LAPORAN PRAKTIKUM

PERTEMUAN 7

Pengenalan C++ : Queue

Nama :

Ahmad Uffi Lestari Ma’ruf (2311104015)

Dosen :

YUDHA ISLAMI SULISTYA, S.Kom., M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. Unguided

Pada program in membuat sebuah Queue yang dirubah nenjadi Linked List pertama nantinya inputan diberikan oleh User berupa **nama** dan **NIM,** lalu bisa mengurutkan dari terkecil ke terbesar.

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

// Struktur untuk node dalam linked list
struct Mahasiswa {
    string nama;
    string nim;
    Mahasiswa* next;
};

class Queue {
private:
    Mahasiswa *front, *back;
    int count;

public:
    Queue() {
        front = back = NULL;
        count = 0;
    }

    bool isEmpty() {
        return front == NULL;
    }

    void enqueue(string nama, string nim) {
        Mahasiswa* newMhs = new Mahasiswa;
        newMhs->nama = nama;
        newMhs->nim = nim;
        newMhs->next = NULL;

        if(isEmpty()) {
            front = back = newMhs;
        } else {
            // Mencari posisi yang tepat berdasarkan NIM
            if(nim < front->nim) {
                newMhs->next = front;
                front = newMhs;
            } else {
                Mahasiswa* current = front;
                Mahasiswa* prev = NULL;

                while(current != NULL && current->nim < nim) {
                    prev = current;
                    current = current->next;
                }

                if(current == NULL) {
                    back->next = newMhs;
                    back = newMhs;
                } else {
                    prev->next = newMhs;
                    newMhs->next = current;
                }
            }
        }
        count++;
        cout << "Mahasiswa berhasil ditambahkan ke antrian" <<
endl}
```

```
 void dequeue() {
        if(isEmpty()) {
            cout << "Antrian kosong" << endl;
            return;
        }

        Mahasiswa* temp = front;
        cout << "Mahasiswa yang dikeluarkan: " << endl;
        cout << "Nama: " << temp->nama << endl;
        cout << "NIM: " << temp->nim << endl;

        front = front->next;
        delete temp;

        if(front == NULL) {
            back = NULL;
        }

        count--;
    }

    int countQueue() {
        return count;
    }

    void clearQueue() {
        while(!isEmpty()) {
            dequeue();
        }
        cout << "Antrian telah dikosongkan" << endl;
    }

    void viewQueue() {
        if(isEmpty()) {
            cout << "Antrian kosong" << endl;
            return;
        }

        cout << "\nDaftar Antrian Mahasiswa:" << endl;
        cout << "=========================" << endl;
        Mahasiswa* current = front;
        int nomor = 1;

        while(current != NULL) {
            cout << nomor << ". Nama: " << current->nama <<
endl;       cout << "   NIM: " << current->nim << endl;
            cout << "-------------------------" << endl;
            current = current->next;
            nomor++;
        }
    }
};
```

```cpp
int main() {
    Queue antrian;
    int pilihan;
    string nama, nim;

    do {
        cout << "\nMenu Antrian Mahasiswa" << endl;
        cout << "1. Tambah Mahasiswa" << endl;
        cout << "2. Keluarkan Mahasiswa" << endl;
        cout << "3. Lihat Antrian" << endl;
        cout << "4. Jumlah Antrian" << endl;
        cout << "5. Kosongkan Antrian" << endl;
        cout << "0. Keluar" << endl;
        cout << "Pilihan: ";
        cin >> pilihan;
        cin.ignore();

        switch(pilihan) {
            case 1:
                cout << "Masukkan Nama: ";
                getline(cin, nama);
                cout << "Masukkan NIM: ";
                getline(cin, nim);
                antrian.enqueue(nama, nim);
                break;

            case 2:
                antrian.dequeue();
                break;

            case 3:
                antrian.viewQueue();
                break;

            case 4:
                cout << "Jumlah antrian: " << antrian.countQueue() <<
endl;           break;

            case 5:
                antrian.clearQueue();
                break;

            case 0:
                cout << "Program selesai" << endl;
                break;

            default:
                cout << "Pilihan tidak valid" << endl;
        }
    } while(pilihan != 0);

    return 0;
}
```

Output yang dihasilkan :