

**LAPORAN PRAKTIKUM**  
**MODUL 8**  
**QUEUE**



**Nama :**

Muhammad Rifqi Al Baqi Ananta (2311104005)

**Dosen :**

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

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

# I. TP

## 1. UNGUIDED

Code:

```
1 #include <iostream>
2 using namespace std;
3
4 struct Mahasiswa {
5     string nama;
6     string nim;
7 };
8
9 struct Node {
10     Mahasiswa data;
11     Node* next;
12 };
13
14 class PriorityQueue {
15 private:
16     Node *front;
17     int count;
18
19 public:
20     PriorityQueue() {
21         front = NULL;
22         count = 0;
23     }
24
25     bool isEmpty() {
26         return front == NULL;
27     }
28
29     void enqueue(string nama, string nim) {
30         Node* newNode = new Node;
31         newNode->data.nama = nama;
32         newNode->data.nim = nim;
33         newNode->next = NULL;
34
35         if(isEmpty() || nim < front->data.nim) {
36             newNode->next = front;
37             front = newNode;
38         } else {
39             Node* current = front;
40             while(current->next != NULL &&
41                 current->next->data.nim <= nim) {
42                 current = current->next;
43             }
44             newNode->next = current->next;
45             current->next = newNode;
46         }
47         count++;
48     }
49
50     void dequeue() {
51         if(isEmpty()) {
52             cout << "Antrian kosong" << endl;
53             return;
54         }
55
56         Node* temp = front;
57         front = front->next;
58         delete temp;
59         count--;
60     }
61 }
```

```
1 void viewQueue() {
2     if(isEmpty()) {
3         cout << "Antrian kosong" << endl;
4         return;
5     }
6
7     Node* current = front;
8     int no = 1;
9     cout << "\nData antrian:" << endl;
10    while(current != NULL) {
11        cout << no << ". Nama: " << current->data.nama
12        << ", NIM: " << current->data.nim << endl;
13        current = current->next;
14        no++;
15    }
16 }
17
18 int countQueue() {
19     return count;
20 }
21
22 void clearQueue() {
23     while(!isEmpty()) {
24         dequeue();
25     }
26 }
27 };
28
29 int main() {
30     PriorityQueue q;
31     string nama, nim;
32     int pilihan;
33
34     do {
35         cout << "\nMenu:" << endl;
36         cout << "1. Tambah Mahasiswa" << endl;
37         cout << "2. Hapus Mahasiswa" << endl;
38         cout << "3. Lihat Antrian" << endl;
39         cout << "4. Kosongkan Antrian" << endl;
40         cout << "0. Keluar" << endl;
41         cout << "Pilihan: ";
42         cin >> pilihan;
43
44         switch(pilihan) {
45             case 1:
46                 cout << "Masukkan Nama: ";
47                 cin.ignore();
48                 getline(cin, nama);
49                 cout << "Masukkan NIM: ";
50                 getline(cin, nim);
51                 q.enqueue(nama, nim);
52                 break;
53             case 2:
54                 q.dequeue();
55                 break;
56             case 3:
57                 q.viewQueue();
58                 cout << "Jumlah Antrian = " << q.countQueue() << endl;
59                 break;
60             case 4:
61                 q.clearQueue();
62                 break;
63         }
64     } while(pilihan != 0);
65
66     return 0;
67 }
```

Output:

```
Data antrian:
1. Nama: bub, NIM: 102
2. Nama: taa, NIM: 120
3. Nama: nan, NIM: 123
Jumlah Antrian = 3

Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa
3. Lihat Antrian
4. Kosongkan Antrian
0. Keluar
Pilihan: █
```

## II. PENJELASAN

Implementasi Priority Queue (antrian prioritas) menggunakan linked list dalam C++ yang digunakan untuk mengelola data mahasiswa. Program ini memiliki struktur Mahasiswa yang terdiri dari nama dan NIM, serta menggunakan struct Node untuk membuat linked list. Class PriorityQueue menyediakan berbagai operasi seperti enqueue (menambah mahasiswa baru dengan prioritas berdasarkan NIM), dequeue (menghapus mahasiswa dari depan antrian), viewQueue (menampilkan semua data mahasiswa dalam antrian), countQueue (menghitung jumlah mahasiswa dalam antrian), dan clearQueue (mengosongkan antrian). Program utama menampilkan menu interaktif yang memungkinkan pengguna untuk melakukan operasi-operasi tersebut, dimana antrian akan otomatis terurut berdasarkan NIM mahasiswa (dari yang terkecil ke terbesar) saat melakukan penambahan data baru.