



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230992</b>
<b>Nama Lengkap</b>	<b>Andriano Kurniawan Ladjeba</b>
<b>Minggu ke / Materi</b>	<b>13 / Rekursif</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.

### MATERI 1

#### Pengertian Rekursif

Fungsi rekursif, juga disebut sebagai "fungsi yang memanggil dirinya sendiri", adalah fungsi yang berisi dirinya sendiri atau fungsi yang mendefinisikan dirinya sendiri. Fungsi rekursif adalah fungsi matematis yang berulang yang memiliki pola yang terstruktur, tetapi biasanya diperlukan perhatian agar fungsi ini dapat berhenti dan tidak menghabiskan memori. Fungsi rekursif harus digunakan dengan hati-hati karena memiliki pola yang terstruktur.

Untuk fungsi rekursif, ada dua blok penting: blok yang berfungsi sebagai titik berhenti dari proses rekursif dan blok yang memanggil dirinya sendiri. Fungsi ini akan berlanjut sampai kondisi berhenti terpenuhi. Rekursif terdiri dari dua bagian:

- Base Case adalah bagian dimana penentu bahwa fungsi rekursif itu berhenti
- Rekursif Case adalah bagian dimana terdapat statement yang akan terus diulang-terus menerus hingga mencapai Base Case

#### Kelebihan dan Kekurangan

Beberapa keunggulan fungsi rekursif adalah sebagai berikut:

1. Kode program lebih singkat dan elegan.
2. Masalah kompleks dapat di breakdown menjadi sub masalah yang lebih kecil di dalam rekursif

Sedangkan kelemahan fungsi rekursif adalah:

1. Memakan memori yang lebih besar karena setiap kali bagian dirinya dipanggil maka dibutuhkan sejumlah ruang memori tambahan.
2. Mengorbankan efisiensi dan kecepatan.
3. Fungsi rekursif sulit dilakukan debugging dan kadang sulit dimengerti.

#### Bentuk Umum dan Studi Kasus

Bentuk umum fungsi rekursif pada Python:

```
def function_name(parameter_list):  
  
function_name(...)
```

Sebenarnya, setiap fungsi rekursif pasti memiliki solusi iteratifnya. Contoh kasus faktorial berikut adalah sebagai berikut: Faktorial adalah menghitung perkalian deret angka  $1 \times 2 \times 3 \times \dots \times n$ . Algoritma yang digunakan untuk menghitung faktorial adalah sebagai berikut:

1. Tanyakan  $n$
2. Siapkan variabel total untuk menampung hasil perkalian faktorial dan set nilai awal dengan 0
3. Loop dari  $i = 1$  hingga  $n$  untuk mengerjakan:
4.  $\text{total} = \text{total} * i$
5. Tampilkan total Dengan menggunakan fungsi rekursif maka faktorial dapat dihitung dengan rumus pada gambar 13.1

$$\text{fact}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot \text{fact}(n - 1) & \text{if } n > 0 \end{cases}$$

Gambar 13.1: Rumus faktorial

Dari rumus 13.1 dapat dibuat pseudocode secara rekursif seperti pada gambar 13.2 Dari pseudocode, maka kode Python yang dapat dibuat adalah:

**Pseudocode (recursive):**

```
function factorial is:
input: integer  $n$  such that  $n \geq 0$ 
output:  $[n \times (n-1) \times (n-2) \times \dots \times 1]$ 

    1. if  $n$  is 0, return 1
    2. otherwise, return  $[n \times \text{factorial}(n-1)]$ 

end factorial
```

Gambar 13.2 Pseudocode rekursif faktorial

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

print(faktorial(4))
```

Hasilnya :

```
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13\pratice13>
h/Semester 2/Prakkrikum Algoritma dan Pemograman\tugas13/pratice13/has.py"
24
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13\pratice13>
```

Bagaimana proses perhtiungan yang terjadi? Berikut adalah gambarannya:

