

Tugas Proceeding
Modul 6 Struktur Data
DOUBLE LINKED LIST BAGIAN 1



Disusun Oleh:
Christian Felix Saliman Sugiono (2311104031)
S1SE0701

Dosen:
Yudha Islami Sulistya
Program Studi S1 Software Engineering
Fakultas Informatika
Telkom University
Purwokerto
2024

Tugas Pendahuluan Modul 6
STRUKTUR DATA - Ganjil 2024/2025
" DOUBLE LINKED LIST BAGIAN 1 "

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 4 adalah Senin, 9 Oktober 2023 pukul 06.00 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan:
TP_MOD_[XX]_NIM_NAMA.pdf

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

B. Unguided Modul 6

1 Library Buku, kodingan dari persoalan ini adalah sebagai berikut ini:

```
#include <iostream>
#include <string>

using namespace std;

struct Node {
    int idBuku;
    string judulBuku;
    string penulisBuku;
    Node* prev;
    Node* next;
};

Node* createNode_2311104031(int id, string judul, string penulis) {
    Node* newNode = new Node();
    newNode->idBuku = id;
    newNode->judulBuku = judul;
    newNode->penulisBuku = penulis;
    newNode->prev = nullptr;
    newNode->next = nullptr;
    return newNode;
}

void tambahBuku_2311104031(Node* head, int id, string judul, string penulis) {
    Node* newNode = createNode_2311104031(id, judul, penulis);
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
    cout << "Buku berhasil ditambahkan!" << endl;
}

void tampilkanDariAwal_2311104031(Node* head) {
    if (head == nullptr) {
        cout << "Daftar buku kosong." << endl;
        return;
    }

    Node* temp = head;
    cout << "Daftar Buku (Dari Awal ke Akhir):" << endl;
    while (temp != nullptr) {
        cout << "ID Buku: " << temp->idBuku << ", Judul Buku: " << temp->judulBuku
            << ", Penulis: " << temp->penulisBuku << endl;
        temp = temp->next;
    }
}

void tampilkanDariAkhir_2311104031(Node* head) {
    if (head == nullptr) {
        cout << "Daftar buku kosong." << endl;
        return;
    }

    Node* temp = head;
    while (temp->next != nullptr) {
        temp = temp->next;
    }

    cout << "Daftar Buku (Dari Akhir ke Awal):" << endl;
    while (temp != nullptr) {
        cout << "ID Buku: " << temp->idBuku << ", Judul Buku: " << temp->judulBuku
            << ", Penulis: " << temp->penulisBuku << endl;
        temp = temp->prev;
    }
}

int main() {
    Node* head = nullptr;
    int pilihan, id;
    string judul, penulis;

    do {
        cout << "\nMenu: " << endl;
        cout << "1. Tambah Buku" << endl;
        cout << "2. Tampilkan Daftar Buku (Awal ke Akhir)" << endl;
        cout << "3. Tampilkan Daftar Buku (Akhir ke Awal)" << endl;
        cout << "4. Keluar" << endl;
        cout << "Pilih opsi: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Masukkan ID Buku: ";
                cin >> id;
                cin.ignore();
                cout << "Masukkan Judul Buku: ";
                getline(cin, judul);
                cout << "Masukkan Penulis Buku: ";
                getline(cin, penulis);
                tambahBuku_2311104031(head, id, judul, penulis);
                break;
            case 2:
                tampilkanDariAwal_2311104031(head);
                break;
            case 3:
                tampilkanDariAkhir_2311104031(head);
                break;
            case 4:
                cout << "Keluar dari program." << endl;
                break;
            default:
                cout << "Pilihan tidak valid!" << endl;
        }
    } while (pilihan != 4);

    return 0;
}
```

