Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**Samuel Natanael**
Nama Lengkap	**71231050**
Minggu ke / Materi	**13 / Recursif**

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

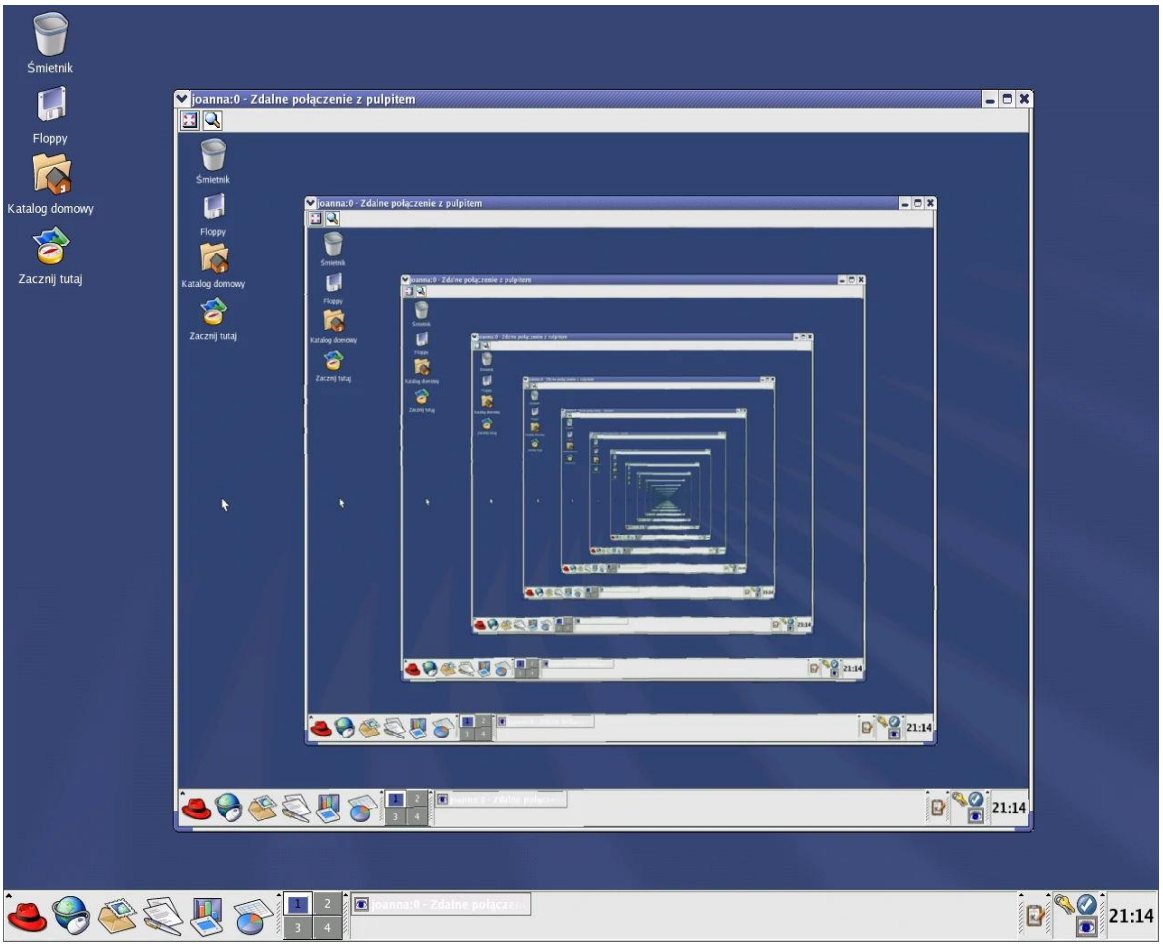
BAGIAN 1: MATERI MINGGU INI (40%)

Pengertian Rekursif

Fungsi rekursif adalah proses di mana sebuah fungsi memanggil dirinya sendiri, baik secara langsung maupun tidak langsung. Dengan menggunakan algoritma rekursif, beberapa masalah bisa diselesaikan dengan lebih mudah.

Fungsi rekursif juga dapat diartikan sebagai fungsi yang berulang kali memanggil dirinya sendiri dalam proses pengolahan data atau pemanggilan fungsi. Dalam beberapa kasus, penggunaan fungsi rekursif bisa lebih mudah dipahami dan lebih sederhana dibandingkan solusi iteratif. Oleh karena itu, penggunaan fungsi rekursif dapat mengurangi jumlah kode yang diperlukan serta membuatnya lebih mudah dibaca.

Dalam sistem pemrograman, rekursif terbagi menjadi dua jenis: fungsi rekursif langsung dan tidak langsung. Fungsi rekursif langsung dibagi lagi menjadi empat jenis, yaitu tail recursion, head recursion, nested recursion, dan tree recursion.



