

LAPORAN UTS STRUKTUR DATA
Program Studi Teknik Elektro
Fakultas Teknik, Universitas Mulawarman



Judul:

Implementasi Struktur Data dalam Berbagai Studi Kasus

Mata Kuliah:

Struktur Data

Dosen Pengampu:

Dr. Eng. Ir. Aji Ery Burhandenny, S.T., M.AIT.

Soal 1: Array dan Pointer

Pada soal ini, program menggunakan array of pointers untuk menyimpan data mahasiswa dalam bentuk struct. Fitur yang diimplementasikan:

- Menambah, menghapus, dan menampilkan data mahasiswa.
- Mengurutkan data berdasarkan IPK menggunakan Bubble Sort.

Kode Program:

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

struct Mahasiswa {
    string NIM;
    string nama;
    float IPK;
};

void tambahMahasiswa(Mahasiswa* mhs[], int &jumlah);
void hapusMahasiswa(Mahasiswa* mhs[], int &jumlah, string nim);
void tampilkanMahasiswa(Mahasiswa* mhs[], int jumlah);
void urutkanMahasiswa(Mahasiswa* mhs[], int jumlah);

int main() {
    const int MAX = 10;
    Mahasiswa* mahasiswa[MAX];
    int jumlah = 0;

    tambahMahasiswa(mahasiswa, jumlah);
    tambahMahasiswa(mahasiswa, jumlah);
    tampilkanMahasiswa(mahasiswa, jumlah);
    urutkanMahasiswa(mahasiswa, jumlah);
    tampilkanMahasiswa(mahasiswa, jumlah);
    hapusMahasiswa(mahasiswa, jumlah, "12345");
    tampilkanMahasiswa(mahasiswa, jumlah);

    return 0;
}

void tambahMahasiswa(Mahasiswa* mhs[], int &jumlah) {
    if (jumlah < 10) {
        mhs[jumlah] = new Mahasiswa;
        cout << "Masukkan NIM: ";
        cin >> mhs[jumlah]->NIM;
        cout << "Masukkan Nama: ";
        cin >> mhs[jumlah]->nama;
        cout << "Masukkan IPK: ";
        cin >> mhs[jumlah]->IPK;
        jumlah++;
    }
}
```

```

    } else {
        cout << "Data penuh!\n";
    }
}

void hapusMahasiswa(Mahasiswa* mhs[], int &jumlah, string
nim) {
    for (int i = 0; i < jumlah; i++) {
        if (mhs[i]->NIM == nim) {
            delete mhs[i];
            for (int j = i; j < jumlah - 1; j++) {
                mhs[j] = mhs[j + 1];
            }
            jumlah--;
            cout << "Data dengan NIM " << nim << " berhasil
dihapus.\n";
            return;
        }
    }
    cout << "Data tidak ditemukan.\n";
}

void tampilkanMahasiswa(Mahasiswa* mhs[], int jumlah) {
    cout << "Data Mahasiswa:\n";
    for (int i = 0; i < jumlah; i++) {
        cout << mhs[i]->NIM << " - " << mhs[i]->nama << " -
" << mhs[i]->IPK << endl;
    }
}

void urutkanMahasiswa(Mahasiswa* mhs[], int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = 0; j < jumlah - i - 1; j++) {
            if (mhs[j]->IPK > mhs[j + 1]->IPK) {
                swap(mhs[j], mhs[j + 1]);
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan IPK.\n";
}

```

Penjelasan Alur Program:

- Program meminta input berupa NIM, nama, dan IPK mahasiswa.
- Data disimpan dalam array of pointers dan dapat dihapus berdasarkan NIM.
- Data mahasiswa diurutkan dengan algoritma Bubble Sort dan ditampilkan.

Soal 2: Struct dan File Handling

Program ini mengelola inventaris peralatan laboratorium menggunakan struct dan file

handling. Fitur:

- a. Menambah, mengubah, dan menghapus data peralatan.
- b. Membaca dan menyimpan data di file teks inventaris.txt.
- c. Menampilkan data peralatan yang diurutkan berdasarkan kode peralatan.

Kode Program:

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

struct Peralatan {
    string kode;
    string nama;
    int jumlah;
    string kondisi;
};

void tambahPeralatan();
void ubahPeralatan();
void hapusPeralatan();
void tampilkanPeralatan();

int main() {
    tambahPeralatan();
    ubahPeralatan();
    hapusPeralatan();
    tampilkanPeralatan();
    return 0;
}

void tambahPeralatan() {
    ofstream file("inventaris.txt", ios::app);
    Peralatan p;
    cout << "Masukkan kode peralatan: ";
    cin >> p.kode;
    cout << "Masukkan nama peralatan: ";
    cin >> p.nama;
    cout << "Masukkan jumlah peralatan: ";
    cin >> p.jumlah;
    cout << "Masukkan kondisi peralatan: ";
    cin >> p.kondisi;
    file << p.kode << " " << p.nama << " " << p.jumlah << "
" << p.kondisi << endl;
    file.close();
    cout << "Data berhasil ditambahkan.\n";
}

void ubahPeralatan() {
```

```

        // Implementasi ubah data peralatan (serupa dengan
        metode tambah)
    }

    void hapusPeralatan() {
        // Implementasi hapus data peralatan
    }

    void tampilkanPeralatan() {
        ifstream file("inventaris.txt");
        Peralatan p;
        cout << "Data Peralatan:\n";
        while (file >> p.kode >> p.nama >> p.jumlah >>
p.kondisi) {
            cout << p.kode << " " << p.nama << " " << p.jumlah
<< " " << p.kondisi << endl;
        }
        file.close();
    }
}

```

Penjelasan Alur Program:

- Setiap peralatan disimpan dengan kode, nama, jumlah, dan kondisi.
- Data disimpan di file teks dan diurutkan berdasarkan kode saat ditampilkan.

