

**Laporan Praktikum**

**Algoritma dan Struktur Data**

Ganjil 2023/2024

Program Studi Teknik Informatika

Institut Teknologi Sumatera

|  |  |
| --- | --- |
| **Modul :** | **Circular Linked List** |
| **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**

Tugas Circular Linked List

**Link repl.it Source Code**

<https://replit.com/@RayhanFadel/CircularLL-TM3#main.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. // Rayhan Fadel Irwanto
2. // 122140236
4. #include <iostream>
5. **using namespace std;**
7. // Definisi struktur untuk node dalam Circular Single Linked List
8. struct Node {
9. int data;
10. **Node\* next;**
11. };
13. // Inisialisasi kepala linked list
14. Node\* head = nullptr;
16. // Fungsi untuk memasukkan angka di awal linked list
17. void insertFirst(int value) {
18. Node\* newNode = new Node;
19. newNode->data = value;
21. if (head == nullptr) {
22. head = newNode;
23. newNode->next = head;
24. } else {
25. **Node\* temp = head;**
27. while (temp->next != head) {
28. temp = temp->next;
29. }
31. temp->next = newNode;
32. newNode->next = head;
33. head = newNode;
34. }
35. **}**
37. // Fungsi untuk memasukkan angka di akhir linked list
38. void insertLast(int value) {
39. Node\* newNode = new Node;
40. **newNode->data = value;**
42. if (head == nullptr) {
43. head = newNode;
44. newNode->next = head;
45. **} else {**
46. Node\* temp = head;
48. while (temp->next != head) {
49. temp = temp->next;
50. **}**
52. temp->next = newNode;
53. newNode->next = head;
54. }
55. **}**
57. void deleteLast() {
58. if (head == nullptr) {
59. cout << "Linked List is empty." << endl;
60. **return;**
61. }
63. Node\* temp = head;
65. **while (temp->next->next != head) {**
66. temp = temp->next;
67. }
69. Node\* toDelete = temp->next;
70. **temp->next = head;**
71. delete toDelete;
72. }
74. void deleteFirst() {
75. **if (head == nullptr) {**
76. cout << "Linked List is empty." << endl;
77. return;
78. }
80. **Node\* temp = head;**
82. while (temp->next != head) {
83. temp = temp->next;
84. }
86. Node\* toDelete = head;
87. temp->next = head->next;
88. head = head->next;
89. delete toDelete;
90. **}**
92. // Fungsi untuk menampilkan data di linked list
93. void displayCircularLinkedList() {
94. if (head == nullptr) {
95. **cout << "Linked List is empty." << endl;**
96. return;
97. }
99. cout << "**\n**Isi circular linkedlist saat ini:" << endl << endl;
101. Node\* current = head;
103. do {
104. cout << "[" << current->data << ", " << current->next << "] ";
106. current = current->next;
107. } while (current != head);
109. cout << endl;
110. **}**
112. int main() {
113. int inputFromUser, size;
115. **cout << "Tentukan panjang linkedlist: ";**
116. cin >> size;
118. for (int i = 0; i < size; i++) {
119. cout << "Masukkan angka: ";
120. **cin >> inputFromUser;**
122. if (inputFromUser % 2 == 0) {
123. insertFirst(inputFromUser);
124. } else {
125. **insertLast(inputFromUser);**
126. }
127. }
129. displayCircularLinkedList();
131. int choice;
133. do {
134. cout<<"======================="<<endl;
135. **cout << "Pilih operasi yang ingin anda lakukan: " << endl;**
136. cout << "1. Hapus data di awal" << endl;
137. cout << "2. Hapus data di akhir" << endl;
138. cout << "3. Keluar Program" << endl;
140. **cout << "Masukkan pilihan anda: ";**
141. cin >> choice;
142. cout<<"======================="<<endl<<endl;
144. switch (choice) {
145. **case 1:**
146. deleteFirst();
147. break;
148. case 2:
149. deleteLast();
150. **break;**
151. case 3:
152. cout << "Terima kasih" << endl;
153. break;
154. default:
155. **cout << "Pilihan anda tidak valid" << endl;**
156. break;
157. }
159. if (choice != 3) {
160. **displayCircularLinkedList();**
161. }
163. cout<<endl;
164. } while (choice != 3);
166. return 0;
167. }

**Dokumentasi Hasil Running**

|  |
| --- |
|  |
| **Gambar 1.** Output Code Circular Linked List Tugas Minggu 3.  Program di atas merupakan implementasi Circular Single Linked List dalam bahasa pemrograman C++. Struktur data ini memiliki sifat melingkar, di mana node terakhir mengarah kembali ke node pertama. Program ini memungkinkan pengguna untuk menentukan panjang linked list, memasukkan angka ke dalamnya, dan melakukan operasi penghapusan elemen di awal atau di akhir linked list. Berikut adalah penjelasan singkatnya:  Pertama, program mendefinisikan struktur 'Node' untuk merepresentasikan node dalam circular linked list dengan dua anggota: 'data' untuk menyimpan nilai dan 'next' untuk menunjuk ke node berikutnya. Kepala linked list diinisialisasi sebagai null.  Terdapat dua fungsi untuk memasukkan angka, yaitu 'insertFirst' untuk memasukkan di awal linked list dan 'insertLast' untuk memasukkan di akhir linked list. Fungsi 'deleteFirst' dan 'deleteLast' digunakan untuk menghapus elemen di awal dan di akhir linked list.  Program kemudian meminta pengguna untuk menentukan panjang linked list dan memasukkan angka-angka. Jika angka genap, maka angka tersebut dimasukkan di awal linked list; jika ganjil, dimasukkan di akhir linked list. Setelah itu, program menampilkan isi circular linked list.  Selanjutnya, program memberikan opsi kepada pengguna untuk melakukan operasi penghapusan di awal atau di akhir linked list, atau keluar dari program. Pilihan pengguna diambil menggunakan perulangan 'do-while' dengan menu yang diberikan. Setiap setelah operasi penghapusan, isi circular linked list ditampilkan kembali.  Program ini memberikan pengalaman interaktif untuk memahami operasi dasar dari Circular Single Linked List, seperti penambahan elemen di awal dan akhir, serta penghapusan elemen di awal dan akhir. |

|  |
| --- |
|  |
|  |