

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



| | |
|--------------------|--|
| NIM | 71200645 |
| Nama Lengkap | Edith Felicia Putri |
| 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
2023

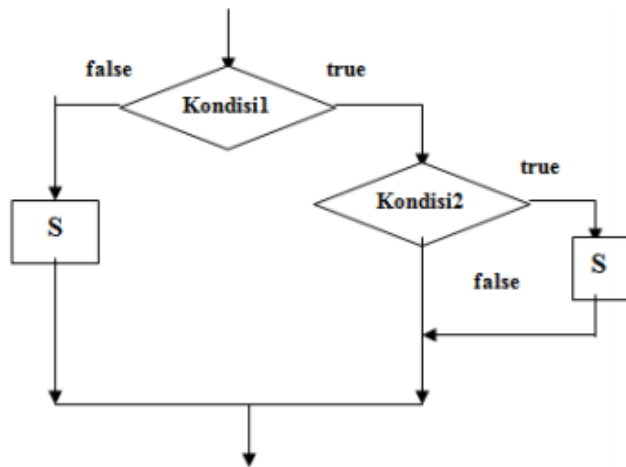
BAGIAN 1: MATERI MINGGU INI (40%)

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

STRUKTUR PERCABANGAN KOMPLEKS

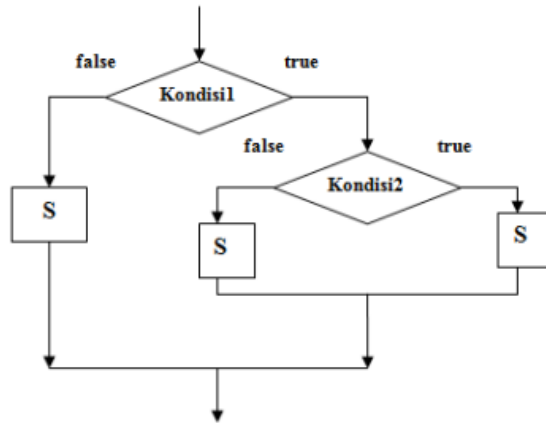
Percabangan dimana kondisi pemilihan tidak hanya satu, tetapi bisa terdiri atas banyak alternatif. Perintah-perintah yang dikerjakan juga bisa lebih dari satu.



Gambar 1. Flowchart Percabangan Kompleks Bentuk 1

SOURCE CODE:

```
x = int(input("Masukkan bilangan 1 < x < 5: "))
if 1 < x < 5:
    if x == 2:
        print("Angka kamu bagus sekali!")
else:
    print("Okaaay")
```

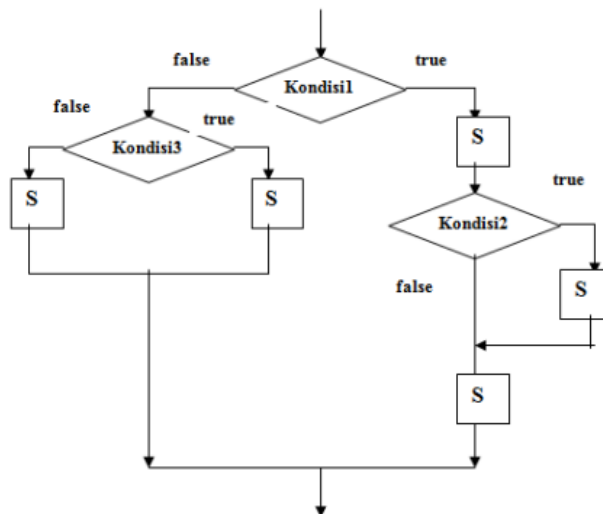


Gambar 2. Flowchart Percabangan Kompleks Bentuk 2

SOURCE CODE:

```

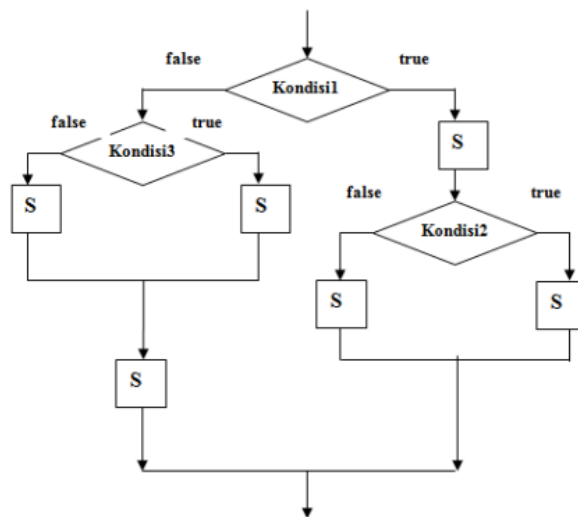
x = int(input("Pilih angka dari 1 - 10: "))
if x >= 1 and x <= 10:
    if x == 5:
        print("Angka yang kuinginkan")
    else:
        print("Bukan angka yang kuinginkan")
else:
    print("Anda salah memasukkan angka")
  
