

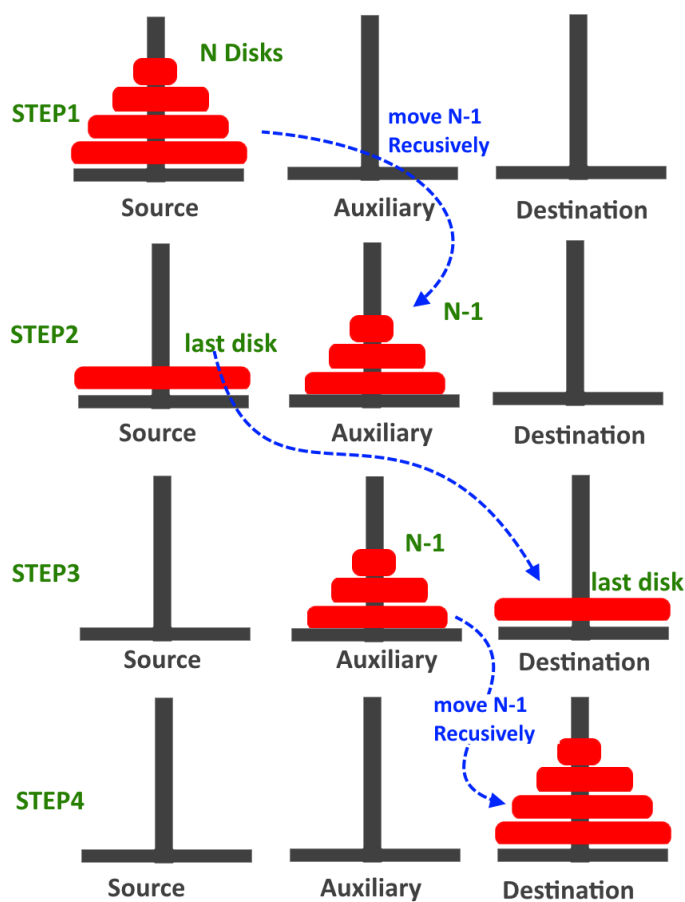
## Tugas Kelompok ke-3

### Week 9

#### Soal 1: Desain dan Analisis Recursive Function

##### Deskripsi Soal

Anda diminta untuk merancang sebuah fungsi rekursif yang menyelesaikan masalah "**Towers of Hanoi**" untuk 4 buah disk. Fungsi harus dapat mengindikasikan langkah-langkah pemindahan disk dari tiang A ke tiang C dengan menggunakan tiang B sebagai perantara.



Gambar 1. Ilustrasi Algoritma Towers of Hanoi

### Instruksi Pengerjaan

1. Buat pseudo code atau flowchart untuk fungsi rekursif yang menyelesaikan Towers of Hanoi.
2. Jelaskan secara terperinci setiap langkah rekursi yang dilakukan oleh fungsi Anda.
3. **Pseudo Code/Flowchart:** Harus menggambarkan pemanggilan rekursif yang jelas dan kondisi berhenti rekursi.
4. **Penjelasan Langkah:** Harus mencakup base case dan cara fungsi memecah masalah menjadi sub-problem yang lebih kecil.
5. **Analisis Kompleksitas:** Berikan analisis tentang kompleksitas waktu dari algoritma berdasarkan jumlah disk.

### Soal 2: Analisis Kesalahan pada Implementasi Recursive Function

#### Deskripsi Soal:

Diberikan pseudo code dari implementasi rekursif yang dirancang untuk menghitung faktorial suatu bilangan tetapi mengandung kesalahan logis. Tugas Anda adalah menganalisis dan menentukan kesalahan dalam pseudo code tersebut dan memberikan solusi yang benar.

#### Cuplikan Pseudo Code:

```
Function factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n)
```

#### Instruksi Pengerjaan:

1. Identifikasi dan jelaskan kesalahan dalam pseudo code yang diberikan.
2. Berikan perbaikan yang benar pada pseudo code.

3. Gambarkan flowchart yang menunjukkan fungsi rekursif yang benar untuk menghitung faktorial.


#### **Catatan Penting**

- Pastikan untuk mencantumkan referensi jika menggunakan sumber dari internet atau bantuan GPT.
- Cantumkan referensi dengan format: Nama sumber, Tautan (jika ada), diakses pada [Tanggal Akses].
- Pekerjaan harus asli dan setiap bentuk plagiarisme akan ditindak sesuai dengan kebijakan akademik.

## Soal 2: Analisis Kesalahan pada Implementasi Recursive Function

### Cuplikan pseudocode dan code

```
Function factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n)
```



```
1  int factorial(int n)  
2  {  
3      if (n == 0)  
4          return 1;  
5      else  
6          return n * factorial(n);  
7  }  
8
```

### 1. Identifikasi dan jelaskan kesalahan dalam pseudo code yang diberikan.

Potongan pseudo-code diatas merupakan sebuah fungsi dengan nama factorial dengan argumen n dengan block code yang berisi

- baris ke-3 berisi conditional if untuk melakukan pengecekan nilai dari n apakah n sama dengan nol
- baris ke-4 memberikan return angka 1 jika n sama dengan nol karena faktorial dari 0 adalah 1.
- baris ke-5-6 berisi else statement yang merupakan operasi perkalian n dengan faktorial(n) yang merupakan fungsi itu sendiri sehingga terjadi rekursif.
- kesalahan dalam program tersebut adalah pada baris ke-6 saat melakukan pemanggilan ke fungsi itu sendiri atau rekursi nilai argumen yang di passing adalah n. Hal ini tidak tepat karena akan menyebabkan infinite loop karena program akan memanggil fungsi itu sendiri dengan nominal n terus menerus sebagai argumen sehingga tidak ada jalan untuk program itu selesai. Oleh karena itu hal ini sangat fatal karena dapat menyebabkan program crash dan stack overflow karena kehabisan memori.

## 2. Berikan perbaikan yang benar pada pseudo code.

- Untuk memperbaiki kode diatas kita dapat mengubah argumen n menjadi n-1 karena memang secara logika untuk menghitung faktorial adalah dengan melakukan perkalian dengan angka yang lebih kecil selisih satu dan seterusnya sampai dengan 1. Berikut adalah cuplikan source code yang sudah dibenarkan yang kami sampaikan dalam file faktorial.c

```
1  int factorial(int n)
2  {
3      if (n == 0)
4          return 1;
5      else
6          return n * factorial(n-1);
7  }
8
```

3. Gambarkan flowchart yang menunjukkan fungsi rekursif yang benar untuk menghitung faktorial.

