

LAPORAN PRAKTIKUM
PERTEMUAN 6
MODUL 6 STD



Nama :

Farrel I'zaz Yuwono (2311104014)

Dosen :

YUDHA ISLAMI SULISTYA

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tugas Pendahuluan

1. Soal 1

```

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node* prev;
};

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

    void insertFirst_2311104014(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;
    }

    void insertLast_2311104014(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;
    }

    void displayForward_2311104014() {
        Node* current = head;
        while (current != nullptr) {
            cout << current->data;
            if (current->next != nullptr) {
                cout << " <-> ";
            }
            current = current->next;
        }
        cout << endl;
    }
};

int main() {
    DoublyLinkedList dll;

    int elemen;
    cout << "Masukkan elemen pertama = ";
    cin >> elemen;
    dll.insertFirst_2311104014(elemen);

    cout << "Masukkan elemen kedua di awal = ";
    cin >> elemen;
    dll.insertFirst_2311104014(elemen);

    cout << "Masukkan elemen ketiga di akhir = ";
    cin >> elemen;
    dll.insertLast_2311104014(elemen);

    cout << "DAFTAR ANGGOTA LIST: ";
    dll.displayForward_2311104014();

    return 0;
}
```

Output :

```
P_Soal_1 }  
Masukkan elemen pertama = 1  
Masukkan elemen kedua di awal = 2  
Masukkan elemen ketiga di akhir = 3  
DAFTAR ANGGOTA LIST: 2 <-> 1 <-> 3
```

2. Soal 2

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node* prev;
};

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

    void insertLast_2311104014(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;
    }

    void deleteFirst_2311104014() {
        if (head == nullptr) {
            cout << "List kosong, tidak ada elemen yang dihapus." << endl;
            return;
        }
        Node* temp = head;
        head = head->next;
        if (head != nullptr) {
            head->prev = nullptr;
        } else {
            tail = nullptr;
        }
        delete temp;
    }

    void deleteLast_2311104014() {
        if (tail == nullptr) {
            cout << "List kosong, tidak ada elemen yang dihapus." << endl;
            return;
        }
        Node* temp = tail;
        tail = tail->prev;
        if (tail != nullptr) {
            tail->next = nullptr;
        } else {
            head = nullptr;
        }
        delete temp;
    }

    void displayForward_2311104014() {
        Node* current = head;
        while (current != nullptr) {
            cout << current->data;
            if (current->next != nullptr) {
                cout << " <-> ";
            }
            current = current->next;
        }
        cout << endl;
    }
};

int main() {
    DoublyLinkedList dll;

    int elemen;
    cout << "Masukkan elemen pertama = ";
    cin >> elemen;
    dll.insertLast_2311104014(elemen);

    cout << "Masukkan elemen kedua di akhir = ";
    cin >> elemen;
    dll.insertLast_2311104014(elemen);

    cout << "Masukkan elemen ketiga di akhir = ";
    cin >> elemen;
    dll.insertLast_2311104014(elemen);

    cout << "DAFTAR ANGGOTA LIST: ";
    dll.displayForward_2311104014();

    dll.deleteFirst_2311104014();
    dll.deleteLast_2311104014();

    cout << "DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: ";
    dll.displayForward_2311104014();

    return 0;
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\06_Double_Linked_List_Bagian_1\TP> cd "d
++ TP_Soal_2.cpp -o TP_Soal_2 } ; if ($?) { .\TP_Soal_2 }
Masukkan elemen pertama = 1
Masukkan elemen kedua di akhir = 2
Masukkan elemen ketiga di akhir = 3
DAFTAR ANGGOTA LIST: 1 <-> 2 <-> 3
DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 2
PS D:\kuliah\strukturdat\tugas kuliah\06_Double_Linked_List_Bagian_1\TP> █
```

3. Soal 3

