

# **PRAKTIKUM STRUKTUR DATA**

## **TUGAS PENDAHULUAN 6**



**Nama :**

Aulia Ahmad Ghaus Adzam (2311104028)

**Dosen :**

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

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **B. Soal Tugas Pendahuluan**

### **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

```

SOAL_01.cpp

#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_2311104028(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->prev = nullptr;
        newNode->next = head;
        if (head != nullptr) {
            head->prev = newNode;
        }
        head = newNode;
        if (tail == nullptr) {
            tail = newNode;
        }
    }

    void insertLast_2311104028(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        newNode->prev = tail;
        if (tail != nullptr) {
            tail->next = newNode;
        }
        tail = newNode;
        if (head == nullptr) {
            head = newNode;
        }
    }

    void display_2311104028() {
        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() {
    DoublyLinkedList dll;
    dll.insertFirst_2311104028(10);
    dll.insertFirst_2311104028(5);
    dll.insertLast_2311104028(20);
    dll.display_2311104028();
    return 0;
}

```

### Penjelasan:

program Doubly Linked List yang mendukung penambahan elemen di awal dan akhir list. Program ini mendefinisikan struktur Node yang terdiri dari data, next (pointer ke node berikutnya), dan prev (pointer ke node sebelumnya), serta class DoublyLinkedList yang memiliki tiga fungsi utama: insertFirst\_2311104028 untuk menambahkan node di awal list dan menjadikannya head (atau tail jika list kosong), insertLast\_2311104028 untuk menambahkan node di akhir list dan menjadikannya tail (atau head jika list kosong), serta display\_2311104028 yang menampilkan semua elemen list dari depan ke belakang. Pada fungsi main, elemen-elemen ditambahkan di awal dan akhir, lalu ditampilkan hasilnya dengan format terhubung <->.

### Output:

```
ums.uxr --stderr=Microsoft-Runtime-Error-11xy3nir.pic --pid=Microsoft-Runtime-Pid-pzviectq.
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20
PS D:\documents\ngoding\STD_Aulia_Ahmad_Ghaus_Adzam_2311104028\06_Double_Linked_List_Bagian_1>
```

## 2. Menghapus Elemen Di Akhir dan di awal 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

```

SOAL_02.cpp

#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_2311104028(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        newNode->prev = tail;
        if (tail != nullptr) {
            tail->next = newNode;
        }
        tail = newNode;
        if (head == nullptr) {
            head = newNode;
        }
    }

    void deleteFirst_2311104028() {
        if (head == nullptr) return;
        Node* temp = head;
        head = head->next;
        if (head != nullptr) {
            head->prev = nullptr;
        } else {
            tail = nullptr;
        }
        delete temp;
    }

    void deleteLast_2311104028() {
        if (tail == nullptr) return;
        Node* temp = tail;
        tail = tail->prev;
        if (tail != nullptr) {
            tail->next = nullptr;
        } else {
            head = nullptr;
        }
        delete temp;
    }

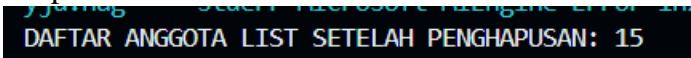
    void display_2311104028() {
        Node* temp = head;
        cout << "DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: ";
        while (temp != nullptr) {
            cout << temp->data;
            if (temp->next != nullptr) {
                cout << " ↔ ";
            }
            temp = temp->next;
        }
        cout << endl;
    }
};

int main() {
    DoublyLinkedList dll;
    dll.insertLast_2311104028(10);
    dll.insertLast_2311104028(15);
    dll.insertLast_2311104028(20);
    dll.deleteFirst_2311104028();
    dll.deleteLast_2311104028();
    dll.display_2311104028();
    return 0;
}

```

**Penjelasan:**

Doubly Linked List ini memungkinkan penghapusan elemen pertama dan terakhir dari list. Program ini memiliki fungsi insertLast\_2311104028 untuk menambahkan elemen di akhir list, deleteFirst\_2311104028 untuk menghapus elemen pertama dengan mengubah head ke node berikutnya, serta deleteLast\_2311104028 untuk menghapus elemen terakhir dengan memindahkan tail ke node sebelumnya. Fungsi display\_2311104028 menampilkan semua elemen yang tersisa dalam list dari depan ke belakang setelah proses penghapusan. Di bagian main, list diisi dengan tiga elemen, lalu elemen pertama dan terakhir dihapus, kemudian hasil akhirnya ditampilkan dengan format terhubung <->.