```
1.
2.   calc_factorial(4)                # 1st call with 4
3.   4 * calc_factorial(3)            # 2nd call with 3
4.   4 * 3 * calc_factorial(2)        # 3rd call with 2
5.   4 * 3 * 2 * calc_factorial(1)    # 4th call with 1
6.   4 * 3 * 2 * 1                    # return from 4th call as number=1
7.   4 * 3 * 2                        # return from 3rd call
8.   4 * 6                            # return from 2nd call
9.   24                              # return from 1st call
```

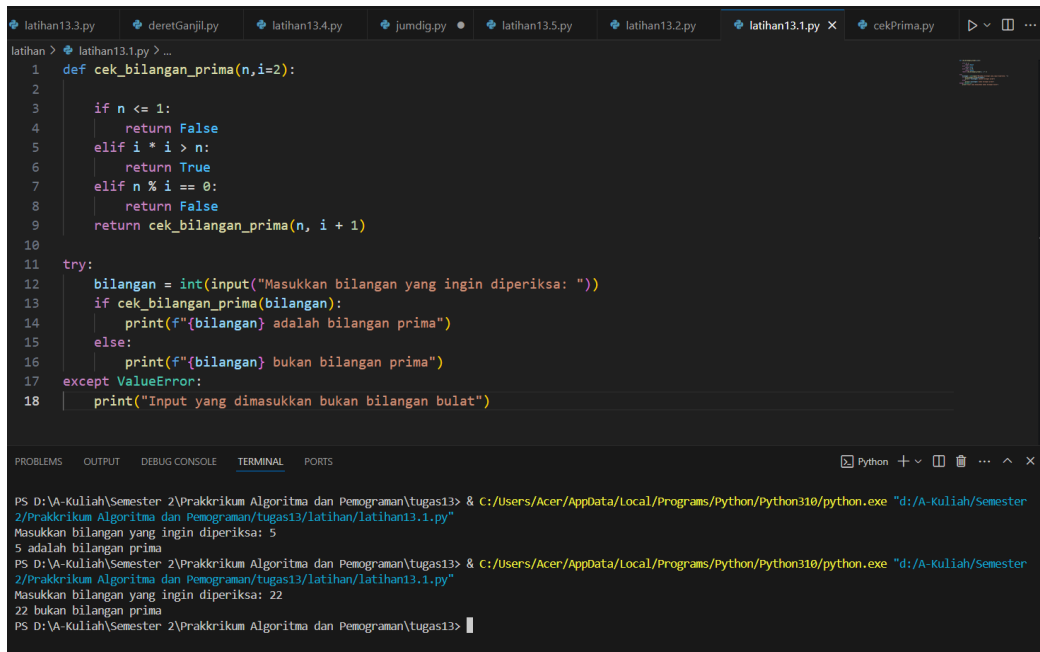
Gambar 13.3: Proses perhitungan faktorial rekursif

## 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 13.1



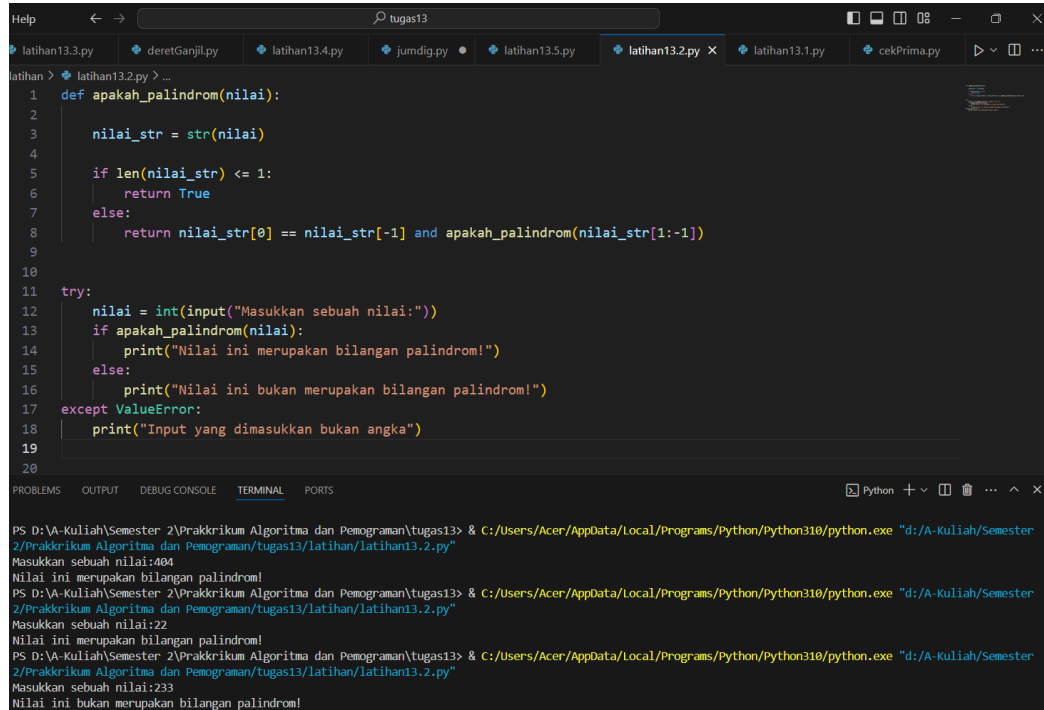
```
1 def cek_bilangan_prima(n,i=2):
2
3     if n <= 1:
4         return False
5     elif i * i > n:
6         return True
7     elif n % i == 0:
8         return False
9     return cek_bilangan_prima(n, i + 1)
10
11 try:
12     bilangan = int(input("Masukkan bilangan yang ingin diperiksa: "))
13     if cek_bilangan_prima(bilangan):
14         print(f"{bilangan} adalah bilangan prima")
15     else:
16         print(f"{bilangan} bukan bilangan prima")
17 except ValueError:
18     print("Input yang dimasukkan bukan bilangan bulat")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:\Users\Acer\AppData\Local\Programs\Python\Python310\python.exe "d:/A-Kuliah/Semester
2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.1.py"
Masukkan bilangan yang ingin diperiksa: 5
5 adalah bilangan prima
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:\Users\Acer\AppData\Local\Programs\Python\Python310\python.exe "d:/A-Kuliah/Semester
2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.1.py"
Masukkan bilangan yang ingin diperiksa: 22
22 bukan bilangan prima
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13>
```

## SOAL 2

### Latihan 13.2



