(6):  
    if i==3:  
        continue  
    print(i)
```

Output:

```
0  
1  
2  
4  
5
```

- **Perulangan Bertingkat**

For

Perulangan kompleks adalah perulangan yang dilakukan di dalam sebuah perulangan.

Perulangan kompleks biasanya digunakan untuk sesuatu yang memiliki pola grid (kotak-kotak) yang memiliki panjang dan lebar.

Contoh:

```
for i in range(10):  
    print()  
    for j in range(i):  
        print(i,end=' ')
```

Perulangan ini bekerja dengan range 10, dimana i akan terus naik dari 0 hingga 10, dan i akan menjadi range pada perulangan ke 2, sehingga perulangan ke 2 akan memiliki range yang terus bertambah hingga range dari perulangan pertama.

Output:

```
1  
22  
333  
4444  
55555  
666666  
7777777  
88888888  
999999999
```

While

```
i = 0
while i < 10:
    print()
    j = 0
    while j < i:
        print(i, end='')
        j += 1
    i += 1
```

Perulangan while bertingkat ini bekerja dengan menginisialisasikan ulang j pada setiap perulangan pertama dilakukan, dan i akan terus bertambah 1 setiap perulangan selesai dilakukan. j akan terus bertambah didalam perulangan ke 2, sehingga akan didapati

Output:

```
1
22
333
4444
55555
666666
7777777
88888888
999999999
```

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

Buatlah program untuk mencari bilangan prima terdekat dari suatu bilangan yang diinputkan oleh pengguna (n) dan nilai bilangan prima tersebut $< n$. Contoh: input $n=12$, maka prima terdekat < 12 adalah 11 Contoh: input $n=21$, maka prima terdekat < 21 adalah 19

Jawab:

```
import math
n = int(input("n = "))

i = n - 1
while i > 1:
    prima = True
    for j in range(2, int(math.sqrt(i)) + 1):
        if i % j == 0:
            prima = False
            break
    if prima:
        print(f"Maka prima terdekat < {n} adalah {i}")
        break
    i -= 1
```

Code ini bekerja dengan meminta inputan pengguna dan ditampung pada n, kemudian gunakan i untuk menampung nilai $n - 1$ (untuk memastikan yang di cek adalah bilangan lebih kecil dari n), kemudian jika i kurang dari 1, maka akan dilakukan perulangan. Prima diinisialisasikan dengan nilai True, kemudian dilakukan perulangan lagi (perulangan bertingkat) dimana range dari perulangan adalah start = 2, stop akar dari i dan step adalah 1. Jika didalam perulangan ditemukan $i \% j$ adalah 0, maka prima adalah False dan perulangan selesai. Setelah perulangan selesai dan tidak ditemukan dilai $i \% j$, maka akan dikembalikan nilai prima terdekat. Setiap perulangan dilakukan, i akan selalu dikurang 1.

Input, Output:

```
n = 12
Maka prima terdekat < 12 adalah 11
```

SOAL 2

Buatlah program untuk menampilkan deret seperti di bawah ini. n diinputkan secara dinamis.

Jawab:

```
1  n = int(input("n = "))
2  bantu = 1
3  for i in range(n, 0, -1):
4      for j in range(i, 0, -1):
5          bantu *= j
6          print(bantu, end=" ")
7          for j in range(i, 0, -1):
8              print(j, end=" ")
9
10     bantu = 1
11     print()
```

Code ini menggunakan n sebagai tampungan input dari user. Variabel bantu disini digunakan sebagai tampungan dari nilai hasil dari perkalian dari 1 deret bilangan. Perulangan menggunakan reverse loop dengan start adalah n, stop adalah 0, dan step adalah -1. Perulangan ke 2 (perulangan bertingkat) dimulai dari i, kemudian stop 0, dan step adalah -1. Perulangan ke 2 digunakan untuk menghasilkan nilai kali dari 1 deret dengan mengalikan bantu dengan nilai j. dan setelah perulangan ke 2 selesai, bantu akan di print dan end=" " digunakan untuk memberi spasi dan tidak membuat baris baru.

Perulangan selanjutnya, masih di dalam perulangan untuk i, dengan j adalah range start i, stop 0, dan step -1, akan mengeluarkan output j.

bantu akan direset menjadi 1 pada akhir perulangan i dan kemudian print() untuk memberikan jarak antar baris.

Input, Output:

```
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
```

SOAL 3

Buatlah program untuk menampilkan deret seperti di bawah ini. n diinputkan secara dinamis.

Jawab:

```
1 tinggi = int(input("Tinggi = "))
2 lebar = int(input("Lebar = "))
3
4 a = 1
5 for i in range(tinggi):
6     for j in range(lebar):
7         print(a, end=" ")
8         a += 1
9     print()
```

Code ini menggunakan tinggi dan lebar sebagai variabel penampung input pengguna, a diinisialisasikan sebagai 1, kemudian perulangan dimulai untuk i memiliki range tinggi, kemudian perulangan ke 2 (perulangan bertingkat), j memiliki range lebar, perulangan j akan menghasilkan output a dan kemudian a akan ditambah dengan 1 setelah perulangan j selesai, print() digunakan untuk memberikan jarak antar baris.

Input, Output:

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

Link GitHub:

https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan_7