



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230987</b>
<b>Nama Lengkap</b>	<b>Prastha Pradipta Purusa</b>
<b>Minggu ke / Materi</b>	<b>06 / Percabangan dan perulangan kompleks</b>

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.

### Struktur Percabangan kompleks

Percabangan adalah kondisi dimana bisa terdapat banyak alternatif pilihan dan perintah juga lebih dari satu. Berikut adalah contohnya:

```
1 def genap(x):  
2     if x % 2 == 0:  
3         return True  
4     else:  
5         return False  
6 print(genap(6))
```

True

```
1 def genap(x):  
2     if x % 2 == 0:  
3         return True  
4     else:  
5         return False  
6 print(genap(3))
```

False

Program tersebut mengecek bilangan genap. Jika memasukan bilangan genap seperti di contoh angka 6 maka program tersebut akan menghasilkan true jika memasukan bilangan ganjil maka akan memunculkan false. Berikut adalah jika dalam perulangan kompleks:

```

1 a = 10
2 b = 5
3 operator = input("Masukkan operator (+, -, *, /): ")
4
5 if operator == '+':
6     hasil = a + b
7     print("Hasil penjumlahan a + b =", hasil)
8 elif operator == '-':
9     hasil = a - b
10    print("Hasil pengurangan a - b =", hasil)
11 elif operator == '*':
12    hasil = a * b
13    print("Hasil perkalian a * b =", hasil)
14 elif operator == '/':
15     if b != 0:
16         hasil = round(a / b, 2)
17         print("Hasil pembagian a / b =", hasil)
18     else:
19         print("Pembagian dengan nol tidak dapat dilakukan.")
20 else:
21     print("Operator tidak valid.")

```

```

Masukkan operator (+, -, *, /): +
Hasil penjumlahan a + b = 15

```

```

Masukkan operator (+, -, *, /): *
Hasil perkalian a * b = 50

```

## Struktur Perulangan kompleks

### Continue

Perintah ini melakukan perulangan dengan menggunakan perintah IF dan FOR. Berikut adalah contoh programnya:

```

1 for char in 'abcdef':
2     if char == 'd':
3         continue
4     print(char)

```

Jika huruf hanya sampai d tetapi jika menggunakan continue maka program akan tetap berjalan sampai huruf f. Continue adalah sebuah pernyataan yang digunakan untuk melanjutkan ke iterasi berikutnya dalam sebuah loop tanpa mengeksekusi kode di bawahnya dalam blok loop.

### Break

Program akan berhenti membaca d karena menggunakan break yang berfungsi untuk menghentikan program. Break adalah sebuah pernyataan yang digunakan untuk menghentikan eksekusi dari sebuah loop. Biasanya, break digunakan di dalam loop (seperti for atau while) untuk keluar dari loop lebih awal daripada kondisi loop normal yang telah ditetapkan.

```

1 for char in 'abcdef':
2     if char == 'd':
3         break
4     print(char)

```

Program akan berhenti membaca d karena menggunakan break yang berfungsi untuk menghentikan program.

## Perulangan Bertingkat

Struktur perulangan kompleks melibatkan perulangan di dalam perulangan, meningkatkan waktu proses secara signifikan. Perulangan kompleks menggunakan format for sebanyak lebih dari 1x. Berikut adalah contoh programnya:

```
1 tinggi_segitiga = 5
2
3 for baris in range(tinggi_segitiga):
4     for kolom in range(baris + 1):
5         print('#', end='')
6     print()
```

```
#
##
###
####
#####
```

Variabel tinggi\_segitiga menyimpan tinggi segitiga yang diinginkan, dalam contoh ini memiliki nilai 5. Program ini menggunakan perulangan for untuk mencetak setiap baris dari segitiga. Di dalam perulangan baris, terdapat perulangan kolom yang mencetak jumlah sesuai dengan nomor baris. Setiap baris memiliki jumlah yang bertambah satu dari baris sebelumnya dan menciptakan pola segitiga. Output dari program ini adalah segitiga dengan tinggi 5, yang terdiri dari baris yang bertambah satu jumlahnya pada setiap barisnya.

