



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230976
Nama Lengkap	Galih Pramana Chandra Prasetya
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

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 digunakan saat terdapat lebih dari satu kondisi pemilihan. Perintah-perintah yang dikerjakan juga bisa lebih dari satu.

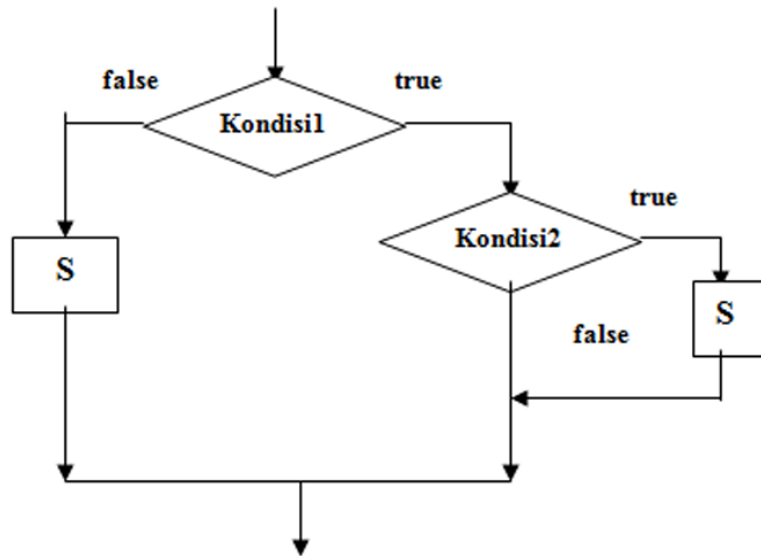
Pada kasus tertentu if bertingkat memiliki keuntungan tersendiri, dengan menggunakan if bertingkat tersebut maka eksekusi perintah menjadi lebih baik dan efisien.

Menggunakan if bertingkat maka tidak semua kondisi if dikerjakan sehingga waktu eksekusi lebih cepat dibandingkan dengan if biasa yang semua kondisi if pasti akan dicoba satu per satu walaupun mungkin pada akhirnya hanya satu saja perintah if yang terpenuhi.

Percabangan kompleks bentuk 1:

```
1  if kondisi1:
2      if kondisi2:
3          S
4          S
5      else:
6          S
7          S
```

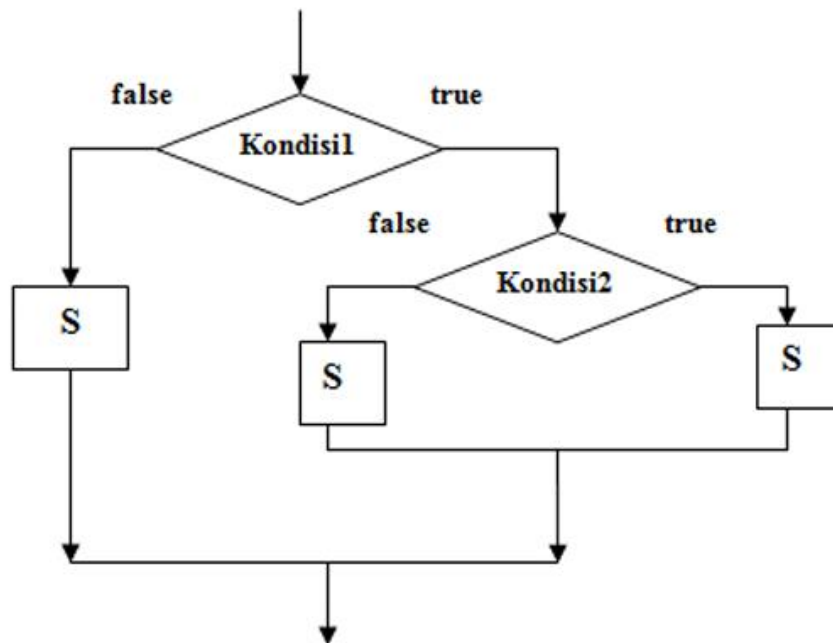
Berikut ini gambar flowchart nya:



(diambil dari modul 6)

Percabangan kompleks bentuk 2:

