Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230985**
Nama Lengkap	**TOMAS BECKET**
Minggu ke / Materi	**06 / Percabangan & 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

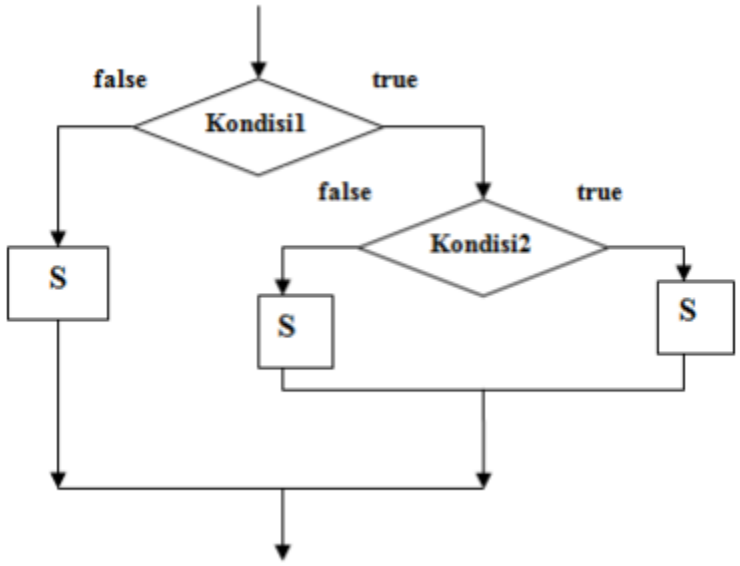
BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1

Struktur percabangan kompleks yaitu percabangan yang dimana kondisi pemilihan tidak hanya satu tetapi terdiri dari banyak kondisi dan perintah yang diprogram lebih dari satu.. contoh :

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

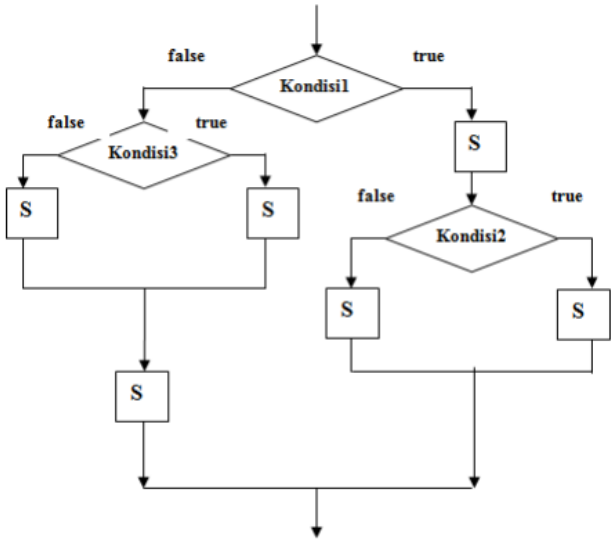
Source code diatas menunjukan bahwa kondisi yang dimiliki pada program tersebut yaitu 2 kondisi. Sehingga jika dilihat dalam bentuk flowchart maka :



Adapun bentuk percabangan menggunakan 3 kondisi sekaligus di percabangan kompleks sekaligus dengan flowchartnya dimana:

Jika kondisi satu memenuhi maka print s, jiak kondisi 2 memenuhi maka print s. namun jika kondisi 2 memenuhi maka gunakan else pada kondisi 2, ketika pada kondisi 1 tidak memenuhi sama sekali maka lanjut ke kondisi 3 dengan mengskip kondisi 2 dikarenaka kondisi 2 akan di check jika memenuhi kondisi 1. Jiak kondisi 3 memenuhi maka print, jika tidak maka else.