Berikut adalah contoh jika menggunakan while:

```
1 i = 1
2 total = 0
3 while i <= 5:
4     total += i
5     print("Nilai i:", i, "Total:", total)
6     i += 1
```

```
Nilai i: 1 Total: 1
Nilai i: 2 Total: 3
Nilai i: 3 Total: 6
Nilai i: 4 Total: 10
Nilai i: 5 Total: 15
```

Pertama-tama, variabel i diinisialisasi dengan nilai 1, sedangkan variabel total diinisialisasi dengan nilai 0. Kemudian, perulangan while dimulai. Perulangan akan terus berjalan selama nilai i kurang dari atau sama dengan 10. Dalam setiap iterasi perulangan, nilai i akan ditambahkan ke dalam total, dan kemudian nilai i dan total saat ini akan dicetak. Ini memungkinkan kita untuk melihat bagaimana nilai total berkembang setiap kali nilai i bertambah.

## Kegiatan Praktikum

Disini akan membuat program jajar genjang:

```
1- def jajar_genjang(lebar, tinggi):  
2-     for i in range(tinggi):  
3-         print(" " * (tinggi - i - 1) + "#" * lebar)  
4- jajar_genjang(2,3)
```

```
##  
##  
##
```

Kita menggunakan fungsi `jajar_genjang()` dengan parameter lebar dan tinggi. Bagian `for` dalam fungsi `jajar_genjang` digunakan untuk melakukan iterasi sebanyak tinggi kali. Setiap iterasi akan mencetak satu baris pola jajargenjang. `Print(" " * (tinggi - i - 1) + "#" * lebar)` untuk mencetak satu baris pola jajargenjang, dengan menggunakan jumlah spas dan karakter pagar yang sesuai untuk membentuk pola jajargenjang yang diinginkan.

Program kedua adalah kita membuat deret palindrome:

```
1- def palindrom(panjang):  
2-     if panjang % 2 == 0 or panjang < 1:  
3-         print("tidak sesuai")  
4-     else:  
5-         n = (panjang + 1) // 2  
6-         for i in range(n, 0, -1):  
7-             print(i, end=" ")  
8-         for i in range(2, n + 1):  
9-             print(i, end=" ")  
10-        print()  
11-  
12- panjang = 7  
13- palindrom(panjang)
```

```
4 3 2 1 2 3 4
```

Fungsi `palindrom` dibuat untuk menghasilkan deret bilangan yang membentuk pola palindrom sesuai dengan panjang yang diberikan sebagai input. Untuk memenuhi syarat pola palindrom, panjang harus merupakan bilangan ganjil minimal 1. Jika panjang tidak memenuhi syarat tersebut, fungsi akan mencetak pesan "dimensi tidak sesuai". Jika panjang memenuhi syarat, fungsi akan menghasilkan deret bilangan yang membentuk pola palindrom dan mencetaknya

Program ketiga adalah membuat segitiga terbalik dengan angka:

```

1 n = 5
2 for i in range(n, 0, -1):
3     for j in range(0, n-i):
4         print(" ", end="")
5     for j in range(0, i):
6         print(i, end="")
7     print()

```

```

55555
4444
333
22
1


```

Kita memasukan angka 5 di program tersebut. Untuk i akan menjadi nilai dan 0,-1 akan menghitung kebawah. For j in range digunakan untuk n-i maka akan melakukan pengurangan kebawah. Lalu di print(" ", end="") untuk membuat baris baru. Lalu for j in range(0,i) digunakan untuk membaca i. Lalu di print(i,end="") digunakan untuk membaca nilai i. print() digunakan untuk membuat baris baru.