Gambar 13.1 Contoh Rekursif (diambil dari https://jagongoding.com/web/php/dasar/fungsi-rekursif/)

Fungsi rekursif akan terus berjalan sampai kondisi berhenti terpenuhi. Oleh karena itu, dalam sebuah fungsi rekursif harus ada dua blok penting: blok yang menentukan titik berhenti dari proses rekursif dan blok yang memanggil dirinya sendiri. Dalam rekursif, terdapat dua bagian utama:

- Base Case adalah bagian yang menentukan kapan fungsi rekursif harus berhenti.
- Recursive Case adalah bagian yang mengandung pernyataan yang akan terus diulang sampai mencapai Base Case.

Kelebihan dan Kekurangan

- Kelebihan rekursif
  Sangat mudah untuk melakukan perulangan dengan batasan yang luas dalam artian melakukan perulangan dalam skala yang besar, dapat melakukan perulangan dengan batasan fungsi.
- Kekurangan
  Tidak bisa melakukan nested loop atau looping bersarang.

Bentuk Umum dan Studi Kasus

Untuk membuat fungsi rekursif, caranya sama saja dengan fungsi biasa. Yang membuat sebuah fungsi menjadi rekursif adalah karena ia memanggil dirinya sendiri.

```
def Hello_World():
    # Base Case
    print('Hello World!')

    # rekursifitas
    return Hello_World()

# Memanggil fungsi helo_dunia
Hello_World()
# Output:
# Hello World!
# Traceback (most recent call last):
#   File "d:\Ngampus\Semester 2\PrakAlPro\tes13.py", line 67, in Hello_World
#     return Hello_World() # <-- rekursifitas
#            ^^^^^^^^^^^^^
#   [Previous line repeated 996 more times]
```

Error tersebut terjadi karena perulangan rekursif sudah dipanggil terlalu banyak dan tidak ada tanda-tanda akan berhenti. Oleh karena itu sistem langsung memberhentikannya secara paksa.

Contoh bentuk rekursif kedua adalah fungsi untuk menampilkan angka 0-10

```
for i in range(10):
  print(i)
```

Output:

0

1

2

3

4

5

6

7

8

9

10

Bentuk diatas menggunakan perulangan for untuk menampilkan angka 0-10. Sekarang mari kita lihat code yang berfungsi mirip tetapi dalam rekursif:

```
def tampilkanAngka(angka):
    if angka == 0:
        print(angka)
    else:
        print(angka)
        return tampilkanAngka(angka-1)

tampilkanAngka(10)
```

output:

10

9

8

7

6

5

4

3

2

1

0

Contoh Soal

1. Buatlah fungsi untuk menyelesaikan perpangkatan dengan rekursif

```python
def hitung_pangkat(bilangan, pangkat):
    if pangkat > 1:
        return bilangan * hitung_pangkat(bilangan, pangkat - 1) # Base Case dan Rekursif
    return bilangan

print(hitung_pangkat(10, 4))
#10000
```

Misalkan kita akan mencari hasil dari 10 ^ 4.

Maka pada program di atas, kita akan membuat iterasi mundur sebanyak 4 kali. Ilustrasinya seperti berikut:

```
Berapa kah 10 ^ 4?

10 ^ 4 = 10 * (10 ^ 3)

Berapa 10 ^ 3?

10 ^ 3 = 10 * (10 ^ 2)

Berapa 10 ^ 2?

10 ^ 2 = 10 * (10 ^ 1)

Berapa 10 ^ 1?

10 ^ 1 = 10
```

Gambar 13.2 Cara perhitungan rekursif perpangkatan (diambil dari https://jagongoding.com/python/latihan-logika/perpangkatan/#cara-4-perulangan-rekursif)

2. Buatlah fungsi untuk menyelesaikan factorial dengan rekursif

```
def hitung_faktorial (n):
  if n > 2:
    return n * hitung_faktorial(n - 1)
  else:
    return 2
n = 5
print(hitung_faktorial(n))
# 120
```

Cara Kerja:

1. Jika nilai n lebih besar dari 2, maka n akan dikalikan dengan hasil dari hitung_faktorial(n – 1)
2. Jika nilai n kurang dari 2, maka akan mengembalikan nilai 2. Ini merupakan base case dari rekursi.
3. Jadi, system perhitungannya adalah 5! = 5*4*3*2 = 120

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

13,1.py > ...

```python
def is_prima(n):
    # Membuat d untuk divider
    d = n-1
    # Fungsi rekursif
    def cek(n, d):
        # Fungsi cek akan membagi n dengan d, lalu rekursif fungsi lagi dengan d-1
        if d == 1:
            # Jika nilai d setelah dikurang 1 menjadi 1, maka n adalah prima
            return True
        if n % d == 0:
            # Jika n bisa dibagi d manapun maka n bukanlah prima
            return False
        else:
            # Rekursif
            return cek(n, d-1)
    return cek(n, d)

print(is_prima(17))
print(is_prima(16))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/13,1.py"
True
False
```

