



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230981>
Nama Lengkap	<Deo Dewanto >
Minggu ke / Materi	14 / Rekursif

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.

### Pengertian Rekursif

Fungsi rekursif adalah fungsi yang berisi dirinya sendiri atau mendefinisikan dirinya sendiri. Fungsi ini sering kali disebut sebagai fungsi yang melakukan panggilan terhadap dirinya sendiri. Fungsi rekursif adalah jenis fungsi matematis yang berulang dengan pola yang terstruktur. Namun, perlu diperhatikan bahwa fungsi ini harus diatur agar bisa berhenti pada suatu titik dan tidak menyebabkan penggunaan memori berlebihan. Penggunaan fungsi rekursif harus dilakukan dengan hati-hati karena dapat menyebabkan Infinite loop.

Fungsi ini akan terus berjalan sampai kondisi berhenti terpenuhi, oleh karena itu dalam sebuah fungsi rekursif perlu terdapat 2 blok penting, yaitu blok yang menjadi titik berhenti dari sebuah proses rekursif dan blok yang memanggil dirinya sendiri. Di dalam rekursif terdapat 2 bagian:

Di dalam fungsi rekursif terdapat dua blok atau bagian, yaitu blok yang menjadi titik henti dari proses rekursif dan blok yang memanggil dirinya sendiri.

- Base Case :Kondisi yang menentukan kapan rekursi harus berhenti.
- Rekursif Case: Bagian rekursif di mana fungsi memanggil dirinya sendiri hingga mencapai Base Case

### Kelebihan dan Kekurangan

#### Kelebihan dari fungsi rekursif:

1. Kode program lebih singkat dan elegan.
2. Masalah kompleks dapat dipecah menjadi submasalah yang lebih kecil di dalam rekursif

#### Kekurangan dari fungsi rekursif

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

### Bentuk Umum Studi Kasus

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

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

Setiap fungsi rekursif pasti memiliki solusi iteratif. Sebagai contoh, mari kita lihat kasus perhitungan faktorial. Faktorial adalah hasil perkalian dari deret angka  $1 \times 2 \times 3 \times \dots \times n$ . Algoritma untuk menghitung faktorial dapat dijelaskan sebagai berikut:

1. Input nilai  $n$ .
2. Siapkan variabel total untuk menyimpan hasil perkalian faktorial dan set nilai awalnya menjadi 1.
3. Lakukan loop dari  $i = 1$  hingga  $n$ , dengan langkah-langkah:
4. total totali.
5. Tampilkan hasil total.

#### 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
```

```
1 def faktorial(n):
2     if n==0 or n==1:
3         return 1
4     else:
5         return faktorial(n-1) * n
6 print(faktorial(4))
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
/usr/local/bin/python3 /Users/macbook/lapra...
macbook@MacBook-Pro-2 laprak13fungsirekursif
rekursif/contoh.py
24
```

Contoh gambar proses perhitungan

```
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
```

## 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 1
def cek_prima(bilangan, pembagi=None):
    if pembagi is None:
        |   pembagi = bilangan - 1

    if bilangan <= 1:
        |   return False
    elif pembagi == 1:
        |   return True
    elif bilangan % pembagi == 0:
        |   return False
    else:
        |   return cek_prima(bilangan, pembagi - 1)

angka = int(input("Masukkan bilangan: "))
if cek_prima(angka):
    |   print(f"{angka} adalah bilangan prima.")
else:
    |   print(f"{angka} bukan bilangan prima.")

✓ 3.3s
2 adalah bilangan prima.
```

### PENJELASAN

Fungsi `cek_prima` digunakan untuk mengecek apakah suatu bilangan adalah bilangan prima menggunakan rekursi. Jika `pembagi` tidak diberikan, fungsi menginisialisasinya dengan `bilangan - 1`. Base case pertama mengembalikan `False` jika bilangan kurang dari atau sama dengan 1. Base case kedua mengembalikan `True` jika `pembagi` sama dengan 1, menandakan bilangan tersebut adalah bilangan prima. Jika bilangan habis dibagi `pembagi`, fungsi mengembalikan `False`, menunjukkan bilangan tersebut bukan bilangan prima. Jika tidak, fungsi memanggil dirinya sendiri dengan `pembagi - 1`. Fungsi ini dipanggil dengan input dari pengguna, dan hasilnya menunjukkan apakah bilangan tersebut adalah bilangan prima atau bukan.

