



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230992
Nama Lengkap	Andriano Kurniawan Ladjeba
Minggu ke / Materi	06 / Perulangan dan Percabangan 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

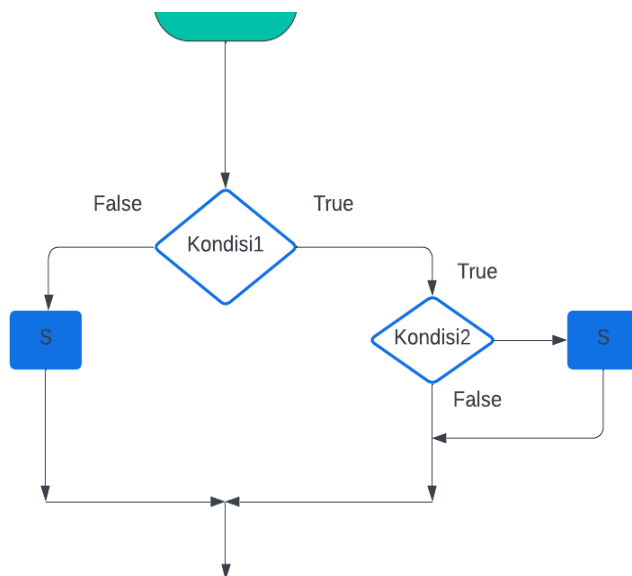
Struktur Percabangan Komplek

Percabangan di mana kondisi pemilihan dapat berupa banyak pilihan. Selain itu, mungkin ada lebih dari satu perintah yang dikerjakan.

Bentuk percabangan kompleks 1:

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

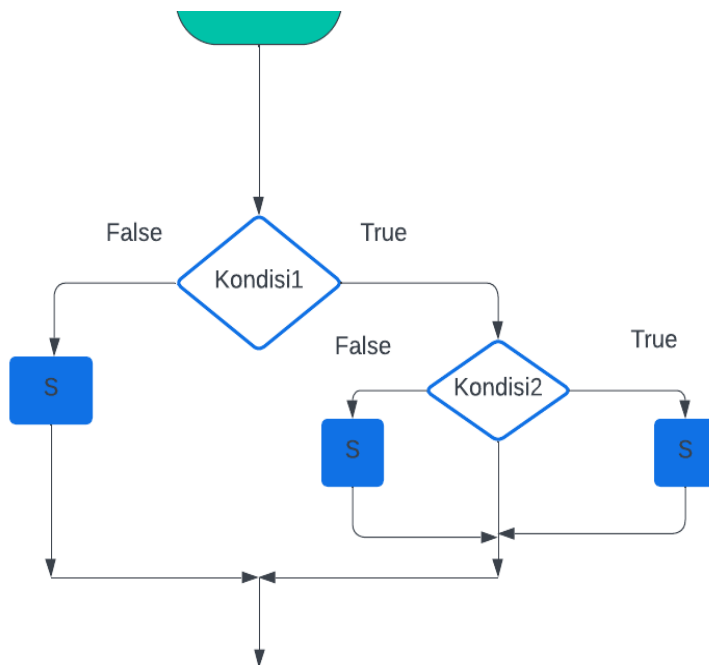
Gambar flowchart bentuk 1



Percabangan kompleks bentuk 2:

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

Gambar flowchart bentuk 2



Percabangan kompleks bentuk 3:

