

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71220939
Nama Lengkap	Sadrakh Satria Wibowo
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.

MATERI 1

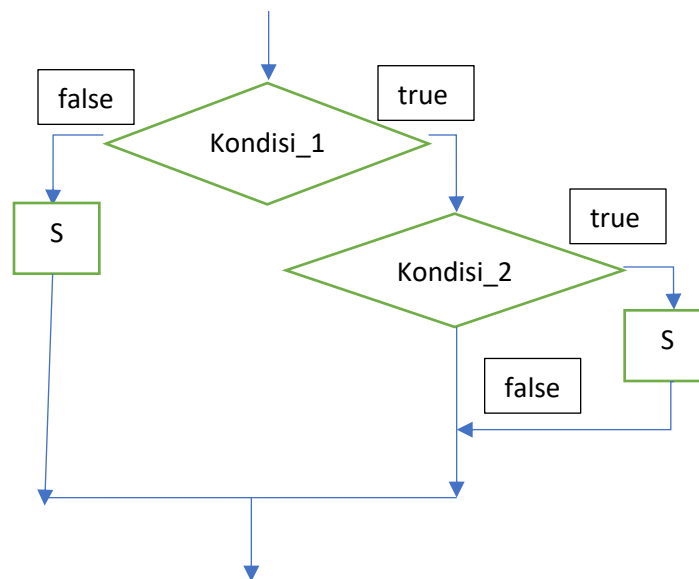
6.3 Materi

6.3.1 Struktur Percabangan Kompleks

Percabangan ini tidak hanya satu pemilihan tetapi bisa lebih banyak alternatif. Selain itu perintah perintah yang dikerjakan bisa lebih dari satu. Contoh percabangan kompleks bentuk pertama :

```
if kondisi_1 :  
    if kondisi_2 :  
        S  
        S  
else :  
    S  
    S
```

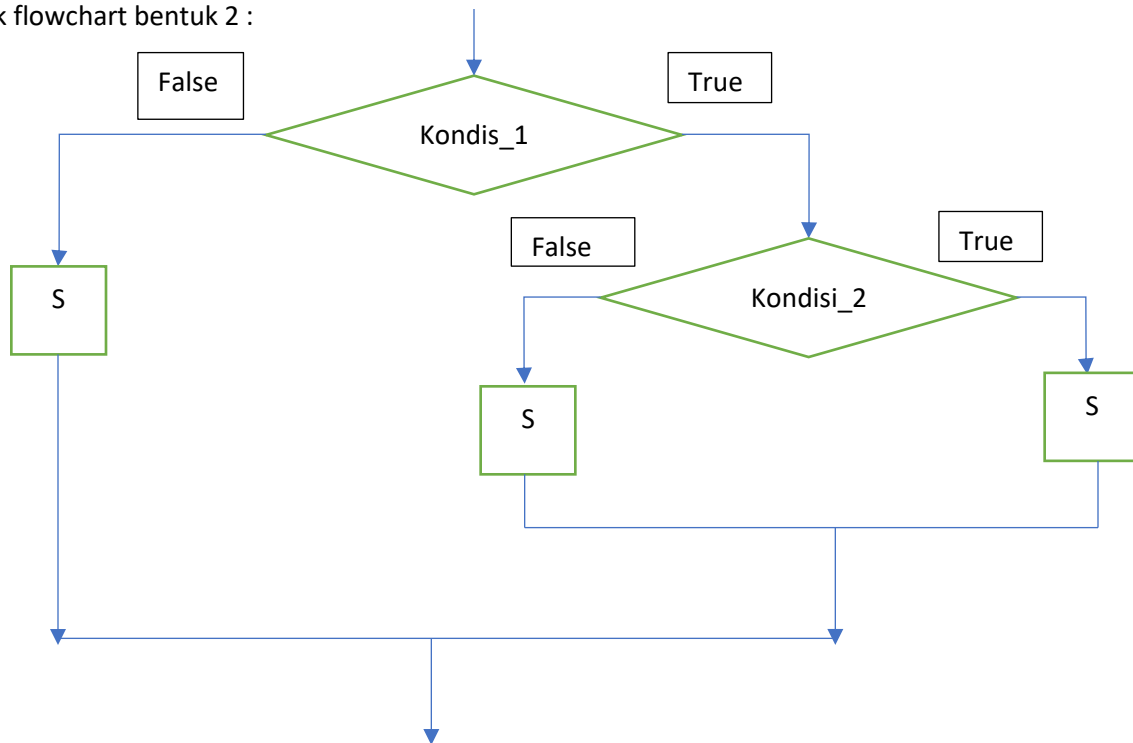
Bentuk flowchart bentuk 1:



Percabangan kompleks bentuk 2:

```
if kondisi_1 :  
    if kondisi_2 :  
        S  
        S  
    else :  
        S  
        S  
else :  
    S  
    S
```

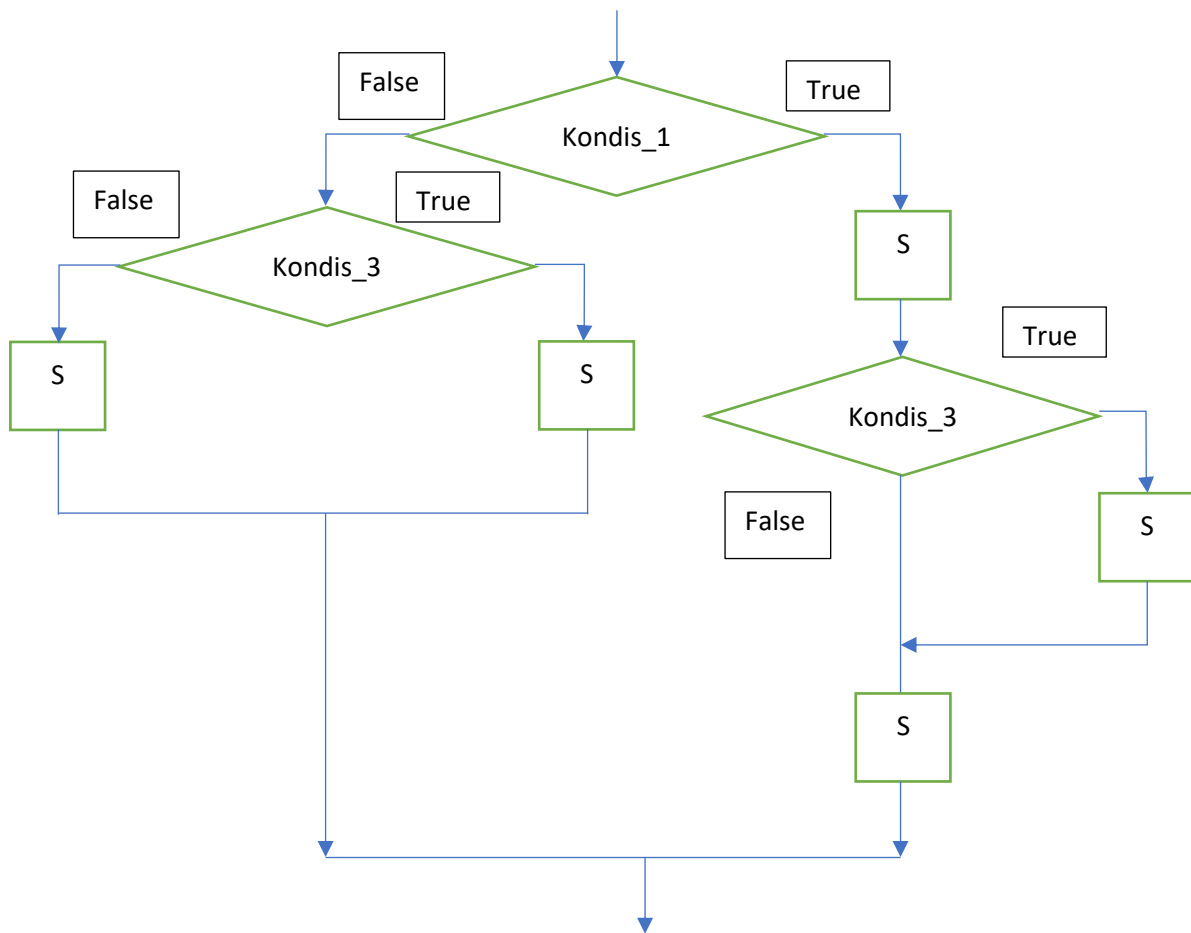
Bentuk flowchart bentuk 2 :