```
def apakah_palindrom(nilai):
    nilai_str = str(nilai)
    if len(nilai_str) <= 1:
        return True
    else:
        return nilai_str[0] == nilai_str[-1] and apakah_palindrom(nilai_str[1:-1])

try:
    nilai = int(input("Masukkan sebuah nilai:"))
    if apakah_palindrom(nilai):
        print("Nilai ini merupakan bilangan palindrom!")
    else:
        print("Nilai ini bukan merupakan bilangan palindrom!")
except ValueError:
    print("Input yang dimasukkan bukan angka")
```

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.2.py"

Masukkan sebuah nilai:404

Nilai ini merupakan bilangan palindrom!

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.2.py"

Masukkan sebuah nilai:22

Nilai ini merupakan bilangan palindrom!

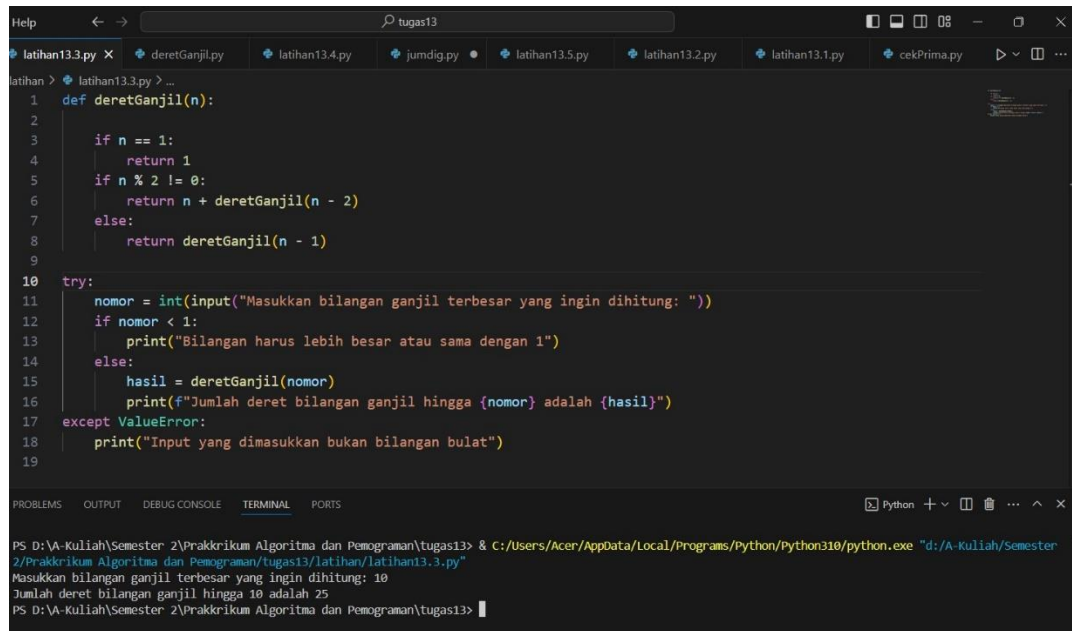
PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.2.py"

Masukkan sebuah nilai:233

Nilai ini bukan merupakan bilangan palindrom!

## SOAL 3

### Latihan 13.3



