# TUGAS PENDAHULUAN MODUL 6

"Single Linked List (Bagian Kedua)"



Disusun Oleh: Isabelle Putri Ardini - 2311104030 SE-07-01

Dosen:

Yudha Islami Sulistya

PROGRAM STUDI SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

#### Soal Tugas Pendahuluan

Soal 1: Menambahkan Elemen di Awal dan Akhir DLL

#### **Deskripsi Soal:**

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

#### Instruksi:

- 1. Implementasikan fungsi 'insertFirst' untuk menambahkan elemen di awal list.
- 2. Implementasikan fungsi 'insertLast' untuk menambahkan elemen di akhir list.
- 3. Tampilkan seluruh elemen dalam list dari depan ke belakang setelah penambahan dilakukan

#### **Contoh Input:**

- Input: Masukkan elemen pertama = 10
- Input: Masukkan elemen kedua di awal = 5
- Input: Masukkan elemen ketiga di akhir = 20

### **Output:**

- DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20

#### Program:

```
#include "DoublyLinkedList.cpp"

int main() {
    DoublyLinkedList dll;

    int elemenPertama = 10;
    int elemenKedua = 5;
    int elemenKetiga = 20;

// Tambah elemen pertama di List
    cout << "Input: Masukkan elemen pertama = " << elemenPertama << endl;
    dll.insertLast(elemenPertama);

// Tambah elemen kedua di awal List
    cout << "Input: Masukkan elemen kedua di awal = " << elemenKedua << endl;
    dll.insertFirst(elemenKedua);

// Tambah elemen ketiga di akhir list
    cout << "Input: Masukkan elemen ketiga di akhir = " << elemenKetiga << endl;
    dll.insertLast(elemenKetiga);

// Tampilkan List dari depan ke belakang
    dll.displayForward();

return 0;
}</pre>
```

### **Hasil Running:**

```
Input: Masukkan elemen pertama = 10
Input: Masukkan elemen kedua di awal = 5
Input: Masukkan elemen ketiga di akhir = 20
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20
```

Soal 2: Menghapus Elemen di Awal dan Akhir DLL

## **Deskripsi Soal:**

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

#### Instruksi:

- 1. Implementasikan fungsi 'deleteFirst' untuk menghapus elemen pertama.
- 2. Implementasikan fungsi 'deleteLast' untuk menghapus elemen terakhir.
- 3. Tampilkan seluruh elemen dalam list setelah penghapusan dilakukan.

## **Contoh Input:**

- Input: Masukkan elemen pertama = 10
- Input: Masukkan elemen kedua di akhir = 15
- Input: Masukkan elemen ketiga di akhir = 20
- Hapus elemen pertama dan terakhir.

#### **Output:**

- DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 15

### Program:

```
#include "DoublyLinkedList.cpp"
int main() {
    DoublyLinkedList dll;
    cout << "Input: Masukkan elemen pertama = 10" << endl;</pre>
    dll.insertLast(10);
    cout << "Input: Masukkan elemen kedua di akhir = 15" << endl;</pre>
    dll.insertLast(15);
    cout << "Input: Masukkan elemen ketiga di akhir = 20" << endl;</pre>
    dll.insertLast(20);
    // Hapus elemen pertama
    cout << "Hapus elemen pertama." << endl;</pre>
    dll.deleteFirst();
    // Hapus elemen terakhir
    cout << "Hapus elemen terakhir." << endl;</pre>
    dll.deleteLast();
    // Tampilkan list setelah penghapusan
    dll.displayForward();
    return 0;
```

# **Hasil Running:**

```
Input: Masukkan elemen pertama = 10
Input: Masukkan elemen kedua di akhir = 15
Input: Masukkan elemen ketiga di akhir = 20
Hapus elemen pertama.
Hapus elemen terakhir.
DAFTAR ANGGOTA LIST: 15
```

Soal 3: Menampilkan Elemen dari Depan ke Belakang dan Sebaliknya

**Deskripsi Soal:** 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.

#### **Instruksi:**

- 1. Implementasikan fungsi untuk menampilkan elemen dari depan ke belakang.
- 2. Implementasikan fungsi untuk menampilkan elemen dari belakang ke depan.
- 3. Tambahkan 4 elemen ke dalam list dan tampilkan elemen tersebut dalam dua arah.

#### **Contoh Input:**

- Input: Masukkan 4 elemen secara berurutan: 1, 2, 3, 4

# **Output:**

- Daftar elemen dari depan ke belakang: 1 < -> 2 < -> 3 < -> 4
- Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1

# Program:

```
#include "DoublyLinkedList.cpp"

int main() {
    DoublyLinkedList dll;

    // Tambah 4 elemen ke list
    cout << "Input: Masukkan 4 elemen secara berurutan: 1, 2, 3, 4" << endl;
    dll.insertLast(1);
    dll.insertLast(2);
    dll.insertLast(3);
    dll.insertLast(4);

    // Tampilkan dari depan ke belakang
    dll.displayForward();

    // Tampilkan dari belakang ke depan
    dll.displayBackward();

    return 0;
}</pre>
```

# **Hasil Running:**

```
Input: Masukkan 4 elemen secara berurutan: 1, 2, 3, 4
DAFTAR ANGGOTA LIST: 1 <-> 2 <-> 3 <-> 4
Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1
```