```

    ● ● ●

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
    Node* prev;
};

class DoublyLinkedList {
private:
    Node* head;
    Node* tail;

public:
    DoublyLinkedList() {
        head = nullptr;
        tail = nullptr;
    }

    void insertLast_2311104014(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;
    }

    void displayForward_2311104014() {
        Node* current = head;
        while (current != nullptr) {
            cout << current->data;
            if (current->next != nullptr) {
                cout << " <-> ";
            }
            current = current->next;
        }
        cout << endl;
    }

    void displayBackward_2311104014() {
        Node* current = tail;
        while (current != nullptr) {
            cout << current->data;
            if (current->prev != nullptr) {
                cout << " <-> ";
            }
            current = current->prev;
        }
        cout << endl;
    }
};

int main() {
    DoublyLinkedList dll;

    int elemen;
    for (int i = 1; i <= 4; i++) {
        cout << "Masukkan elemen ke-" << i << ": ";
        cin >> elemen;
        dll.insertLast_2311104014(elemen);
    }

    cout << "Daftar elemen dari depan ke belakang: ";
    dll.displayForward_2311104014();

    cout << "Daftar elemen dari belakang ke depan: ";
    dll.displayBackward_2311104014();

    return 0;
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\06_Double_Linked_List_Bagian_1\TP> cd
++ TP_Soal_3.cpp -o TP_Soal_3 } ; if ($?) { .\TP_Soal_3 }
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Daftar elemen dari depan ke belakang: 1 <-> 2 <-> 3 <-> 4
Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1
PS D:\kuliah\strukturdat\tugas kuliah\06_Double_Linked_List_Bagian_1\TP> |
```

Penjelasan TP

1. Soal 1

- Struktur Node: Struktur Node digunakan untuk merepresentasikan elemen dari linked list. Setiap Node memiliki data (int data) dan pointer (next dan prev) yang menunjuk ke node berikutnya dan sebelumnya.
- Class DoublyLinkedList: Kelas DoublyLinkedList memiliki dua pointer (head dan tail) yang menunjuk ke awal dan akhir dari list.
- Method insertFirst_2311104014: Metode ini menambahkan elemen baru di awal list. Jika list kosong, elemen ini akan menjadi tail juga.
- Method insertLast_2311104014: Metode ini menambahkan elemen baru di akhir list. Jika list kosong, elemen ini akan menjadi head juga.
- Method displayForward_2311104014: Metode ini menampilkan semua elemen dari awal hingga akhir list.
- Function main: Dalam main, list dibuat, elemen ditambahkan di awal dan di akhir, lalu list ditampilkan.

2. Soal 2

- Struktur Node: Struktur Node memiliki data (int data) dan pointer (next dan prev) untuk menunjuk ke node berikutnya dan sebelumnya dalam list.
- Class DoublyLinkedList: Kelas ini memiliki pointer head dan tail yang menunjuk ke awal dan akhir list. Ada beberapa fungsi utama dalam kelas ini:

- insertLast_2311104014(int value): Menambahkan elemen baru di akhir list. Jika list kosong, elemen ini akan menjadi head.
- deleteFirst_2311104014(): Menghapus elemen pertama. Jika list hanya memiliki satu elemen, list menjadi kosong setelah penghapusan.
- deleteLast_2311104014(): Menghapus elemen terakhir. Jika list hanya memiliki satu elemen, list juga akan kosong setelah penghapusan.
- displayForward_2311104014(): Menampilkan elemen dari awal hingga akhir list dengan format elemen <-> elemen.
- Function main: Dalam main, list dibuat dan tiga elemen ditambahkan di akhir. Setelah ditampilkan, elemen pertama dan terakhir dihapus, lalu list ditampilkan kembali untuk menunjukkan hasil setelah penghapusan.

3. Soal 3

- Struktur Node: Struktur Node berisi data (int data) dan dua pointer (next dan prev) yang menunjuk ke node berikutnya dan sebelumnya dalam list.
- Class DoublyLinkedList:
 - insertLast_2311104014(int value): Menambahkan elemen baru di akhir list. Jika list kosong, elemen ini menjadi head.
 - displayForward_2311104014(): Menampilkan elemen dari awal hingga akhir list dalam format elemen <-> elemen.
 - displayBackward_2311104014(): Menampilkan elemen dari akhir hingga awal list dalam format elemen <-> elemen.
- Function main: Di dalam fungsi main, program meminta pengguna untuk memasukkan empat elemen yang akan ditambahkan ke akhir list. Kemudian program menampilkan list dalam dua arah: dari depan ke belakang dan dari belakang ke depan.