



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71220876
Nama Lengkap	DEVINA ELISSE PUTRI
Minggu ke / Materi	06 / Pengantar Python

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

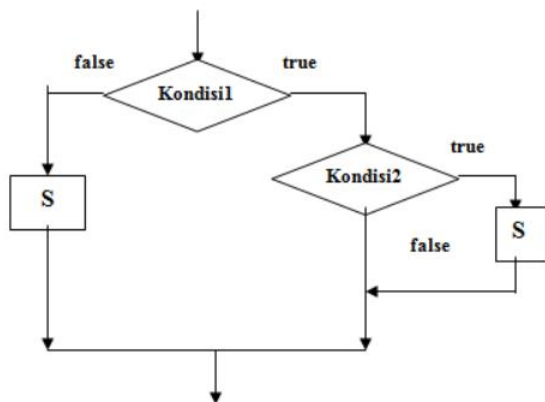
BAGIAN 1: MATERI MINGGU INI (40%)

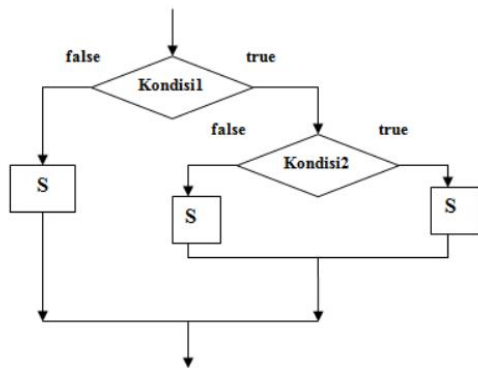
Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6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

MATERI 1

1.

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



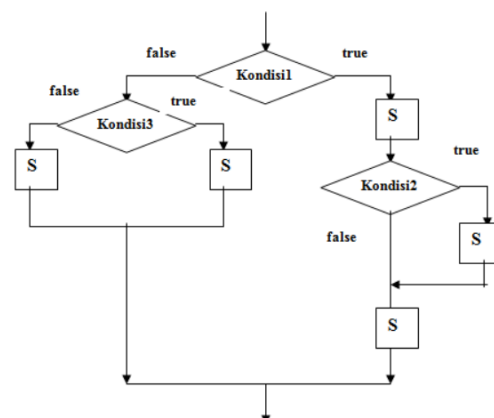


```

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

```

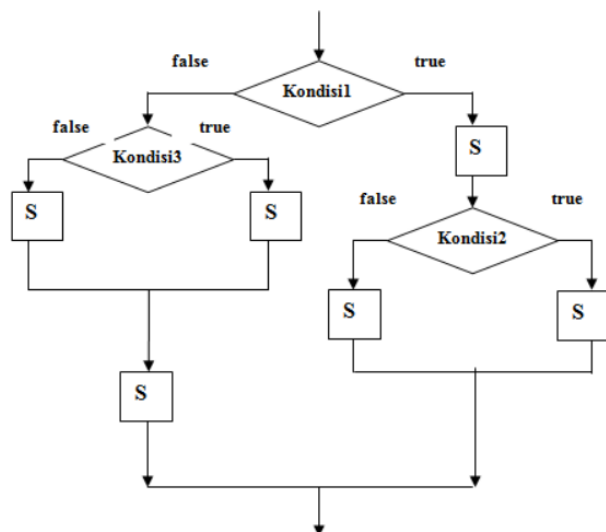
1  if kondisi1:
2      S
3      if kondisi2:
4          S
5          S
6  else:
7      if kondisi3:
8          S
9          S
10     else:
11         S
12         S
  
```



```

1  if kondisi1:
2      S
3      if kondisi2:
4          S
5          S
6      else:
7          S
8          S
9  else:
10     if kondisi3:
11         S
12         S
13     else:
14         S
15         S
16 S

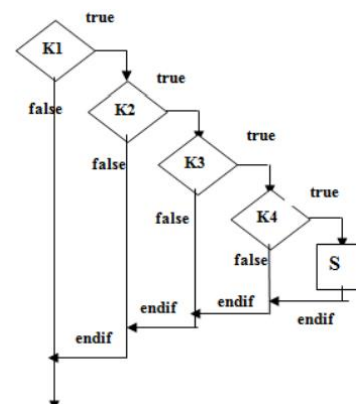
```



```

1  if kondisi1:
2      if kondisi2:
3          if kondisi3:
4              if kondisi4:
5                  S

```



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

```
1 if (kondisi1):
2     instruksi1
3 elif(kondisi2):
4     instruksi2
5 elif(kondisi3):
6     instruksi3
7 elif(kondisi4):
8     instruksi4
```

Bedakan dengan:

```
1 if (kondisi1):
2     instruksi1
3 if(kondisi2):
4     instruksi2
5 if(kondisi3):
6     instruksi3
7 if(kondisi4):
8     instruksi4
```

MATERI 2

