

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2023/2024
Program Studi Teknik Informatika
Institut Teknologi Sumatera



Modul : Circular DLL

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, Double Circular Linked List adalah suatu struktur yang mengorganisir elemen-elemen data dalam urutan linear, dengan setiap elemen (node) memiliki dua pointer, yaitu pointer ke node sebelumnya dan pointer ke node berikutnya. Double Circular Linked List berbeda dari Linked List biasa karena ujung belakangnya terhubung ke ujung depan, membentuk suatu lingkaran. Konsep ini memungkinkan pengaksesan data baik dari awal hingga akhir atau sebaliknya dengan mudah.

FUNGSI DOUBLE CIRCULAR LINKED LIST:

Double Circular Linked List memberikan fleksibilitas dalam penyimpanan dan manipulasi data. Struktur ini memudahkan operasi penambahan dan penghapusan elemen di ujung depan atau belakang dengan efisien. Lingkaran pada struktur ini memungkinkan iterasi melalui semua elemen tanpa batas.

OPERASI DASAR DOUBLE CIRCULAR LINKED LIST:

- Enqueue (Add): Menambahkan elemen ke dalam Double Circular Linked List, baik di ujung depan atau belakang.
- Dequeue (Remove): Menghapus elemen dari Double Circular Linked List, baik di ujung depan atau belakang.
- Iterasi: Melakukan traversal pada seluruh elemen Double Circular Linked List, baik dari awal hingga akhir atau sebaliknya.

Link repl.it Source Code

<https://onlinegdb.com/9PDSm366B>

Source Code

```
1. //Aulia Putri Sayidina
2. //122140060
3.
4. #include <iostream>
5. using namespace std;
6.
7. // Struktur Node untuk elemen dalam Circular Double Linked List
8. struct Node {
9.     string data;
10.    Node* next;
11.    Node* prev;
12.};
13.
14.// Struktur Circular Double Linked List
15.struct CircularDoubleLinkedList {
16.    Node* head;
17.
18.    // Konstruktor untuk menginisialisasi Circular Double Linked List
19.    CircularDoubleLinkedList() : head(nullptr) {}
20.};
21.
22.// Fungsi untuk membuat node baru
23.Node* createNode(string value) {
24.    Node* newNode = new Node;
25.    newNode->data = value;
26.    newNode->next = nullptr;
27.    newNode->prev = nullptr;
28.    return newNode;
29.}
30.
31.// Fungsi untuk menambahkan elemen ke awal Circular Double Linked List
32.void insertFirst(CircularDoubleLinkedList& list, string value) {
33.    Node* newNode = createNode(value);
34.}
```

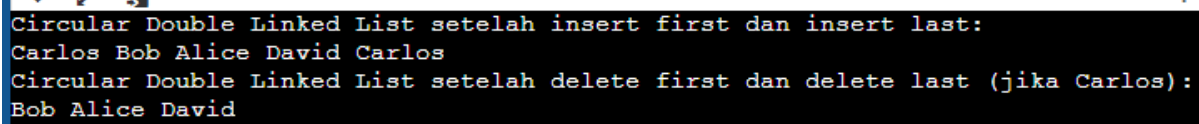
```
35.     if (list.head == nullptr) {
36.         list.head = newNode;
37.         list.head->next = list.head;
38.         list.head->prev = list.head;
39.     } else {
40.         newNode->next = list.head;
41.         newNode->prev = list.head->prev;
42.         list.head->prev->next = newNode;
43.         list.head->prev = newNode;
44.         list.head = newNode;
45.     }
46. }
47.
48. // Fungsi untuk menambahkan elemen ke akhir Circular Double Linked List
49. void insertLast(CircularDoubleLinkedList& list, string value) {
50.     Node* newNode = createNode(value);
51.
52.     if (list.head == nullptr) {
53.         list.head = newNode;
54.         list.head->next = list.head;
55.         list.head->prev = list.head;
56.     } else {
57.         newNode->next = list.head;
58.         newNode->prev = list.head->prev;
59.         list.head->prev->next = newNode;
60.         list.head->prev = newNode;
61.     }
62. }
63.
64. // Fungsi untuk menghapus elemen dari awal Circular Double Linked List
65. void deleteFirst(CircularDoubleLinkedList& list) {
66.     if (list.head == nullptr) {
67.         cout << "Circular Double Linked List kosong. Tidak dapat menghapus elemen."
68.         << endl;
69.         return;
70.     }
71.     Node* temp = list.head;
72.
73.     if (list.head->next == list.head) {
74.         list.head = nullptr;
75.     } else {
76.         list.head->prev->next = list.head->next;
77.         list.head->next->prev = list.head->prev;
78.         list.head = list.head->next;
```

```
79.     }
80.
81.     delete temp;
82. }
83.
84. // Fungsi untuk menghapus elemen dari akhir Circular Double Linked List
85. void deleteLast(CircularDoubleLinkedList& list) {
86.     if (list.head == nullptr) {
87.         cout << "Circular Double Linked List kosong. Tidak dapat menghapus elemen."
            << endl;
88.         return;
89.     }
90.
91.     Node* temp = list.head->prev;
92.
93.     if (temp == list.head) {
94.         list.head = nullptr;
95.     } else {
96.         temp->prev->next = list.head;
97.         list.head->prev = temp->prev;
98.     }
99.
100.    delete temp;
101. }
102.
103. // Fungsi untuk menampilkan Circular Double Linked List
104. void displayList(CircularDoubleLinkedList& list) {
105.     if (list.head == nullptr) {
106.         cout << "Circular Double Linked List kosong." << endl;
107.         return;
108.     }
109.
110.    Node* current = list.head;
111.    do {
112.        cout << current->data << " ";
113.        current = current->next;
114.    } while (current != list.head);
115.    cout << endl;
116. }
117.
118. int main() {
119.     CircularDoubleLinkedList myList;
120.
121.     // Menambahkan elemen ke awal dan akhir Circular Double Linked List
122.     insertFirst(myList, "Alice");
```

Praktikum Algoritma dan Struktur Data Ke-VI — Circular DLL

```
123.         insertFirst(myList, "Bob");
124.         insertFirst(myList, "Carlos");
125.         insertLast(myList, "David");
126.         insertLast(myList, "Carlos");
127.
128.         // Menampilkan Circular Double Linked List
129.         cout << "Circular Double Linked List setelah insert first dan insert
last:" << endl;
130.         displayList(myList);
131.
132.         // Menghapus elemen dari awal dan akhir Circular Double Linked List
133.         deleteFirst(myList);
134.
135.         // Menghapus elemen terakhir jika namanya "Carlos"
136.         if (myList.head != nullptr && myList.head->prev->data == "Carlos") {
137.             deleteLast(myList);
138.         }
139.
140.         // Menampilkan Circular Double Linked List setelah delete first dan
delete last (jika Carlos)
141.         cout << "Circular Double Linked List setelah delete first dan delete
last (jika Carlos):" << endl;
142.         displayList(myList);
143.
144.         return 0;
145.     }
146.
```

Dokumentasi Hasil Running



```
Circular Double Linked List setelah insert first dan insert last:  
Carlos Bob Alice David Carlos  
Circular Double Linked List setelah delete first dan delete last (jika Carlos):  
Bob Alice David
```

Gambar 1. Hasil Running Code Tugas Circular DLL

Referensi

Modul perkuliahan Algoritma Struktur Data – Circular DLL dari ITERA