

**Laporan Praktikum**

**Algoritma dan Struktur Data**

Ganjil 2023/2024

Program Studi Teknik Informatika

Institut Teknologi Sumatera

|  |  |
| --- | --- |
| **Modul :** | **Stack Queue** |
| **Nama :** | **Rayhan Fadel Irwanto** |
| **NIM :** | **122140236** |
| **Kelas (Kelas Asal) :** | **ASD RA** |

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](https://highlight.hohli.com/index.php)** untuk merapikan kode yang sudah Praktikan buat ke dalam Laporan Praktikum.

**Materi Praktikum**

Stack Queue

**Link repl.it Source Code**

<https://replit.com/@RayhanFadel/queue-LM4#latihan.cpp>

**Source Code**

Gunakan [**Syntax Highlighter**](https://highlight.hohli.com/index.php)untuk merapikan Source Code yang dipindahkan dari text editor anda ke dokumen ini.

1. #include <iostream>
2. using namespace std;
4. const int MAX\_SIZE = 10; // Maximum queue size
6. struct Queue {
7. int queueArray[MAX\_SIZE];
8. int front, rear;
10. **Queue() { createEmpty(); }**
12. bool isEmpty() { return front == -1; }
14. bool isFull() { return (rear + 1) % MAX\_SIZE == front; }
16. void createEmpty() {
17. front = -1;
18. rear = -1;
19. }
21. void add(int value) {
22. if (isFull()) {
23. cout << "Queue is full. Cannot add more elements." << endl;
24. return;
25. **}**
27. if (isEmpty()) {
28. front = 0;
29. rear = 0;
30. **} else {**
31. rear = (rear + 1) % MAX\_SIZE;
32. }
34. queueArray[rear] = value;
35. **}**
37. void remove() {
38. if (isEmpty()) {
39. cout << "Queue is empty. Cannot remove elements." << endl;
40. **return;**
41. }
43. if (front == rear) {
44. createEmpty();
45. **} else {**
46. front = (front + 1) % MAX\_SIZE;
47. }
48. }
50. **void printQueue() {**
51. if (isEmpty()) {
52. cout << "Queue is empty." << endl;
53. return;
54. }
56. cout << "Queue elements:" << endl;
58. int i = front;
59. while (i != rear) {
60. **cout << queueArray[i] << " -> index " << i << endl;**
61. i = (i + 1) % MAX\_SIZE;
62. }
64. cout << queueArray[rear] << " -> index " << rear << endl;
65. **cout << endl;**
66. }
68. int countNode() {
69. if (isEmpty()) {
70. **return 0;**
71. }
73. if (front <= rear) {
74. return rear - front + 1;
75. **} else {**
76. return MAX\_SIZE - front + rear + 1;
77. }
78. }
79. };
81. int main() {
82. int size, numFromUser;
84. Queue queue;
86. cout << "Enter the number of elements to add to the queue: ";
87. cin >> size;
89. if (size > MAX\_SIZE) {
90. **cout << "Number of elements exceeds the maximum capacity." << endl;**
91. return 1;
92. }
94. for (int i = 0; i < size; i++) {
95. **cout << "Enter element " << i + 1 << ": ";**
96. cin >> numFromUser;
98. queue.add(numFromUser);
99. }
101. cout << "Front element: " << queue.queueArray[queue.front] << endl;
103. queue.printQueue();
105. **cout << "Number of nodes in the queue: " << queue.countNode() << endl;**
107. return 0;
108. }

**Dokumentasi Hasil Running**

|  |
| --- |
|  |
| **Gambar 1.** Output Code Queue Latihan Minggu 4.  Program di atas adalah implementasi struktur data antrian (queue) dalam bahasa pemrograman C++. Struktur 'Queue' menggunakan array 'queueArray' untuk menyimpan elemen-elemen dalam antrian. Terdapat fungsi-fungsi seperti 'isEmpty' dan 'isFull' untuk memeriksa apakah antrian kosong atau penuh, 'add' untuk menambahkan elemen, 'remove' untuk menghapus elemen, 'printQueue' untuk menampilkan isi dari antrian, 'countNode' untuk menghitung jumlah elemen dalam antrian, dan 'createEmpty' untuk mengosongkan antrian.  Pada bagian utama program, pengguna diminta untuk memasukkan jumlah elemen yang ingin ditambahkan ke dalam antrian. Kemudian, elemen-elemen tersebut dimasukkan ke dalam antrian. Program menampilkan elemen paling depan dari antrian, isi dari antrian, serta jumlah total elemen dalam antrian.  Dengan struktur data antrian ini, program dapat menambah, menghapus, dan menampilkan elemen-elemen antrian dengan mudah. |

|  |
| --- |
|  |
|  |