```
def deretGanjil(n):
    if n == 1:
        return 1
    if n % 2 != 0:
        return n + deretGanjil(n - 2)
    else:
        return deretGanjil(n - 1)

try:
    nomor = int(input("Masukkan bilangan ganjil terbesar yang ingin dihitung: "))
    if nomor < 1:
        print("Bilangan harus lebih besar atau sama dengan 1")
    else:
        hasil = deretGanjil(nomor)
        print(f"Jumlah deret bilangan ganjil hingga {nomor} adalah {hasil}")
except ValueError:
    print("Input yang dimasukkan bukan bilangan bulat")
```

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakkrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.3.py"

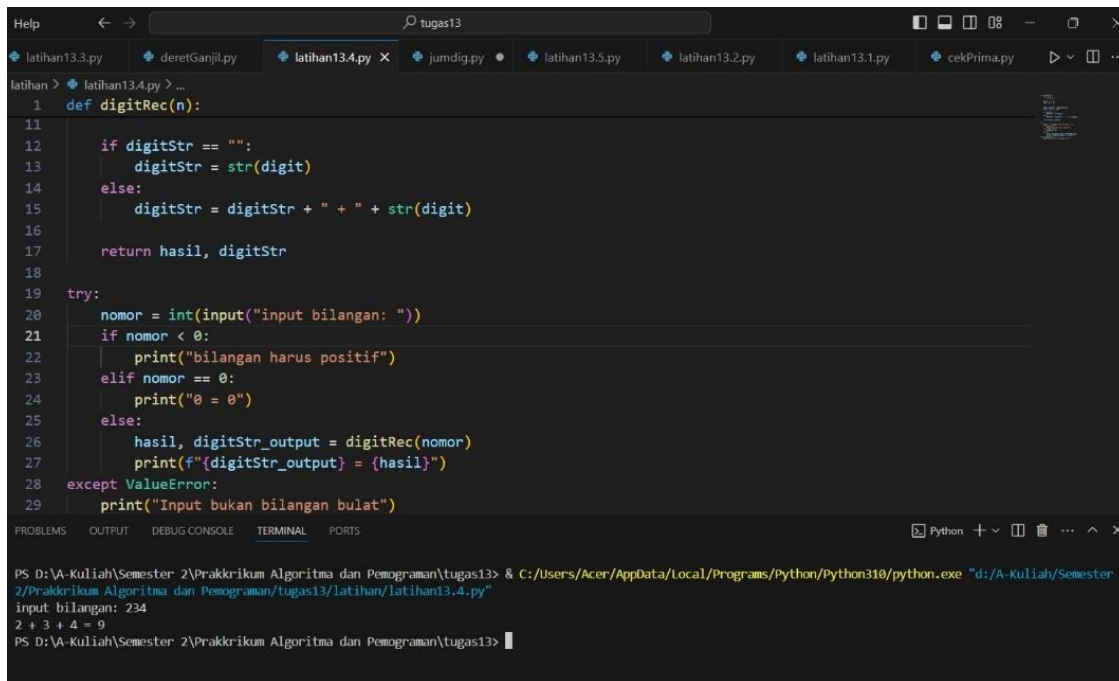
Masukkan bilangan ganjil terbesar yang ingin dihitung: 10

Jumlah deret bilangan ganjil hingga 10 adalah 25

PS D:\A-Kuliah\Semester 2\Prakkrikum Algoritma dan Pemograman\tugas13>

## SOAL 4

### Latihan 13.4