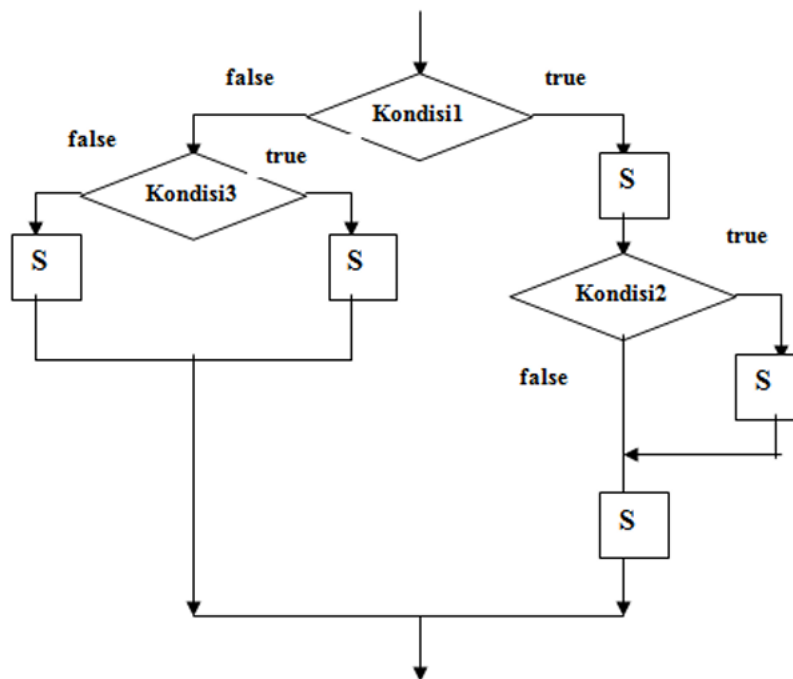
```
1  if kondisi1:
2      if kondisi2:
3          S
4          S
5      else:
6          S
7          S
8  else:
9      S
10     S
```



(diambil dari modul 6)

Percabangan kompleks bentuk 3:

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

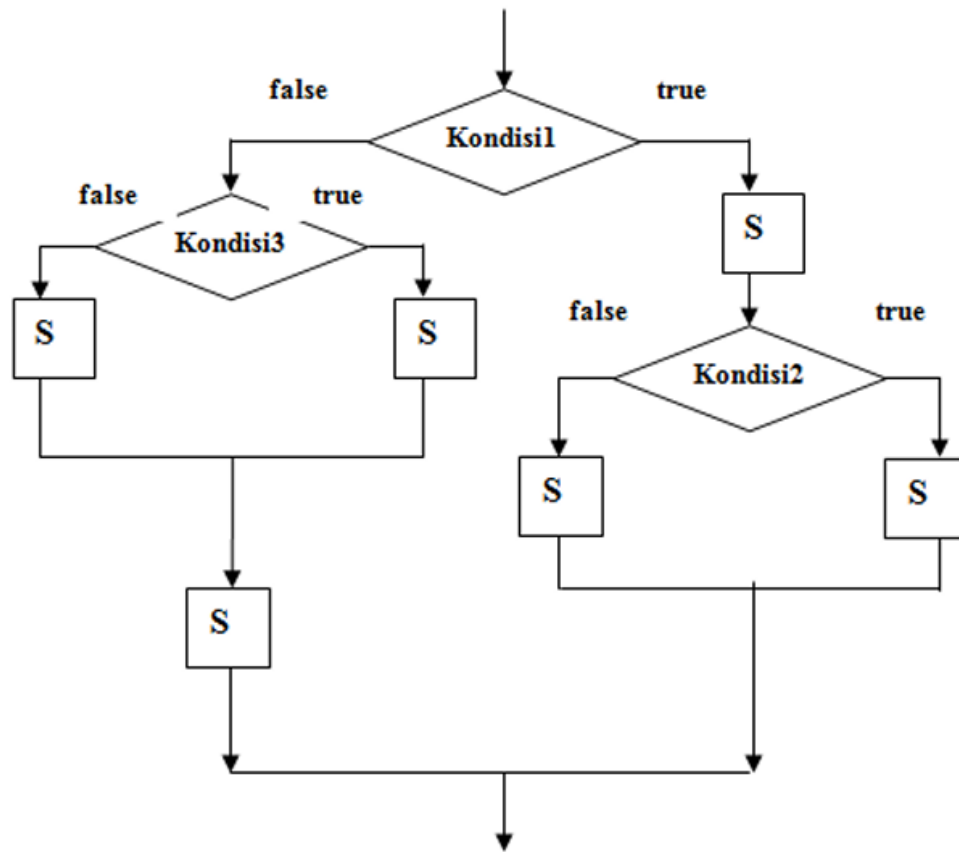


(diambil dari modul 6)