### SOAL 2

```
#Latihan 2
def palindrom(kalimat):
    |   if len(kalimat) < 2:
    |       |   return True
    |   elif kalimat[0] == kalimat[-1]:
    |       |   return palindrom(kalimat[1:-1])
    |   return False

kal = input("masukkan kalimat = ")
kal = kal.replace(' ', '').lower()
if palindrom(kal):
    |   print("kalimat tersebut palindrom")
else:
    |   print("kalimat tersebut bukan palindrom")

✓ 6.3s
kalimat tersebut palindrom
```

## PENJELASAN

Pertama kita buat fungsi Bernama palindrom dengan parameter kalimat. Kemudian dalam fungsi tersebut kita buat base case Panjang kalimat kurang dari 2, karena sudah pasti palindrom. Kemudian pada elif kita beri kondisi array index 0 kalimat sama dengan array index terakhir kalimat maka fungsi ini akan rekursi dan memanggil dirinya sendiri jika index kalimat awal dan akhir tidak sama akan

## SOAL 3

```
#Latihan 3
def deret_ganjil(x):
    if x <= 0:
        return 0
    else:
        return (2 * x - 1) + deret_ganjil(x - 1)

bil = int(input("masukkan bilangan = "))
print(deret_ganjil(bil))
```

✓ 1.1s

16

## PENJELASAN

Pertama kita buat fungsi deret ganjil dengan parameter 'x'. Kita beri base casenya dengan kondisi  $x \leq 0$  karena sudah pasti hasilnya 0. Kemudian pada else kita returnkan nilai jumlah deret ganjil dengan rekursi yang bisa dilihat di source code.

## SOAL 4

```
#Latihan 4
def jumlah_digit(bilangan):
    if bilangan < 10:
        return bilangan, str(bilangan)
    else:
        hasil, representasi = jumlah_digit(bilangan // 10)
        sisa = bilangan % 10
        return sisa + hasil, representasi + " + " + str(sisa)

def print_jumlah_digit(bilangan):
    hasil, representasi = jumlah_digit(bilangan)
    print(f"{representasi} = {hasil}")

print_jumlah_digit(234)
```

✓ 0.0s

2 + 3 + 4 = 9

## PENJELASAN

Kode ini terdiri dari dua fungsi: 'jumlah\_digit' dan 'print jumlah\_digit'. Fungsi 'jumlah\_digit' menghitung jumlah digit bilangan secara rekursif dan membuat string representasi dari proses penjumlahan. Fungsi 'print jumlah\_digit' memanggil 'jumlah\_digit', kemudian mencetak string

representasi penjumlahan diikuti dengan tanda sama dengan (=) dan hasil akhirnya. Misalnya, saat `print_jumlah_digit(234)` dijalankan, hasilnya akan mencetak "2+ 3+ 4 = 9".

## SOAL 5

```
#Latihan 5
def kombinasi(n, k):
    if k == 0 or k == n:
        return 1
    else:
        return kombinasi(n-1, k-1) + kombinasi(n-1, k)

n = 5
k = 2
print(f"C({n}, {k}) = {kombinasi(n, k)}")
```

✓ 0.0s

C(5, 2) = 10

## PENJELASAN

Fungsi `kombinasi(n, k)` digunakan untuk menghitung kombinasi, yaitu jumlah cara memilih  $k$  objek dari  $n$  objek tanpa memperhatikan urutan. Basis rekursif fungsi ini adalah jika  $k$  sama dengan 0 atau  $k$  sama dengan  $n$ , maka hasilnya adalah 1, karena ada tepat satu cara untuk memilih semua atau tidak memilih sama sekali dari  $n$  objek. Untuk kasus rekursif, fungsi menggunakan sifat kombinasi:  $C(n, k) = C(n-1, k-1) + C(n-1, k)$ . Ini berarti bahwa untuk menghitung jumlah cara memilih  $k$  objek dari  $n$ , kita bisa menjumlahkan jumlah cara memilih  $k-1$  objek dari  $n-1$  dan jumlah cara memilih  $k$  objek dari  $n-1$ . Contoh penggunaan fungsi ini diberikan dengan menghitung kombinasi dari 5 objek yang dipilih 2, yang ditampilkan sebagai  $C(5, 2)$ ,

LINK GITHUB : <https://github.com/Deodewanto07/LAPRAK14.git>