

LAPORAN PRAKTIKUM
PERTEMUAN 6
Double Linked List Bagian 1



Nama :

Andika Rifki Pratama (2311104011)

Dosen :

Yudha Islami Sulistya,
S.Kom.,M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

B. Soal Tugas Pendahuluan

Code :

```
TP > tepe.cpp > DLL > DLL0
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7      Node* prev;
8
9      Node(int val) {
10         data = val;
11         next = nullptr;
12         prev = nullptr;
13     }
14 };
15
16 class DLL {
17 private:
18     Node* head;
19     Node* tail;
20
21 public:
22     DLL() {
23         head = nullptr;
24         tail = nullptr;
25
26         tambahAkhir_2311104011(9);
27         tambahAkhir_2311104011(82);
28         tambahAkhir_2311104011(11);
29     }
30
31     void tambahAwal_2311104011(int data) {
32         Node* newNode = new Node(data);
33         if (head == nullptr) {
34             head = tail = newNode;
35         } else {
36             newNode->next = head;
37             head->prev = newNode;
38             head = newNode;
39         }
40     }
41
42     void tambahAkhir_2311104011(int data) {
43         Node* newNode = new Node(data);
44         if (tail == nullptr) {
45             head = tail = newNode;
46         } else {
47             tail->next = newNode;
48             newNode->prev = tail;
49             tail = newNode;
50         }
51     }
52
53     void hapusAwal_2311104011() {
54         if (head == nullptr) {
55             cout << "LIST KOSONG" << endl;
56             return;
57         }
58         Node* temp = head;
59         if (head == tail) {
60             head = tail = nullptr;
61         } else {
62             head = head->next;
63             head->prev = nullptr;
64         }
65         delete temp;
```

```

86     cout << "Elemen pertama berhasil dihapus." << endl;
87 }
88
89 void hapusAkhir_2311104011() {
90     if (tail == nullptr) {
91         cout << "List Kosong" << endl;
92         return;
93     }
94     Node* temp = tail;
95     if (head == tail) {
96         head = tail = nullptr;
97     } else {
98         tail = tail->prev;
99         tail->next = nullptr;
100     }
101     delete temp;
102     cout << "Elemen berhasil dihapus." << endl;
103 }
104
105 void showDariAwal_2311104011() {
106     Node* temp = head;
107     if (temp == nullptr) {
108         cout << "LIST KOSONG" << endl;
109         return;
110     }
111     while (temp != nullptr) {
112         cout << temp->data;
113         if (temp->next != nullptr) cout << " <-> ";
114         temp = temp->next;
115     }
116     cout << endl;
117 }
118
119 void showDariAkhir_2311104011() {
120     Node* temp = tail;
121     if (temp == nullptr) {
122         cout << "List Kosong" << endl;
123         return;
124     }
125     while (temp != nullptr) {
126         cout << temp->data;
127         if (temp->prev != nullptr) cout << " <-> ";
128         temp = temp->prev;
129     }
130     cout << endl;
131 }
132
133 int main() {
134     DLL list;
135     int pilihan, nilai;
136
137     while (true) {
138         cout << "\n== MENU ==\n" << endl;
139         cout << "1. Tambah di Awal" << endl;
140         cout << "2. Tambah di Akhir" << endl;
141         cout << "3. Hapus dari Awal" << endl;
142         cout << "4. Hapus dari Akhir" << endl;
143         cout << "5. Tampilkan dari Akhir" << endl;
144         cout << "6. Keluar" << endl;
145         cout << "==" List saat ini ==\n" << endl;
146         list.showDariAwal_2311104011();
147         cout << "Pilih opsi: ";
148
149         int pilihan;
150         cin >> pilihan;
151
152         switch (pilihan) {
153             case 1:
154                 cout << "Masukkan nilai untuk ditambah di awal: ";
155                 cin >> nilai;
156                 list.tambahAwal_2311104011(nilai);
157                 break;
158             case 2:
159                 cout << "Masukkan nilai untuk ditambah di akhir: ";
160                 cin >> nilai;
161                 list.tambahAkhir_2311104011(nilai);
162                 break;
163             case 3:
164                 list.hapusAwal_2311104011();
165                 break;
166             case 4:
167                 list.hapusAkhir_2311104011();
168                 break;
169             case 5:
170                 cout << "List dari Akhir: ";
171                 list.showDariAkhir_2311104011();
172                 break;
173             case 6:
174                 cout << "Bye." << endl;
175                 return 0;
176             default:
177                 cout << "Opsi tidak valid, coba lagi." << endl;
178                 break;
179         }
180     }
181
182     return 0;
183 }