SOAL 2

13,2.py > ...

```python
def palindrome(k):
    # Jika hanya memiliki 1 huruf maka otomatis palindrome
    if len(k) <= 1:
        return True
    # Mengecek huruf pertama dan terakhir
    elif k[0] == k[-1]:
        # Melakukan rekursif tetapi huruf pertama dan terakhir dihilangkan
        return palindrome(k[1:-1])
    else:
        # Jika tidak sama, maka bukan palindrome
        return False
print(palindrome("kasurrusak"))
print(palindrome("kasurrsak"))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/13,2.py"
True
False
PS D:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 3

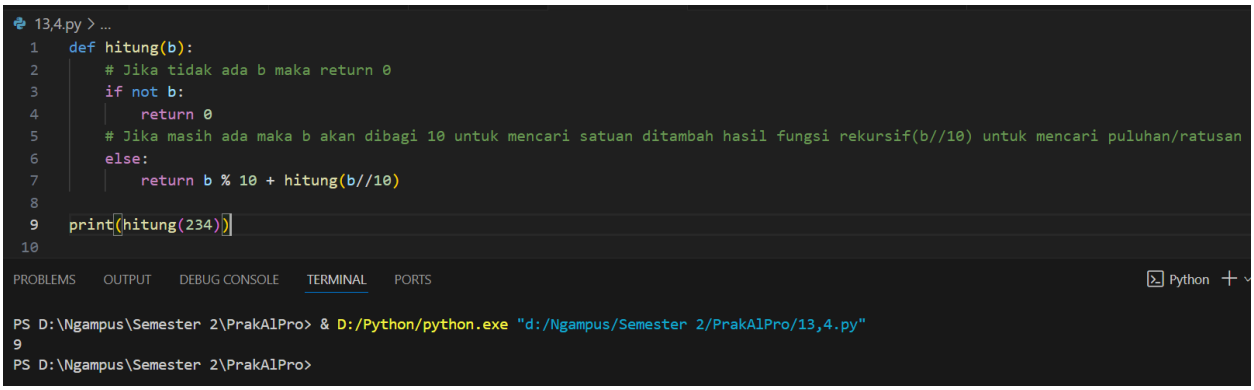
13,3.py > ...

```python
def ganjil(n, awal = None):
    # Awal bilangan ganjil = 1
    awal = 1

    # Fungsi rekursif untuk menambah ganjil
    def hitung(n, awal):
        # Jika bilangan awal setelah ditambah melebihi batas n, return 0
        if awal > n:
            return 0
        # Jika belom melebihkan batas, awal akan ditambah dengan fungsi rekursif hitung awal ditambah 2
        if n >= awal:
            return awal + hitung(n, awal+2)
    return hitung(n, awal)

print(ganjil(10))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

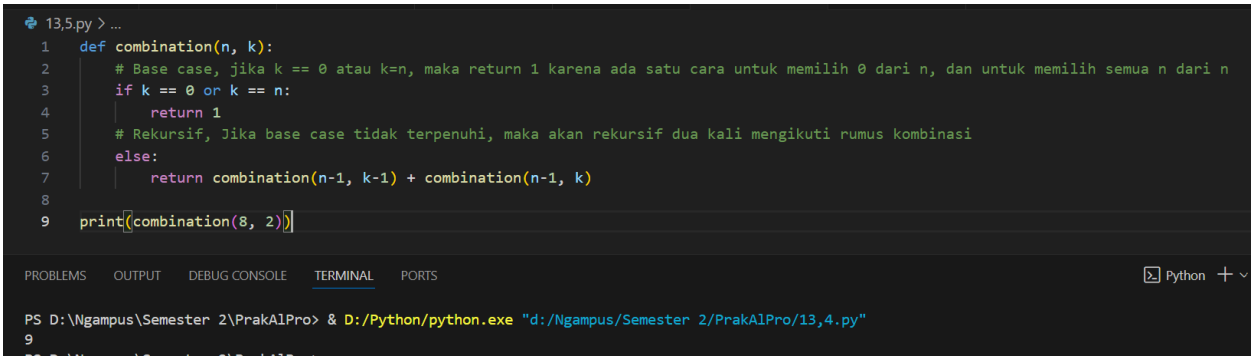
```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/13,3.py"
25
```

SOAL 4



```
13,4.py > ...
1   def hitung(b):
2       # Jika tidak ada b maka return 0
3       if not b:
4           return 0
5       # Jika masih ada maka b akan dibagi 10 untuk mencari satuan ditambah hasil fungsi rekursif(b//10) untuk mencari puluhan/ratusan
6       else:
7           return b % 10 + hitung(b//10)
8
9   print(hitung(234))
10

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/13,4.py"
9
PS D:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 5



```
13,5.py > ...
1   def combination(n, k):
2       # Base case, jika k == 0 atau k=n, maka return 1 karena ada satu cara untuk memilih 0 dari n, dan untuk memilih semua n dari n
3       if k == 0 or k == n:
4           return 1
5       # Rekursif, Jika base case tidak terpenuhi, maka akan rekursif dua kali mengikuti rumus kombinasi
6       else:
7           return combination(n-1, k-1) + combination(n-1, k)
8
9   print(combination(8, 2))

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    Python

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/13,4.py"
9
```

Github : https://github.com/SamuelN1508/PrakAlPro13.git