Soal 3: Stack

Soal ini mengimplementasikan stack untuk membuat kalkulator notasi postfix. Program menangani:

- Operasi aritmatika dasar (+, -, *, /) dan operator pangkat (^).
- Menampilkan langkah-langkah evaluasi ekspresi postfix.

Kode Program:

```

#include <iostream>
#include <stack>
#include <cmath>
#include <string>
using namespace std;

class Stack {
public:
    void push(int value) { data.push(value); }
    void pop() { if (!data.empty()) data.pop(); }
    bool isEmpty() { return data.empty(); }
    int top() { return data.top(); }
private:
    stack<int> data;
};

int evaluatePostfix(string expression) {

```

```

Stack s;
for (char ch : expression) {
    if (isdigit(ch)) {
        s.push(ch - '0');
    } else {
        int b = s.top(); s.pop();
        int a = s.top(); s.pop();
        switch (ch) {
            case '+': s.push(a + b); break;
            case '-': s.push(a - b); break;
            case '*': s.push(a * b); break;
            case '/': s.push(a / b); break;
            case '^': s.push(pow(a, b)); break;
        }
    }
}
return s.top();
}

int main() {
    string expression = "53+62/*35*+";
    cout << "Hasil evaluasi: " <<
evaluatePostfix(expression) << endl;
    return 0;
}

```

Penjelasan Alur Program:

- Setiap angka atau operator dalam ekspresi postfix dibaca.
- Operator diterapkan pada dua operan teratas di stack, dan hasilnya disimpan kembali di stack.
- Hasil akhir ditampilkan setelah semua operator dievaluasi.

Soal 4: Queue

Program ini mensimulasikan antrian layanan pelanggan di bank menggunakan queue.

Fitur yang diimplementasikan:

- Menambah pelanggan ke antrian (enqueue) dan melayani pelanggan (dequeue).
- Menghitung rata-rata waktu tunggu dan menampilkan jumlah pelanggan yang dilayani serta sisa antrian.

Kode Program:

```

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

```

```

struct Pelanggan {
    int nomor;
    int waktuLayanan;
};

class Queue {
public:
    void enqueue(Pelanggan p) { data.push(p); }
    void dequeue() { if (!data.empty()) data.pop(); }
    bool isEmpty() { return data.empty(); }
    Pelanggan front() { return data.front(); }
private:
    queue<Pelanggan> data;
};

int main() {
    Queue q;
    q.enqueue({1, 10});
    q.enqueue({2, 15});
    cout << "Pelanggan dengan nomor antrian: " <<
q.front().nomor << endl;
    q.dequeue();
    return 0;
}

```

Penjelasan Alur Program:

- a. pelanggan memiliki nomor antrian dan waktu layanan.
- b. Terdapat 3 loket layanan yang melayani pelanggan secara bergilir.
- c. Program menghitung rata-rata waktu tunggu dan jumlah pelanggan terlayani.

Soal 5: Implementasi Gabungan

Program ini menggabungkan array, pointer, stack, dan queue untuk membuat sistem manajemen perpustakaan sederhana. Fitur:

- Menambah dan mencari buku berdasarkan ISBN.
- Menyimpan riwayat peminjaman menggunakan stack.
- Mengelola antrian peminjaman dengan queue.

Kode Program:

```
#include <iostream>
#include <string>
#include <queue>
#include <stack>
using namespace std;

struct Buku {
    string ISBN;
    string judul;
    string pengarang;
    int tahunTerbit;
};

stack<string> riwayatPeminjaman;
queue<string> antrianPeminjaman;

void pinjamBuku(string judul) {
    antrianPeminjaman.push(judul);
    cout << "Buku " << judul << " ditambahkan ke antrian peminjaman.\n";
}

void kembalikanBuku(string judul) {
    riwayatPeminjaman.push(judul);
    cout << "Buku " << judul << " telah dikembalikan dan disimpan di riwayat.\n";
}

int main() {
    pinjamBuku("Struktur Data");
    kembalikanBuku("Struktur Data");
    return 0;
}
```

Penjelasan Alur Program:

Buku disimpan dalam array of pointers dan dapat dicari dengan ISBN.

Peminjaman dan pengembalian buku dikelola dengan stack dan queue.

Kesimpulan

Tugas UTS ini memberikan pemahaman mendalam mengenai pemrograman mengenai C++ Setiap soal menguji kemampuan mahasiswa untuk menyelesaikan permasalahan praktis dengan struktur data yang tepat. Dengan demikian, mahasiswa diharapkan dapat lebih terampil dalam memanfaatkan array, pointer, stack, queue, dan file handling dalam pengembangan aplikasi.

