Hasil Pertama

#include <iostream>

using namespace std;

// Struktur simpul

struct Node {

int data;

Node\* prev;

Node\* next;

};

// Kelas Double Linked List

class DoubleLinkedList {

private:

Node\* head;

Node\* tail;

public:

// Konstruktor

DoubleLinkedList() {

head = nullptr;

tail = nullptr;

}

// Method untuk menyisipkan simpul di ujung awal

void insertAtBeginning(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->prev = nullptr;

newNode->next = head;

if (head != nullptr)

head->prev = newNode;

else

tail = newNode;

head = newNode;

}

// Method untuk menyisipkan simpul di ujung akhir

void insertAtEnd(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->next = nullptr;

newNode->prev = tail;

if (tail != nullptr)

tail->next = newNode;

else

head = newNode;

tail = newNode;

}

// Method untuk mencari sebuah simpul tertentu

Node\* search(int value) {

Node\* current = head;

while (current != nullptr) {

if (current->data == value)

return current;

current = current->next;

}

return nullptr;

}

// Method untuk menghapus simpul tertentu

void remove(int value) {

Node\* nodeToDelete = search(value);

if (nodeToDelete == nullptr) {

cout << "Simpul dengan nilai " << value << " tidak ditemukan." << endl;

return;

}

if (nodeToDelete->prev != nullptr)

nodeToDelete->prev->next = nodeToDelete->next;

else

head = nodeToDelete->next;

if (nodeToDelete->next != nullptr)

nodeToDelete->next->prev = nodeToDelete->prev;

else

tail = nodeToDelete->prev;

delete nodeToDelete;

cout << "Simpul dengan nilai " << value << " berhasil dihapus." << endl;

}

// Method untuk menghapus semua simpul

void removeAll() {

Node\* current = head;

while (current != nullptr) {

Node\* temp = current;

current = current->next;

delete temp;

}

head = nullptr;

tail = nullptr;

cout << "Semua simpul berhasil dihapus." << endl;

}

// Method untuk menampilkan data

void display() {

Node\* current = head;

cout << "Isi Double Linked List: ";

while (current != nullptr) {

cout << current->data << " ";

current = current->next;

}

cout << endl;

}

};

int main() {

DoubleLinkedList dll;

int choice, value;

do {

cout << "\nMenu:\n";

cout << "1. Menyisipkan sebagai simpul ujung awal\n";

cout << "2. Menyisipkan sebagai simpul terakhir\n";

cout << "3. Mencari sebuah simpul tertentu\n";

cout << "4. Menghapus simpul tertentu\n";

cout << "5. Menghapus semua simpul\n";

cout << "6. Menampilkan data\n";

cout << "7. Keluar\n";

cout << "Pilihan Anda: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Masukkan nilai yang ingin disisipkan: ";

cin >> value;

dll.insertAtBeginning(value);

break;

case 2:

cout << "Masukkan nilai yang ingin disisipkan: ";

cin >> value;

dll.insertAtEnd(value);

break;

case 3:

cout << "Masukkan nilai yang ingin dicari: ";

cin >> value;

if (dll.search(value) != nullptr)

cout << "Simpul dengan nilai " << value << " ditemukan." << endl;

else

cout << "Simpul dengan nilai " << value << " tidak ditemukan." << endl;

break;

case 4:

cout << "Masukkan nilai yang ingin dihapus: ";

cin >> value;

dll.remove(value);

break;

case 5:

dll.removeAll();

break;

case 6:

dll.display();

break;

case 7:

cout << "Terima kasih!" << endl;

break;

default:

cout << "Pilihan tidak valid. Silakan coba lagi." << endl;

}

} while (choice != 7);

return 0;

}