```



Gambar 3. Flowchart Percabangan Kompleks Bentuk 3

SOURCE CODE:

```
print("Selamat datang di restoran!")
x = input("Memulai pesan? y/n: ")
if x == "y":
    print("1. Keju Mozarella")
    print("2. Hamburger")
    pesanan = int(input("Masukkan angka pesanan Anda: "))
    if pesanan == 1:
        print("Terima kasih atas Keju Mozarellanya!")
    elif pesanan == 2:
        print("Terima kasih atas Hamburgernya!")
    else:
        print("Tidak ada menu yang lain")
else:
    if x == "n":
        print("Terima kasih sudah mendatangi toko!")
    else:
        print("Anda salah memasukkan pilihan")
```



Gambar 4. Flowchart Percabangan Kompleks Bentuk 4

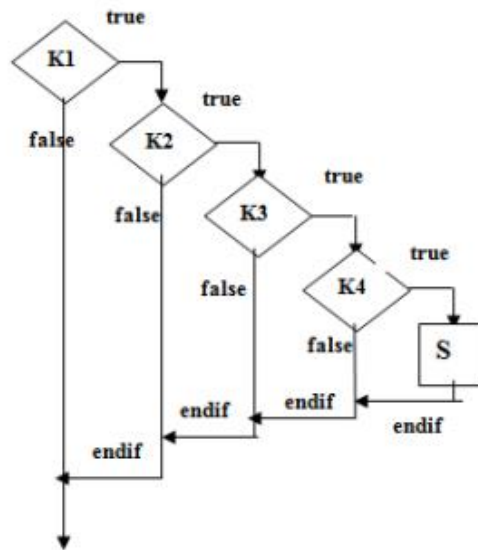
SOURCE CODE:

```
x = int(input("Masukkan angka antara 1 - 10: "))
if 1 <= x <= 10:
    y = int(input("Masukkan angka antara 50 - 100: "))
    if 50 <= y <= 100:
        y = y + 50
    print("nilai y:", y)
```

```

else:
    if y > 100:
        print("Nilai y sudah lebih dari 100:", y)
    else:
        print("Nilai y kurang dari 50:", y)
else:
    if x <= 0:
        print("Nilai x memasukkan nilai negatif atau nol")
    else:
        print("Nilai x sudah lebih dari 10")
print("Terima kasih sudah memasukkan angka")

```



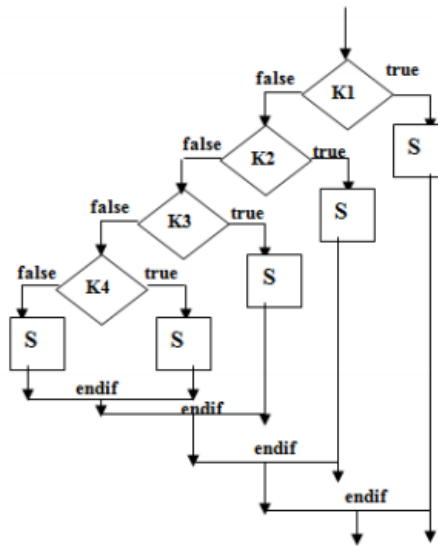
Gambar 5. Flowchart Percabangan Kompleks Bentuk 5

SOURCE CODE:

```

x = int(input("Masukkan angka antara 1 - 10: "))
if 1 <= x <= 10:
    if x < 8:
        if x < 5:
            if x == 2:
                print("Angka kesukaanku!")
else:
    print("Angka yang Anda masukkan salah")

```



Gambar 6. Flowchart Percabangan Kompleks Bentuk 6

SOURCE CODE:

```

x = int(input("Masukkan angka: "))
if x > 100:
    print("Semakin tinggi")
else:
    if x < 0:
        print("Sudah negatif")
    else:
        if 10 <= x <= 100:
            print("Pertengahan")
        else:
            if x == 1:
                print("Hello 1")
            else:
                print("Bukan yang pertama")
  
```

MATERI 2

STRUKTUR PERULANGAN KOMPLEKS

Tidak semua bentuk WHILE dapat diubah menjadi bentuk FOR, sementara semua bentuk FOR dapat diubah ke bentuk WHILE.

Break

Perintah ini digunakan untuk menghentikan proses perulangan yang sedang terjadi. Biasanya disebabkan oleh suatu kondisi tertentu yang diimplementasikan menggunakan perintah IF.

Continue

Perintah *continue* menyebabkan proses perulangan kembali ke awal mula, dengan mengabaikan statement-statement berikutnya setelah *continue*. Biasanya perintah *continue* juga diimplementasikan menggunakan perintah IF.

Perulangan Bertingkat

Bentuk perulangan dimana dalam suatu perulangan terdapat perulangan lain. Bagian for i menjadi *outer loop*, sedangkan bagian for j menjadi *inner loop*. Masalah yang dapat diselesaikan dengan perulangan kompleks adalah masalah matriks yang menggunakan array 2 dimensi, masalah game board seperti catur dan minesweeper, ataupun masalah pengolahan citra digital seperti algoritma untuk mendeteksi tepi citra, algoritma untuk mengubah citra berwarna menjadi *grayscale*, atau berbagai hal lain. Intinya, masalah yang dapat diselesaikan menggunakan perulangan kompleks biasanya memiliki pola grid (kotak-kotak) yang memiliki lebar dan panjang.

