

Tugas Proceeding
Modul 8 Struktur Data

QUEUE



Disusun Oleh:

Christian Felix Saliman Sugiono (2311104031)

S1SE0701

Dosen:

Yudha Islami Sulistya

Program Studi S1 Software Engineering

Fakultas Informatika

Telkom University

Purwokerto

2024

Tugas Pendahuluan Modul 8
STRUKTUR DATA - Ganjil 2024/2025
"QUEUE"

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 4 adalah Senin, 9 Oktober 2023 pukul 06.00 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan:
TP_MOD_[XX]_NIM_NAMA.pdf

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

Unguided Modul 8

Implementasi Queue pada C++

array to linked list, antrian mahasiswa dengan nama dan NIM, prioritas nim angka kecil

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

struct Mahasiswa {
    string nama;
    string NIM;
    Mahasiswa* next;
};

Mahasiswa* front = nullptr;
Mahasiswa* back = nullptr;

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

void enqueueAntrian(string nama, string NIM) {
    Mahasiswa* newNode = new Mahasiswa();
    newNode->nama = nama;
    newNode->NIM = NIM;
    newNode->next = nullptr;

    if (isEmpty()) {
        front = back = newNode;
    } else if (newNode->NIM < front->NIM) {
        newNode->next = front;
        front = newNode;
    } else {
        Mahasiswa* temp = front;
        while (temp->next != nullptr && temp->next->NIM < newNode->NIM) {
            temp = temp->next;
        }
        newNode->next = temp->next;
        temp->next = newNode;
        if (newNode->next == nullptr) {
            back = newNode;
        }
    }
}

void dequeueAntrian() {
    if (isEmpty()) {
        cout << "Antrian kosong brow" << endl;
    } else {
        Mahasiswa* temp = front;
        front = front->next;
        delete temp;
        if (front == nullptr) {
            back = nullptr;
        }
    }
}

int countQueue() {
    int count = 0;
    Mahasiswa* temp = front;
    while (temp != nullptr) {
        count++;
        temp = temp->next;
    }
    return count;
}

void clearQueue() {
    while (!isEmpty()) {
        dequeueAntrian();
    }
    cout << "Antrian dikosongkan." << endl;
}
```

Dari guided yang telah kita buat dalam praktikum untuk mengubah array menjadi linked list kita akan membutuhkan struct mahasiswa yang dimana struct berfungsi sebagai kanvas yang kosong untuk menempatkan nama dan NIM mahasiswa, setelah pembuatan struct kita perlu membuat deklarasi bahwa bagian depan (front) dan belakang (back) menjadi nullptr sebagai deklarasi bahwa linked list kosong dan siap untuk di isi, di dalam method method lainnya kita juga akan merubah kodingan agar selaras dengan linked list dan bukan array lagi seperti pada enqueue antrian kita mengubah array mahasiswa menjadi node Mahasiswa dengan nim nama dan deklarasi next.

lalu untuk menginsertkan berdasarkan urutan kita akan membuat perulangan didalam method enqueue ini dengan perulangan yang mengecek besar nim mahasiswa dan mengurutkan yang kecil dengan posisi yang lebih awal, dan untuk membuat method tersebut kita akan mengecek apakah depan belakang dan node baru nilainya sama, jika tidak maka node baru akan dimasukan didepan, jika bukan juga maka nim akan dimasukan dalam kondisi else dimana ia akan dicek dengan temp node dan dimasukan sesuai dengan urutan sesuai dengan besarnya, kemudian kita akan membuat method dequeue antrian yang dimana fungsi method ini adalah menghapus satu elemen dari queue dengan aturan first in first out, atau yang diatas (front) akan keluar terlebih dahulu maka kita akan mengeluarkan elemen paling depan dari queue di fungsi ini, untuk menghitung panjang queue kita akan inisialisasi hitungan dari 0, dan melakukan perulangan while hingga list bersifat nullptr yang dimana dia akan mengembalikan panjang queue, untuk clear queue kita akan menghapus semua elemen dari queue ddengan perulangan while yang memanfaatkan method dequeue. Lalu kita akan masuk

kedalam method berikutnya yakni view queue.

```

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

int main() {
    int choice;
    string nama, NIM;

    do {
        cout << "\n1. Tambah Antrian\n2. Hapus Antrian\n3. Lihat Antrian\n4. Bersihkan Antrian\n5. Keluar\n";
        cout << "Pilih opsi: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Masukkan Nama Mahasiswa: ";
                cin.ignore();
                getline(cin, nama);
                cout << "Masukkan NIM Mahasiswa: ";
                cin >> NIM;
                enqueueAntrian(nama, NIM);
                break;
            case 2:
                dequeueAntrian();
                break;
            case 3:
                viewQueue();
                cout << "Jumlah antrian = " << countQueue() << endl;
                break;
            case 4:
                clearQueue();
                break;
            case 5:
                cout << "Keluar dari program." << endl;
                break;
            default:
                cout << "Opsi tidak valid. Silakan coba lagi." << endl;
                break;
        }
    } while (choice != 5);

    return 0;
}

```

pada method view queue ini kita akan membuat permissalan, apabila queue kosong maka kita akan mengeprint data kosong, apabila tidak maka kita akan membuat perulangan while untuk setiap linked list untuk tampil data queue, lalu kita akan masuk kedalam kelas main, didalam kels main kita akan inisialisasi pilihan dengan int, lalu string nama dan NIM, kemudian kita akan menggunakan do untuk membuat menu pilihan, didalam sini kita akan membuat 5 menu, tambah antrian, hapus antrian, lihat data, bersihkan antrian, keluar program, setelah itu kita akan masuk kedalam switch case yang dimana kita menyesuaikan switch case sesuai dengan pilihan opsinya, kita akan memasukan method-method sesuai dengan apa yang ada pada opsi, setelah itu semua kita akan menjalankan program dengan output seperti berikut ini:

```

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 1
Masukkan Nama Mahasiswa: Maulana
Masukkan NIM Mahasiswa: 2311104034

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 1
Masukkan Nama Mahasiswa: DM Dokuro
Masukkan NIM Mahasiswa: 2311104010

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 1
Masukkan Nama Mahasiswa: Skellytron
Masukkan NIM Mahasiswa: 2311104001

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 3
Data antrian teller:
1. Nama: Skellytron, NIM: 2311104001
2. Nama: DM Dokuro, NIM: 2311104010
3. Nama: Maulana, NIM: 2311104034
Jumlah antrian = 3

```

```

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 2

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 3
Data antrian teller:
1. Nama: DM Dokuro, NIM: 2311104010
2. Nama: Maulana, NIM: 2311104034
Jumlah antrian = 2

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 3
Data antrian teller:
1. Nama: DM Dokuro, NIM: 2311104010
2. Nama: Maulana, NIM: 2311104034
Jumlah antrian = 2

```

```

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 4
Antrean dikosongkan.

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 3
Data antrian teller: kosong
Jumlah antrian = 0

1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilih opsi: 5
Keluar dari program.

```