LAPORAN PRAKTIKUM

PERTEMUAN 8

STACK

Nama :

Andika Rifki Pratama (2311104011)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

B. Soal Unguided

1. **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

Unguided > no3.cpp > Node > next

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

struct Node {
    string data;
    int NIM;
    Node* next;
};

Node* front = nullptr;

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

void enqueueAntrian(string data, int NIM) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->NIM = NIM;
    newNode->next = nullptr;

    if (isEmpty()) {
        front = newNode;
    } else {
        Node* temp = front;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}
void dequeueAntrian() {
    if (isEmpty()) {
        cout << "Antrian kosong!" << endl;
    } else {
        Node* current = front;
        front = front->next;
        delete current;
    }
}

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

void clearQueue() {
    while (!isEmpty()) {
        dequeueAntrian();
    }
}

void viewQueue() {
    cout << "Data antrian:" << endl;
    Node* temp = front;
    int position = 1;
    while (temp != nullptr) {
        cout << position << ". " << temp->data << "NIM :"<< temp->NIM << endl;
        temp = temp->next;
        position++;
```

```
    }
    if (position == 1) {
        cout << "Antrian kosong!" << endl;
    }
}


void sortQueue(){
    if (isEmpty()){
        cout << "Antrian kosong!" << endl;
        return;
    }
        Node* current = front;
        while (current != nullptr) {
            Node* lowest = current;
            Node* nextNode = current-> next;
            while (nextNode != nullptr) {
                if (nextNode->NIM < lowest->NIM){
                    lowest = nextNode;
                }
                nextNode = nextNode-> next;
            }

    if (lowest != current){
        swap(lowest->NIM, current-> NIM);
        swap(lowest->data, current-> data);

    } current = current->next;
    }
}

int main() {
    int total;
    cout << "Masukan jumlah mahasiswa : ";
    cin >> total;

    for (int i = 0; i < total; i++){
        string nama;
        int NIM;
        cout << "Masukan nama mahasiswa : " << endl;
        cin >> nama;
        cout << "Masukan NIM mahasiswa : " << endl;
        cin >> NIM;
        enqueueAntrian(nama, NIM);
    }
    cout <<"\nList Data setelah di sorting" << endl;
    viewQueue();
}
```

Penjelasan :

Pada bagian awal program terdapat struct node untuk pendeklarasian variable NIM dan data, serta terdapat struct Node* next;

Kemudian terdapat Node* front = nullptr, ini befungsi untuk memastikan value awal dari list adalah kosong atau nill. Kemudian tedapat method isEmpty untuk memeriksa apakah list kosong atau tidak.

Pada line ke 16 sampai dengan 31 terdapat fungsi untuk memasukan nilai value dan node baru pada list, pada method ini akan memeriksa apakah list kosong kemudian akan menjalankan program dengan membangun Node temo sebagai list di depan, kemudian memeriksa Ketika temporary next adalah kosong maka akan membuat node baru setelah list yang terpakai.

Pada dequeue akan memeriksa apakah list kosong, kemudian akan menunjuk pointer kepada list sebelumnya dan akan dihapuskan.

Selanjutnya terdapat fungsi count queue untuk menghitung jumlah total node yang ada.

Selanjutnya terdapat clearqueue yang memiliki fungsi yang sama dengan dequeue.

Kemudian terdapat viewQueue, ini akan melakukan perulangan untuk memberikan output berupa data dan NIM.

Selanjutnya adalah sort, pada method ini akan membandingkan value NIM dari setiap node yang kemudian akan mengubah posisiinya supaya list terurut.

```
992
Masukan nama mahasiswa :
DIka
Masukan NIM mahasiswa :
882
Masukan nama mahasiswa :
Oki
Masukan NIM mahasiswa :
124
Masukan nama mahasiswa :
Ming
Masukan NIM mahasiswa :
612

Data before sorted
Data antrian:
1. Dika
2. Oki
3. Ming
4. DIka
5. Andi

Data after sorted
Data antrian:
1. Dika
2. Oki
3. Ming
4. DIka
5. Andi
PS D:\Campuss\SMT 3\Praktikum Struktur Data\08_Queue\Unguided>
```