



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230978</b>
<b>Nama Lengkap</b>	<b>Jonathan Satriani Gracio Andrianto</b>
<b>Minggu ke / Materi</b>	<b>06 / 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

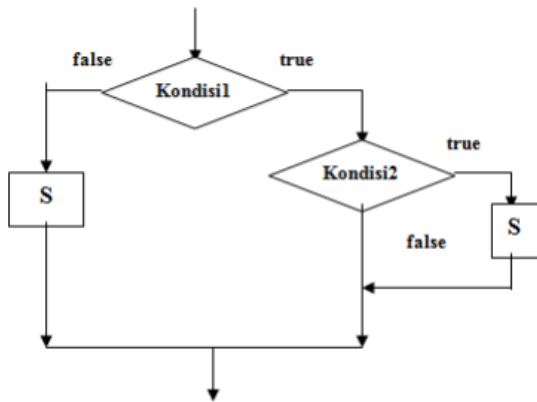
## BAGIAN 1: MATERI MINGGU INI (40%)

### Struktur Percabangan Kompleks

Struktur percabangan kompleks adalah percangan dimana kondisi pemilihan lebih dari satu atau terdiri dari banyak alternatif. Contoh bentuk percabangan kompleks :

```
if kondisi1:  
    if kondisi2:  
        S  
        S  
else:  
    S  
    S
```

Contoh bentuk percabangan kompleks

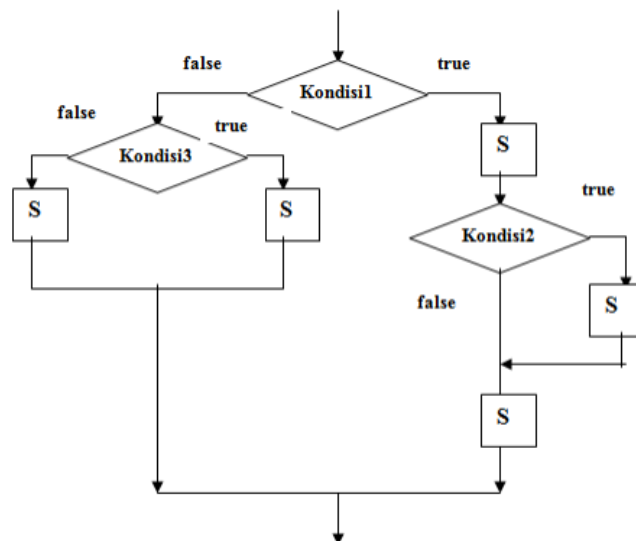


Contoh bentuk percabangan kompleks

Contoh lain bentuk percabangan kompleks :

```
if kondisi1:  
    S  
    if kondisi2:  
        S  
        S  
else:  
    if kondisi3:  
        S  
        S  
    else:  
        S  
        S
```

Contoh lain bentuk percabangan kompleks



Flowchart contoh lain bentuk percabangan kompleks

Percabangan kompleks berguna untuk menangani situasi yang kompleks atau yang memiliki banyak alternatif. Percabangan kompleks juga ini mengurangi kebutuhan untuk mengevaluasi setiap kondisi secara berurutan, karena dibagi ke dalam cabang-cabang tersendiri.

## Struktur Perulangan Kompleks

### Break

Perintah ini digunakan untuk menghentikan proses perulangan yang sedang terjadi ketika suatu kondisi terpenuhi. Contohnya :

```
for i in range(100):  
    print(i)  
    if i==10:  
        break
```

*Contoh penggunaan break*

Pada perulangannya, sudah ditentukan kalau ia akan mencetak angka 1 sampai 100. Tetapi karena ada perintah break ketika `i == 10`, maka program hanya mencetak angka 1 sampai 10

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

*Output*

```
for i in range(1000):  
    if i==10:  
        break  
    print(i)
```

*Contoh lain penggunaan break*

Program ini sama seperti program diatas, tetapi karena Perintah break diletakan sebelum perintah print nilai i. Hal tersebut menyebabkan nilai `i == 10` tidak dicetak.

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

*Output*

### Continue

Perintah continue berfungsi untuk melanjutkan proses perulangan ke iterasi berikutnya dan mengabaikan statement-statement dalam iterasinya saat ini. Contohnya :

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i)
```

0  
1  
2  
4

*Output*

*Contoh penggunaan continue*

Perulangan diatas seharusnya mencetak angka 1 sampai 5, tetapi karena ada perintah continue pada i == 3, alhasil perintah print i = 3 pun diabaikan. Makanya kita bisa lihat di output diatas angka 3 tidak dicetak atau dilewatkan

## Perulangan Bertingkat

Perulangan bertingkat atau perulangan kompleks adalah bentuk perulangan dimana ada perulangan didalam perulangan sama halnya seperti percabangan kompleks yang memiliki 'if' didalam 'if'. Contoh bentuk perulangan kompleks :

```
for i in range(m):
    for j in range(n):
        <lakukan perintah ini di inner>
        <lakukan perintah itu di inner>
    <lakukan perintah lain di outer>
    <lakukan perintah lain lagi di outer>
```

*Contoh bentuk perulangan kompleks for*

