



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

| | |
|---------------------------|-------------------------------------|
| NIM | 71230997 |
| Nama Lengkap | Christ Jevicto Ajimas Kirana |
| Minggu ke / Materi | 14 / Fungsi 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.

MATERI 1

1.1 Pengertian Rekursif

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk berkali-kali untuk menemukan hasil, namun kita harus berhati-hati dalam memakai fungsi rekursif karena fungsi rekursif bisa jadi tidak dapat berhenti dan menjadi infinite loop yang akan menyebabkan program mengalami hung up. Contoh :

```
[Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
```

Fungsi ini tidak akan berhenti jika tidak ada kondisi yang terpenuhi, oleh karena itu fungsi rekursif memiliki 2 poin penting dalam pembuatannya, yaitu blok yang menjadi titik berhentinya fungsi dan blok yang memanggil dirinya sendiri. Di dalam rekursif memiliki 2 bagian:

- **Base Case** = adalah dimana rekursif dapat berhenti
- **Rekursif case** = adalah bagian dimana terdapat statement yang akan diulang hingga base case terpenuhi.

1.2 Kelebihan dan Kekurangan

Rekursif case memiliki kekurangan dan kelebihan.

Kekurangan rekursif yaitu:

- Memakan memori yang besar, karena setiap kali memanggil fungsi maka fungsi tersebut membutuhkan memory tambahan
- Mengorbankan efisiensi dan kecepatan
- Fungsi rekursif cenderung sulit untuk di debugging karena kadang fungsi sulit untuk dimengerti

Sedangkan kelebihan rekursif yaitu:

- Code program, lebih elegan dan singkat
- Masalah yang besar dapat dibagi menjadi masalah masalah kecil

1.3 Bentuk Umum dan Studi Kasus

Dalam fungsi rekursif pasti memiliki Solusi iteratif atau berulang ulang. Contoh pada soal menghitung angka $1 \times 2 \times 3 \times 4 \times \dots \times n$

Algoritma nya adalah

1. Menanyakan n
2. Siapkan variabel total untuk menampung hasil perkalian factorial dan set nilai awal dengan 0
3. Loop dari $i = 1$ hingga n untuk mengerjakan
4. $Total = total \text{ dikali } i$ ($total * i$)
5. Print total

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

Visualisasi rumus =

```

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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\PRALPRO\71230997_minggu14> & C:/Users/ch
 ata/Local/Programs/Python/Python311/python.exe
 PRO/71230997_minggu14/latman.py
 24

Contoh pada python =

MATERI 2

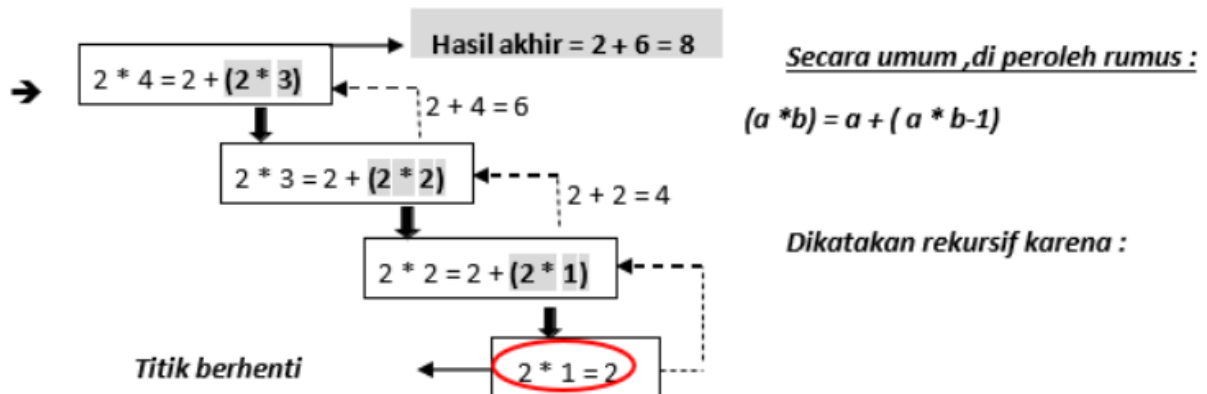
Kegiatan Praktikum

Kasus 1.1

Perkalian Menggunakan Fungsi Rekursif

Membuat program perkalian 2 angka yang menggunakan fungsi rekrusif misal angka 2 dengan 4 dengan metode seperti $2 * 4 = 2 + 2 + 2 + 2 = 8$

Dengan rumus =



Hasil=