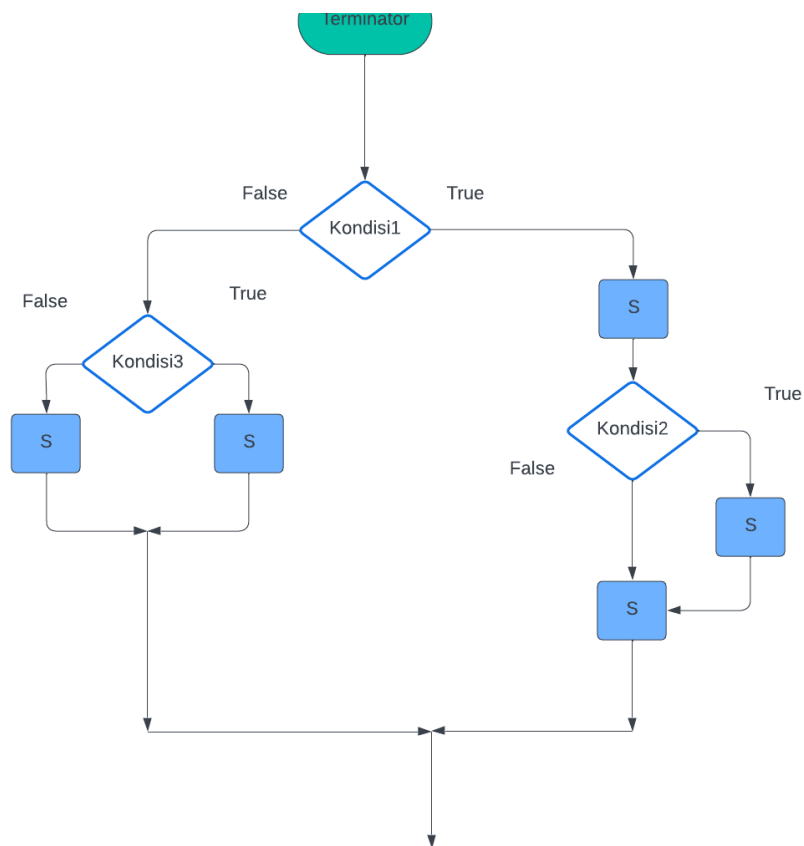
```
if kondisi1:  
    S  
    if kondisi2:  
        S  
        S  
else:  
    S
```

```

if kondisi3:
    S
    S
else:
    S
    S

```

Gambar flowchart bentuk 3



Percabangan kompleks bentuk 4:

```

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

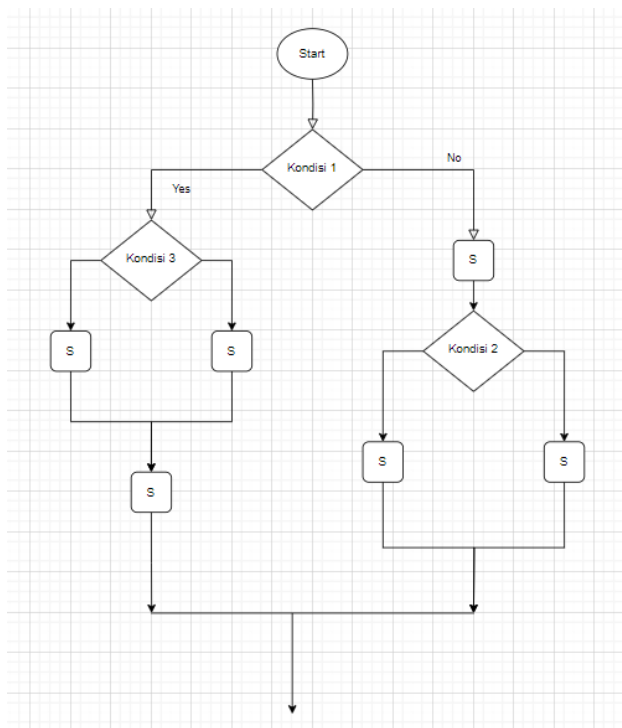
```

```

S
else:
    if kondisi3:
        S
        S
    else:
        S
        S
S