sama dengan single linked list, hal pertama yang perlu kita lakukan adalah untuk membuat struct node yang dimana struct ini berisi judul buku, penulis buku dan id buku, lalu kita akan membuat method untuk create node yang dimana method ini berfungsi untuk membuat node baru untuk setiap struct buku yang dibuat oleh user, dan node ini dapat berjumlah sesuai keinginan user, lalu kita akan membuat method untuk menambahkan buku kedalam list kita yang dimana kita menggunakan metode insert last agar urutan buku yang di insert sesuai ordenya, kemudian kita akan membuat method tampilBuku untuk mengeprint buku, di dalam sini kita membuat 2 method yakni tampil dari awal -> ahir dan ahir -> awal sesuai dengan keinginan user ia ingin menggunakan output yang mana. Kemudian kkita akan masuk kedalam main programnya, yang dimana kita mengset head dengan nullptr sebagai deklarasi list kosong, baru kemudian kita membuat deklarasi string judul, dan penulis dan int pilihan (untuk menu) dan id bukunya, untuk menunya kita akan menampilkan 4 pilihan yakni menambah buku, tampil dari awal ke ahir, tampil ahir ke awal dan end program/keluar program program akan berjalan sesuai pilihan dari user, dan kita akan menggunakan switch case untuk memasukkan pilihan ke method yang sesuai dengan menu yang telah kita buat, untuk hasil runningnya dalah sebagai berikut ini:

```
Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 1
Masukkan ID Buku: 44105
Masukkan Judul Buku: Lord Of Rings
Masukkan Penulis Buku: Jf Kendrick
Buku berhasil ditambahkan!

Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 1
Masukkan ID Buku: 33015
Masukkan Judul Buku: Friendship over War
Masukkan Penulis Buku: Ruth shroedinger
Buku berhasil ditambahkan!

Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 1
Masukkan ID Buku: 11903
Masukkan Judul Buku: Capitalism key to wealth
Masukkan Penulis Buku: Aventureline The Augin
Buku berhasil ditambahkan!

Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 2
Daftar Buku (Dari Awal ke Akhir):
ID Buku: 44105, Judul Buku: Lord Of Rings, Penulis: Jf Kendrick
ID Buku: 33015, Judul Buku: Friendship over War, Penulis: Ruth shroedinger
ID Buku: 11903, Judul Buku: Capitalism key to wealth, Penulis: Aventureline The Augin

Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 3
Daftar Buku (Dari Akhir ke Awal):
ID Buku: 11903, Judul Buku: Capitalism key to wealth, Penulis: Aventureline The Augin
ID Buku: 33015, Judul Buku: Friendship over War, Penulis: Ruth shroedinger
ID Buku: 44105, Judul Buku: Lord Of Rings, Penulis: Jf Kendrick

Menu:
1. Tambah Buku
2. Tampilkan Daftar Buku (Awal ke Akhir)
3. Tampilkan Daftar Buku (Akhir ke Awal)
4. Keluar
Pilih opsi: 4
Keluar dari program.
```

TP modul 6

1 Penambahan Elm pada awal dan ahir DLL, kodingan dari permintaan soal adalah sebagai berikut ini:

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

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

Node* createNode_2311104031(int data) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->prev = nullptr;
    newNode->next = nullptr;
    return newNode;
}

void insertFirst_2311104031(Node*& head, int data) {
    Node* newNode = createNode_2311104031(data);
    if (head == nullptr) {
        head = newNode;
    } else {
        newNode->next = head;
        head->prev = newNode;
        head = newNode;
    }
}

void insertLast_2311104031(Node*& head, int data) {
    Node* newNode = createNode_2311104031(data);
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
}

void display_2311104031(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data;
        if (temp->next != nullptr)
            cout << " <-> ";
        temp = temp->next;
    }
    cout << endl;
}

int main() {
    Node* head = nullptr;
    insertFirst_2311104031(head, 10);
    insertFirst_2311104031(head, 5);
    insertLast_2311104031(head, 20);

    cout << "DAFTAR ANGGOTA LIST: ";
    display_2311104031(head);

    return 0;
}
```

sama dengan single linked list untuk permulaan kita akan membuat struct sebagai tempat penampungan data dari list yang akan kita buat, kemudian kita akan membuat method untuk create node, yakni pembuatan node baru untuk setiap input yang akan kita lakukan dengan jumlahnya yakni sebanyak x (inputan data), kemudian kita akan membuat 2 inputan untuk list ini yakni input dari first dan insert dari last, yang dimana node yang dibuat akan dimasukan dari depan atau dari belakang, sesuai dengan method yang akan kita gunakan pada main, setelah pembuatan method untuk insert, langkah berikutnya adalah untuk print hasil dari DLL kita dengan method display yang dimana ia akan mengeprint semua elemen yang terdapat pada DLL, setelah pembuatan method display langkah berikutnya adalah untuk membuat kelas main sebagai tempat penggunaan method yang baru kita buat, seperti biasa kita akan mengset head dengan nilai nullptr sebagai deklarasi bahwa list kosong kemudian baru kita akan membuat insert dengan penggunaan kedua method kita yang di kodingan daya mencontohkan 3 bilangan 10, 5, dan 20 yang dimana 10 dan 5 di insert first dan 20 di insert ke last, kemudian kita mengahiri kodingan dengan mengeprint isi dari list yang outputnya seperti berikut