**Output:**

```
DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 15
```

**3. Menampilkan Elemen dari depan 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

```

SOAL_03.cpp

#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_2311104028(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        newNode->prev = tail;
        if (tail != nullptr) {
            tail->next = newNode;
        }
        tail = newNode;
        if (head == nullptr) {
            head = newNode;
        }
    }

    void displayForward_2311104028() {
        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 displayBackward_2311104028() {
        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() {
    DoublyLinkedList dll;
    dll.insertLast_2311104028(1);
    dll.insertLast_2311104028(2);
    dll.insertLast_2311104028(3);
    dll.insertLast_2311104028(4);
    dll.displayForward_2311104028();
    dll.displayBackward_2311104028();
    return 0;
}

```

**Penjelasan:**

Struktur Node digunakan untuk menyimpan data dan pointer next serta prev. Fungsi insertLast\_2311104028 menambahkan elemen di akhir list, displayForward\_2311104028 menampilkan elemen dari depan ke belakang dengan memulai dari head, sedangkan displayBackward\_2311104028 menampilkan elemen dari belakang ke depan dengan memulai dari tail. Pada bagian main, empat elemen ditambahkan secara berurutan, dan hasil list ditampilkan dalam dua arah untuk menunjukkan bahwa Doubly Linked List mendukung akses dua arah.

Output:

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

**C. Latihan Unguided**

1. Buatlah program di C++ yang mengelola daftar buku menggunakan Double Linked List. Setiap node dalam list akan menyimpan informasi mengenai buku berupa :

- ID Buku
- Judul Buku
- Penulis Buku.

Program harus dapat melakukan penambahan buku di akhir list, dan menampilkan daftar buku dari awal ke akhir maupun dari akhir ke awal.

Spesifikasi Program:

1. Buatlah struktur Node untuk menyimpan data buku dengan elemen:

- ID Buku (integer)
- Judul Buku (string)
- Penulis Buku (string)

2. Implementasikan fungsi untuk:

- Menambahkan buku di akhir linked list.
- Menampilkan semua buku dari awal ke akhir.
- Menampilkan semua buku dari akhir ke awal.



```

LATIHAN_01.cpp

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

struct Node {
    int id;
    string judul;
    string penulis;
    Node* next;
    Node* prev;
};

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

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

    void insertLast_2311104028(int id, string judul, string penulis) {
        Node* newNode = new Node();
        newNode->id = id;
        newNode->judul = judul;
        newNode->penulis = penulis;
        newNode->next = nullptr;
        newNode->prev = tail;
        if (tail != nullptr) {
            tail->next = newNode;
        }
        tail = newNode;
        if (head == nullptr) {
            head = newNode;
        }
    }

    void displayForward_2311104028() {
        Node* temp = head;
        cout << "Daftar Buku (Dari Awal ke Akhir):" << endl;
        while (temp != nullptr) {
            cout << "ID Buku: " << temp->id << ", Judul: " << temp->judul << ", Penulis: " << temp->penulis << endl;
            temp = temp->next;
        }
    }

    void displayBackward_2311104028() {
        Node* temp = tail;
        cout << "Daftar Buku (Dari Akhir ke Awal):" << endl;
        while (temp != nullptr) {
            cout << "ID Buku: " << temp->id << ", Judul: " << temp->judul << ", Penulis: " << temp->penulis << endl;
            temp = temp->prev;
        }
    }
};

int main() {
    DoublyLinkedList dll;
    int id;
    string judul, penulis;
    char choice;

    do {
        cout << "Masukkan ID Buku: ";
        cin >> id;
        cin.ignore();
        cout << "Masukkan Judul Buku: ";
        getline(cin, judul);
        cout << "Masukkan Penulis Buku: ";
        getline(cin, penulis);

        dll.insertLast_2311104028(id, judul, penulis);

        cout << "Tambah buku lain? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');

    cout << endl;
    dll.displayForward_2311104028();
    cout << endl;
    dll.displayBackward_2311104028();

    return 0;
}

```

**Penjelasan:**

Struct Node menyimpan data Dari Buku berupa int id, String judul, String Penulis Pointer Next dan prev, lalu setelah itu ada Void insertlast untuk menambahkan buku di akhir list dan void displayforward untuk men cout dari awal ke akhir dan void displaybakeward sebaliknya

**Output:**

```
Masukkan ID Buku: 1
Masukkan Judul Buku: Alan Wake
Masukkan Penulis Buku: Alan
Tambah buku lain? (y/n): n

Daftar Buku (Dari Awal ke Akhir):
ID Buku: 1, Judul: Alan Wake, Penulis: Alan

Daftar Buku (Dari Akhir ke Awal):
ID Buku: 1, Judul: Alan Wake, Penulis: Alan
PS D:\documents\ngoding\STD_Aulia_Ahmad_Ghaus_Adzam_2311104028\06_Double_Linked_List_Bagian_1> []
```