```

Gambar flowchart bentuk 4



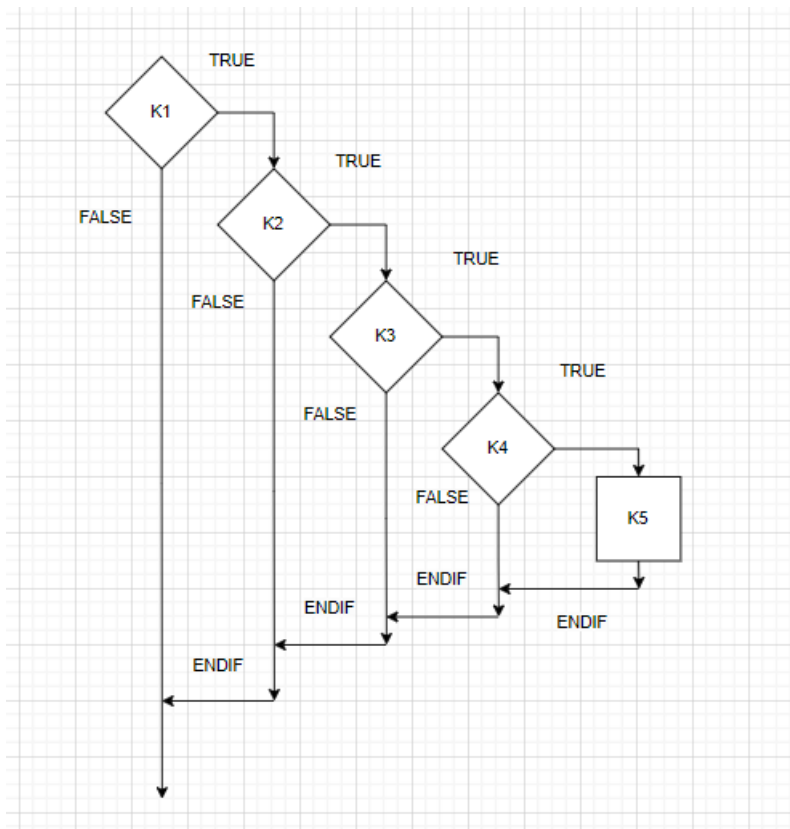
Percabangan kompleks bentuk 5:

```

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

```

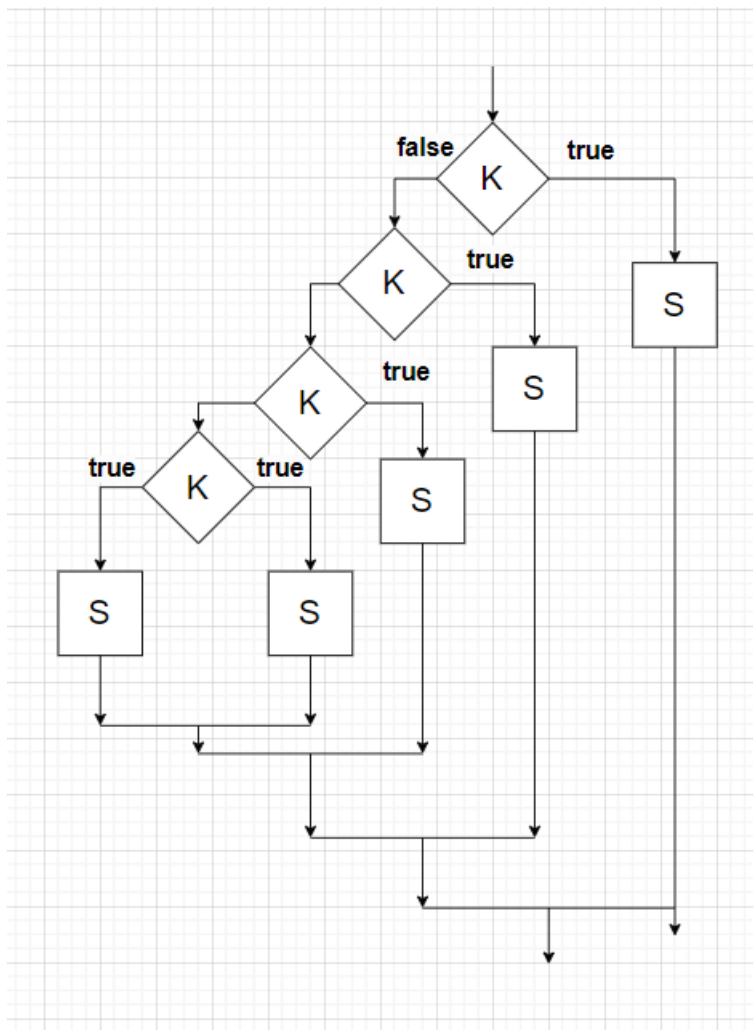
Gambar flowchart bentuk 5



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
```

Gambar flowchart bentuk 6



batasan-batasan nilai yang sudah ditentukan sejak awal

```
if (kondisi1):  
    instruksi1  
elif(kondisi2):  
    instruksi2  
elif(kondisi3):  
    instruksi3  
elif(kondisi4):  
    instruksi4
```

Coba bedakan dengan:

```
if (kondisi1):  
    instruksi1  
if(kondisi2):  
    instruksi2  
if(kondisi3):  
    instruksi3  
if(kondisi4):  
    instruksi4
```

Struktur Perulangan Kompleks Break

Perintah ini digunakan untuk menghentikan perulangan. Ini biasanya terjadi karena kondisi tertentu yang diimplementasikan melalui perintah IF.

