



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71200581</b>
<b>Nama Lengkap</b>	<b>Dewangga Yuka Pratama</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%)

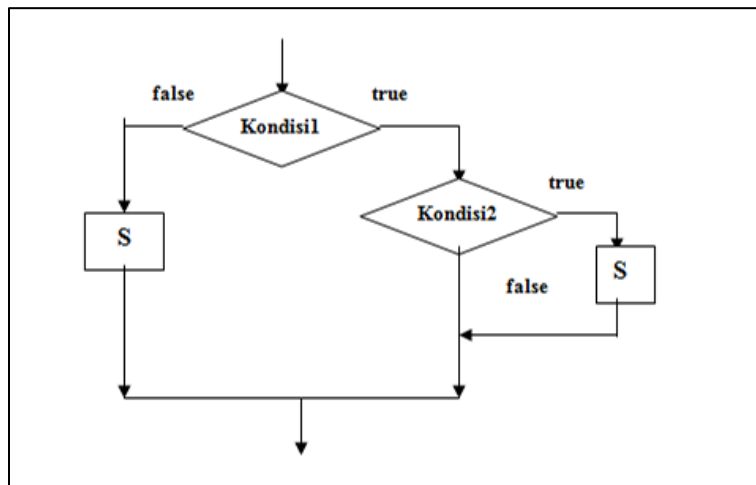
Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

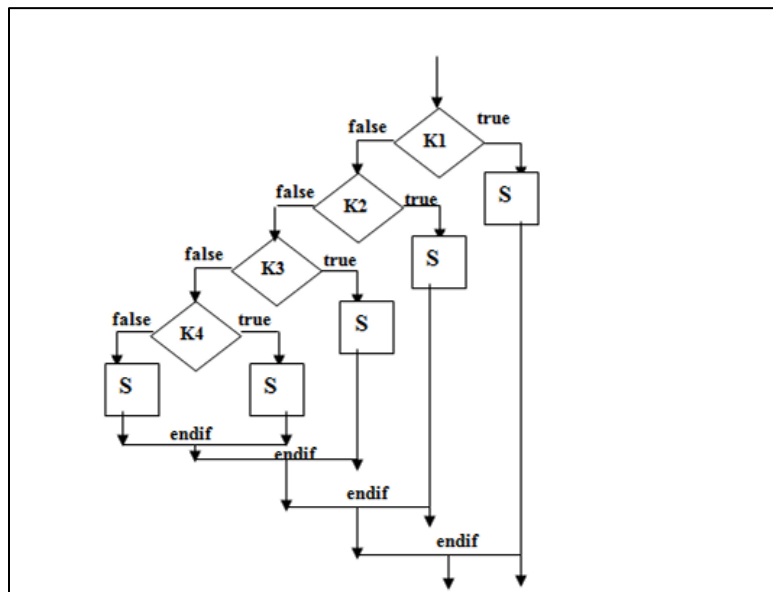
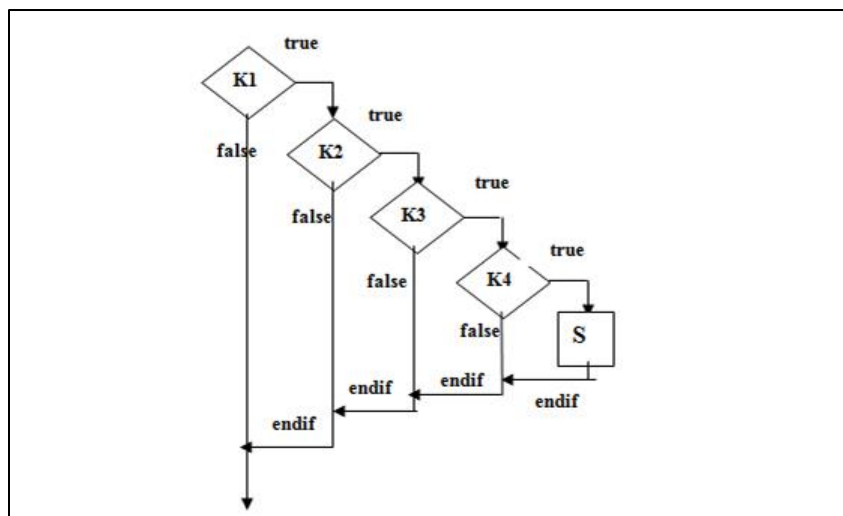
### Struktur Percabangan Kompleks