```
1 def perkalian(a, b):
2     if b == 1:
3         print(f'{a} = ', end='')
4         return a
5     else:
6         print(f'{a} + ', end='')
7         return a + perkalian(a, b-1)
8
9 print(perkalian(2, 4))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\PRALPRO\71230997_minggu14> & C:/Users/ch
2 + 2 + 2 + 2 = 8
```

Kasus 1.2

Bagaimana jika mencari pangkat dengan fungsi rekursif

Hasil =

```
tabnine: test | explain | document | ask
1 def perkalian(a, b):
2     if b == 1:
3         print(f'{a} = ', end='')
4         return a
5     else:
6         print(f'{a} * ', end='')
7         return a * perkalian(a, b-1)
8
9 print(perkalian(2, 4))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\PRALPRO\71230997_minggu14> & C:/Users/ch
2 * 2 * 2 * 2 = 16
```

Kasus 1.3

Membuat suatu bilangan fibonacci dengan user menginputkan Batasan atau n dari fibonacci tersebut

```
tabnine: test | explain | document | ask
1 def fibonacci(n):
2     if n == 1 or n == 2:
3         return 1
4     else:
5         return(fibonacci(n-1)) + fibonacci(n-2)
6
7 n = int(input('masukan sebuah bilangan= '))
8 print(fibonacci(n))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMM

```
PS D:\PRALPRO\71230997_minggu14> & C:/Users/chris/AppData/
masukan sebuah bilangan= 7
13
```

Kasus 1.4

Membuat konfersi suatu angka menjadi basis yang di inginkan menggunakan fungsi rekrusif

```
tabnine: test | explain | document | ask
1 def toBasis(n,base):
2     convertString = "0123456789ABCDEF"
3     if n < base:
4         return convertString[n]
5     else:
6         return toBasis(n//base,base) + convertString[n%base]
7
8 print("Silahkan masukkan bilangan dan basis")
9 angka = int(input("Bilangan : "))
10 basis = int(input("Basis (2/8/16) : "))
11 print(toBasis(angka, basis))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\PRALPRO\71230997_minggu14> & C:/Users/chris/AppData/Local/Programs
Silahkan masukkan bilangan dan basis
Bilangan : 9
Basis (2/8/16) : 8
11
```

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

```
tabnine: test | explain | document | ask
def mencaribilanganprima(n,p=None):
    if p is None:
        p = n - 1
    if n < 2:
        return False
    if p == 1:
        return True
    if n % p == 0:
        return False
    return mencaribilanganprima(n, p-1)

bilangan = int(input('inputkan suatu bilangan = '))
if mencaribilanganprima(bilangan):
    print(f'{bilangan} adalah bilangan prima')
else:
    print(f'{bilangan} bukan bilangan prima')

✓ 0.0s

29 adalah bilangan prima
```

Penjelasan = n adalah inputan dari user untuk mencari bilangan prima atau bukan dengan menggunakan fungsi rekrusif

Pertama-tama user akan memberi input suatu bilangan lalu bilangan tersebut akan masuk ke dalam fungsi yang dimana jika p adalah none maka p adalah $n - 1$ lalu fungsi akan berulang terus menerus hingga $p = 1$ jika $p = 1$ maka bilangan yang di inputkan adalah bilangan prima, jika bilangan yang diinputkan kurang dari 2 dan dapat di modulo dengan p yang menghasilkan 0 dalam perulangan fungsi maka bilangan tersebut bukan bilangan Prima.