Percabangan kompleks bentuk 3:

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

Bentuk flowchart bentuk 3 :

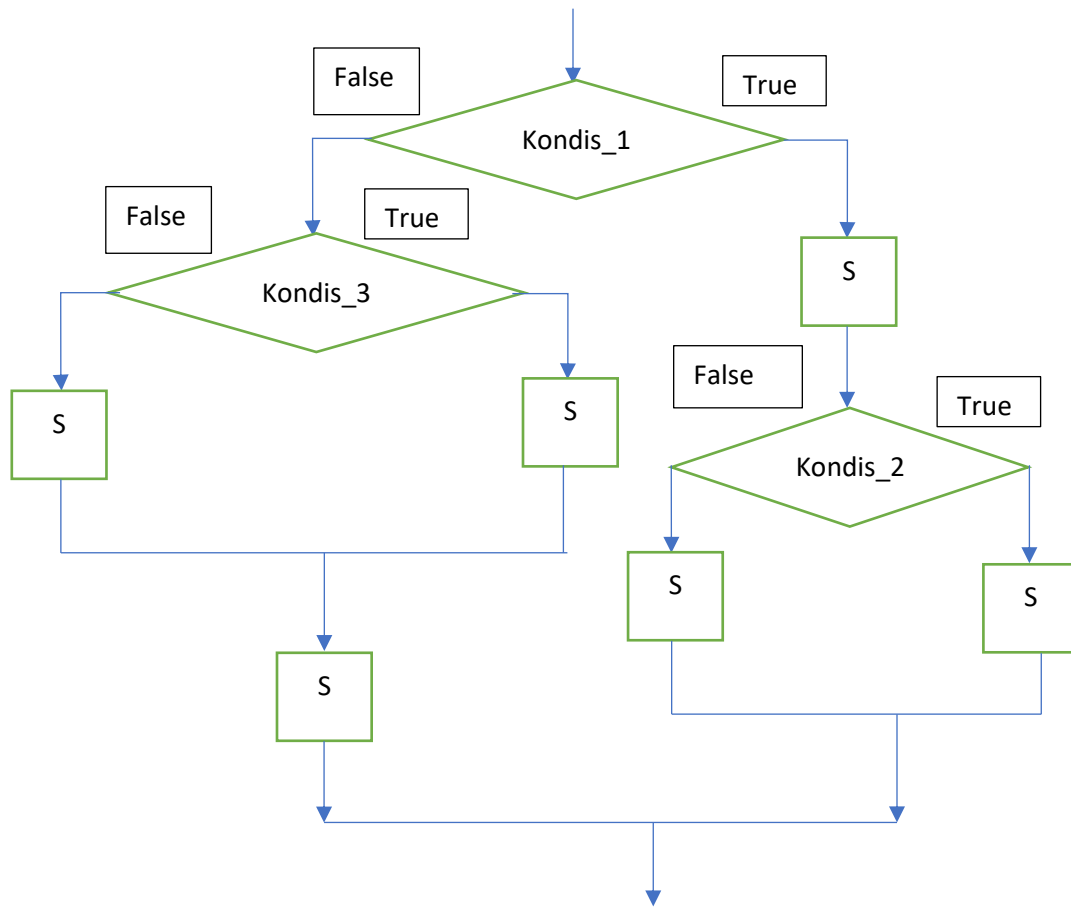


Percabangan kompleks bentuk 4 :

```

if kondisi 1:
    S
    if kondisi 2:
        S
        S
    else :
        S
        S
else :
    if Kondisi 3 :
        S
        S
    else :
        S
        S
S
  
```

Bentuk flowchart bentuk 4 :