```
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20
PS C:\Users\jerry\OneDrive\Documents\
```

2 Penghapusan Elm dari awal dan ahir DLL, untuk kodinganya dapat dilihat seperti berikut ini:

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

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

Node* createNode_2311104031(int data) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->prev = nullptr;
    newNode->next = nullptr;
    return newNode;
}

void insertLast_2311104031(Node*& head, int data) {
    Node* newNode = createNode_2311104031(data);
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
}

void deleteFirst_2311104031(Node*& head) {
    if (head == nullptr) return;
    Node* temp = head;
    if (head->next != nullptr) {
        head = head->next;
        head->prev = nullptr;
    } else {
        head = nullptr;
    }
    delete temp;
}

void deleteLast_2311104031(Node*& head) {
    if (head == nullptr) return;
    Node* temp = head;
    if (temp->next == nullptr) {
        head = nullptr;
        delete temp;
    } else {
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->prev->next = nullptr;
        delete temp;
    }
}

void display_2311104031(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data;
        if (temp->next != nullptr)
            cout << " <-> ";
        temp = temp->next;
    }
    cout << endl;
}

int main() {
    Node* head = nullptr;
    insertLast_2311104031(head, 10);
    insertLast_2311104031(head, 15);
    insertLast_2311104031(head, 20);

    deleteFirst_2311104031(head);
    deleteLast_2311104031(head);

    cout << "DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: ";
    display_2311104031(head);

    return 0;
}
```

Sebagai langkah awal untuk memulai DLL adalah untuk membuat struct node sebagai tempat penyimpanan data yang akan dimasukkan kedalam list kemudian kita akan membuat method create node sebagai deklarasi pembuatan node dari tiap variabel yang akan kita buat dengan struct, lalu kita masuk ke dalam method insert last agar inputnyaurut sesuai inputan yang kita buat dan bukan terbalik, setelah pembuatan insert kita akan membuat 2 macam penghapusan yakni delete first dan delete last untuk penghapusan elemen dari list, lalu kita akan membuat fungsi untuk print list sebagai method output dari kodingan ini, kemudian kita akan masuk ke dalam main untuk menjalankan kodingan kita ini, di dalam main ini kita akan mendeklarasikan head sebagai null untuk membuat list sebagai kosong (empty), lalu baru kita memasukan elemen elemen kita dengan method insert, dari saya memilih 3 nilai yakni 10,15,20, kemudian kita akan langsung menghapus nilai pertama dan terakhir menggunakan method delete first dan last, dan untuk melihat hasil kita akan menggunakan method display untuk menunjukkan outputnya, yakni sebagai berikut ini:

DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 15
PS C:\Users\jerry\OneDrive\Documents\Belajar

3 Penampilan elemen dari awal -> ahir dan sebaliknya, kodingan dari soal ini adalah sebagai berikut:

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

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

Node* createNode_2311104031(int data) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->prev = nullptr;
    newNode->next = nullptr;
    return newNode;
}

void insertLast_2311104031(Node*& head, int data) {
    Node* newNode = createNode_2311104031(data);
    if (head == nullptr) {
        head = newNode;
    } else {
        Node* temp = head;
        while (temp->next != nullptr) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->prev = temp;
    }
}

void displayForward_2311104031(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data;
        if (temp->next != nullptr)
            cout << " <-> ";
        temp = temp->next;
    }
    cout << endl;
}

void displayBackward_2311104031(Node* head) {
    if (head == nullptr) return;

    Node* temp = head;
    while (temp->next != nullptr) {
        temp = temp->next;
    }

    while (temp != nullptr) {
        cout << temp->data;
        if (temp->prev != nullptr)
            cout << " <-> ";
        temp = temp->prev;
    }
    cout << endl;
}

int main() {
    Node* head = nullptr;
    insertLast_2311104031(head, 1);
    insertLast_2311104031(head, 2);
    insertLast_2311104031(head, 3);
    insertLast_2311104031(head, 4);

    cout << "Daftar elemen dari depan ke belakang: ";
    displayForward_2311104031(head);

    cout << "Daftar elemen dari belakang ke depan: ";
    displayBackward_2311104031(head);

    return 0;
}
```

Sama dengan kedua kodingan sebelumnya, kita akan deklarasi struct terlebih dahulu sebagai permulaan dari list, yang kemudian akan di follow up dengan pembuatan method untuk create node yang dimana create node ini akan digunakan sebagai pembuatan node dari setiap data yang akan di insert pada list, lalu kita akan membuat method untuk insert elemen ke dalam list menggunakan insertlast agar data urut sesuai orde dari pemasukanya, langkah berikutnya adalah membuat 2 macam display yakni dari awal ke ahir dan ahir ke awal, yang dimana keduanya memiliki string node yang berbeda beda sesuai dengan apa yang diminta oleh methodnya, kemudian kita akan masuk ke dalam main yang dimana kita akan menggunakan method yang telah kita buat untuk menjalankan kodingan display, sebagai awal kita akan mengset head dengan null ptr sebagai deklarasi list kosong, yang kemudian akan dimasukan beberapa data angka (saya memilih angka 1-4), kemudian kita akan meng display list dengan 2 method urutan yang telah kita buat, untuk outputnya adalah sebagai berikut ini:

```
Daftar elemen dari depan ke belakang: 1 <-> 2 <-> 3 <-> 4
Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1
PS C:\Users\jerry\OneDrive\Documents\Belajar koding\C++\P
```