

Laporan Praktikum Algoritma dan Struktur Data

Praktikum 05 : Queue



Oleh:

Muhammad Dimas Ardiansyah / 5223600019

Program Studi Teknologi Game

Departemen Teknik Multimedia Kreatif

Politeknik Elektronika Negeri Surabaya

2024

Tahapan Praktikum:

1. Queue Menggunakan Linked List

Program tersebut adalah implementasi sederhana dari struktur data antrian (queue) menggunakan linked list dalam C++. Antrian mengikuti aturan FIFO (First In First Out). Program ini memiliki fungsi untuk menambahkan elemen ke antrian ('enqueue'), menghapus elemen dari antrian ('dequeue'), serta melihat elemen pertama tanpa menghapusnya ('peekFront'). Program menguji implementasi antrian dengan memasukkan 100 elemen ke antrian, menghapus 50 elemen, dan mencetak sisa elemen dalam antrian.

```
Output Clear  
/tmp/nrzSCyohUB.o  
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86  
87 88 89 90 91 92 93 94 95 96 97 98 99 100  
  
=== Code Execution Successful ===
```

2. Enqueue pada Queue Linked List

Program tersebut adalah implementasi sederhana dari struktur data antrian (queue) dalam C++. Fungsi 'enqueue' menambahkan elemen baru ke antrian, dan menggunakan linked list untuk penyimpanan. Variabel 'count' digunakan untuk melacak jumlah elemen dalam antrian.

```
main.cpp  
  
1 void Queue::enqueue(int newValue) {  
2     Node* newNode = new Node(newValue, nullptr);  
3     if (head == nullptr) {  
4         head = tail = newNode;  
5     } else {  
6         tail->next = newNode;  
7         tail = newNode;  
8     }  
9     count++;  
10 }
```

3. Dequeue pada Queue Linked List

Program tersebut merupakan implementasi dari struktur data antrian (queue) menggunakan bahasa pemrograman C++. Fungsi 'dequeue' digunakan untuk menghapus elemen dari antrian dan mengembalikan nilainya. Jika antrian kosong, sebuah pengecualian (exception) akan dilempar. Variabel 'count' melacak jumlah elemen dalam antrian.

main.cpp

```
1 int Queue::dequeue() {  
2     if (empty()) {  
3         throw invalid_argument("QueueEmptyException");  
4     }  
5     int extractedValue = head->value;  
6     Node* tempNode = head;  
7     head = head->next;  
8     delete tempNode;  
9     count--;  
10    return extractedValue;  
11 }  
12
```

4. Implementasi Queue Menggunakan 2 Stack

Program tersebut adalah implementasi antrian (queue) menggunakan dua tumpukan (stack) dalam bahasa C++. Fungsi `enqueue` menambahkan elemen ke antrian, sementara fungsi `dequeue` menghapus dan mengembalikan elemen dari antrian sesuai dengan aturan FIFO. Dua tumpukan (`stack1` dan `stack2`) digunakan untuk menyimpan elemen antrian. Program diuji dengan menambahkan dan menghapus beberapa elemen dari antrian.

Output

/tmp/4MubLEu0dr.o

1

11

111

=== Code Execution Successful ===

Tugas:

1. Stack Using a Queue

Program tersebut adalah implementasi tumpukan (stack) menggunakan struktur data antrian (queue) dalam C++. Ketika nilai baru dimasukkan, program memindahkan seluruh elemen yang ada ke belakang untuk membuat nilai baru menjadi elemen teratas tumpukan. Fungsi `push` menambahkan nilai, `pop` menghapus elemen teratas, `top` mengembalikan nilai teratas tanpa menghapusnya, dan `empty` memeriksa apakah tumpukan kosong. Program menguji operasi tumpukan dengan menambahkan nilai dan menampilkan nilai teratas sebelum dan setelah penghapusan.

Output

```
/tmp/YLXK3Kj2qx.o
```

```
Top: 3
```

```
Top: 2
```

```
=== Code Execution Successful ===
```

2. Josephus Problem

Program tersebut merupakan implementasi solusi untuk masalah Josephus menggunakan linked list dalam bahasa C++. Masalah Josephus melibatkan penghapusan berurutan dari sekelompok orang sampai hanya satu orang yang tersisa.

Output

```
/tmp/biCQcrUSeh.o
```

```
Masukkan jumlah orang: 10
```

```
Masukkan nilai langkah: 2
```

```
Orang yang selamat adalah: 5
```

```
=== Code Execution Successful ===
```