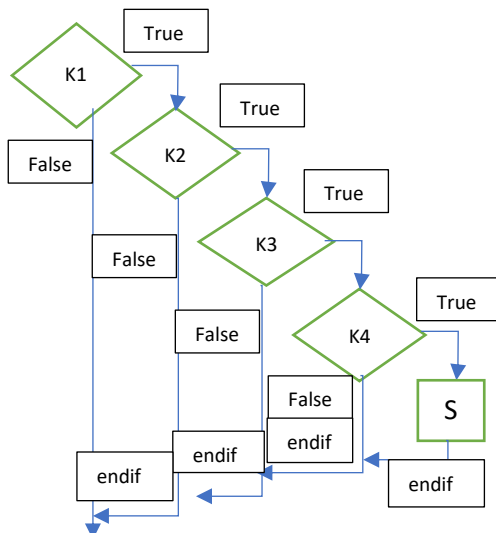


Percabangan kompleks bentuk 5 :

```

if kondisi_1:
    if kondisi_2 :
        if kondisi_3 :
            if kondisi_4 :
                S
  
```

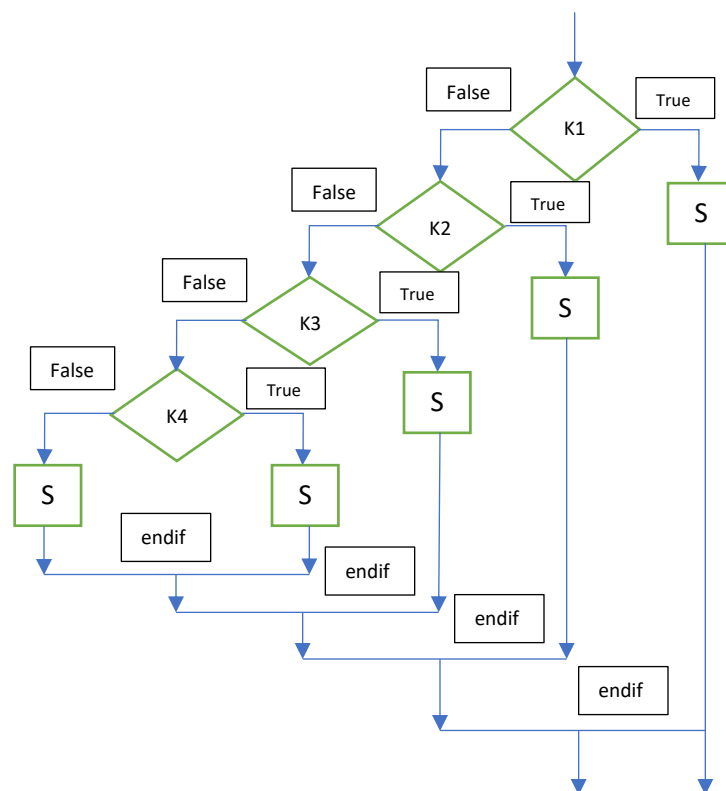
Bentuk flowchart bentuk 5 :



Percabangan kompleks bentuk 6:

```
if kondisi 1 :  
    S  
else :  
    if kondisi 2 :  
        S  
    else :  
        if kondisi 3 :  
            S  
        else :  
            if kondisi 4 :  
                S  
            else :  
                S
```

Pada kasus if pada bentuk percabangan ini IF bertingkat memiliki keuntungan tersendiri. Dengan cara menggunakan IF bertingkat ini maka eksekusi perintah menjadi lebih baik dan efisien. Selain itu juga tidak semua kondisi IF di kerjakan sehingga waktu eksekusi lebih cepat. Pada IF biasa semua kondisi IF pasti akan dicoba satu persatu walaupun akhirnya hanya satu perintah IP yang dapat terpenuhi, dengan hal itu dapat memperlambat proses. Contoh kasus untuk IF bertingkat adalah kasus konversi nilai angka menjadi nilai huruf dengan Batasan nilai yang sudah terpenuhi :



