



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230997</b>
<b>Nama Lengkap</b>	<b>Christ Jevicto Ajimas Kirana</b>
<b>Minggu ke / Materi</b>	<b>6/Percabangan dan Perulangan Kompleks</b>

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

#### 1.1 Struktur Percabangan Kompleks

Ada beberapa bentuk percabangan yang kompleks =

1. Bentuk 1

```
# Bentuk 1
if kondisi1:
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 terpenuhi tapi kondisi2 tidak terpenuhi
```

2. Bentuk 2

```
# Bentuk 2
if kondisi1:
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 terpenuhi tapi kondisi2 tidak terpenuhi
else:
    # Lakukan sesuatu jika kondisi1 tidak terpenuhi
```

3. Bentuk 3

```
# Bentuk 3
if kondisi1:
    # Lakukan sesuatu jika kondisi1 terpenuhi
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 terpenuhi tapi kondisi2 tidak terpenuhi
        if kondisi3:
            # Lakukan sesuatu jika kondisi1 dan kondisi3 terpenuhi
        else:
            # Lakukan sesuatu jika kondisi1 terpenuhi tapi kondisi3 tidak terpenuhi
else:
    # Lakukan sesuatu jika kondisi1 tidak terpenuhi
```

4. Bentuk 4

```
# Bentuk 4
if kondisi1:
    # Lakukan sesuatu jika kondisi1 terpenuhi
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 terpenuhi tapi kondisi2 tidak terpenuhi
else:
    # Lakukan sesuatu jika kondisi1 tidak terpenuhi
    if kondisi3:
        # Lakukan sesuatu jika kondisi3 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 dan kondisi3 tidak terpenuhi
        if kondisi4:
            # Lakukan sesuatu jika kondisi1, kondisi3, dan kondisi4 terpenuhi
        else:
            # Lakukan sesuatu jika kondisi1, kondisi3, dan kondisi4 tidak terpenuhi
```

5. Bentuk 5

```
# Bentuk 5
if kondisi1:
    #lakukan sesuatu jika kondisi terpenuhi
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 terpenuhi
        if kondisi3:
            # Lakukan sesuatu jika kondisi1, kondisi2, dan kondisi3 terpenuhi
            if kondisi4:
                # Lakukan sesuatu jika kondisi1, kondisi2, kondisi3, dan kondisi4 terpenuhi
```

6. Bentuk 6

```
# Bentuk 6
if kondisi1:
    # Lakukan sesuatu jika kondisi1 terpenuhi
else:
    # Lakukan sesuatu jika kondisi1 tidak terpenuhi
    if kondisi2:
        # Lakukan sesuatu jika kondisi1 tidak terpenuhi tapi kondisi2 terpenuhi
    else:
        # Lakukan sesuatu jika kondisi1 dan kondisi2 tidak terpenuhi
        if kondisi3:
            # Lakukan sesuatu jika kondisi1 dan kondisi2 tidak terpenuhi tapi kondisi3 terpenuhi
        else:
            # Lakukan sesuatu jika kondisi1, kondisi2, dan kondisi3 tidak terpenuhi
            if kondisi4:
                # Lakukan sesuatu jika kondisi1, kondisi2, kondisi3, dan kondisi4 terpenuhi
            else:
                # Lakukan sesuatu jika kondisi1, kondisi2, kondisi3, dan kondisi4 tidak terpenuhi
```

## 1.2 Struktur Perulangan Kompleks

- **Break**

Perintah break digunakan untuk menghentikan proses perulangan yang dikarenakan oleh kondisi tertentu yang biasanya menggunakan perintah if. Ada 2 cara penggunaan break

1. Cara 1

```
for i in range(10):
    print(i, end=" ")
    if i == 5:
        break
```

[2] ✓ 0.0s

... 0 1 2 3 4 5

Penjelasan = Program diset untuk memberi output dari 0-10, karena ada perintah break di angka 5 maka hanya akan menampilkan angka sampai 5, hal ini karena perintah print diletakkan di atas break.

2. Cara ke-2

```
for i in range(10):
    if i == 5:
        break
    print(i, end=" ")
```

✓ 0.0s

0 1 2 3 4

Penjelasan = Program diset untuk memberi output dari 0-10, karena ada perintah break di angka 5 maka hanya akan menampilkan angka sampai 4, hal ini dikarenakan perintah print diletakkan di bawah break.

- **Continue**

Perintah continue digunakan untuk melompati pernyataan dalam suatu loop, yang berarti jika pernyataan terpenuhi maka perintah loop 'continue' tidak akan di eksekusi dan loop akan tetap berlanjut

```
for i in range(10):
    if i == 5:
        continue
    print(i, end=" ")
```

✓ 0.0s

0 1 2 3 4 6 7 8 9

Penjelasan = Program di set untuk memberikan output dari 0-10 karena ada pernyataan di angka 5 dan diberikan perintah continue, maka ketika pernyataan terpenuhi perintah loop 'continue' akan di eksekusi, oleh karena itu outputnya tidak ada angka 5.

