

**LAPORAN PRAKTIKUM**

**Modul 8**

**“QUEUE”**



**Disusun Oleh:**

**Benedictus Qsota Noventino Baru - 2311104029  
S1SE07A**

**Dosen :**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM**

**PURWOKERTO**

**2024**

## LATIHAN MODUL

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Node {
6     string nama;
7     string nim;
8     Node* next;
9 };
10
11 Node* front = nullptr;
12 Node* back = nullptr;
13
14 bool isEmpty() {
15     return front == nullptr;
16 }
17
18 void enqueueAntrian(string nama, string nim) {
19     Node* newNode = new Node();
20     newNode->nama = nama;
21     newNode->nim = nim;
22     newNode->next = nullptr;
23
24     if (isEmpty()) {
25         front = back = newNode;
26     } else {
27         back->next = newNode;
28         back = newNode;
29     }
30 }
31
32 void dequeueAntrian() {
33     if (isEmpty()) {
34         cout << "Antrian kosong" << endl;
35     } else {
36         Node* temp = front;
37         front = front->next;
38         delete temp;
39
40         if (front == nullptr) {
41             back = nullptr;
42         }
43     }
44 }
45
46 void clearQueue() {
47     while (!isEmpty()) {
48         dequeueAntrian();
49     }
50 }
51
52 int countQueue() {
53     int count = 0;
54     Node* temp = front;
55     while (temp != nullptr) {
56         count++;
57         temp = temp->next;
58     }
59     return count;
60 }
61
62 void viewQueue() {
63     if (isEmpty()) {
64         cout << "Antrian kosong" << endl;
65         return;
66     }
67
68     Node* temp = front;
69     cout << "Data antrian teller:" << endl;
70     int position = 1;
71     while (temp != nullptr) {
72         cout << position++ << ". Nama: " << temp->nama << ", NIM: " << temp->nim << endl;
73         temp = temp->next;
74     }
75 }
```

```

1 void enqueuePriority(string nama, string nim) {
2     Node* newNode = new Node();
3     newNode->nama = nama;
4     newNode->nim = nim;
5     newNode->next = nullptr;
6
7     if (isEmpty() || nim < front->nim) {
8         newNode->next = front;
9         front = newNode;
10        if (back == nullptr) {
11            back = newNode;
12        }
13    } else {
14        Node* temp = front;
15        while (temp->next != nullptr && temp->next->nim < nim) {
16            temp = temp->next;
17        }
18        newNode->next = temp->next;
19        temp->next = newNode;
20        if (newNode->next == nullptr) {
21            back = newNode;
22        }
23    }
24 }
25
26 int main() {
27     int choice;
28     string nama, nim;
29
30     do {
31         cout << "\nMenu:\n";
32         cout << "1. Tambah Antrian\n";
33         cout << "2. Tambah Antrian dengan Prioritas NIM\n";
34         cout << "3. Hapus Antrian\n";
35         cout << "4. Lihat Antrian\n";
36         cout << "5. Hapus Semua Antrian\n";
37         cout << "6. Hitung Antrian\n";
38         cout << "0. Keluar\n";
39         cout << "Pilih menu: ";
40         cin >> choice;
41
42         switch (choice) {
43             case 1:
44                 cout << "Masukkan Nama: ";
45                 cin >> nama;
46                 cout << "Masukkan NIM: ";
47                 cin >> nim;
48                 enqueueAntrian(nama, nim);
49                 break;
50             case 2:
51                 cout << "Masukkan Nama: ";
52                 cin >> nama;
53                 cout << "Masukkan NIM: ";
54                 cin >> nim;
55                 enqueuePriority(nama, nim);
56                 break;
57             case 3:
58                 dequeueAntrian();
59                 break;
60             case 4:
61                 viewQueue();
62                 break;
63             case 5:
64                 clearQueue();
65                 break;
66             case 6:
67                 cout << "Jumlah antrian = " << countQueue() << endl;
68                 break;
69             case 0:
70                 cout << "Keluar dari program." << endl;
71                 break;
72             default:
73                 cout << "Pilihan tidak valid." << endl;
74         }
75     } while (choice != 0);
76
77     clearQueue(); // Membersihkan semua node sebelum keluar
78     return 0;
79 }
80

```

## Penjelasan Program

1. **Menggunakan Linked List:** Queue diubah menggunakan **Node** yang berisi nama, NIM, dan pointer **next** untuk node berikutnya.
2. **Enqueue dengan Prioritas:** Pada fungsi **enqueuePriority**, NIM yang lebih kecil akan dimasukkan lebih awal. Proses ini dilakukan dengan membandingkan nilai NIM dari setiap node.
3. **Fungsi-fungsi Lain:** Fungsi **isEmpty**, **dequeueAntrian**, **clearQueue**, **countQueue**, dan **viewQueue** dimodifikasi agar bekerja dengan struktur linked list.
4. **Input Dinamis:** Program menampilkan menu interaktif untuk memungkinkan pengguna menambahkan antrian dengan atau tanpa prioritas, melihat antrian, dan menghapus antrian.