Perbedaannya :

```

if(kondisi 1):
    instruksi 1
elif(kondisi 2):
    instruksi 2
elif(kondisi 3):
    instruksi 3
elif(kondisi 4):
    instruksi 4

```

Bedakan dengan gambar di bawah ini :

```

if(kondisi 1):
    instruksi 1
if(kondisi 2):
    instruksi 2
if(kondisi 3):
    instruksi 3
if(kondisi 4):
    instruksi 4

```

6.3.2 Struktur perulangan kompleks

1. Break :

Digunakan untuk menghentikan proses perulangan yang sedang terjadi. Terjadi karena suatu kondisi tertentu yang diimplementasikan menggunakan perintah IF. Berikut contoh perintah break beserta flowchartnya :

```

for i in range(5000):
    print(i)
    if i == 6:
        break

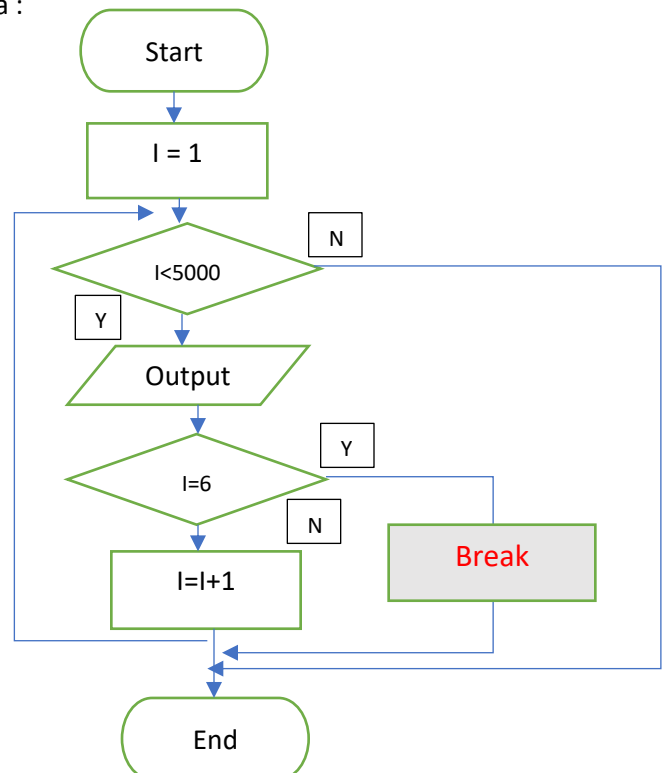
```

Output:

```

0
1
2
3
4
5
6
PS E:\>

```



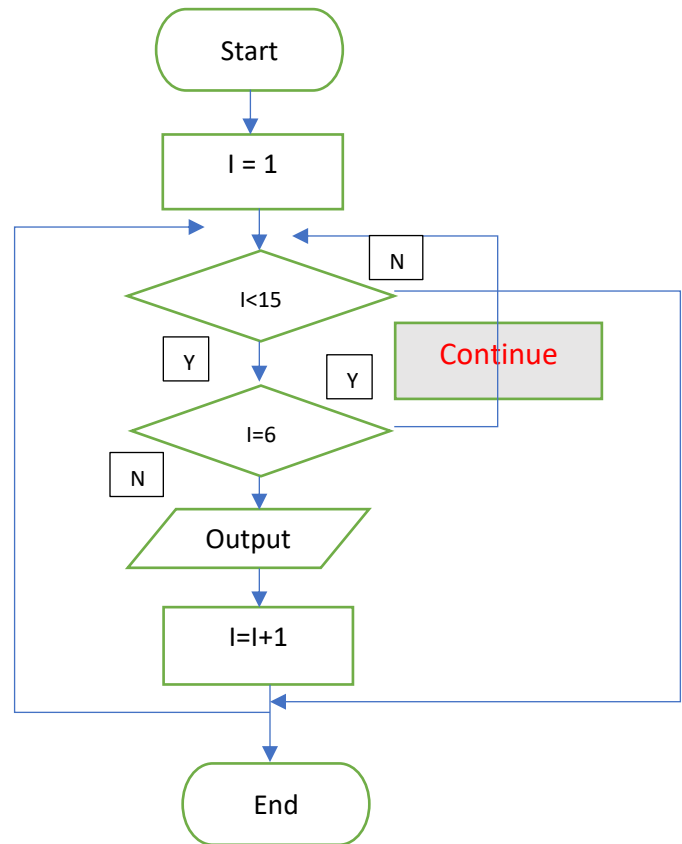
2. Continue:

Menyebabkan proses perulangan kembali ke awal, dengan mengabaikan statement selanjutnya setelah continue. Contoh perintah continue beserta flowchartnya :

```
for i in range(15):  
    if i == 6:  
        continue  
    print(i)
```

Output :

```
0  
1  
2  
3  
4  
5  
7  
8  
9  
10  
11  
12  
13  
14  
PS E:\>
```



3. Perulangan Bertingkat:

Struktur perulangan kompleks adalah bentuk perulangan yang dimana dalam suatu perulangan terdapat sebuah perulangan lain, sehingga terjadilah sebuah perulangan bertingkat yang memakan banyak waktu. Suatu masalah yang dapat diselesaikan dengan perulangan kompleks adalah dengan masalah matriks yang menggunakan array 2 dimensi. Berikut adalah contoh perulangan kompleks :

Program yang pertama :

```
for i in range(x) :  
    <lakukan perintah yang ini>  
    <lakukan perintah yang itu>
```

Program yang kedua :

```
for j in range(y) :  
    <lakukan perintah yang ini>  
    <lakukan perintah yang itu>
```

Dalam program yang pertama ini mengerjakan perulangan sebanyak 15 kali sedangkan pada program kedua mengerjakan sebanyak 6 kali. Program ini berjalan sendiri sendiri.

Jika pada program yang kedua dimasukkan ke dalam perulangan pada program pertama, maka akan menjadi sebuah program dibawah ini :

```
for i in range(x) :  
    for j in range(y) :  
        <lakukan perintah ini di bagian inner>  
        <lakukan perintah itu di bagian inner>  
    <lakukan perintah lain di bagian outer>  
<lakukan perintah lain lagi di outer>|
```

Pada bagian for i menjadi outer loop, dan pada bagian for j menjadi inner loop. Dapat kita lihat alur pengerjaannya sebagai berikut ini : untuk setiap 1x sampai dengan x outer loop dijalankan akan dikerjakan y kali inner loop.

```
while(i<=x) :  
    while(j<=y) :  
        <lakukan perintah ini dibagian inner>  
        <lakukan perintah ini dibagian inner>  
    <lakukan perintah lain dibagian outer>  
<lakukan perintah lain di outer>|
```

Pada bagian while i menjadi outer loop, dan pada bagian while j menjadi inner loop. Alur dalam pengerjaannya adalah sebagai berikut : untuk 1x sampai dengan x outer loop dijalankan akan dikerjakan y kali inner loop.

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 prima(n):
    prima = False
    counter = 0
    for i in range(1,n+1):
        if n % i == 0:
            counter += 1
        if counter > 2:
            prima = False
            break
        else:
            prima = True
    return prima

bilangan_angka= int(input("input n : "))
for i in range(bilangan_angka-1,0,-1):
    if prima(i):
        print("%d ialah bilangan prima terdekat dari %d"%(i,bilangan_angka))
        break
```

Penjelasan :

Pada soal 6.1 pada soal latihan mandiri ini kita disuru mencari bilangan prima terdekat dari bilangan yang diinput, dengan itu saya menggunakan rumus percabangan dan perulangan kompleks. Saya pertama memasukan rumus def prima(n):. setelah itu saya memasukkan prima = false dan counter = 0. Setelah itu saya mulai masuk ketahap for i in range(1,n+1):. Setelah itu saya memasukkan if n % i == 0:. Dan saya lanjutkan dengan counter += 1. Stelah itu saya masuk ke if counter > 2: . Setelah itu saya masukkan perinta prima = false dan saya enter dan saya masukkan perintah break. Setelah melakukan if saya menutup dengan perintah else, dan saya lanjutkan dengan masukkan perintah prima = true setelah itu saya enter dan saya mengetik perintah return prima. Setelah itu saya enter untuk ganti baris, dan saya lanjutkan dengan melakukan perintah input dengan cara mengetik bilangan_angka = int(input("input n: ")). Setelah itu saya lanjutkan dengan for i in range(bilangan_angka-1,0,-1):. Setelah itu saya enter dan saya ketik perintah if prima(i):. Setelah itu ketik perintah print("%d ialah bilangan prima terdekat dari %d"%(i,bilangan_angka)). Setelah itu saya enter dan saya ketik perintah break.

