

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
Minggu ke / Materi	13 / 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
2023

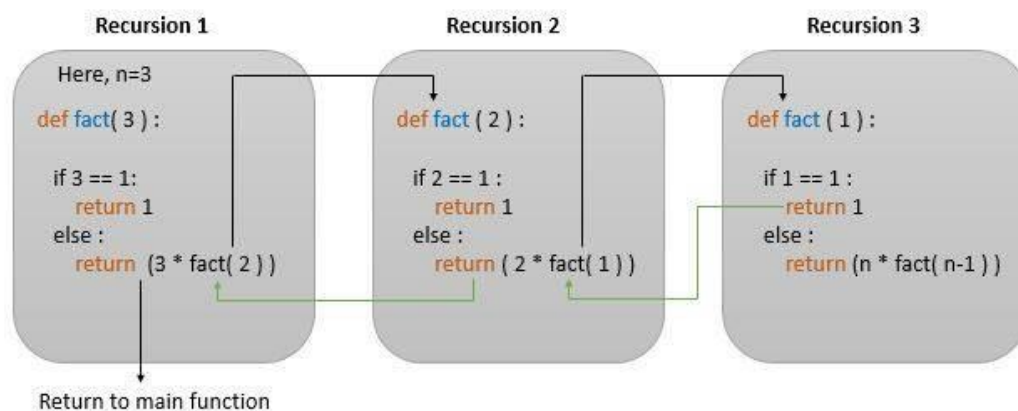
BAGIAN 1: MATERI MINGGU INI (40%)

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 adalah fungsi yang berisi dirinya sendiri atau fungsi yang mendefinisikan dirinya sendiri. Fungsi ini sering disebut sebagai **fungsi yang memanggil dirinya sendiri**. Fungsi rekursif merupakan fungsi matematis yang berulang dan memiliki pola yang terstruktur, namun biasanya fungsi ini perlu diperhatikan agar fungsi ini dapat berhenti dan tidak menghabiskan memori.



Gambar 1. rekursif (<https://id.linkedin.com/pulse/bahasa-python-function-recursion-def-aris-ihwan>)

Tujuan Rekursif

1. Dapat digunakan untuk memecahkan masalah yang kompleks menjadi sub masalah yang lebih sederhana.
2. Algoritma ini sering digunakan dalam implementasi struktur data seperti pohon, graf, atau daftar terhubung.
3. Serta dapat digunakan untuk melakukan pengulangan secara efisien.

Jenis Algoritma Rekursif

- Faktorial: untuk menghitung faktorial dari suatu bilangan.
- Tower of Hanoi: untuk memindahkan tumpukan cakram dari satu tiang ke tiang lainnya.
- DFS of Graph: untuk melakukan pencarian pada struktur data graf secara rekursif.
- Esponential: untuk menghitung hasil dari suatu operasi eksponensial dengan memanggil diri sendiri.

source: <https://fikti.umsu.ac.id/algoritma-rekursif-pengertiantujuan-dan-jenisnya/>

MATERI 2

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.

Pemanggilan rekursif.

```
def halo():  
    print("Hello World")  
    halo()
```

```
halo()
```

Terjadi error karena perulangan terlalu banyak dan tidak ada tanda-tanda akan berhenti.

Membuat deret angka dengan rekursif.

```
def perulangan(angka):  
    if angka > 1:  
        print(angka, end="")  
        angka -= 1  
        perulangan(angka)  
    else:  
        print(angka)  
  
pengguna = int(input("Masukkan angka:"))  
perulangan(pengguna)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Program  
s/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py  
Masukkan angka:5  
54321
```

Perulangan for loop dengan rekursif.

```
def tampilkanAngka (batas, i = 1):  
    print(f'Perulangan ke {i}')
```

```

    if (i < batas):
        tampilkanAngka(batas, i + 1)

tampilkanAngka(5)

```

OUTPUT:

```

PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py
Perulangan ke 1
Perulangan ke 2
Perulangan ke 3
Perulangan ke 4
Perulangan ke 5

```

Perpangkatan dengan rekursif.

```

def power(n, pangkat):
    if pangkat == 1:
        print(f"{n} = ", end="")
        return n
    elif pangkat == 0:
        return 1
    else:
        print(f"{n} * ", end="")
        return n * power(n, pangkat - 1)

print(power(2, 4))

```

OUTPUT:

```

PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py
2 * 2 * 2 * 2 = 16