## 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


**Jupyter**
Prastha Pradipta Purusa(71230987) Laprak 6 Last Checkpoint: 25 men

File Edit View Insert Cell Kernel Widgets Help











Code

```

In [1]: n = int(input("Masukkan angka: "))

        for x in range(n - 1, 1, -1):
            for y in range(2,x):
                if x % y == 0:
                    break
            else:
                bilangan_prima_terdekat = x
                break

        if bilangan_prima_terdekat:
            print(f"Prima terdekat < {n} adalah {bilangan_prima_terdekat}")
        else:
            print(f"Tidak ada bilangan prima yang kurang dari {n}")

```

Masukkan angka: 12  
Prima terdekat < 12 adalah 11

```
In [2]: n = int(input("Masukkan angka: "))

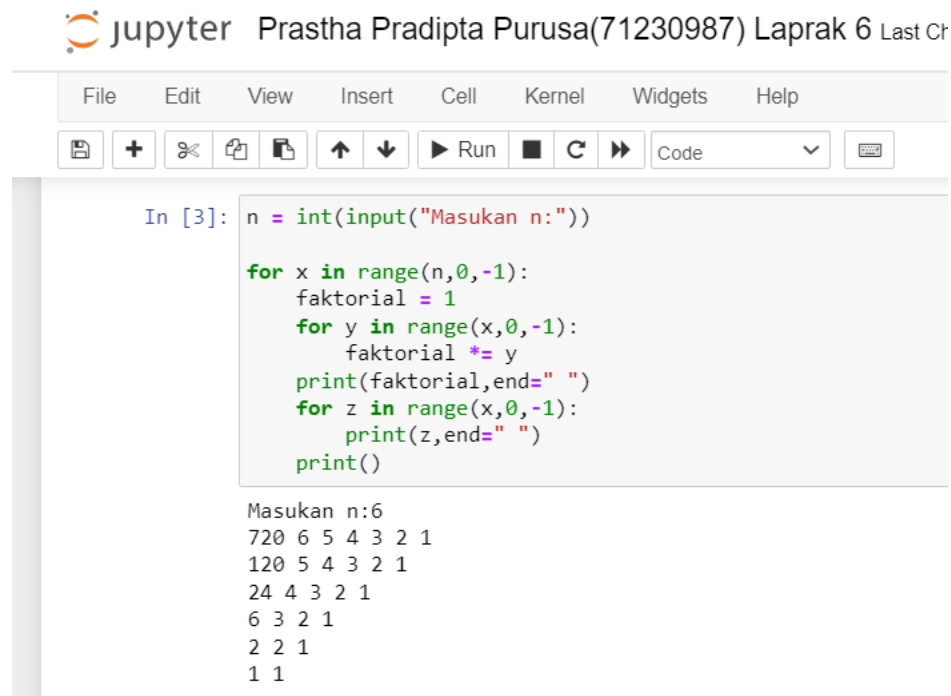
for x in range(n - 1, 1, -1):
    for y in range(2, x):
        if x % y == 0:
            break
    else:
        bilangan_prima_terdekat = x
        break

if bilangan_prima_terdekat:
    print(f"Prima terdekat < {n} adalah {bilangan_prima_terdekat}")
else:
    print(f"Tidak ada bilangan prima yang kurang dari {n}")
```

```
Masukkan angka: 21
Prima terdekat < 21 adalah 19
```

Kita mencari bilangan prima terdekat dari bilangan yang kita input. Pertama kita memberi variabel `n` untuk nilai. `for x in range(n-1,1,-1)` digunakan untuk menghitung `n-1` hingga 2 kali tetapi secara mundur sebanyak 1. `for y in range(2, x)` berfungsi untuk mengiterasi nilai `y` dari 2 hingga `x - 1`, memeriksa apakah `x` dapat dibagi habis oleh nilai `y`, untuk menentukan apakah `x` adalah bilangan prima atau bukan. Jika `x % y = 0` maka program akan berhenti berjalan. Jika `bilangan_prima_terdekat` adalah `x` maka program akan berhenti berjalan dan akan `print(f"Prima terdekat < {n} adalah {bilangan_prima_terdekat}")` maka akan memunculkan bilangan prima terdekat dari yang kita inputkan. Jika tidak ada bilangan prima terdekat maka akan `print(f"Tidak ada bilangan prima yang kurang dari {n}")`