```
latihan-6-1 (variable) prima: Literal[False]
1 def
2     prima = False
3     counter = 0
4     for i in range (1,n+1):
5         if n % i == 0:
6             counter += 1
7         if counter > 2:
8             prima = False
9             break
10        else:
11            prima = True
12    return prima
13
14    bilangan_angka= int(input("input n : "))
15    for i in range(bilangan_angka-1,0,-1):
16        if prima(i):
17            print("%d ialah bilangan prima terdekat dari %d"%(i,bilangan_angka))
18            break
19
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Kuliah\Semester 4\Prak Alpro> & "E:/Coding Tool/python.exe" "d:/Kuliah/Semester 4\Prak Alpro\latihan-6-1.py"
input n : 12
11 ialah bilangan prima terdekat dari 12
PS D:\Kuliah\Semester 4\Prak Alpro> & "E:/Coding Tool/python.exe" "d:/Kuliah/Semester 4\Prak Alpro\latihan-6-1.py"
input n : 21
19 ialah bilangan prima terdekat dari 21
PS D:\Kuliah\Semester 4\Prak Alpro>
```

SOAL 2

Source code :

```
def faktorial(n):
    total_seluruhnya = 1
    for i in range(1, n + 1):
        total_seluruhnya *= i
    return(total_seluruhnya)
```

```

n = 6
for i in range(n , 0, -1):
    print(faktorial(i), end = " ")
    for j in range(i , 0 , -1):
        print(j, end = " ")
    print("")

```

Penjelasan :

Pada soal 6.2 dalam soal Latihan mandiri ini kita disuru membuat program untuk menampilkan suatu deret seperti di gambar pada soal tersebut. Dengan itu saya dalam melakukan program pada no 6.2 ini saya menggunakan rumus def faktorial(n):. Setelah itu saya memasukkan total_seluruhnya = 1. Setelah itu saya klik enter dan saya melakukan perintah for dengan mengetik for i in range(1, n + 1):. Dan setelah saya klik enter saya memasukkan rumus total_seluruhnya*= i. setelah itu saya lanjutkan dengan perintah return(total_seluruhnya). Setelah itu saya enter untuk mengganti baris. Setelah itu saya masukkan input dengan mengetik n = 6. Setelah itu saya enter dan saya memasukkan perintah for i in range(n , 0, -1):. Setelah itu saya lanjutkan dengan memasukkan perintah print(faktorial(i), end = " "). Setelah itu saya enter kembali dan memasukkan rumus for lagi dengan variabel j untuk membedakannya. Saya mengetik for j in range(i , 0 , -1):. Setelah itu saya enter dan memasukkan perintah print(j, end = " "). Dan saya enter kembali dan memasukkan perintah print("").

The screenshot shows a code editor with the following Python code:

```

1 def faktorial(n):
2     total_seluruhnya = 1
3     for i in range(1, n + 1):
4         total_seluruhnya*= i
5     return(total_seluruhnya)
6
7 n = 6
8 for i in range(n , 0, -1):
9     print(faktorial(i), end = " ")
10    for j in range(i , 0 , -1):
11        print(j, end = " ")
12    print("")
13

```

Below the code editor is a terminal window showing the output of the program:

```

PS D:\Kuliah\Semester 4\Prak Alpro> & "E:/Coding Tool/python.exe" ...
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:\Kuliah\Semester 4\Prak Alpro>

```

SOAL 3

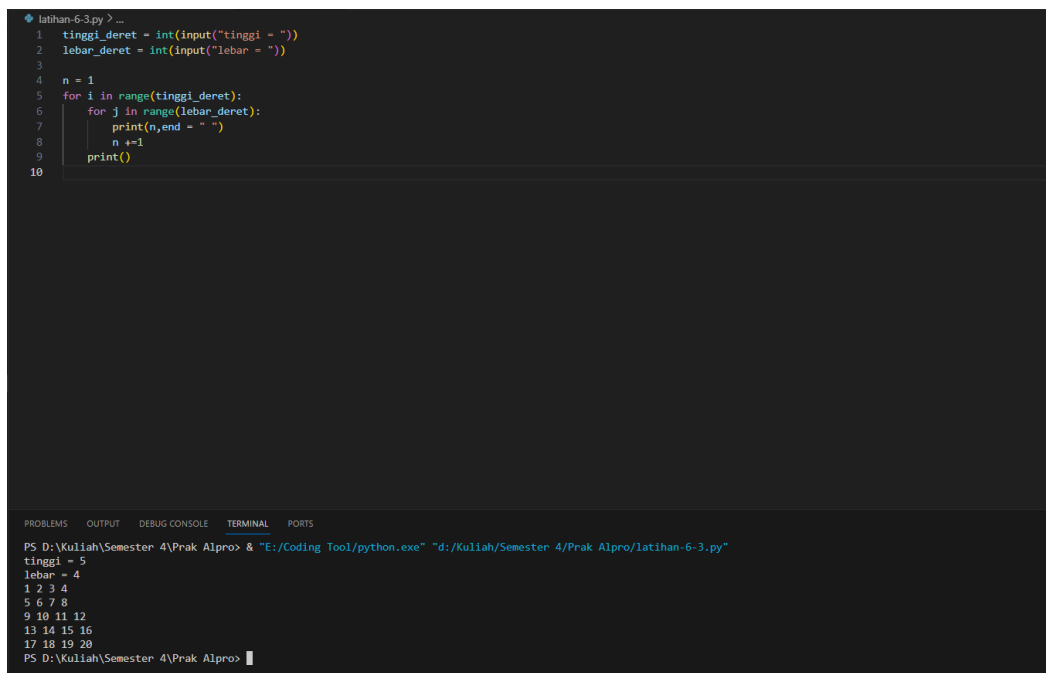
Source code :

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

```
n = 1
for i in range(tinggi_deret):
    for j in range(lebar_deret):
        print(n,end = " ")
        n +=1
    print()
```

Penjelasan :

Pada soal 6.3 pada soal Latihan mandiri ini kita disuru membuat sebuah program menampilkan deret seperti yang ada pada gambar di soal. Dalam soal ini saya melakukan codingan dengan menggunakan rumus yang pertama dengan memasukkan input sebuah tinggi deret dengan cara seperti `tinggi_deret = int(input("tinggi = "))`. Setelah itu saya lanjutkan dengan memasukkan input sebuah lebar deret dengan cara `lebar_deret = int(input("lebar = "))`. Setelah itu saya enter dan mengetik sebuah inputan `n = 1`. Setelah itu saya melakukan enter dan melanjutkan dengan melakukan perintah `for i in range(tinggi_deret):`. Dan saya enter lagi dan melakukan perintah `for j in range(lebar_deret):`. Setelah itu saya melakukan perintah `print(n,end = " ")` dan setelah itu saya enter. Setelah itu saya melakukan perintah `n +=1`. Setelah itu saya enter, pada tahap terakhir saya memasukkan perintah `print()`.



```
latihan-6-3.py >--
1 tinggi_deret = int(input("tinggi = "))
2 lebar_deret = int(input("lebar = "))
3
4 n = 1
5 for i in range(tinggi_deret):
6     for j in range(lebar_deret):
7         print(n,end = " ")
8         n +=1
9     print()
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Kuliah\Semester 4\Prak Alpro> & "E:/Coding Tool/python.exe" "d:/Kuliah/Semester 4/Prak Alpro/latihan-6-3.py"
tinggi = 5
lebar = 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
PS D:\Kuliah\Semester 4\Prak Alpro>
```