```
for i in range(1000):  
    print(i)  
if i==10:  
    break
```

Output:

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

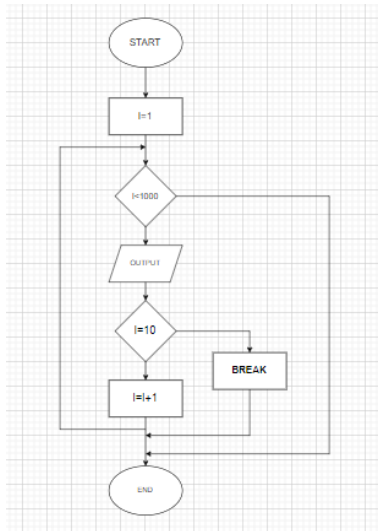
Karena perintah break diberikan pada saat kondisi $i=10$, angka 10 tetap ditampilkan karena perintah untuk mencetak diletakkan sebelum perintah break, program di atas tetap menampilkan angka 1 hingga 10.

Contoh program break lain:

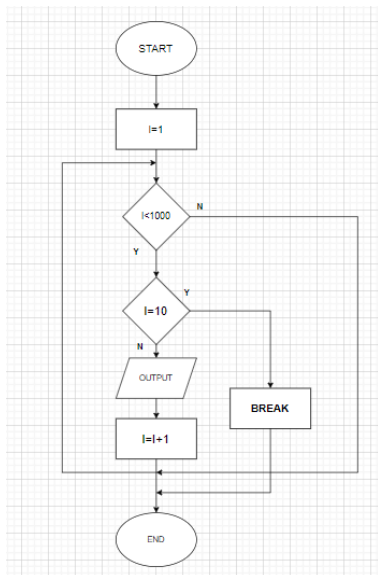
```
for i in range(1000):  
    if i==10:  
        break  
print(i)
```


Walaupun telah diatur dari 1 hingga 1000, program di atas akan menampilkan angka 1 hingga 9. Ini disebabkan oleh fakta bahwa perintah break diberikan pada saat kondisi $i=10$. Karena perintah untuk mencetak diletakkan sesudah perintah break, angka 10 tidak akan ditampilkan.

Flowchart Program Menggunakan Break



Flowchart Program Menggunakan Break 2



Continue

Perintah continue mengembalikan proses ke awal, mengabaikan statement-statement berikutnya setelah continue. Perintah IF biasanya digunakan untuk menerapkan perintah continue juga.

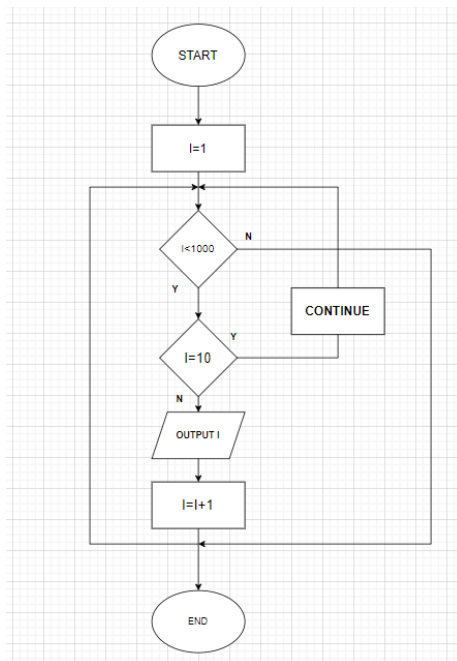
```
for i in range(10):  
    if i==5:
```

