

LAPORAN PRAKTIKUM

PERTEMUAN 8

QUEUE



Nama :

Resita Istania Purwanto (231104037)

Dosen :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. UNGUIDED

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

Jawab:

1. Soal 1

Kode:

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  Codeium: Refactor | Explain
6  struct Mahasiswa {
7      string nama;
8      string nim;
9      Mahasiswa* next; // Penunjuk ke mahasiswa berikutnya
10 };
11
12 Codeium: Refactor | Explain
13 class Queue {
14     protected:
15         Mahasiswa* front;
16         Mahasiswa* back;
17     public:
18         Queue() : front(nullptr), back(nullptr) {}
19
20     Codeium: Refactor | Explain | Generate Function Comment | X
21     bool isEmpty() {
22         return front == nullptr;
23     }
24
25     Codeium: Refactor | Explain | Generate Function Comment | X
26     void enqueue(string nama, string nim) {
27         Mahasiswa* newMahasiswa = new Mahasiswa;
28         newMahasiswa->nama = nama;
29         newMahasiswa->nim = nim;
30         newMahasiswa->next = nullptr;
31
32         // Jika antrian kosong, masukkan mahasiswa baru
33         if (isEmpty()) {
```

```

31         front = back = newMahasiswa;
32     } else {
33         back->next = newMahasiswa;
34         back = newMahasiswa;
35     }
36 }
37
38 Codeium: Refactor | Explain | Generate Function Comment | X
39 void dequeue() {
40     if (isEmpty()) {
41         cout << "Antrian kosong!" << endl;
42     } else {
43         Mahasiswa* temp = front;
44         front = front->next;
45         delete temp;
46     }
47 }
48
49 Codeium: Refactor | Explain | Generate Function Comment | X
50 void viewQueue() {
51     if (isEmpty()) {
52         cout << "Antrian kosong!" << endl;
53     } else {
54         Mahasiswa* temp = front;
55         int i = 1;
56         while (temp != nullptr) {
57             cout << i++ << ". " << temp->nama << " - " << temp->nim << endl;
58             temp = temp->next;
59         }
60     }
61 }

```

```

61 Codeium: Refactor | Explain | Generate Function Comment | X
62 int countQueue() {
63     int count = 0;
64     Mahasiswa* temp = front;
65     while (temp != nullptr) {
66         count++;
67         temp = temp->next;
68     }
69     return count;
70 }
71
72 Codeium: Refactor | Explain | Generate Function Comment | X
73 void clearQueue() {
74     while (!isEmpty()) {
75         dequeue();
76     }
77 }
78
79 };

```

2. Soal 2

```

1 #include <iostream>
2 #include <limits>
3 #include "nol.cpp"
4
5 using namespace std;
6
7 Codeium: Refactor | Explain | Generate Function Comment | X
8 int main() {
9     Queue q;
10    string nama, nim;
11    int pilihan;
12
13    do {
14        cout << "\nMenu Antrian Mahasiswa:" << endl;
15        cout << "1. Tambah Antrian" << endl;
16        cout << "2. Hapus Antrian" << endl;
17        cout << "3. Lihat Antrian" << endl;
18        cout << "4. Hitung Jumlah Antrian" << endl;
19        cout << "5. Kosongkan Antrian" << endl;
20        cout << "6. Keluar" << endl;
21
22        cout << "Pilih menu: ";
23        cin >> pilihan;
24
25        if (cin.fail()) {
26            cin.clear();
27            cin.ignore(numeric_limits<streamsize>::max(), '\n');
28            cout << "Input tidak valid! Silakan pilih angka yang benar." << endl;
29            continue;
30        }
31
32        cin.ignore(); // Menghapus newLine setelah memilih menu
33
34        switch (pilihan) {

```

```

34         case 1:
35             cout << "Masukkan Nama Mahasiswa: ";
36             getline(cin, nama);
37             cout << "Masukkan NIM Mahasiswa: ";
38             getline(cin, nim);
39             q.enqueue(nama, nim);
40             break;
41         case 2:
42             q.dequeue();
43             break;
44         case 3:
45             q.viewQueue();
46             break;
47         case 4:
48             cout << "Jumlah antrian: " << q.countQueue() << endl;
49             break;
50         case 5:
51             q.clearQueue();
52             break;
53         case 6:
54             cout << "Terima kasih! Program selesai." << endl;
55             break;
56         default:
57             cout << "Pilihan tidak valid! Silakan pilih angka yang valid." << endl;
58     }
59 } while (pilihan != 6);
60
61 return 0;
62 }
63

```

3. Soal 3