Percabangan kompleks bentuk 4:

```

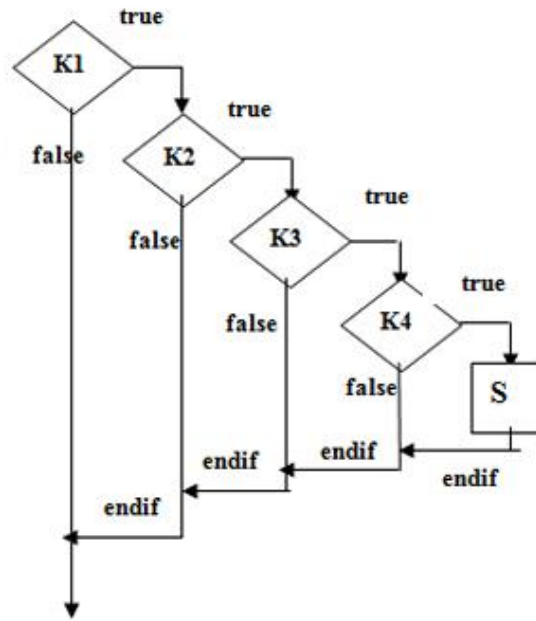
if kondisi1:
    S
    if kondisi2:
        S
        S
    else:
        S
        S
else:
    if kondisi3:
        S
        S
    else:
        S
        S
S
  
```



(diambil dari modul 6)

Percabangan kompleks bentuk 5:

```
if kondisi1:
    if kondisi2:
        if kondisi3:
            if kondisi4:
                S
```

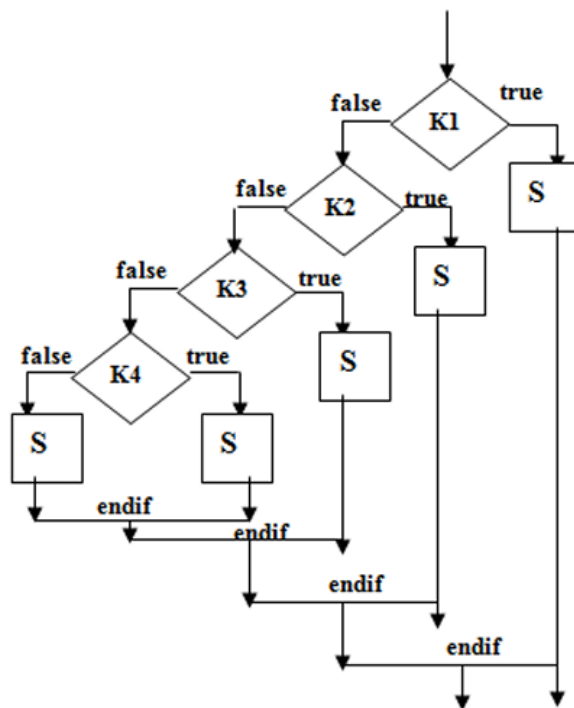


(diambil dari
modul 6)

Percabangan kompleks bentuk 6:

```

if kondisi1:
    S
else:
    if kondisi2:
        S
    else:
        if kondisi3:
            S
        else:
            if kondisi4:
                S
            else:
                S
  
```



(diambil

dari modul 6)

Struktur Perulangan Kompleks


Break

Fungsi break adalah perintah khusus yang dipakai untuk memaksa sebuah perulangan berhenti sebelum waktunya. Biasanya disebabkan oleh suatu kondisi tertentu yang diimplementasikan menggunakan perintah if.

Contoh penggunaan break:

```

i = 1
while i <= 10:
    print(i, " X ", i, " = ", i*i)
    if i == 6:
        break
    i += 1
  
```

```
1 X 1 = 1
2 X 2 = 4
3 X 3 = 9
4 X 4 = 16
5 X 5 = 25
6 X 6 = 36
```


Pada contoh diatas terdapat kode program dengan perulangan while untuk menghasilkan daftar perkalian 1 x 1, 2 x 2 hingga 10 x 10. Didalam perulangan while ini, pada baris ke 4 terdapat pemeriksaan kondisi if i = 6. Jika di kondisi ini terpenuhi maka perulangan langsung dihentikan ketika sudah sampai angka 6.

Continue

Continue berfungsi untuk lompat (skip) ke iterasi selanjutnya tanpa harus mengeksekusi sisa kode yang ada di bawahnya. Biasanya perintah continue juga diimplementasikan menggunakan perintah IF.

Contoh penggunaan continue:

```
for i in range(1, 10):
    if i == 5:
        continue
    print(i)
```



```
1
2
3
4
6
7
8
9
```

Ketika kondisi dimana i sama dengan 5 sudah terpenuhi, maka perintah continue akan dijalankan. Pemanggilan ini akan menyebabkan iterasi berlanjut ke nilai i selanjutnya tanpa mencetak angka 5.

