

LAPORAN PRAKTIKUM
Modul 06
“DOUBLE LINKED LIST (BAGIAN PERTAMA)”



Disusun Oleh:
M.Faris (2311104017)
SE-07-01

Dosen :
Yudha Islami Sulistiya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO
2024

A. Soal Tugas Pendahuluan

1. Buatlah program yang mengizinkan pengguna menambahkan elemen ke dalam Doubly Linked List di awal dan di akhir list

Code :

```
* Soal_no1.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node *prev;
7      Node *next;
8  };
9  class doubleLinkedList {
10 public:
11     Node *head;
12     Node *tail;
13     doubleLinkedList() : head(nullptr), tail(nullptr){}
14     void insertFirst(int value){
15         Node * newNode = new Node();
16         newNode->data = value;
17         newNode->prev = nullptr;
18         newNode->next = head;
19
20         if (head == nullptr){
21             tail = newNode;
22         } else {
23             head->prev = newNode;
24         }
25         head = newNode;
26     }
27     void insertLast(int value){
28         Node *newNode = new Node();
29         newNode->data = value;
30         newNode->next = nullptr;
31         newNode->prev = tail;
32
33         if(tail == nullptr){
34             head = newNode;
35         } else {
36             tail->next = newNode;
37         }
38         tail = newNode;
39     }
}
```

```

void tampil(){
    if (head == nullptr){
        cout << "Daftar kosong\n";
        return;
    }
    Node *temp = head;
    cout << "Daftar anggota List: ";
    while (temp != nullptr){
        cout << temp->data;
        if (temp->next != nullptr){
            cout << "<->";
        }
        temp = temp->next;
    }
    cout << endl;
}

};

int main(){
    doubleLinkedList dll;

    int elemenPertama, elemenAwal, elemenAkhir;

    cout << "Masukan elemen pertama: ";
    cin >> elemenPertama;
    dll.insertLast(elemenPertama);

    cout << "Masukan elemen kedua diawal: ";
    cin >> elemenAwal;
    dll.insertFirst(elemenAwal);

    cout << "Masukan elemen ketiga diakhir: ";
    cin >> elemenAkhir;
    dll.insertLast(elemenAkhir);

    dll.tampil();

    return 0;
}

```

Output:

```

Masukan elemen pertama: 10
Masukan elemen kedua diawal: 5
Masukan elemen ketiga diakhir: 20
Daftar anggota List: 5<->10<->20

```

Penjelasan :

Code tersebut adalah double linked list (DLL) yang mengimplementasikan InsertFirst untuk menambahkan elemen baru di awal dan insertLast untuk menambahkannya diakhir. Metode tampil() digunakan untuk menampilkan semua elemen dalam list dengan format yang menunjukkan hubungan antar node.

Pad fungsi main ,user diminta untuk memasukkan tiga elemen, satu di akhir list, satu di awal dan satu lagi di akhir. Lalu memanggil metode tampil() untuk menampilkan isi list.

2. Buatlah program yang memungkinkan pengguna untuk menghapus elemen pertama dan elemen terakhir dalam Doubly Linked List.

Code:

```
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node *prev;
    Node *next;
};
class doubleLinkedList {
public:
    Node *head;
    Node *tail;
    doubleLinkedList() : head(nullptr), tail(nullptr){}
    void insertLast(int value){
        Node *newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        newNode->prev = tail;

        if(tail == nullptr){
            head = newNode;
        } else {
            tail->next = newNode;
        }
        tail = newNode;
    }
    void deletefirst(){
        if (head == nullptr){
            cout << "daftar kosogn tidak daapat menghapus";
            return;
        }
        Node *temp = head;
        if (head == tail){
            head = tail = nullptr;
        } else {
            head = head->next;
            head->prev = nullptr;
        }
        delete temp;
    }
}
```

```

void deletelast(){
    if (tail == nullptr){
        cout << "Daftar ksoogn, tidak dapat menghapus";
        return;
    }
    Node *temp = tail;
    if (head == tail){
        head = tail = nullptr;
    } else {
        tail = tail->prev;
        tail->next = nullptr;
    }
    delete temp;
}

void tampil(){
    if (head == nullptr){
        cout << "Daftar kosong\n";
        return;
    }
    Node *temp = head;
    cout << "Daftar anggota List: ";
    while (temp != nullptr){
        cout << temp->data;
        if (temp->next != nullptr){
            cout << "<->";
        }
        temp = temp->next;
    }
    cout << endl;
}
};