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



MATERI 2

Sama halnya dengan percabangan kompleks, perulangan sendiri memiliki hal yang kompleks juga seperti perulangan bertingkat contoh for i in range bilangan, dan for j in range bilangan, dimana fungsi tersebut memiliki 2 perulangan yang berjalan sendiri-sendiri. Maka ketika perulangan kedua dimasukan ke perulangan 1 maka menjadi :

```
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 code seperti dibawah ini :

Variabel n digunakan untuk input user memasukan sebuah bilangan atau angka dan fungsi for i in range (n, 0, -1) akan membuat baris mulai dari angka terbesar dan kekecil dari atas kebawah, sedangkan for j in range(i, 0, -1) akan membuat kolom dengan angka yang sama sesuai i dengan berjejer ke arah kanan sesuai angka yang dimilikinya.

```
n = int(input("masukan n = "))
for i in range(n, 0, -1):
    for j in range(i, 0, -1):
        print(i," ", end='')
    print()
```

Hasil output:

```
masukan n = 5
5  5  5  5  5
4  4  4  4
3  3  3
2  2
1
```

MATERI 3

Adapun source code dengan penggunaan perulangan dan percabangan, contohnya seperti dibawa ini

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

Awalnya program meminta user untuk memasukan sebuah nilai ke variabel n menggunakan fungsi input. Kemudian Loop luar (for i in range(1, n+1)) berfungsi untuk mengulang sebanyak n kali, dimulai dari 1 hingga n (inklusif). Variabel iterasi i digunakan untuk menentukan baris yang sedang dicetak.

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

Di dalam loop luar, terdapat sebuah kondisi (if i%2 == 1:) yang memeriksa apakah i (nomor baris saat ini) adalah ganjil atau genap.

Jika Ganjil (if i%2 == 1): Program menjalankan loop dalam pertama yang mencetak angka dari 1 hingga n secara berurutan pada baris yang sama. Ini terjadi untuk setiap baris ganjil.

```
else:
    for j in range(n, 0, -1):
        print(j," ", end='')
```

Jika genap maka else yaitu dengan mencetak n hingga 1 secara berurutan pada baris yang sama.

Hasil Output :

```
masukan n = 5
1  2  3  4  5
5  4  3  2  1
1  2  3  4  5
5  4  3  2  1
1  2  3  4  5
```

MATERI 4

Sedangkan pada source code ini menggunakan perulangan kompleks dan percabangan menggunakan ternery operator

```
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()
```

Sama seperti sebelumnya, variabel n digunakan untuk menyimpan hasil input user yaitu sebuah bilangan atau angka. Fungsi for i in range(0, n+1) digunakan untuk membuat baris kebawah sesuai n. sedangkan pada bagia for i in range(1, n-i+1) akan membuat angka urut mulai dari 1 hingga 5 dan dikurang -1 untuk baris kebawah.

Pada fungsi percabangan ini lah untuk menentukan mana angka ganjil dan angka genap yang akan di print sebagai X atau O nantinya.

Hasil output:

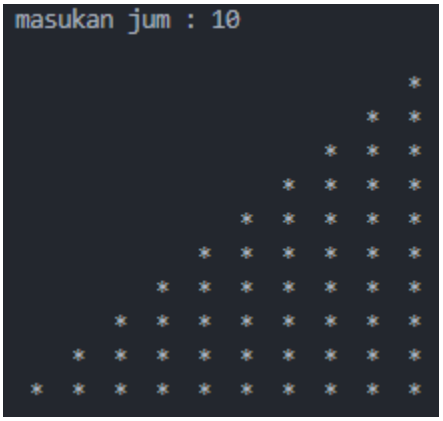
```
Masukkan n = 5
X O X O X
X O X O
X O X
X O
X
```

MATERI 5

```
hasil = ""

x = int(input("masukan jum : "))
bar = x

while bar >= 0:

    kol = bar
    while kol > 0:
        hasil += "    "
        kol -= 1

    kanan = 1
    while kanan < (x - (bar-1)):
        hasil += " * "
        kanan += 1