Struktur percabangan kompleks merupakan struktur kontrol dalam pemrograman yang memungkinkan program untuk menjalankan berbagai instruksi berdasarkan beberapa kondisi yang saling terkait. Struktur ini lebih kompleks daripada struktur percabangan sederhana seperti if dan else karena melibatkan lebih dari dua kondisi dan memungkinkan pengujian kondisi secara bertingkat.

Percabangan ini terjadi ketika kita memiliki struktur percabangan di dalam struktur percabangan lainnya. Ini berguna ketika kita perlu mengevaluasi beberapa kondisi secara berurutan atau berjenjang. Percabangan dengan banyak kondisi terjadi ketika kita memiliki lebih dari dua pilihan. Contohnya adalah menggunakan pernyataan if-elif-else dalam Python. Ini memungkinkan kita untuk mengevaluasi banyak kondisi dan memilih jalur yang sesuai berdasarkan kondisi tersebut.

Ini merupakan beberapa gambaran dari bagaimana percabangan digambarkan :





Ini contoh dari percabangan kompleks:

```
2
3 total_belanja = 150000
4 member = True
5 poin = 100
6
7 if total_belanja > 100000:
8     if member:
9         if poin >= 100:
10             diskon = 0.2
11         else:
12             diskon = 0.1
13     else:
14         diskon = 0.15
15 else:
16     diskon = 0
17
18 print("Diskon yang Anda dapatkan:", diskon * 100, "%")
19
```

Pada contoh ini, program akan menentukan besarnya diskon berdasarkan total belanja, keanggotaan, dan jumlah poin yang dimiliki. Jika total belanja lebih dari Rp100.000, maka program akan mempertimbangkan apakah pelanggan adalah anggota atau tidak, dan jumlah poin yang dimilikinya. Jika total belanja kurang dari Rp100.000, maka tidak ada diskon yang diberikan.

## Struktur Perulangan Kompleks

Struktur perulangan kompleks biasa digunakan untuk menggambarkan penggunaan struktur kontrol perulangan yang lebih kompleks, seperti penggunaan kondisi yang kompleks dalam perulangan. Sebagai contoh adalah sebagai berikut :

```
2
3 tinggi = 5
4 for i in range(tinggi):
5     for j in range(i+1):
6         print('*', end='')
7     print()
```

Contoh ini menghasilkan pola segitiga sesuai dengan variabel tingginya.

Pada perulangan kompleks ini terdapat juga perintah break dan continue. Break digunakan untuk menghentikan loop secara paksa, bahkan jika kondisi loop belum selesai. Ketika break dieksekusi, program akan keluar dari loop dan melanjutkan eksekusi dari pernyataan setelah loop sedangkan continue digunakan untuk melompati sisa iterasi loop dan langsung ke iterasi berikutnya. Dengan kata lain, jika continue dieksekusi, blok kode di dalam loop setelah continue akan dilewati, dan loop akan melanjutkan dengan iterasi berikutnya. Contohnya seperti berikut :

```
2
3 for i in range(5):
4     if i == 3:
5         break
6     print(i)
7
```

Atau

```
2
3 for i in range(5):
4     if i == 3:
5         continue
6     print(i)
7
```

Contoh diatas menggunakan for namun juga dapat menggunakan while.

## 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

```
2
3 def prima(bil):
4     if bil < 2:
5         return False
6     for i in range(2, int(bil ** 0.5) + 1):
7         if bil % i == 0:
8             return False
9     return True
10
11 def cari_prima(n):
12     prima_terdekat = None
13     for i in range(n - 1, 1, -1):
14         if prima(i):
15             prima_terdekat = i
16             break
17     return prima_terdekat
18
19 n = int(input("Masukkan bilangan (n): "))
20 prima_terdekat = cari_prima(n)
21
22 if prima_terdekat:
23     print(f"Bilangan prima terdekat yang kurang dari {n} adalah {prima_terdekat}.")
24 else:
25     print(f"Tidak ditemukan bilangan prima yang kurang dari {n}.")
```

```
Masukkan bilangan (n): 21
Bilangan prima terdekat yang kurang dari 21 adalah 19.
```