```

```

int main(){
    doubleLinkedList dll;
    int elemenPertama, elemenKedua, elemenKetiga;
    cout << "Masukan elemen pertama: ";
    cin >> elemenPertama;
    dll.insertLast(elemenPertama);

    cout << "Masukan elemen kedua diawal: ";
    cin >> elemenKedua;
    dll.insertLast(elemenKedua);

    cout << "Masukan elemen ketiga diakhir: ";
    cin >> elemenKetiga;
    dll.insertLast(elemenKetiga);

    dll.tampil();
    dll.deletefirst();
    dll.deletelast();

    cout << "daftar setelah penghapusan";
    dll.tampil();

    return 0;
}

```

Output:

```
Masukan elemen pertama: 10
Masukan elemen kedua diawal: 15
Masukan elemen ketiga diakhir: 20
Daftar anggota List: 10<->15<->20
daftar setelah penghapusanDaftar anggota List: 15
Press any key to continue
```

Penjelasan :

Code tersebut adalah double linked list (DLL) yang mengimplementasikan fungsi delete first untuk menghapus elemen pertama dari list ,jika list kosong maka penghapusan tidak bisa dilakukan. Dan delete last untuk menghapus elemen terakhir dari list. Di program juga ada metode insertLast untuk menambahkan elemen baru di akhir list dan metode tampil () untuk menampilkan semua elemen dalam list

Pada fungsi main user diminta untuk memasukkan tiga elemen. Lalu program memanggil deleteFirst dan deleteLast untuk menghapus elemen pertama dan terakhir sebelum menampilkan isi list yang telah diperbarui.

3. Buatlah program yang memungkinkan pengguna memasukkan beberapa elemen ke dalam Doubly Linked List. Setelah elemen dimasukkan, tampilkan seluruh elemen dalam list dari depan ke belakang, kemudian dari belakang ke depan.

Code :

```
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node *prev;
    Node *next;
};
class doubleLinkedList {
public:
    Node *head;
    Node *tail;
    doubleLinkedList() : head(nullptr), tail(nullptr){}
    void insertLast(int value){
        Node *newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        newNode->prev = tail;

        if(tail == nullptr){
            head = newNode;
        } else {
            tail->next = newNode;
        }
        tail = newNode;
    }
    void tampilDariDepan(){
        if (head == nullptr){
            cout << "daftar kosong";
            return;
        }
        Node *temp = head;
        cout << "Daftar elemen dari depan ke belakang";
        while (temp != nullptr){
            cout << temp->data;
            if(temp->next != nullptr){
                cout << " <-> ";
            }
            temp = temp->next;
        }
        cout << endl;
    }
};
```

```

void tampilkanBelakang(){
    if (tail == nullptr){
        cout << "daftar kosong";
        return;
    }
    Node *temp = tail;
    cout << "Daftar elemen dari belakang ke depan: ";
    while (temp != nullptr){
        cout << temp->data;
        if(temp->prev != nullptr){
            cout << "<->";
        }
        temp = temp->prev;
    }
    cout << endl;
}

};

int main(){
    doubleLinkedList dll;
    int elemen;

    for (int i = 1; i <= 4; i++){
        cout << "Masukan elemen ke-" << i << ": ";
        cin >> elemen;
        dll.insertLast(elemen);
    }

    dll.tampilkanDariDepan();

    dll.tampilkanBelakang();

    return 0;
}

```

Output:

```

Masukan elemen ke-1:1
Masukan elemen ke-2:2
Masukan elemen ke-3:3
Masukan elemen ke-4:4
Daftar elemen dari depan ke belakang: 1<->2<->3<->4
Daftar elemen dari belakang ke depan: 4<->3<->2<->1
Press any key to continue

```

Penjelasan:

Code tersebut adalah program double linked list (DLL) untuk menampilkan list dari depan ke belakang maupun belakang ke depan. Pertama ada metode insertLast untuk menambahkan elemen baru di akhir list. Metode tampilkanDariDepan() untuk menampilkan elemen list dari depan ke belakang, jika list kosong maka akan mencetak list kosong. Metode tampilkanBelakang() untuk menampilkan elemen dari belakang ke depan.

Pada fungsi main, program akan meminta user untuk memasukkan empat elemen kemudian ditambahkan ke list. Setelah itu metode tampilkanDariDepan() dan metode tampilkanBelakang() dipanggil untuk menampilkan isi list dari kedua arah.