

Tugas Pendahuluan Modul 8

STRUKTUR DATA - Ganjil 2024/2025

Queue

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 2 adalah Senin, 30 September 2024 pukul 07.30 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. Codingan diupload di Github dan upload Laporan di Lab menggunakan format **PDF** dengan ketentuan: **TP_MOD_[XX]_NIM_NAMA.pdf**

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

LAPORAN PRAKTIKUM
PERTEMUAN 8
QUEUE



Nama :

Tsaqif Hisyam Saputra (2311104024)

Dosen :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK FAKULTAS
INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

A. Tujuan

- Mahasiswa mampu menjelaskan definisi dan konsep dari queue
- Mahasiswa mampu menerapkan operasi tambah, menghapus pada queue
- Mahasiswa mampu menerapkan operasi tampil data pada queue

B. Tools

Visual Studio Code dengan C++ Extensions Pack & Codeblocks

C. Latihan – Unguided

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Node {
6     string nama;
7     int nim;
8     Node* next;
9 };
10
11 class Queue {
12 private:
13     Node* front;
14     Node* back;
15
16 public:
17     Queue() : front(nullptr), back(nullptr) {}
18
19     bool isEmpty() {
20         return front == nullptr;
21     }
22
23     void enqueue(string nama, int nim) {
24         Node* newNode = new Node(nama, nim, nullptr);
25
26         if (isEmpty() || nim < front->nim) {
27             newNode->next = front;
28             front = newNode;
29             if (back == nullptr) {
30                 back = newNode;
31             }
32         } else {
33             Node* current = front;
34             while (current->next != nullptr && current->next->nim <= nim) {
35                 current = current->next;
36             }
37             newNode->next = current->next;
38             current->next = newNode;
39             if (newNode->next == nullptr) {
40                 back = newNode;
41             }
42         }
43     }
44
45     void dequeue() {
46         if (isEmpty()) {
47             cout << "Antrian kosong" << endl;
48         } else {
49             Node* temp = front;
50             front = front->next;
51             delete temp;
52             if (front == nullptr) {
53                 back = nullptr;
54             }
55         }
56     }
57
58     void viewQueue() {
59         if (isEmpty()) {
60             cout << "Antrian kosong" << endl;
61         } else {
62             Node* current = front;
63             cout << "Data antrian teller:" << endl;
64             int position = 1;
65             while (current != nullptr) {
66                 cout << position++ << ". " << current->nama << " (NIM: " << current->nim << ") " << endl;
67                 current = current->next;
68             }
69         }
70     }
71
72     void clearQueue() {
73         while (!isEmpty()) {
74             dequeue();
75         }
76         cout << "Antrian telah dikosongkan." << endl;
77     }
78
79     int countQueue() {
80         int count = 0;
81         Node* current = front;
82         while (current != nullptr) {
83             count++;
84             current = current->next;
85         }
86         return count;
87     }
88
89     ~Queue() {
90         clearQueue();
91     }
92 };
93
94 int main() {
95     Queue queue;
96     int choice;
97
98     do {
99         cout << "Menu Antrian:\n";
100         cout << "1. Tambah Mahasiswa ke Antrian\n";
101         cout << "2. Hapus Mahasiswa dari Antrian\n";
102         cout << "3. Lihat Antrian\n";
103         cout << "4. Berikan Antrian\n";
104         cout << "5. Keluar\n";
105         cout << "Pilihan: ";
106         cin >> choice;
107
108         switch (choice) {
109             case 1: {
110                 string nama;
111                 int nim;
112                 cout << "Masukkan Nama Mahasiswa: ";
113                 cin >> nama;
114                 getline(cin, nama);
115                 cout << "Masukkan NIM Mahasiswa: ";
116                 cin >> nim;
117                 queue.enqueue(nama, nim);
118                 break;
119             }
120             case 2: {
121                 queue.dequeue();
122                 break;
123             }
124             case 3: {
125                 queue.viewQueue();
126                 cout << "Jumlah antrian = " << queue.countQueue() << endl;
127                 break;
128             }
129             case 4: {
130                 queue.clearQueue();
131                 break;
132             }
133             case 5:
134                 cout << "Keluar dari program." << endl;
135                 break;
136             default:
137                 cout << "Pilihan tidak valid. Coba lagi." << endl;
138         }
139     } while (choice != 5);
140
141     return 0;
142 }
```

Dalam program di atas dikelola antrian mahasiswa menggunakan struktur data linke list. Setiap mahasiswa memiliki nama dan nim, dengan antrian diurutkan menaik berdasarkan nim. Program tersebut juga menyediakan operasi utama seperti: menambah mahasiswa (enqueue) yang menempatkan elemen secara terurut, menghapus mahasiswa dari depan (dequeue), menampilkan seluruh antrian yang dilakukan oleh (viewQueue), lalu yang menghapus semua elemen yaitu (clearQueue), dan yang menghitung jumlah elemen (countQueue). Menu interaktif memungkinkan pengguna memilih fungsi ini hingga mereka memilih "Keluar". Antrian otomatis dihapus saat program berakhir untuk menghindari kebocoran memori, menggunakan destruktur yang memanggil clearQueue.

Output:

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 1
Masukkan Nama Mahasiswa: Andrew
Masukkan NIM Mahasiswa: 1234
```

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 1
Masukkan Nama Mahasiswa: Alex
Masukkan NIM Mahasiswa: 4321
```

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 3
Data antrian teller:
1. Andrew (NIM: 1234)
2. Alex (NIM: 4321)
Jumlah antrian = 2
```

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 2
```

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 4
Antrian telah dikosongkan.
```

```
Menu Antrian:
1. Tambah Mahasiswa ke Antrian
2. Hapus Mahasiswa dari Antrian
3. Lihat Antrian
4. Bersihkan Antrian
5. Keluar
Pilihan: 5
Keluar dari program.
Antrian telah dikosongkan.
```

Semoga Selalu diberi kemudahan^^