## Tugas Kelompok ke-3

## Week 9

* LO 1: Apply syntax and functions in the C language
* LO 2: Analyzing a program to solve problems using the C language.
* LO 3: Evaluate the kind of algorithms, syntax, and function for problem-solving

**Relevansi**

|  |  |  |
| --- | --- | --- |
| Week | Topic | Relevansi |
| Week 6 | Function and Recursion | Langsung digunakan dalam desain dan analisis rekursif (Towers of Hanoi & Faktorial) |
| Week 7 | Structures & Union | Diperlukan untuk struktur data array of struct dalam implementasi file pencarian (Soal 3) |
| Week 9 | File Processing and Searching | Langsung mendukung implementasi pencarian data siswa dari file pada Soal 3 |

**Studi Kasus**

Anda dan tim Anda adalah tim pengembang algoritma di **Unit Penelitian Algoritma Komputasi** di Universitas Bina Nusantara. Divisi Anda baru saja mendapatkan proyek dari **Museum Sains dan Matematika Indonesia** untuk membuat simulasi visual algoritma klasik yang akan dipasang di area pameran edukasi interaktif.

Untuk mendukung program literasi algoritma bagi siswa sekolah menengah, Anda dan tim ditugaskan untuk mendesain dua komponen:

1. **Simulasi Towers of Hanoi untuk 4 cakram** (Lihat Gambar 1)
2. **Kalkulator faktorial berbasis rekursif dengan penjelasan flowchart yang tepat**

Simulasi ini akan dijadikan demonstrasi langsung di layar sentuh agar pengunjung memahami konsep rekursi melalui visualisasi langkah demi langkah.

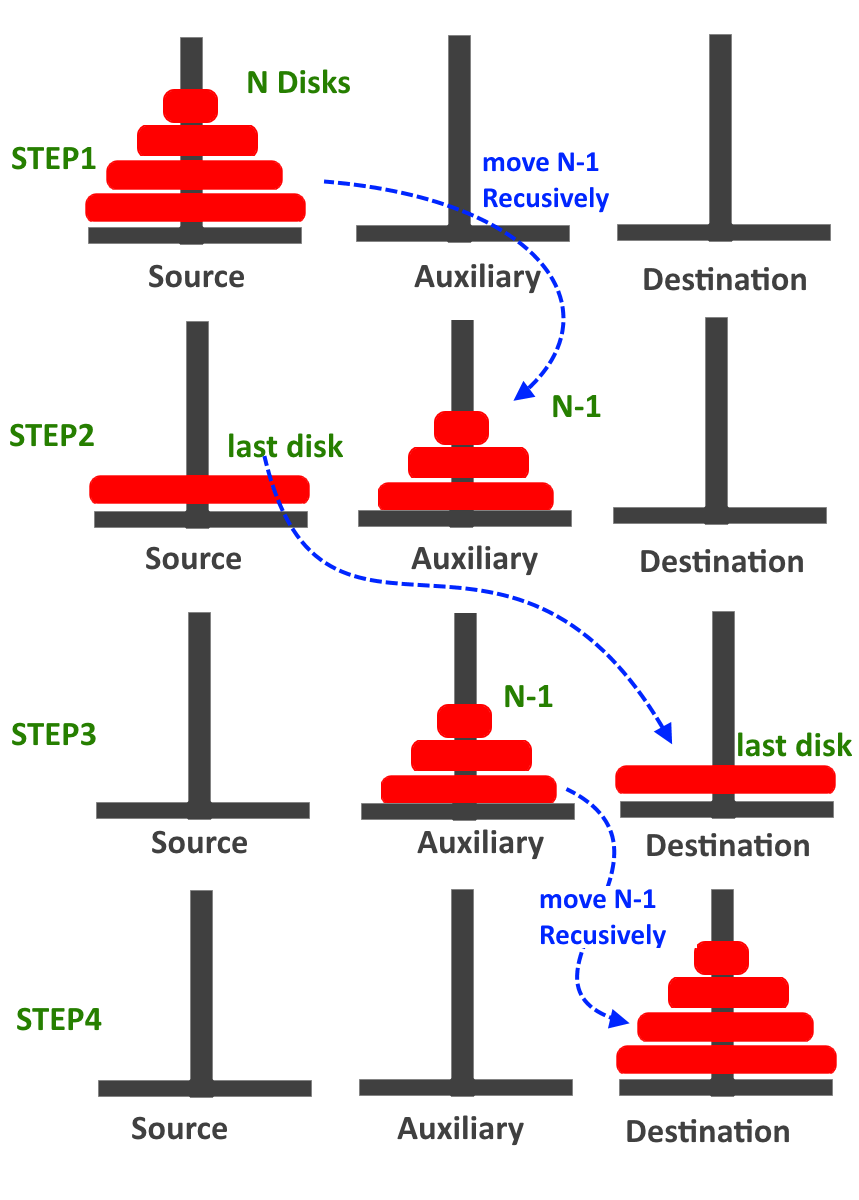
**Soal 1: Desain dan Analisis Fungsi Rekursif - Towers of Hanoi (Bobot 40%)**