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

print(hasil)
```

Dan juga ada fungsi perluangan menggunakan while untuk membuat sebuah segitiga siku-siku ke arah kanan

Hasil output :



BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

1.1 SOURCE CODE

```python
def prima(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def cari_prima(num):
    for i in range(num - 1, 1, -1):
        if prima(i):
            return i
    return None

n = int(input("Masukkan bilangan n: "))
prima_terdekat = cari_prima(n)

if prima_terdekat:
    print(f"Bilangan prima terdekat < {n} adalah {prima_terdekat}")
else:
    print(f"Tidak ditemukan bilangan prima terdekat < {n}")
```

1.2 PENJELASAN

Pertama digunakan fungsi prima(num) untuk mengambil sebuah nilai integer num dan memeriksa apakah nilai num merupakan bilangan prima atau bukan.

```python
if num < 2:
    return False
```

Dan if num < 2 digunakan untuk mengembalikan nilai True jika num merupakan bilangan prima dan False sebaliknya.

Dan karena bilangan prima dimulai dari angka 2 dan tidak kurang dari angka 2 maka digunakan < 2

```python
for i in range(2, int(num ** 0.5) + 1):
    if num % i == 0:
        return False
return True
```

Dan fungsi iterasi dari angka 2 sampai akar kuadrat num int(untuk memeriksa apakah num dapat dibagi oleh angka selain 1 dan dirinya sendiri. Jika ditemukan bilangan yang memenuhi, maka num bukan bilangan prima dan fungsi mengembalikan False

```
def cari_prima(num):
    for i in range(num - 1, 1, -1):
        if prima(i):
            return i
    return None
```

Sedangkan fungsi cari_prima(num) digunakan untuk mengambil sebuah integer num sebagai input dan memeriksa apakah itu merupakan bilangan prima. Fungsi ini melakukan iterasi dari num - 1 sampai 2, mencari bilangan prima pertama yang ditemukan dengan memanggil fungsi prima(i). Jika bilangan prima ditemukan, fungsi akan langsung mengembalikan bilangan prima tersebut. Jika tidak ditemukan bilangan prima dalam rentang tersebut, fungsi mengembalikan None.

```
n = int(input("Masukkan bilangan n: "))
prima_terdekat = cari_prima(n)

if prima_terdekat:
    print(f"Bilangan prima terdekat < {n} adalah {prima_terdekat}")
else:
    print(f"Tidak ditemukan bilangan prima terdekat < {n}")
```

kode di bawah definisi fungsi meminta pengguna untuk memasukkan sebuah bilangan n, mencari bilangan prima terdekat yang lebih kecil dari n menggunakan fungsi cari_prima(n), dan mencetak hasilnya. Jika bilangan prima terdekat ditemukan, program akan mencetak "Bilangan prima terdekat < n adalah prima_terdekat". Jika tidak, program akan mencetak "Tidak ditemukan bilangan prima terdekat < n".

1.3 HASIL OUTPUT

```
Masukkan bilangan n: 12
Bilangan prima terdekat < 12 adalah 11
```

```
Masukkan bilangan n: 27
Bilangan prima terdekat < 27 adalah 23
```

SOAL 2

2.1 SOURCE CODE

```
n = int(input("masukan bilangan n = "))

for i in range(n, 0, -1):
    faktorial = 1
    for j in range(i, 0, -1):
        faktorial *= j

    angka_menurun = ' '.join(str(k) for k in range(i, 0, -1))
    print(f"{faktorial} {angka_menurun}")
```

2.2 PENJELASAN

```
n = int(input("masukan bilangan n = "))
```

Pertama-tama, program meminta user untuk memasukkan sebuah bilangan bulat **n** menggunakan fungsi **input()**. Kemudian, nilai tersebut diubah menjadi integer menggunakan **int()**.