```
1 for i in range(1000):
2     print(i)
3     if i==10:
4         break
```

Output:

```
0
1
2
3
4
5
6
7
8
9
10
```

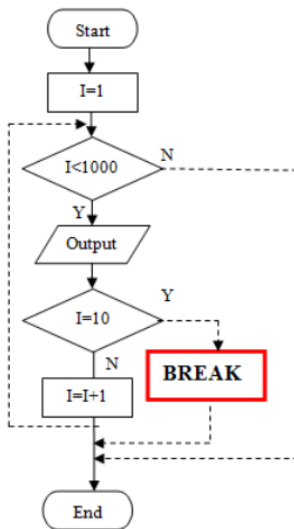
Program tersebut akan mencetak angka 1 hingga 10 dalam perulangan, meskipun telah diatur hingga 1000, karena adanya perintah break saat i=10, sehingga angka 10 tetap dicetak karena perintah cetak diletakkan sebelum break.

```
1 for i in range(1000):  
2     if i==10:  
3         break  
4     print(i)
```

Output:

0
1
2
3
4
5
6
7
8
9

Program tersebut mencetak angka 1 hingga 9 dalam perulangan, meskipun perulangan telah diatur hingga 1000. Ini karena adanya perintah break saat i=10. Angka 10 tidak dicetak karena perintah cetak berada setelah perintah break.

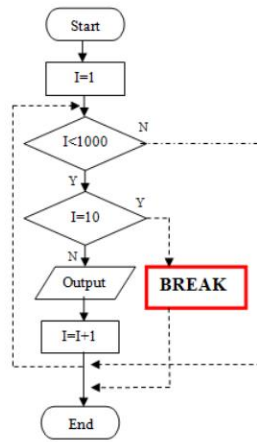


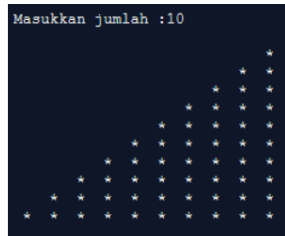
Perintah continue mengembalikan perulangan ke awal, mengabaikan pernyataan setelahnya, seringkali diimplementasikan dengan if.

```
1 for i in range(10):  
2     if i==5:  
3         continue  
4     print(i)
```

Output:

0
1
2
3
4
6
7
8
9





Gambar 6.10: Soal Segitiga

```

hasil = ""

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)

```

```

1  n=int(input("Masukkan n = "))
2  for i in range(1,n+1):
3      if i%2==1:
4          for j in range(1,n+1):
5              print(j," ",end='')
6      else:
7          for j in range(n,0,-1):
8              print(j," ",end='')
9      print()

```

■ **Contoh 6.3** Anda diminta untuk membuat suatu deret dengan tampilan sebagai berikut:

```

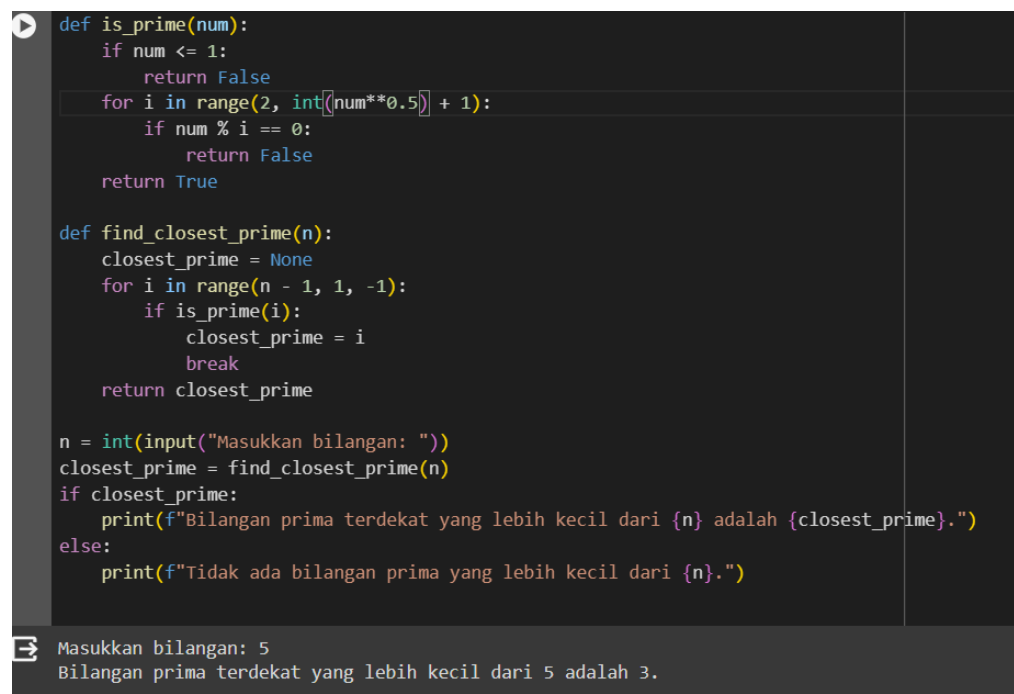
n=6
X 0 X 0 X 0
X 0 X 0 X
X 0 X 0
X 0 X
X 0
X

```


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



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

def find_closest_prime(n):
    closest_prime = None
    for i in range(n - 1, 1, -1):
        if is_prime(i):
            closest_prime = i
            break
    return closest_prime

n = int(input("Masukkan bilangan: "))
closest_prime = find_closest_prime(n)
if closest_prime:
    print(f"Bilangan prima terdekat yang lebih kecil dari {n} adalah {closest_prime}.")
else:
    print(f"Tidak ada bilangan prima yang lebih kecil dari {n}.")
```