```

Penjumlahan dengan rekursif.

```

def soal_1(x, y):
    if y == 1:
        print(f"{x} = ", end="")
        return x
    else:
        print(f"{x} + ", end="")
        return x + soal_1(x, y-1) # memanggil fungsi lagi

print(soal_1(3, 4))

```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py
3 + 3 + 3 + 3 = 12
```

Rumus Permutasi:

$$P_r = \frac{n!}{(n - r)!}$$

Menghitung permutasi menggunakan faktorial.

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

def permutasi(n, r):
    return faktorial(n) // faktorial(n - r)

print(f"Permutasi P({10}, {4}) = {permutasi(10, 4)}")
```

Menghitung permutasi menggunakan library Python.

```
from numpy import prod
from itertools import permutations

def permutasi(m,n):
    return prod(range(m, 1, -1)) / prod(range(m-n, 1, -1))

print(permutasi(10, 4))
```

Menghitung permutasi secara manual.

```
def permutasi_manual(m, n):
    if n == 0:
        return 1
    else:
        return (permutasi_manual(m, n-1) * (m-n+1))

print(permutasi_manual(10, 4))
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py
5040.0
5040
Permutasi P(10, 4) = 5040
```

Konversi angka menjadi nilai biner,

```
def konversi(angka, basis):
    biner = "0123456789ABCDEF"
    if angka < basis:
        return biner[angka]
    else:
        return konversi(angka//basis, basis) + biner[angka%basis]

print(konversi(8, 2))
print(konversi(10, 16))
```

biner berisi karakter-karakter yang akan digunakan sebagai representasi bilangan dalam basis tertentu. Jika $\text{angka} < \text{basis}$, maka fungsi mengembalikan karakter dari string biner sesuai dengan nilai angka. Jika, tidak, fungsi melakukan rekursif dengan parameter $\text{angka} // \text{basis}$, dan basis untuk mendapatkan digit-digit konversi dari angka, lalu menambahkannya dengan karakter dari biner yang sesuai dengan sisa pembagian $\text{angka} \% \text{basis}$ untuk mendapatkan digit konversi.

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_13/Modul.py
1000
A
PS D:\71200645_PrakAlpro_13>
```

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 Vidi adalah adik Tono yang sedang belajar bilangan prima. Vidi mengalami kesulitan untuk menentukan suatu bilangan bilangan prima atau bukan. Untuk membantu adiknya Tono kemudian membuat program untuk pengecekan bilangan prima dengan menggunakan fungsi rekursif. Bantulah Tono untuk menyelesaikan tugas tersebut. ■

Pengerjaan soal dilakukan dengan menyesuaikan fungsi dengan basis angka termasuk bilangan prima. Jika $n < 2$ atau bisa dibagi dengan angka selain satu dan dirinya sendiri, artinya bilangan tersebut bukan prima. Fungsi menggunakan dua parameter, yaitu n untuk angka yang akan diperiksa, dan i sebagai pembagi yang default dimulai dari dua. Jika $n == 2$, maka prima. Jika n tidak menyisakan sisa dibagi dengan i , maka bukan prima. Jika i dikali i lebih besar dari n , maka tidak ada pembagi yang ditemukan, maka prima. Jika tidak ada syarat yang terpenuhi, maka fungsi rekursif dengan $i + 1$ untuk memeriksa pembagi berikutnya.

SOURCE CODE:

```
def prima(n, i=2):
    if n <= 2:
        return n == 2
    if n % i == 0:
        return False
    if i * i > n:
        return True

    return prima(n, i + 1)

angka = 43
if prima(angka):
    print(f"{angka} adalah bilangan prima.")
else:
    print(f"{angka} bukan bilangan prima.")
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_13/Latihan 1.1.py"
43 adalah bilangan prima.
```

SOAL 2

Latihan 13.2 Buatlah fungsi rekursif mengetahui suatu kalimat adalah palindrom atau bukan! ■

Pengerjaan soal dilakukan dengan menghapus spasi dan huruf alfanumerik, lalu mengecilkan setiap huruf pada kalimat. Jika panjang kalimat ≤ 1 , maka palindrom. Jika huruf pertama pada kalimat tidak sama dengan huruf akhir, maka bukan palindrom. Dilakukan pemanggilan fungsi rekursif pada kalimat tanpa huruf pertama dan huruf akhir.

SOURCE CODE:

```
def palindrom(kalimat):
    kalimat = ''.join(i.lower() for i in kalimat if i.isalnum())

    if len(kalimat) <= 1:
        return True
    if kalimat[0] != kalimat[-1]:
        return False
    return palindrom(kalimat[1:-1])

kalimat = "nama Ibu ini Ubiaman"
if palindrom(kalimat):
    print("Kalimat tersebut adalah palindrom.")
else:
    print("Kalimat tersebut bukan palindrom.")
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe "d:/71200645_PrakAlpro_13/Latihan
1.2.py"
Kalimat tersebut adalah palindrom.
```

SOAL 3

Latihan 13.3 Buatlah fungsi rekursif untuk menghitung jumlah deret ganjil dari $1 + 3 + 7 + \dots + n!$

Pengerjaan soal dilakukan dengan fungsi dengan parameter n, angka ganjil atau genap terakhir dari penjumlahan. Jika $n == 1$, maka dikembalikan satu. Jika nilai n yang diterima berupa nilai genap berarti tidak ada sisa dibagi dengan dua, maka nilai n dikurang 1. Mengembalikan nilai penjumlahan dengan n ditambah fungsi rekursif (n-2) untuk mendapatkan nilai ganjil sebelumnya.

SOURCE CODE:

```
def deret_ganjil(n):
    if n == 1:
        return 1
    elif n % 2 == 0:
        n -= 1
    return n + deret_ganjil(n - 2)

print(f"Jumlah deret ganjil adalah {deret_ganjil(9)}")
```


OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_13/Latihan 1.3.py"
Jumlah deret ganjil adalah 25
```

SOAL 4

Latihan 13.4 Buatlah fungsi rekursif untuk mengetahui jumlah digit dari suatu bilangan. Seperti misalnya tulisan: "234" maka jumlah digitnya adalah $2+3+4 = 9$!

Pengerjaan soal dilakukan dengan mengambil nilai n dari pengguna, untuk dihitung penjumlahan per-digit-nya. Jika nilai yang dimasukkan hanya satu digit, maka dikembalikan nilai itu saja. Fungsi mengambil digit terakhir dari n dengan $n \% 10$ (sisa dari pembagian n dengan 10) dan menambahkan ke fungsi jumlah_angka dengan $n//10$ (n tanpa digit terakhir). Fungsi ini berlanjut secara rekursif sampai $n < 10$.

SOURCE CODE:

```
def jumlah_angka(n):
    if n < 10:
        return n
    else:
        return n % 10 + jumlah_angka(n // 10)

print(jumlah_angka(234))
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_13/Latihan 1.4.py"
Hasil = 9
PS D:\71200645_PrakAlpro_13>
```

```
print(234 % 10) # 4
print(234 // 10) # 23
# 23 > 10, sehingga rekursif lagi
print(23 % 10) # 3
print(23 // 10) # 2
# Hasil 4 + 3 + 2 = 9
```

SOAL 5

Latihan 13.5 Buatlah fungsi rekursif untuk menghitung kombinasi!

Pengerjaan soal dilakukan dengan membuat fungsi rekursif untuk perhitungan faktorial dengan parameter n . Lalu pada fungsi rekursif kombinasi diambil dua parameter, yaitu n dan r . Dapat dikerjakan sesuai rumus kombinasi dengan faktorial:

$$C_r^n = \frac{n!}{r! (n-r)!}$$

atau dengan rumus $C(n,r) = C(n-1, k-1) + C(n-1, r)$. Jika $r == 0$ atau $r == n$, dikembalikan satu karena untuk memilih kombinasi dapat dengan tidak memilih apa pun atau memilih semuanya. Jika tidak, fungsi mengembalikan jumlah kombinasi dari dua kasus: ketika elemen ke-n dipilih (jumlah elemen yang harus dipilih berkurang menjadi $(r-1)$ dan ketika elemen ke-n tidak dipilih (jumlah elemen yang harus dipilih tetap r).

SOURCE CODE:

Menghitung kombinasi dengan faktorial.

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

def kombinasi(n, r):
    if r == 0 or r == n:
        return 1
    else:
        return faktorial(n) // (faktorial(r) * faktorial(n - r))

print(f"Kombinasi C({5}, {3}) = {kombinasi(5, 3)}")
```

Menghitung kombinasi secara manual.

```
def kombinasi(n, r):
    if r == 0 or r == n:
        return 1
    else:
        return kombinasi(n - 1, r - 1) + kombinasi(n - 1, r)

print(f"Kombinasi C({5}, {3}) = {kombinasi(5, 3)}")
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_13> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_13/Latihan 1.5.py"
Kombinasi C(5, 3) = 10
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_13