

PRAKTIKUM STRUKTUR DATA

UNGUIDED 8



Nama :

Aulia Ahmad Ghaus Adzam (2311104028)

Dosen :

Yudha Islami Sulistya, S.
Kom.,M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

B. UNGUIDED

1. Ubahlah penerapan konsep queue pada bagian guided dari array menjadi linked list
2. Dari nomor 1 buatlah konsep antri dengan atribut Nama mahasiswa dan NIM Mahasiswa
3. Modifikasi program pada soal 1 sehingga mahasiswa dapat diprioritaskan berdasarkan NIM (NIM yang lebih kecil didahulukan pada saat output).

Noted : Untuk data mahasiswa dan nim dimasukan oleh user

JAWABAN:

- **Penjelasan Jawaban No 1 – 3 dan kode nya:**

karena soal no 1 sampai 3 itu satu kesatuan alias menyatu jadi saya akan jelaskan menjadi satu saja pertama program dibuat dalam versi linked list dari yang tadinya array, pertama kita buat/deklarasikan struct data sih Mahasiswa nya dan kita buat pointer front sama rear nya yang value default nya itu nullptr:

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

// Data Mahasiswa
struct Mahasiswa {
    string nama;
    int nim;
    Mahasiswa* next;
};

Mahasiswa* front = nullptr;
Mahasiswa* rear = nullptr;
```

lalu kita buat function Boolean isEmpty untuk mengecek apakah data dalam queue ini kosong atau enggak:

```
bool isEmpty() {
    return front == nullptr;
}
```

Lalu kita buat void enqueueMahasiswa yang berupa berisikan parameter Nama

Mahasiswa dan NIM Mahasiswa, void ini akan digunakan untuk mengepush Data Mahasiswa ke dalam queue:

```
// BUAT PUSH KE QUEUE
void enqueueMahasiswa(string nama, int nim) {
    Mahasiswa* newMahasiswa = new Mahasiswa();
    newMahasiswa->nama = nama;
    newMahasiswa->nim = nim;
    newMahasiswa->next = nullptr;

    if (isEmpty()) {
        front = rear = newMahasiswa;
    } else {
        rear->next = newMahasiswa;
        rear = newMahasiswa;
    }
    cout << "Mahasiswa " << nama << " dengan NIM " << nim << " telah ditambahkan ke dalam queue." << endl;
}
```

Lalu setelah itu kita buat void dequeueMahasiswa untuk mengeluarkan data Mahasiswa:

```
// HAPUS DARI QUEUE
void dequeueMahasiswa() {
    if (isEmpty()) {
        cout << "Antrian kosong, tidak ada yang bisa dihapus." << endl;
    } else {
        Mahasiswa* temp = front;
        cout << "Mahasiswa " << front->nama << " dengan NIM " << front->nim << " telah dikeluarkan dari queue." << endl;
        front = front->next;
        delete temp;
    }

    if (front == nullptr) {
        rear = nullptr;
    }
}
```

Lalu setelah itu kita buat void viewQueue agar informasi nya ter display melalui Format yang ada di dalam void ini:

```
// NGDISPLAY
void viewQueue() {
    if (isEmpty()) {
        cout << "Antrian kosong." << endl;
    } else {
        Mahasiswa* temp = front;
        cout << "Data antrian mahasiswa:" << endl;
        int count = 1;
        while (temp != nullptr) {
            cout << count++ << ". Nama: " << temp->nama << ", NIM: " << temp->nim << endl;
            temp = temp->next;
        }
    }
}
```

Setelah itu kita buat kan main dan kita menggunakan switch case karena sesuai Ketentuan soal nya user input secara manual: jika user input 1 untuk menambahkan Ke dalam queue kalo 2 untuk hapus kalo 3 untuk view dan kalo 4 untuk keluar

```
int main() {
    int pilihan;
    string nama;
    int nim;

    do {
        cout << "\nMenu Queue Mahasiswa:" << endl;
        cout << "1. Tambah Mahasiswa ke Queue" << endl;
        cout << "2. Hapus Mahasiswa dari Queue" << endl;
        cout << "3. Lihat Isi Queue" << endl;
        cout << "4. Keluar" << endl;
        cout << "Pilih opsi: ";
        cin >> pilihan;

        if(cin.fail()) {
            cin.clear();
            cin.ignore(1000, '\n');
            cout << "Input tidak valid. Silakan masukkan angka antara 1-4." << endl;
            continue;
        }

        cin.ignore();

        switch (pilihan) {
            case 1:
                cout << "Masukkan Nama Mahasiswa: ";
                getline(cin, nama);
                cout << "Masukkan NIM Mahasiswa: ";
                cin >> nim;

                if(cin.fail()) {
                    cin.clear();
                    cin.ignore(1000, '\n'); // Biar Gak Ngebug LOOPING TERUS
                    cout << "NIM tidak valid. Silakan masukkan angka." << endl;
                    continue;
                }

                enqueueMahasiswa(nama, nim);
                break;
            case 2:
                dequeueMahasiswa();
                break;
            case 3:
                viewQueue();
                break;
            case 4:
                cout << "Keluar dari program." << endl;
                break;
            default:
                cout << "Pilihan tidak valid. Silakan coba lagi." << endl;
        }
    } while (pilihan != 4);
    return 0;
}
```

Output:

Berikut contoh dari output jika user memilih 1

```
Menu Queue Mahasiswa:
1. Tambah Mahasiswa ke Queue
2. Hapus Mahasiswa dari Queue
3. Lihat Isi Queue
4. Keluar
Pilih opsi: 1
Masukkan Nama Mahasiswa: Aulia
Masukkan NIM Mahasiswa: 28
Mahasiswa Aulia dengan NIM 28 telah ditambahkan ke dalam queue.

Menu Queue Mahasiswa:
1. Tambah Mahasiswa ke Queue
2. Hapus Mahasiswa dari Queue
3. Lihat Isi Queue
4. Keluar
Pilih opsi: 1
Masukkan Nama Mahasiswa: Fulan
Masukkan NIM Mahasiswa: 41
Mahasiswa Fulan dengan NIM 41 telah ditambahkan ke dalam queue.
```

Berikut contoh dari user memilih 3 yaitu melihat data di dalam queue

```
Menu Queue Mahasiswa:
1. Tambah Mahasiswa ke Queue
2. Hapus Mahasiswa dari Queue
3. Lihat Isi Queue
4. Keluar
Pilih opsi: 3
Data antrian mahasiswa:
1. Nama: Aulia, NIM: 28
2. Nama: Fulan, NIM: 41
```

Berikut contoh jika user memilih no 2 yaitu hapus data dari queue karena system Queue itu FIFO First in First out maka data yang pertama dimasukan terlebih Dahulu yang akan dikeluarkan lalu data setelah nya layaknya Ngantri aja

```
Menu Queue Mahasiswa:
1. Tambah Mahasiswa ke Queue
2. Hapus Mahasiswa dari Queue
3. Lihat Isi Queue
4. Keluar
Pilih opsi: 2
Mahasiswa Aulia dengan NIM 28 telah dikeluarkan dari queue.

Menu Queue Mahasiswa:
1. Tambah Mahasiswa ke Queue
2. Hapus Mahasiswa dari Queue
3. Lihat Isi Queue
4. Keluar
Pilih opsi: 2
Mahasiswa Fulan dengan NIM 41 telah dikeluarkan dari queue.
1. Nama: Aulia, NIM: 28
2. Nama: Fulan, NIM: 41
```