Perulangan Bertingkat

Perulangan bertingkat adalah sebuah perulangan yang berada atau terletak di dalam perulangan yang lain, sehingga terjadi perulangan bertingkat yang mengakibatkan waktu proses semakin lama.

Contoh perulangan bertingkat:

```
for i in range(3):    # Perulangan luar
    for j in range(2): # Perulangan dalam
        print("Nilai i:", i, ", Nilai j:", j)
```

Perulangan for “for i in range(3)” mengatur variabel i dari 0 hingga 2, dan perulangan for “for j in range(2)” mengatur variabel j dari 0 hingga 1. Setiap kombinasi nilai i dan j dicetak. Oleh karena perulangan dalam berada di dalam perulangan luar, setiap iterasi dari perulangan luar akan menjalankan perulangan dalam. Sehingga program mencetak semua kombinasi nilai i dan j yang mungkin.

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 is_prime(n):  
    if n <= 1:  
        return False  
    for i in range(2, int(n**0.5) + 1):  
        if n % i == 0:  
            return False  
    return True  
  
N = int(input("Masukkan bilangan: "))  
for i in range(N - 1, 1, -1):  
    if is_prime(i):  
        print(f"input n = {N}, maka prima terdekat < {N} adalah {i}")  
        break
```

Output:

```
➞ Masukkan bilangan: 12  
input n = 12, maka prima terdekat < 12 adalah 11
```

```
➞ Masukkan bilangan: 21  
input n = 21, maka prima terdekat < 21 adalah 19
```

Penjelasan:


- Fungsi **is_prime** digunakan untuk memeriksa apakah suatu bilangan adalah bilangan prima.
- Fungsi **is_prime** mengembalikan True jika bilangan yang diberikan adalah bilangan prima dan False jika tidak.
- Pengguna diminta untuk memasukkan bilangan N.
- Memakai loop **for** untuk mengiterasi dari N - 1 ke bawah hingga 2.
- Pada setiap iterasi, program memeriksa apakah bilangan yang sedang dicek adalah bilangan prima menggunakan fungsi **is_prime**.
- Jika iya, program mencetak bahwa bilangan prima terdekat yang lebih kecil dari N adalah bilangan tersebut, dan kemudian program berhenti.
- Jika tidak, program akan terus mencari hingga menemukan bilangan prima yang sesuai.

SOAL 2

Source code:

```
 def faktor(n):  
    hasil = 1  
    for i in range(1, n + 1):  
        hasil = hasil * i  
    return hasil  
  
def faktorial(n):  
    i = bilangan  
    while i > 0:  
        print(faktor(i), end=" ")  
        for j in range(i, 0, -1):  
            print(j, end=" ")  
        i = i - 1  
        print()  
  
bilangan = int(input("bilangan = "))  
faktorial(bilangan)
```

Output:

```
 bilangan = 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 **faktor(n)**:
 - Menghitung nilai faktorial dari bilangan n.
- Fungsi **faktorial(n)**:
 - Mencetak deret faktorial untuk bilangan dari n hingga 1.
- Bagian Utama Program:
 - Meminta pengguna untuk memasukkan nilai bilangan.
 - Memanggil fungsi faktorial(bilangan) untuk mencetak deret faktorial.

SOAL 3

Source code:

```
#Latihan Mandiri 6.3
def deret(tinggi, lebar):
    nilai = 1
    for i in range(1, tinggi+1):
        for j in range(1, lebar+1):
            print(nilai, end=" ")
            nilai = nilai + 1
        print()

tinggi = int(input("Masukkan tinggi deret: "))
lebar = int(input("Masukkan lebar deret: "))

deret(tinggi, lebar)
```

Output:

```
➞ 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
```

Penjelasan:

- Fungsi **deret(tinggi, lebar)**:
 - mencetak deret angka berdasarkan tinggi dan lebar yang diinput
 - Loop **for i in range(tinggi)** digunakan untuk mengontrol jumlah baris dalam deret.
 - **for j in range(lebar)**, mengontrol jumlah angka dalam setiap baris deret.
- Bagian Utama Program:
 - Meminta pengguna untuk menginput nilai tinggi dan lebar deret.
 - Memanggil fungsi **deret(tinggi, lebar)** untuk menampilkan deret angka sesuai dengan input pengguna.

Link Github:

<https://github.com/GalihPramana/Praktikum-Alpro-71230976.git>