```
    continue
print(i)
```

Output:

0
1
2
3
4
6
7
8
9

Program di atas tidak menampilkan angka 5 karena pada saat angka 5 akan ditampilkan, perintah `continue` dijalankan, yang berarti perintah mencetak di bagian bawahnya tidak akan dilakukan dan akan langsung dilanjutkan ke perulangan berikutnya!



Perulangan Bertingkat

Jenis per-ulangan di mana ada banyak perulangan di dalam satu perulangan, menyebabkan waktu proses yang lebih lama. Ini dikenal sebagai struktur perulangan kompleks. Ada banyak algoritma yang tepat untuk struktur perulangan kompleks. Masalah matriks yang menggunakan array dua dimensi, masalah game board seperti catur dan minesweeper, dan masalah pengolahan gambar digital seperti algoritma untuk mendeteksi tepi gambar atau mengubah gambar berwarna menjadi grayscale adalah beberapa masalah yang dapat diselesaikan

dengan perulangan kompleks. Pada dasarnya, masalah yang dapat diselesaikan melalui perulangan kompleks biasanya memiliki pola grid, atau kotak, yang memiliki panjang dan lebar.

Program 1:

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

Program 2:

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

Program 1 melakukan perulangan sebanyak sepuluh kali, sedangkan program 2 melakukan perulangan sebanyak lima kali. Jika perulangan dari program 2 "dimasukkan" ke dalam perulangan program 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>
```

Bagian for i berfungsi sebagai loop luar, dan bagian for j berfungsi sebagai loop dalam. Prosesnya adalah sebagai berikut: untuk setiap 1x s/d m loop luar dijalankan, loop dalam akan dikerjakan n kali.

Untuk sintaks dalam bentuk sama saja. Perhatikan contoh berikut:

```
while(i<=m):  
    while(j<=n):  
        <lakukan perintah ini di inner>  
        <lakukan perintah itu di inner>  
    <lakukan perintah lain di outer>  
    <lakukan perintah lain lagi di outer>
```

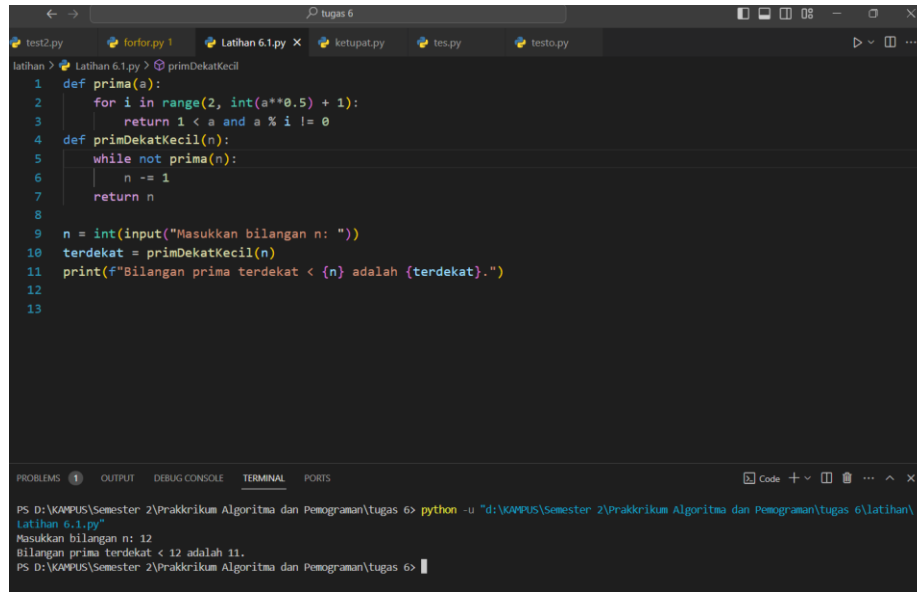
Bagian while i berfungsi sebagai loop luar, dan bagian while j berfungsi sebagai loop dalam. Prosesnya adalah sebagai berikut: untuk setiap 1x s/d m loop luar dijalankan, loop dalam akan dikerjakan n kali.

