

# **LAPORAN PRAKTIKUM**

## **BAB 7**

Disusun Guna Memenuhi Tugas Mata Kuliah Struktur Data  
Dosen Pengampu: Asep Jamaludin, S.Si., M.Kom.



**Disusun Oleh:**

(Nazril Adrian)

(2410631170097)

**KELAS (2 B)**

**INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS SINGAPERBANGSA KARAWANG**

**2025**

## **SOAL PRAKTIKUM**

- Buatlah program pemecahan masalah Hanoi menggunakan struktur data Stack (boleh stack array atau linked list).
- Program harus memindahkan lempengan dari menara A ke menara C dengan bantuan menara B.
- Gunakan data 10, 20, 30.
- Tidak boleh menggunakan library Stack bawaan C++
- Tambahkan komentar untuk menjelaskan alur program.

## LANGKAH PENYELESAIAN

a. Deklarasi dan inisialisasi

```
#include <iostream>
using namespace std;

// Stack A
int stackA[10];
int topA = -1;

// Stack B
int stackB[10];
int topB = -1;

// Stack C
int stackC[10];
int topC = -1;
```

- Program menggunakan 3 stack array: stackA, stackB, dan stackC.
- Setiap stack mempunyai variabel top masing-masing (topA, topB, topC) untuk menandai posisi data teratas.
- Inisialisasi semua top dengan -1 sebagai tanda bahwa stack kosong.

b. Fungsi Push dan Pop

```
// Push data ke Stack A
void pushA(int data) {
    topA++;
    stackA[topA] = data;
}

// Push data ke Stack B
void pushB(int data) {
    topB++;
    stackB[topB] = data;
}

// Push data ke Stack C
void pushC(int data) {
    topC++;
    stackC[topC] = data;
}
```

```

// Pop data dari Stack A
int popA() {
    int data = stackA[topA];
    topA--;
    return data;
}

// Pop data dari Stack B
int popB() {
    int data = stackB[topB];
    topB--;
    return data;
}

// Pop data dari Stack C
int popC() {
    int data = stackC[topC];
    topC--;
    return data;
}

```

- pushA(), pushB(), pushC() digunakan untuk menambahkan data ke masing-masing stack.
- popA(), popB(), popC() digunakan untuk mengambil data dari masing-masing stack.
- Fungsi push menaikkan top terlebih dahulu sebelum menyimpan data.
- Fungsi pop mengambil data dari top, lalu menurunkan top.

c. Fungsi moveBatu()

```

// Fungsi untuk memindahkan batu
void moveBatu(char asal, char tujuan) {
    int batu;

    // Pop dari stack asal
    if (asal == 'A') batu = popA();
    else if (asal == 'B') batu = popB();
    else if (asal == 'C') batu = popC();

    // Push ke stack tujuan
    if (tujuan == 'A') pushA(batu);
    else if (tujuan == 'B') pushB(batu);
    else if (tujuan == 'C') pushC(batu);

    cout << "\nPindahkan batu " << batu << " dari " << asal << " ke " << tujuan << endl;
}

```

- Fungsi ini digunakan untuk memindahkan satu batu dari stack asal ke stack tujuan.
- Menggunakan karakter ('A', 'B', 'C') untuk menentukan dari stack mana ke stack mana.

- Fungsi akan pop dari stack asal dan push ke stack tujuan.
- Setiap perpindahan, dicetak output perintah perpindahan.

d. Penjelasan Output

- Stack A diisi dengan batu-batu dengan ukuran 30, 20, 10 (besar ke kecil).
- **Output program** akan mencetak **setiap perpindahan batu** dari satu tiang ke tiang lain, sesuai aturan Menara Hanoi.
- Program **menampilkan kondisi** dari Stack A, Stack B, dan Stack C setelah setiap langkah pemindahan.
- **Batu dengan nilai 10** selalu dipindahkan terlebih dahulu, karena batu kecil harus di atas batu besar (aturan Hanoi).
- Program juga mencetak kondisi **akhir stack** setelah semua batu berhasil dipindahkan ke Stack C.

```
int main() {
    // Isi Stack A (besar ke kecil)
    pushA(30);
    pushA(20);
    pushA(10);

    cout << "\nBatu A: " << stackA[0] << " " << stackA[1] << " " << stackA[2] << endl;

    // Langkah-langkah Menara Hanoi
    moveBatu('A', 'C'); // Pindahkan batu 10 dari A ke C
    cout << "Batu A: " << stackA[0] << " " << stackA[1] << endl;
    cout << "Batu B: " << "-" << endl;
    cout << "Batu C: " << stackC[0] << endl;

    moveBatu('A', 'B'); // Pindahkan batu 20 dari A ke B
    cout << "Batu A: " << stackA[0] << endl;
    cout << "Batu B: " << stackB[0] << endl;
    cout << "Batu C: " << stackC[0] << endl;

    moveBatu('C', 'B'); // Pindahkan batu 10 dari C ke B
    cout << "Batu A: " << stackA[0] << endl;
    cout << "Batu B: " << stackB[0] << " " << stackB[1] << endl;
    cout << "Batu C: " << "-" << endl;

    moveBatu('A', 'C'); // Pindahkan batu 30 dari A ke C
    cout << "Batu A: " << "-" << endl;
    cout << "Batu B: " << stackB[0] << " " << stackB[1] << endl;
    cout << "Batu C: " << stackC[0] << " " << stackC[1] << endl;

    cout << "\nHasil akhir setelah perpindahan dilakukan: " << endl;
    cout << "Batu A: " << "-" << endl;
    cout << "Batu B: " << "-" << endl;
    cout << "Batu C: " << stackC[0] << " " << stackC[1] << " " << stackC[2] << endl;

    return 0;
}
```

## **Github**

<https://github.com/Nazril17/STRUKTUR-DATA.git>