```

1  #include <iostream>
2  #include <limits>
3  #include "nol.cpp"
4
5  using namespace std;
6
7  Codeium: Refactor | Explain
8  class QueueWithPriority : public Queue {
9  public:
10     Codeium: Refactor | Explain | Generate Function Comment | X
11     void enqueueWithPriority(string nama, string nim) {
12         Mahasiswa* newMahasiswa = new Mahasiswa;
13         newMahasiswa->nama = nama;
14         newMahasiswa->nim = nim;
15         newMahasiswa->next = nullptr;
16
17         // Jika antrian kosong, masukkan mahasiswa baru
18         if (isEmpty()) {
19             front = back = newMahasiswa;
20         } else {
21             // Jika NIM baru lebih kecil dari NIM pertama, masukkan di depan
22             if (nim < front->nim) {
23                 newMahasiswa->next = front;
24                 front = newMahasiswa;
25             } else {
26                 // Cari posisi yang tepat untuk mahasiswa baru berdasarkan NIM
27                 Mahasiswa* temp = front;
28                 while (temp->next != nullptr && temp->next->nim < nim) {
29                     temp = temp->next;
30                 }
31                 newMahasiswa->next = temp->next;
32                 temp->next = newMahasiswa;
33
34                 // Jika mahasiswa baru ditambahkan di akhir, update 'back'

```

```

33         if (temp->next == nullptr) {
34             back = newMahasiswa;
35         }
36     }
37 }
38 }
39 };
40
Codeium: Refactor | Explain | Generate Function Comment | X
41 int main() {
42     QueueWithPriority q;
43     string nama, nim;
44     int pilihan;
45
46     do {
47         cout << "\nMenu Antrian Mahasiswa:" << endl;
48         cout << "1. Tambah Antrian" << endl;
49         cout << "2. Hapus Antrian" << endl;
50         cout << "3. Lihat Antrian" << endl;
51         cout << "4. Hitung Jumlah Antrian" << endl;
52         cout << "5. Kosongkan Antrian" << endl;
53         cout << "6. Keluar" << endl;
54
55         cout << "Pilih menu: ";
56         cin >> pilihan;
57
58         if(cin.fail()) {
59             cin.clear();
60             cin.ignore(numeric_limits<streamsize>::max(), '\n');
61
62             cout << "Input tidak valid! Silakan pilih angka yang benar." << endl;
63             continue;
64         }
65
66         cin.ignore(); // Menghapus newline setelah memilih menu
67
68         switch (pilihan) {
69             case 1:
70                 cout << "Masukkan Nama Mahasiswa: ";
71                 getline(cin, nama);
72                 cout << "Masukkan NIM Mahasiswa: ";
73                 getline(cin, nim);
74                 q.enqueueWithPriority(nama, nim); // Menambah mahasiswa dengan prioritas
75                 break;
76             case 2:
77                 q.dequeue();
78                 break;
79             case 3:
80                 q.viewQueue();
81                 break;
82             case 4:
83                 cout << "Jumlah antrian: " << q.countQueue() << endl;
84                 break;
85             case 5:
86                 q.clearQueue();
87                 break;
88             case 6:
89                 cout << "Terima kasih! Program selesai." << endl;
90                 break;
91             default:
92                 cout << "Pilihan tidak valid! Silakan pilih angka yang valid." << endl;
93                 break;
94         } while (pilihan != 6);
95
96         return 0;
97     }

```

Output:

```
Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 1
Masukkan Nama Mahasiswa: Resita
Masukkan NIM Mahasiswa: 1234

Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 1
Masukkan Nama Mahasiswa: Istania
Masukkan NIM Mahasiswa: 1235

Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 3
1. Resita - 1234
2. Istania - 1235

Menu Antrian Mahasiswa:
```

```
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 4
Jumlah antrian: 2

Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 5

Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
6. Keluar
Pilih menu: 2
Antrian kosong!

Menu Antrian Mahasiswa:
1. Tambah Antrian
2. Hapus Antrian
3. Lihat Antrian
4. Hitung Jumlah Antrian
5. Kosongkan Antrian
```

```
6. Keluar
Pilih menu: 6
Terima kasih! Program selesai.
```

Penjelasan:

- a. no1.cpp - Kelas Queue (Linked List):
 - antrian mahasiswa menggunakan linked list.
 - Ada dua pointer: front (depan) dan back (belakang).
 - Metode:
 1. enqueue(): Menambah mahasiswa ke antrian.
 2. dequeue(): Menghapus mahasiswa dari depan antrian.
 3. viewQueue(), countQueue(), dan clearQueue(): Melihat, menghitung, dan mengosongkan antrian.
- b. no2.cpp - Program Utama
 - Menyediakan menu untuk:
 - Menambah mahasiswa.
 - Menghapus mahasiswa.
 - Melihat antrian.
 - Menghitung jumlah mahasiswa.
 - Mengosongkan antrian.
 - Input nama dan NIM mahasiswa.
- c. no3.cpp
 - Mengurutkan antrian berdasarkan NIM (yang lebih kecil diprioritaskan).
 - enqueueWithPriority(): Menambahkan mahasiswa sesuai prioritas NIM.