Langkah :

1. Buat fungsi prima
2. Periksa apakah bil kurang dari 2. Jika iya, maka fungsi langsung mengembalikan False karena bilangan prima harus lebih besar dari 1.
3. melakukan iterasi dari 2 hingga akar kuadrat dari bil (dilambangkan dengan  $\text{int}(\text{bil}^{**} 0.5) + 1$ ). Iterasi dimulai dari 2 karena setiap bilangan lebih kecil daripada 2 pasti bukan bilangan prima.
4. memeriksa apakah bil habis dibagi oleh bilangan lain selain 1 dan dirinya sendiri. Jika ya, maka bil bukan bilangan prima dan fungsi mengembalikan False. Jika tidak ada bilangan yang membagi habis bil, maka bil adalah bilangan prima dan fungsi mengembalikan True.
5. Buat fungsi cari\_prima() untuk menemukan prima terdekat
6. Fungsi ini melakukan iterasi mundur dimulai dari n-1 (bilangan sebelum n) hingga 2 (karena bilangan prima terkecil adalah 2).
7. Pada setiap iterasi, fungsi memanggil fungsi prima untuk memeriksa apakah bilangan tersebut adalah bilangan prima.
8. Jika ditemukan bilangan prima, nilai tersebut disimpan sebagai bilangan prima terdekat dalam variabel prima\_terdekat dan iterasi dihentikan dengan break.
9. Melakukan percabangan untuk menentukan bilangan tersebut prima atau bukan
- 10.

## SOAL 2

```
2
3 def faktor(bil):
4     if bil == 0:
5         return 1
6     else:
7         return bil * faktor(bil - 1)
8
9 n = int(input("Masukkan nilai n: "))
10
11 for i in range(n, 0, -1):
12     print(faktor(n), end=" ")
13     for j in range(i, 0, -1):
14         print(j, end=" ")
15     print()
16     n -= 1
```

```
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
```

Langkah:

1. Membuat fungsi faktor
2. Fungsi faktor akan akan mengembalikan 1 jika argumen adalah 0. Jika tidak, fungsi akan mengembalikan hasil perkalian antara bil dan pemanggilan rekursif dari faktor(bil - 1).
3. Membuat fungsi input untuk nilai n
4. Membuat Loop for yang akan berjalan dari n hingga 1 dengan langkah -1. Variabel i akan digunakan sebagai pembantu untuk mengatur iterasi.
5. Print faktor(n) untuk mencetak nilai dari n
6. Loop lagi untuk mencetak deret angka dari i hingga 1 dengan langkah -1.

## Soal 3

```
1
2 def tampil_deret(tinggi, lebar):
3     bil_awal = 1
4     for i in range(1, tinggi + 1):
5         for j in range(1, lebar + 1):
6             print(bil_awal, end=" ")
7             bil_awal += 1
8         print()
9
10 tinggi = int(input("Masukkan tinggi deret: "))
11 lebar = int(input("Masukkan lebar deret: "))
12 tampil_deret(tinggi, lebar)
```

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

Langkah:

1. Buat fungsi tampil\_deret() dengan parameter tinggi dan lebar
2. Variabel bil\_awal digunakan untuk menyimpan bilangan awal deret
3. Loop for pertama digunakan untuk mengontrol tinggi dari deret yang akan ditampilkan. Loop ini akan berjalan dari 1 hingga nilai tinggi yang dimasukkan.
4. Loop for kedua digunakan untuk mengontrol lebar dari deret yang akan ditampilkan di setiap baris. Loop ini akan berjalan dari 1 hingga nilai lebar yang dimasukkan.
5. print(bil\_awal, end=" ") akan mencetak nilai bil\_awal di baris yang sama dengan spasi yang telah ditetapkan.
6. Bil\_awal += 1 akan membuat tiap angka ditampilkan nilai awal akan ditambahkan dengan 1 agar deret terus bertambah
7. Memasukkan inputan
8. Panggil fungsi