```

Penjelasan :

Pada program ini pertama terdapat struct untuk membuat node baru. Kemudian terdapat class DLL, pada class ini terdapat deklarasi variable untuk yang di privat.

Pada bagian public terdapat cclass DLL yang berisi penjelasan variable dan juga inputan data awal.

Setelahnya terdapat method tambah awal, fungsinya yaitu untuk menambahkan nilai baru di awal list, cara kerjanya yaitu dengan memeriksa apakah bagian depan list kosong atau tidak, apabila kosong maka akan langsung membuat baru, dan apabila tidak maka akan menggeser nilai supaya head kosong. Cara kerjanya sama dengan fungsi tambahAkhir.

Kemudian terdapat hapusAwal, fungsi ini berfungsi untuk menghapus nilai atau value yang berada di posisi paling depan dari list, fungsi ini cara kerjanya sama dengan fungsi hapusAkhir.

Method showAwal dan Akhir, fungsi ini berfungsi untuk menunjukan list dari awal atau akhir.

Pada bagian main terdapat fungsi while yang akan memberikan output berupa kata2 untuk indicator inputan, setelahnya terdapat switch case untuk memasukan imputan atau sebagai pemecah untuk system memilih inputan yang dimaksud. Kemudian setelah dipilih maka akan Kembali ke menu utama.

UNGUIDED

```

Unguided > C:\no1.cpp > DLL > displayFromEnd_23111040110
#include <iostream>
#include <string>
using namespace std;

struct Node {
    int bookID;
    string bookTitle;
    string bookWriter;
    Node* next;
    Node* prev;

    Node(int id, string title, string writer) {
        bookID = id;
        bookTitle = title;
        bookWriter = writer;
        prev = nullptr;
        next = nullptr;
    }
};

class DLL {
private:
    Node* head;
    Node* tail;
public:
    DLL() {
        head = nullptr;
        tail = nullptr;
    }

    void addBook_2311104011(int bookID, string bookTitle, string bookWriter) {
        Node* newNode = new Node(bookID, bookTitle, bookWriter);

        if (tail == nullptr) {
            head = tail = newNode;
        } else {
            tail->next = newNode;
            newNode->prev = tail;
            tail = newNode;
        }
        cout << "Buku ditambahkan " << bookTitle << endl;
    }

    void displayFromStart_2311104011() {
        if (head == nullptr) {
            cout << "Error! Daftar buku kosong." << endl;
            return;
        }

        Node* temp = head;
        cout << "Daftar buku " << endl;
        while (temp != nullptr) {
            cout << "ID: " << temp->bookID << ", Judul: " << temp->bookTitle
                << ", Penulis: " << temp->bookWriter << endl;
            temp = temp->next;
        }
    }

    void displayFromEnd_2311104011() {
        if (tail == nullptr) {
            cout << "Daftar buku kosong." << endl;
            return;
        }

        Node* temp = tail;
        cout << "Daftar buku (akhir ke awal):" << endl;
        while (temp != nullptr) {
            cout << "ID: " << temp->bookID << ", Judul: " << temp->bookTitle
                << ", Penulis: " << temp->bookWriter << endl;
            temp = temp->prev;
        }
    }
};

int main() {
    DLL bookList;

    bookList.addBook_2311104011(1, "The Sylvester System", "Shay Cormack");
    bookList.addBook_2311104011(2, "Morphology Of The Dawn", "Arno Dorian");
    bookList.addBook_2311104011(3, "Birds Among The Pigeons", "Agatha Christie");

    cout << endl;
    bookList.displayFromStart_2311104011();
    cout << endl;
    bookList.displayFromEnd_2311104011();

    return 0;
}

```