SOURCE CODE:

```
for i in range(5):  
    for j in range(2):  
        print("Hello, Moza")  
    print("Goodbye, Andra")
```

OUTPUT:

```
python.exe d:/71200645_PrakAlpro_6/modul.  
py  
Hello, Moza  
Hello, Moza  
Goodbye, Andra  
Hello, Moza  
Hello, Moza  
Goodbye, Andra  
Hello, Moza  
Hello, Moza  
Goodbye, Andra  
Hello, Moza  
Hello, Moza  
Goodbye, Andra  
Hello, Moza  
Hello, Moza  
Goodbye, Andra  
PS D:\71200645_PrakAlpro_6> █
```

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

Latihan 6.1 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 ■

Pengerjaan soal dilakukan dengan pengecekan apakah angka kurang dari atau sama dengan 2, karena tidak ada bilangan prima yang lebih kecil dari angka 2. Kalau angka lebih dari 2, dilanjutkan ke dalam *for i loop* dari angka-1 hingga 2 dengan pengurangan 1. Bilangan prima dinyatakan *True*. *For j loop* dilakukan untuk memeriksa apakah angka berupa prima atau tidak. Kalau ada bilangan yang membagi i ($i \% j == 0$), maka i itu bukan bilangan prima, dan variabel prima diubah menjadi *False*, lalu *looping* pun dihentikan dengan *break*. Bilangan yang terakhir kali di-*loop* dengan variabel prima = *True* adalah bilangan prima yang akan di-*return*.

SOURCE CODE:

```
def prima_terdekat():
    angka = int(input("Masukkan bilangan: "))
    if angka <= 2:
        return None

    for i in range(angka - 1, 1, -1):
        prima = True
        for j in range(2, int(i ** 0.5) + 1):
            if i % j == 0:
                prima = False
                break
        if prima:
            print(f"Bilangan prima terdekat < {angka} adalah {i}")
            return i

prima_terdekat()
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_6> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_6/Latihan 6.1.py"
Masukkan bilangan: 12
Bilangan prima terdekat < 12 adalah 11
PS D:\71200645_PrakAlpro_6> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_6/Latihan 6.1.py"
Masukkan bilangan: 21
Bilangan prima terdekat < 21 adalah 19
PS D:\71200645_PrakAlpro_6> █
```


SOAL 2

Latihan 6.2 Buatlah program untuk menampilkan deret seperti di bawah ini. n diinputkan secara dinamis

```
contoh: 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
```

Pengerjaan soal dilakukan dengan memasukkan jumlah n (baris sebanyak berapa). Variabel jumlah digunakan untuk menyimpan perkalian nantinya dan diakhir setiap *inner loop* akan mengembalikan nilainya menjadi 1 lagi. Lalu *for loop i* sebanyak nilai n, dan *for loop j* dari nilai n sampai 1 dengan pengurangan 1, selama looping dilakukan perkalian nilai-nilai dalam variabel *temporary j*. Kemudian selesai *looping* akan di-print. Dilakukan *for loop k* dari nilai n sampai 1 dengan pengurangan 1 lagi dengan menge-print nilai variabel *temporary k*. Dilakukan pengurangan nilai n diakhir *looping* untuk berganti ke nilai n selanjutnya.

SOURCE CODE:

```
def deret_bilangan():
    nilai = int(input("n = "))
    jumlah = 1
    for i in range (nilai): # untuk buat baris
        for j in range(nilai, 0, -1): # Berhenti di-1
            jumlah = jumlah * j # hasil perkalian (6 x 5 x 4 x 3 x 2 x 1)
        print(jumlah, end=" ")
        for k in range(nilai, 0, -1):
            print(k, end=" ")
        jumlah = 1 # Untuk me-reset nilai variable hitung
        nilai -= 1
        print()
deret_bilangan()
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_6> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_6/Latihan 6.2.py"
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
PS D:\71200645_PrakAlpro_6> █
```

SOAL 3

Latihan 6.3 Buatlah program untuk menampilkan deret seperti di bawah ini. n diinputkan secara dinamis

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

Pengerjaan soal dilakukan dengan memasukkan nilai tinggi dan lebar dari pengguna. Tinggi menjadi *outer loop* (i) yang menentukan sebanyak apa baris dari program, dan lebar menjadi *inner loop* (j), lalu deretan angka akan bertambah satu di setiap *looping*-nya.

SOURCE CODE:

```
def deret_bilangan() :
    tinggi = int(input("Masukkan tinggi = "))
    lebar = int(input("Masukkan lebar = "))
    deret = 1
    for i in range (tinggi) :
        for j in range (lebar) :
            print(deret, end=" ")
            deret += 1
        print()
    deret_bilangan()
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_6> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python
.exe "d:/71200645_PrakAlpro_6/Latihan 6.3.py"
"
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
PS D:\71200645_PrakAlpro_6> █
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_6