**Deskripsi Masalah:**

Tugas Anda adalah merancang fungsi rekursif yang dapat menyelesaikan **Towers of Hanoi** dengan **4 cakram**. Fungsi harus menampilkan langkah-langkah perpindahan dari tiang A ke tiang C, menggunakan tiang B sebagai penyangga sementara.

**Instruksi Pengerjaan:**

1. Buat **pseudo code atau flowchart** untuk fungsi rekursif Towers of Hanoi.
2. Jelaskan secara terperinci setiap langkah rekursi, termasuk kondisi dasar (*base case*) dan cara fungsi memecah masalah menjadi sub-masalah yang lebih kecil.
3. Analisis dan sebutkan **kompleksitas waktu** algoritma berdasarkan jumlah cakram.



Gambar 1. Ilustrasi Algoritma Towers of Hanoi

**Soal 2: Analisis Kesalahan pada Implementasi Fungsi Rekursif Faktorial (Bobot 30%)**

**Deskripsi Masalah:**

Pada modul lain, Anda menemukan **pseudo code** perhitungan faktorial rekursif yang tidak sempurna. Tugas Anda:

* Menganalisis kesalahan logika pada pseudo code tersebut.
* Memberikan versi pseudo code yang benar.
* Menggambarkan flowchart rekursif yang benar.

**Cuplikan Pseudo Code yang Bermasalah:**

|  |
| --- |
| 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 3: Implementasi dan Analisis Pencarian Data Siswa dari File (Bobot 30%)**

**Deskripsi Masalah:** Museum Sains dan Matematika Indonesia juga ingin menambahkan fitur pencarian interaktif dalam simulasi edukatif yang Anda buat. Pengunjung dapat mencari nama siswa dari daftar data yang disimpan dalam file teks.  
File berisi data siswa dengan format:

A close-up of a text

AI-generated content may be incorrect.

Contoh:

A number with numbers on it

AI-generated content may be incorrect.

Tugas Anda:

1. Tulis pseudo code atau flowchart untuk membaca data dari file dan melakukan pencarian berdasarkan nama.
2. Implementasikan algoritma pencarian **linear search** dan tambahkan validasi jika nama tidak ditemukan.
3. Bandingkan secara konseptual waktu pencarian jika file berisi 10 data, 100 data, dan 1000 data. Apa rekomendasi Anda jika datanya semakin besar?

**Petunjuk Teknis:**

* Gunakan pendekatan **input file teks** (misalnya siswa.txt)
* Pencarian berbasis input nama dari pengguna
* Data dibaca ke array of struct (jika diimplementasikan)

**Bobot dan Penilaian:**

* Pseudo code / flowchart: 10%
* Implementasi algoritma linear search: 10%
* Analisis performa dan rekomendasi: 10%