## SOAL 2



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "Prastha Pradipta Purusa(71230987) Laprak 6 Last Cr". Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Under the menu bar is a toolbar with icons for saving, adding cells, deleting, copying, pasting, undo, redo, and running code. The main area contains a code cell with the following Python code:

```
In [3]: n = int(input("Masukan n:"))

for x in range(n,0,-1):
    faktorial = 1
    for y in range(x,0,-1):
        faktorial *= y
    print(faktorial,end=" ")
    for z in range(x,0,-1):
        print(z,end=" ")
    print()
```

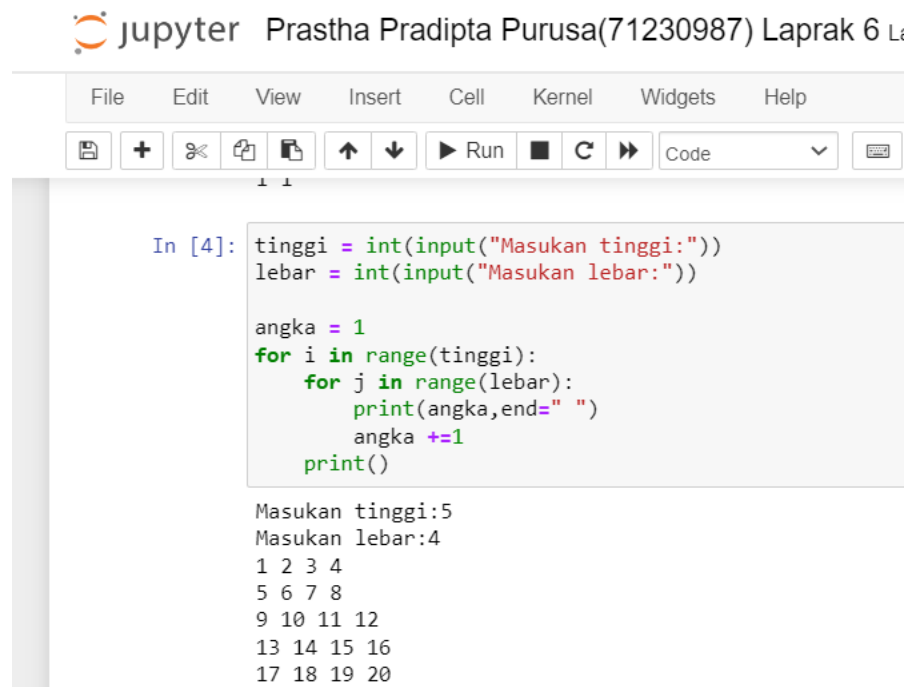
The output of the code is as follows:

```
Masukan 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
```

Kita akan membuat program factorial dari bilangan. Hasil di kiri adalah perkalian dari semua bilangan di kanan. Kita menggunakan n untuk input. For x in range(n,0,-1) adalah untuk menghitung kebawah sebanyak 1. Factorial dimulai dari nilai 1 supaya tidak ada 0. For y in range(x,0,-1) digunakan untuk nilai x menghitung dikurang sebanyak 1. Factorial \*= y berfungsi untuk mengkalikan semua bilangan di nilai y. print(faktorial,end=" ") berfungsi untuk mengeluarkan nilai factorial. For z in range(x,0,-1) berfungsi untuk memunculkan nilai x dibelakang hasil factorial. Print(z,end=" ") digunakan untuk memunculkan nilai z dan end=" " digunakan untuk memberi spasi. Print() digunakan untuk membuat baris baru



### SOAL 3



Jupyter Prastha Pradipta Purusa(71230987) Laprak 6 L

```
In [4]: tinggi = int(input("Masukan tinggi:"))
lebar = int(input("Masukan lebar:"))

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

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

Kita membuat program membuat deret. Pertama kita membuat input untuk tinggi dan lebar. Kita memulai dengan angka = 1 supaya tidak dari 0. For i in range(tinggi) untuk membuat tingginya terlebih dahulu. For j in range(lebar) digunakan untuk membuat lebar. Untuk mengeluarkan angka menggunakan print(angka,end=" "). Angka += 1 berfungsi untuk menambahkan angka sebanyak 1. Print() berfungsi untuk membuat baris baru.