Masukkan bilangan: 5
Bilangan prima terdekat yang lebih kecil dari 5 adalah 3.

- Fungsi ini bertujuan untuk memeriksa apakah suatu bilangan `num` adalah bilangan prima atau bukan.
- Pertama-tama, fungsi memeriksa apakah `num` kurang dari atau sama dengan 1. Jika ya, maka `num` bukan bilangan prima, dan fungsi akan mengembalikan `False`.
- Selanjutnya, dilakukan iterasi dari 2 hingga akar kuadrat dari `num` menggunakan `for` loop.

- Dalam setiap iterasi, fungsi memeriksa apakah ``num`` habis dibagi dengan ``i``. Jika ya, maka ``num`` bukan bilangan prima, dan fungsi akan mengembalikan ``False``.
- Jika tidak ditemukan pembagi selain 1 dan ``num`` itu sendiri, fungsi akan mengembalikan ``True``, menandakan bahwa ``num`` adalah bilangan prima.

2. Fungsi ``find_closest_prime(n)``:

- Fungsi ini bertujuan untuk mencari bilangan prima terdekat yang lebih kecil dari bilangan ``n`` yang diinputkan.
- Fungsi menggunakan fungsi ``is_prime(num)`` untuk memeriksa apakah suatu bilangan merupakan bilangan prima.
- Dimulai dari bilangan yang lebih kecil dari ``n`` (yaitu ``n - 1``), fungsi akan mengecek setiap bilangan secara berurutan dari ``n - 1`` hingga 2.
- Jika fungsi ``is_prime()`` mengembalikan ``True`` untuk suatu bilangan, maka bilangan tersebut akan dianggap sebagai bilangan prima terdekat dan akan disimpan dalam variabel ``closest_prime``. Setelah itu, loop akan dihentikan dengan ``break``.

3. Baris kode utama:

- Program meminta pengguna untuk memasukkan bilangan ``n``.
- Kemudian, program akan mencari bilangan prima terdekat yang lebih kecil dari ``n`` menggunakan fungsi ``find_closest_prime(n)``.
- Jika bilangan prima terdekat ditemukan (``closest_prime`` tidak bernilai ``None``), maka program akan mencetak pesan yang menyatakan bilangan prima terdekat yang ditemukan. Jika tidak ditemukan, program akan mencetak pesan yang menyatakan bahwa tidak ada bilangan prima yang lebih kecil dari ``n``.

SOAL 2

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

def print_series(n):
    for i in range(n, 0, -1):
        for j in range(i, 0, -1):
            print(j, end=" ")
        print()

n = int(input("Masukkan nilai n: "))
for i in range(n, 1, -1):
    print(factorial(i), end=" ")
    print_series(i)
print("1")
```

Penjelasan singkat:

1. Fungsi `factorial(n)` digunakan untuk menghitung faktorial dari suatu bilangan `n`.
2. Fungsi `print_series(n)` digunakan untuk mencetak deret angka dari `n` hingga 1.
3. Program meminta input dari pengguna untuk nilai `n`.
4. Kemudian, program mencetak deret seperti yang diminta dengan menghitung faktorial dari `n` dan mencetak deret angka menggunakan fungsi `print_series()`.

SOAL 3

```
def print_pattern(tinggi, lebar):
    for i in range(1, tinggi + 1):
        for j in range((i - 1) * lebar + 1, i * lebar + 1):
            print(j, end=" ")
        print()

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

print_pattern(tinggi, lebar)
```

1. Fungsi `print_pattern(tinggi, lebar)` digunakan untuk mencetak pola seperti yang diminta.
2. Dalam setiap baris, angka dimulai dari `(i-1)*lebar + 1` hingga `i*lebar`. Ini karena setiap baris dimulai dari angka sebelumnya ditambah satu dan berlanjut hingga angka sebelumnya ditambah lebar baris.
3. Program meminta input dari pengguna untuk tinggi dan lebar deret.

4. Kemudian, program mencetak pola menggunakan fungsi ``print_pattern()``.