

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2023/2024
Program Studi Teknik Informatika
Institut Teknologi Sumatera



Modul : Queue

Nama : Aulia Putri Sayidina

NIM : 122140060

Kelas (Kelas Asal) : RC

Instruksi sederhana :

- Disarankan kepada **Praktikan Algoritma Struktur Data** untuk mengeditnya menggunakan Google Docs agar tidak berantakan dan rapi,
- Silahkan mengganti **Nama Modul** baik yang ada pada **Cover** dan **Header** sesuai dengan materi praktikum,
- Gunakan text styling seperti **Heading 1**, **Normal Text** yang telah terformat / Text Style lainnya yang digunakan untuk menjaga estetika laporan,
- Gunakan **Syntax Highlighter** untuk merapikan kode yang sudah Praktikan buat ke dalam Laporan Praktikum.

Materi Praktikum

PENDAHULUAN:

Dalam struktur data, Queue (Antrian) adalah suatu struktur yang mengorganisir elemen-elemen data secara linear dengan prinsip FIFO (First-In-First-Out). Elemen pertama yang dimasukkan ke dalam antrian akan menjadi elemen pertama yang diambil keluar. Implementasi queue umumnya menggunakan linked list untuk mendukung penambahan dan penghapusan data secara dinamis.

FUNGSI QUEUE:

Queue memberikan solusi untuk masalah pengelolaan data yang mengantri untuk diproses secara teratur. Dengan prinsip FIFO, elemen yang pertama kali datang akan diproses terlebih dahulu.

Queue terdiri dari dua operasi dasar:

- Enqueue (Add): Menambahkan elemen ke ujung belakang antrian.
- Dequeue (Remove): Menghapus elemen dari ujung depan antrian.

Link repl.it Source Code

<https://onlinegdb.com/LxsyPa2KIo>

Source Code

```
1. //Aulia Putri Sayidina
2. //122140060
3.
4. #include <iostream>
5. #include <cctype> // for character type checking
6. using namespace std;
7.
8. const int MAX_SIZE = 100; // Maximum queue size
9.
10. struct Queue {
11.     int queueArray[MAX_SIZE];
12.     int front, rear;
13. }
```

Praktikum Algoritma dan Struktur Data Ke-VI — Queue

```
14.     Queue() {
15.         front = -1;
16.         rear = -1;
17.     }
18.
19.     bool isEmpty() {
20.         return front == -1 && rear == -1;
21.     }
22.
23.     bool isFull() {
24.         return rear == MAX_SIZE - 1;
25.     }
26.
27.     void enqueue(int value) {
28.         if (isFull()) {
29.             cout << "Queue is full. Cannot add more elements." << endl;
30.             return;
31.         }
32.
33.         if (isEmpty()) {
34.             front = rear = 0;
35.         } else {
36.             rear = (rear + 1) % MAX_SIZE;
37.         }
38.
39.         queueArray[rear] = value;
40.     }
41.
42.     int dequeue() {
43.         if (isEmpty()) {
44.             cout << "Queue is empty. Cannot remove elements." << endl;
45.             return -1;
46.         }
47.
48.         int dequeuedValue = queueArray[front];
49.
50.         if (front == rear) {
51.             front = rear = -1;
52.         } else {
53.             front = (front + 1) % MAX_SIZE;
54.         }
55.
56.         return dequeuedValue;
57.     }
58.
```

Praktikum Algoritma dan Struktur Data Ke-VI — Queue

```
59.     void printQueue() {
60.         if (isEmpty()) {
61.             cout << "Queue is empty." << endl;
62.             return;
63.         }
64.
65.         cout << "Queue contents:" << endl;
66.
67.         int i = front;
68.         while (i != rear) {
69.             cout << queueArray[i] << " -> index " << i << " (" <<
char(queueArray[i]) << ")" << endl;
70.             i = (i + 1) % MAX_SIZE;
71.         }
72.         cout << queueArray[rear] << " -> index " << i << " (" <<
char(queueArray[rear]) << ")" << endl;
73.     }
74. };
75.
76. int main() {
77.     string input;
78.     Queue queue;
79.
80.     cout << "Enter a string: ";
81.     cin >> input;
82.
83.     for (int i = 0; i < input.length(); i++) {
84.         char c = input[i];
85.
86.         if (islower(c)) {
87.             queue.enqueue(int(c));
88.             cout << "ENQUEUE: " << int(c) << " (" << c << ")" << endl;
89.         } else if (c == '-') {
90.             int dequeuedValue = queue.dequeue();
91.
92.             if (dequeuedValue != -1) {
93.                 cout << "DEQUEUE: " << dequeuedValue << " (" << char(dequeuedValue)
<< ")" << endl;
94.             }
95.         }
96.     }
97.
98.     queue.printQueue(); // Display remaining characters in the queue
99.     return 0;
100. }
```

101.

Dokumentasi Hasil Running

```
Enter a string: aulia-  
ENQUEUE: 97 (a)  
ENQUEUE: 117 (u)  
ENQUEUE: 108 (l)  
ENQUEUE: 105 (i)  
ENQUEUE: 97 (a)  
DEQUEUE: 97 (a)  
Queue contents:  
117 -> index 1 (u)  
108 -> index 2 (l)  
105 -> index 3 (i)  
97 -> index 4 (a)  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Gambar 1. Hasil Running Code Tugas Queue

Referensi

Modul perkuliahan Algoritma Struktur Data – Queue dari ITERA