```
def digitRec(n):
    if digitStr == "":
        digitStr = str(digit)
    else:
        digitStr = digitStr + " " + str(digit)
    return hasil, digitStr

try:
    nomor = int(input("input bilangan: "))
    if nomor < 0:
        print("bilangan harus positif")
    elif nomor == 0:
        print("0 = 0")
    else:
        hasil, digitStr_output = digitRec(nomor)
        print(f"{digitStr_output} = {hasil}")
except ValueError:
    print("Input bukan bilangan bulat")
```

PS D:\A-Kuliah\Semester 2\Prakrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.4.py"

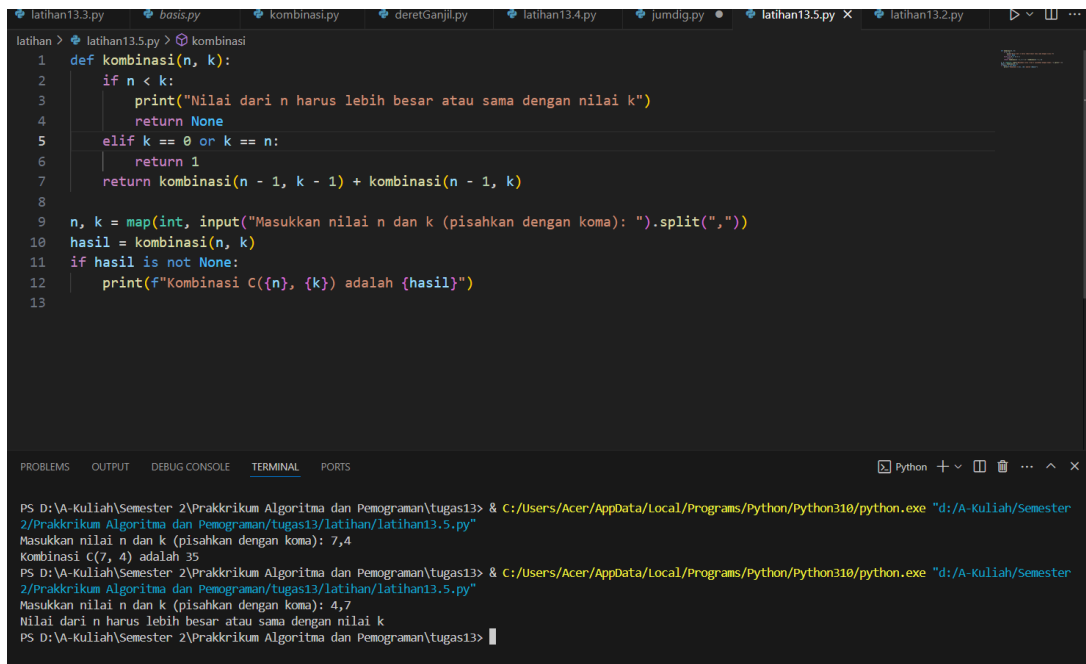
input bilangan: 234

2 + 3 + 4 = 9

PS D:\A-Kuliah\Semester 2\Prakrikum Algoritma dan Pemograman\tugas13>

## SOAL 5

### Latihan 13.5



```
def kombinasi(n, k):
    if n < k:
        print("Nilai dari n harus lebih besar atau sama dengan nilai k")
        return None
    elif k == 0 or k == n:
        return 1
    return kombinasi(n - 1, k - 1) + kombinasi(n - 1, k)

n, k = map(int, input("Masukkan nilai n dan k (pisahkan dengan koma): ").split(","))
hasil = kombinasi(n, k)
if hasil is not None:
    print(f"Kombinasi C({n}, {k}) adalah {hasil}")
```

PS D:\A-Kuliah\Semester 2\Prakrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.5.py"

Masukkan nilai n dan k (pisahkan dengan koma): 7,4

Kombinasi C(7, 4) adalah 35

PS D:\A-Kuliah\Semester 2\Prakrikum Algoritma dan Pemograman\tugas13> & C:/Users/Acer/AppData/Local/Programs/Python/Python310/python.exe "d:/A-Kuliah/Semester 2/Prakrikum Algoritma dan Pemograman\tugas13\latihan\latihan13.5.py"

Masukkan nilai n dan k (pisahkan dengan koma): 4,7

Nilai dari n harus lebih besar atau sama dengan nilai k

PS D:\A-Kuliah\Semester 2\Prakrikum Algoritma dan Pemograman\tugas13>

Github : <https://github.com/EchoGinDev/tugasAlpro13.git>