```
for i in range(n, 0, -1):
    faktorial = 1
    for j in range(i, 0, -1):
        faktorial *= j
```

Program kemudian memulai perulangan pertama menggunakan for i in range(n, 0, -1). Perulangan ini dimulai dari n dan berakhir di 1 (inklusif) dengan langkah mundur sebesar 1 setiap kali iterasi dilakukan.

```
faktorial = 1
```

Pada setiap iterasi perulangan pertama, program menginisialisasi variabel faktorial dengan nilai 1, yang akan digunakan untuk menyimpan hasil faktorial dari angka yang sedang diproses.

```
for j in range(i, 0, -1):
    faktorial *= j
```

Kemudian, program memulai perulangan kedua menggunakan for j in range(i, 0, -1). Perulangan ini dimulai dari nilai i (yang saat itu merupakan nilai dari perulangan pertama) dan berakhir di 1 (inklusif) dengan langkah mundur sebesar 1 setiap kali iterasi dilakukan. Pada setiap iterasi perulangan kedua, program mengalikan nilai faktorial dengan nilai j (yang saat itu merupakan nilai dari perulangan kedua). Ini dilakukan untuk menghitung faktorial dari angka yang sedang diproses.

```
angka_menurun = ' '.join(str(k) for k in range(i, 0, -1))
print(f"{faktorial} {angka_menurun}")
```

Setelah selesai perulangan kedua, program membuat string angka_menurun yang berisi angka-angka dari i hingga 1 dalam urutan menurun, dipisahkan oleh spasi. Ini dilakukan dengan menggunakan fungsi ' '.join() bersama dengan generator untuk mengonversi angka-angka tersebut menjadi string.

2.3 HASIL OUTPUT

```
masukan bilangan 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
```

```
masukan bilangan n = 8
40320 8 7 6 5 4 3 2 1
5040 7 6 5 4 3 2 1
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
```

SOAL 3

3.1 SOURCE CODE

```
tinggi = int(input("tinggi = "))
lebar = int(input("lebar = "))
angka = 1

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

3.2 PENJELASAN

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

User akan memasukkan nilai tinggi dan lebar dari persegi panjang yang ingin dibuat. Nilai-nilai ini diinput melalui fungsi input(), di mana int(input("tinggi = ")) meminta user memasukkan tinggi, dan int(input("lebar = ")) untuk lebar. Input ini kemudian dikonversi menjadi tipe data integer.

```
angka = 1
```

ariabel angka diinisialisasi dengan nilai 1. Variabel ini akan digunakan untuk mencetak angka yang bertambah dalam pola.

```
for i in range(tinggi):
```

Perulangan pertama (for i in range(tinggi)): bertanggung jawab atas tinggi dari pola. Perulangan ini berjalan sebanyak nilai tinggi yang diinput pengguna.

```
for j in range(lebar):
    print(angka, end=" ")
    angka += 1
```

Dalam setiap iterasi dari perulangan luar, perulangan kedua (for j in range(lebar)): dijalankan. Perulangan ini bertanggung jawab atas lebar dari pola. Dalam setiap iterasi, perulangan ini mencetak nilai variabel angka dan kemudian menambahkannya dengan 1 (angka += 1). Hal ini memastikan bahwa setiap angka yang dicetak bertambah secara berurutan.

```
print(angka, end=" ")
```

Dalam setiap iterasi perulangan dalam, kode print(angka, end=" ") mencetak nilai angka saat itu, dengan menambahkan spasi setelahnya dan tidak membuat baris baru karena parameter end=" ".

```
print()
```

Setelah setiap iterasi perulangan dalam selesai (setelah mencetak angka sepanjang lebar), print() dipanggil tanpa argumen untuk mencetak baris baru. Hal ini memastikan bahwa setiap baris dari pola dicetak di baris baru, sesuai dengan tinggi yang diinginkan.

3.3 HASIL OUTPUT

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

```
tinggi = 6
lebar = 5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
26 27 28 29 30
```