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



```
1 def prima(a):
2     for i in range(2, int(a**0.5) + 1):
3         return 1 < a and a % i != 0
4 def primDekatKecil(n):
5     while not prima(n):
6         n -= 1
7     return n
8
9 n = int(input("Masukkan bilangan n: "))
10 terdekat = primDekatKecil(n)
11 print(f"Bilangan prima terdekat < {n} adalah {terdekat}.")
12
13
```

PS D:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6> python -u "d:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6\latihan\latihan 6.1.py"

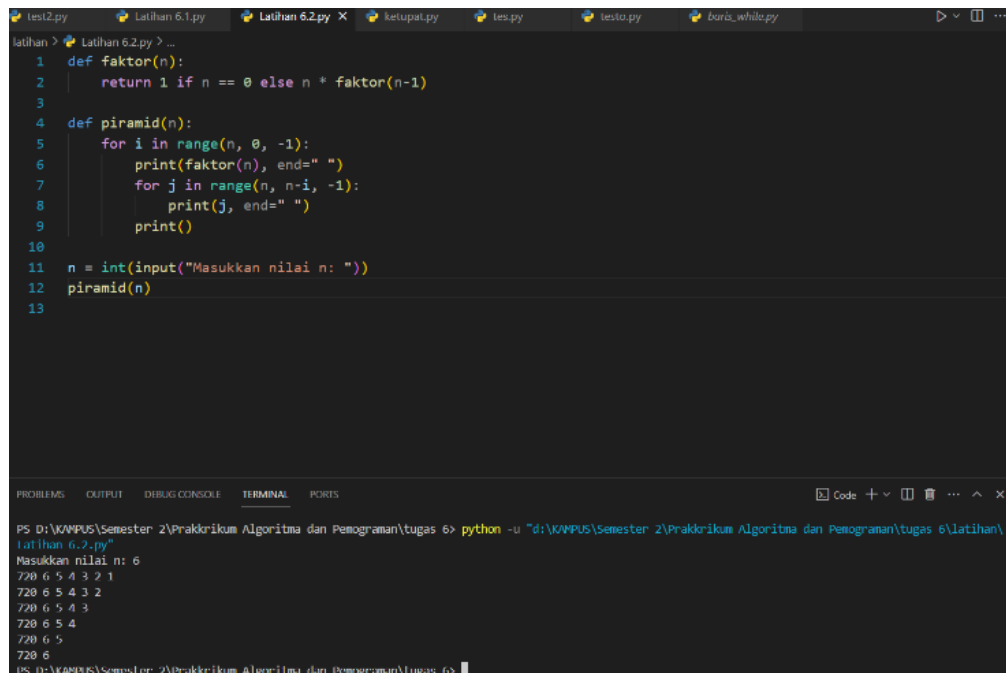
Masukkan bilangan n: 12

Bilangan prima terdekat < 12 adalah 11.

PS D:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6>

SOAL 2

Latihan 6.2



```
1 def faktor(n):
2     return 1 if n == 0 else n * faktor(n-1)
3
4 def piramid(n):
5     for i in range(n, 0, -1):
6         print(faktor(n), end=" ")
7         for j in range(n, n-i, -1):
8             print(j, end=" ")
9         print()
10
11 n = int(input("Masukkan nilai n: "))
12 piramid(n)
13
```

PS D:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6> python -u "d:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6\latihan\latihan 6.2.py"

Masukkan nilai n: 6

720 6 5 4 3 2 1

720 6 5 4 3 2

720 6 5 4 3

720 6 5 4

720 6 5

720 6

PS D:\KAMPUS\Semester 2\Praktikum Algoritma dan Pemrograman\tugas 6>

SOAL 3

Latihan 6.3

```
latihan > Latihan 6.3.py > ...
1 def jajar_genjang(tinggi, lebar):
2     for i in range(tinggi):
3         # berfungsi untuk memiringkan
4         print("*" * i, end='')
5
6         for j in range(1, lebar + 1):
7             # print atas bawah
8             if i + 0 or i + lebar - 1:
9                 print(i * lebar + j, end=" ")
10            print()
11        print("dimensi tidak sesuai")
12
13
14 tinggi = int(input("Masukkan tinggi = "))
15 lebar = int(input("Masukkan Lebar = "))
16 1
17
18 jajar_genjang(tinggi, lebar)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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
dimensi tidak sesuai
PS D:\KAMPUS\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas 6>

Github: <https://github.com/EchoGinDev/tugasAlpro6>