- **Perulangan Bertingkat**

Perulangan bertingkat akan dilakukan jika kita memiliki perulangan di dalam perulangan lainnya. Perulangan bertingkat biasanya digunakan untuk menyelesaikan masalah yang melibatkan struktur data yang teratur seperti matriks. Dalam pemograman, perulangan bertingkat digunakan untuk mengatur loop di sepanjang baris dan kolom, dalam struktur grid atau pola lainnya.

Contoh dalam perintah for =

```
for i in range(2):
    for j in range(2):
        print ('l', end="") #lakukan perintah ini di dalam loop dalam
        print("x", end="") #lakukan perintah ini di luar loop dalam
```

✓ 0.0s

llxllx

Penjelasan = bagian 'for i in range' merupakan perulangan luar dan bagian 'for j in range' merupakan perulangan dalam

Contoh dalam While=

```
i = 0
while i < 2:
    j = 0
    while j < 2:
        print("l", end="") # Lakukan perintah ini di dalam loop dalam
        j+=1
    print("x", end="") # Lakukan perintah lain di luar loop dalam
    i+=1
```

✓ 0.0s

llxllx

Penjelasan = bagian 'while i<2' merupakan perulangan luar dan bagian 'while j<2' merupakan perulangan dalam

## MATERI 2

### Kegiatan Praktikum

2.1

```
for i in range(1,5+1):
    for j in range(1, i+1):
        print(i, end = " ")
    print()

✓ 0.0s
```

1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5

2.2

```
n = 5
for i in range(1, n+1):
    if i % 2 == 0:
        for j in range(1, n+1):
            print(j, end = " ")
    else:
        for j in range(n, 0, -1):
            print(j, end = " ")
    print()

✓ 0.0s
```

5 4 3 2 1  
1 2 3 4 5  
5 4 3 2 1  
1 2 3 4 5  
5 4 3 2 1

2.3

```
for i in range(0, 6+1):
    for j in range(1, 6-i+1):
        print("X", end = " ") if j%2 == 1 else print("[", end = " ")
    print()

✓ 0.0s
```

X [] X [] X []  
X [] X [] X  
X [] X []  
X [] X  
X []  
X  
X

```

x = 10
bar = x

for bar in range(bar, -1, -1):
    for kol in range(bar, 0, -1):
        hasil += " "
    for kanan in range(1, x - (bar-1)):
        hasil += "*"
    hasil = hasil + "\n"
print(hasil)

```

✓ 0.0s

```

      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
*****

```

2.4

```
*****
```

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

```
def mbp(angka):  
    if angka <=1:  
        return False  
    for i in range(2, int(angka**0.5)+1):  
        if angka % i == 0:  
            return False  
    return True  
  
def mbpt(p):  
    if p <=2:  
        return None  
    for i in range(p-1, 1, -1):  
        if mbp(i):  
            break  
    return i  
  
p = int(input('masukan angka= '))  
  
bpt = mbpt(p)  
  
if bpt != None:  
    print (f"bilangan prima terdekat kurang dari {p} adalah {bpt}")  
else:  
    print (f"tidak ada bilangan prima terdekat kurang dari {p}")
```

Penjelasan = pengguna menginput nilai untuk p lalu masuk ke fungsi def mbpt jika  $p \leq 2$  maka hasilnya none. Jika hasilnya bukan none maka masuk ke fungsi range yang akan mengirimkan tiap angka yang ada dalam range tersebut ke fungsi def mbp jika dalam range fungsi mbp menemukan angka prima pertama maka akan mengirimkan True ke dalam fungsi def mbpt lalu menjadi mbp(i) lalu fungsi langsung berhenti. Lalu program akan mengeprint fungsi mbpt(p).



## SOAL 2

```
def kali(hasil):
    if hasil == 0 or hasil == 1:
        return 1
    else:
        return hasil * kali(hasil-1)

tabnine: test | explain | document | ask
def deret(p):
    for i in range(p, 0, -1):
        print(kali(i), end=" ")
        for j in range(i, 0, -1):
            print(j, end=" ")

        print()

p = int(input("masukan bilangan= "))

deret(p)
```

Penjelasan = pengguna akan memberikan input yang masuk ke p, lalu p akan dihitung mundur hingga 1 dalam setiap hitung mundur akan mencetak fungsi def kali. Lalu mencetak bilangan mundur melalui for j in range dari i hingga 1. Setelah mencetak perkalian dan hitung mundur untuk i, lalu akan berpindah ke baris baru menggunakan print().

## SOAL 3

```
def deret(lebar, tinggi):
    mulai = 1
    hasil = "selesai"
    for i in range(tinggi):
        for j in range(lebar):
            print(mulai, end=" ")
            mulai += 1
        print()
    return hasil

lebar = int(input("masukan lebar= "))
tinggi = int(input("masukan tinggi= "))

print(deret(lebar, tinggi))
```

Penjelasan = pengguna akan memberikan 2 input untuk lebar dan tinggi, lalu memasukan nilai lebar dan tinggi untuk argument kedalam fungsi deret. Lalu hasil dicetak menggunakan pernyataan i.

LINK GITHUB= <https://github.com/Yeeemeki/praktikum-algoritma-dan-pemograman.git>