```
while(i<=m):
    while(j<=n):
        <lakukan perintah ini di inner>
        <lakukan perintah itu di inner>
    <lakukan perintah lain di outer>
    <lakukan perintah lain lagi di outer>
```

*Contoh bentuk perulangan kompleks while*

Bagian i menjadi outer loop, sedangkan bagian j menjadi inner loop. Alur pengerjaannya adalah sebagai berikut: untuk setiap 1x s/d m outer loop dijalankan akan dikerjakan n kali inner loop.

## Kegiatan Praktikum

### Contoh 6.1

```
n=int(input("Masukkan n = "))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(i," ",end='')
    print()
```

*Source Code*

```
Masukkan n = 4
1
2 2
3 3 3
4 4 4 4
```

*Output*

Penjelasan :

n=int(input("Masukkan n = ")) → meminta user memasukan input

for i in range(1,n+1): → perulangan outer

for j in range(1,i+1): → perulangan inner

print(i," ",end='') → mencetak nilai i sebanyak j kali

print() → mencetak spasi ke bawah sebanyak i kali

### Contoh 6.3

```
n=int(input("Masukkan n = "))
for i in range(0,n+1):
    for j in range(1,n-i+1):
        print("X",end='') if j%2==1 else print("O",end='')
    print()
```

Source Code

```
Masukkan n = 6
XOXOXO
XOXOX
XOXO
XOX
XO
X
```

Output

Penjelasan :

`n=int(input("Masukkan n = "))` → meminta user menginput nilai n

`for i in range(0,n+1):` → perulangan outer

`for j in range(1,n-i+1):` → perulangan inner

`print("X",end='') if j%2==1 else print("O",end='')` → print 'X' jika j modulus 2 == 1 atau j adalah ganjil, dan print 'O' jika j adalah bilangan genap

`print()` → mencetak spasi kebawah

#### Contoh 6.4

```
x = int(input("Masukkan jumlah :"))
bar = x
# Looping Baris
while bar >= 0:

    # Looping Kolom Spasi Kosong
    kol = bar
    while kol > 0:
        hasil += "  "
        kol -= 1

    # Looping Kolom Bintang
    kanan = 1
    while kanan < (x - (bar-1)):
        hasil += " * "
        kanan += 1

    hasil = hasil + "\n"
    bar -= 1

print (hasil)
```

Source Code

```
Masukkan jumlah :4

      *
     * *
    * * *
   * * * *
```

Output

## 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

```
def prima(bilangan) :  
    if bilangan <= 1 :  
        return False  
    for i in range (2, int(bilangan ** 0.5) + 1) :  
        if bilangan % i == 0 :  
            return False  
        else :  
            return True  
  
def primaterdekat(bil2) :  
    for i in range (bil2 - 1, 1, -1) :  
        if prima(i) :  
            return i  
  
bil2 = int(input("Masukkan bilangan = "))  
hasil = primaterdekat(bil2)  
print(f"Bilangan prima terdekat dari {bil2} adalah {hasil}")
```

#### Output

```
Masukkan bilangan = 12  
Bilangan prima terdekat dari 12 adalah 11
```

#### Penjelasan

- Fungsi prima(bilangan) bertujuan untuk mencari bilangan prima yang ada. Jika bilangan False maka bilangan tersebut bukan prima, melainkan jika bilangan mengembalikan nilai True maka bilangan adalah prima
- Fungsi primaterdekat(bil2) bertujuan untuk mencari bilangan prima yang kurang dari dan terdekat dari 'bil2' atau input yang dimasukan user

## SOAL 2

### Source Code

```
def angka(n):
    for i in range(n, 0, -1):
        print(str(faktor(i)) + " ", end="")
        for j in range(i, 0, -1):
            print(str(j) + " ", end="")
        print()

def faktor(bil):
    if bil == 1 or bil == 0:
        return 1
    return bil * faktor(bil - 1)

n = int(input("Masukkan n = "))
angka(n)
```

### Output

```
Masukkan 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 angka(n) bertujuan untuk mengecek factorial dan mencetak deret angka
- Fungsi faktor(bil) bertujuan untuk menghitung factorial dari variable 'bil'. Jika nilai 'bil' = 1 atau 0 maka fungsi akan mengembalikan nilai 1 yaitu factorial dari 1 dan 0. Jika nilai 'bil' bukan 1 atau 0 maka akan dilakukan perkalian terus menerus antara nilai 'bil' dan factorial dari 'bil - 1'



### SOAL 3

#### Source Code

```
tinggi = int(input("masukkan tinggi = "))
lebar = int(input("masukkan lebar = "))
i = 0
j = 1
l = tinggi * lebar
while i < l :
    print(j, " ", end = "")
    i += 1
    j += 1
    if i % (lebar) == 0 :
        print()
```

#### Output

```
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

```
tinggi = int(input("masukkan tinggi = "))
lebar = int(input("masukkan lebar = "))
i = 0
j = 1
l = tinggi * lebar
```

Bagian ini hanya untuk meminta input user, memasang nilai counter dan memberi rumus luas

```
while i < l :
    print(j, " ", end = "")
    i += 1
    j += 1
    if i % (lebar) == 0 :
        print()
```

Jadi perulangan terjadi saat nilai 'i' kurang dari nilai 'l' atau lebar. 'j' adalah nilai yang dicetak untuk menjadi lebar. Ada kondisi dimana jika 'i' modulus lebar == 0, atau 'i' kelipatan lebar maka akan dicetak sebuah spasi dan menjadi baris baru atau tingginya.

LINK GITHUB

<https://github.com/JonathanAndrianto123/JonathanAndrianto123.git>