SOAL 2

```
tabnine: test | explain | document | ask
def palindrome(s, start, end):
    if start >= end:
        return True
    if s[start] != s[end]:
        return False
    return palindrome(s, start + 1, end - 1)

tabnine: test | explain | document | ask
def membersihkan(s):
    bersih = []
    for i in s:
        if i.isalnum():
            bersih.append(i.lower())
    return ''.join(bersih)

tabnine: test | explain | document | ask
def adalahpalindrome(s):
    bersihbanget = membersihkan(s)
    return palindrome(bersihbanget, 0, len(bersihbanget)-1)

kalimat = "A man, a plan, a canal, Panama"
if adalahpalindrome(kalimat):
    print(f'{kalimat} adalah palindrome')
else:
    print(f'{kalimat} bukan palindorme')
```

✓ 0.0s

A man, a plan, a canal, Panama adalah palindrome

Penjelasan =

1. **Fungsi palindrome:** Mengecek secara rekursif apakah string adalah palindrome dengan membandingkan karakter dari ujung ke tengah.
2. **Fungsi membersihkan:** Membersihkan string dari karakter non-alfanumerik dan mengubah semua huruf menjadi kecil.
3. **Fungsi adalahpalindrome:** Menggunakan **membersihkan** dan **palindrome** untuk memeriksa apakah string yang telah dibersihkan adalah palindrome.
4. **Blok utama:** Memeriksa apakah **kalimat** adalah palindrome dan mencetak hasilnya.

SOAL 3

```
def jumlahganjil(n):
    if n < 1:
        return 0
    if n % 2 == 0:
        n -= 1
    if n == 1:
        return 1
    return n + jumlahganjil(n-2)

n = int(input('masukan sebuah bilangan = '))
hasil = jumlahganjil(n)
if jumlahganjil(n):
    print(f"Jumlah deret bilangan ganjil hingga {n} adalah {hasil}.")

✓ 1.2s

Jumlah deret bilangan ganjil hingga 9 adalah 25.
```

Penjelasan =

1. **Fungsi jumlahganjil:** Menghitung jumlah bilangan ganjil dari 1 hingga **n** secara rekursif. Menangani kasus **n** genap dengan mengurangi 1.
2. **Blok utama:** Mengambil input **n**. Menghitung dan mencetak jumlah bilangan ganjil hingga **n**.

SOAL 4

```
def jumlahdaridigit(n):
    if n == 0:
        return 0
    if n <= 10:
        return n
    return n % 10 + jumlahdaridigit(n // 10)

bilangan = int(input("masukan sebuah bilangan = "))
hasil = jumlahdaridigit(bilangan)
print(f'jumlah dari {bilangan} adalah {hasil}')

✓ 1.7s

jumlah dari 234 adalah 9
```

Penjelasan =

1. **Fungsi jumlahdaridigit:** Menghitung jumlah digit-digit dari bilangan **n** secara rekursif. Jika **n** adalah 0, mengembalikan 0. Jika **n** kurang dari atau sama dengan 10, mengembalikan **n**. Jika tidak, mengembalikan digit terakhir (**n % 10**) ditambah jumlah digit dari hasil pembagian **n** dengan 10 (**jumlahdaridigit(n // 10)**).

2. **Blok utama:** Mengambil input bilangan **n** dari pengguna. Menghitung hasil jumlah digit menggunakan **jumlahdaridigit**. Mencetak hasil jumlah digit dari bilangan tersebut.

SOAL 5

```
tabnine: test | explain | document | ask
def kombinasi(n, k):
    if k == 0 or k == n:
        return 1
    return kombinasi(n-1, k-1) + kombinasi(n-1, k)

lima = (int(input('masukan sebuah bilangan = ')))
dua = (int(input('masukan sebuah bilangan = ')))
hasil = kombinasi(lima, dua)
print (f"hasil kombinasi dari ({lima}, {dua}) adalah {hasil}")

✓ 2.0s

hasil kombinasi dari (10, 8) adalah 45
```

Penejelasan =

1. **Fungsi `kombinasi`:** Menghitung nilai kombinasi $C(n, k)$ secara rekursif. Basis kasus: jika $(k = 0)$ atau $(k = n)$, mengembalikan 1. Kasus rekursif: menggunakan rumus rekursif $C(n, k) = C(n-1, k-1) + C(n-1, k)$.
2. **Blok utama:** Mengambil dua input bilangan. Menghitung kombinasi dari kedua bilangan menggunakan fungsi `kombinasi`. Mencetak hasilnya.

LINK GITHUB = <https://github.com/Yeeemeki/praktikum-algoritma-dan-pemograman.git>