

Tugas Proceeding  
Modul 4 Struktur Data  
SINGLE 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 4  
STRUKTUR DATA - Ganjil 2024/2025  
"SINGLE 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 3

1 Pembuatan linked list untuk kodingan dengan syarat syarat seperti pada modul kita dapat membuat kodingan seperti berikut ini:

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

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

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

    void insertFront(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = head;
        head = newNode;
    }

    void insertBack(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* temp = head;
            while (temp->next != nullptr) {
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }

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

int main() {
    LinkedList list;

    list.insertFront(10);
    list.insertBack(20);
    list.insertFront(5);

    cout << "Isi Dari SLL adalah: ";
    list.printList();

    return 0;
}
```

hal pertama yang perlu kita lakukan yakni untuk membuat struct untuk node yang akan berisikan data (berupa int) dan cara penghubungnya dengan next node, lalu kita akan membuat kelas untuk SLL yang akan berisi node awal (head) yang akan kita set awalnya sebagai null ptr karena kita belum mengisinya, lalu kita akan membuat 3 metod insert pada LinkedList yaitu untuk insert di depan linked list dan di ahir linked list juga, setelahh kita membuat kedua method untuk insert node langkah berikutnya adalah untuk membuat method untuk print linkedlist yang akan menunjukan data yang berada dalam linked list, setelah kita membuat ketiga metod ini kita akan menggunakannya dalam int main yang dimanana kita akan insert nilai 10,20, dan 5 yang dimana urutanya akan menampilkan 5 10 dan 20, lalu setelah kita menginputkan nilai kita akan mengeprint linked list dengan ouput seperti berikut ini:

```
Isi Dari SLL adalah: 5 -> 10 -> 20
PS C:\Users\jerry\OneDrive\Documents
```

2 Penghapusan node pada linked list, kodingan untuk permasalahan ini dapat dilihat pada string kode berikut ini:

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

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

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

    void insertFront(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = head;
        head = newNode;
    }

    void insertBack(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* temp = head;
            while (temp->next != nullptr) {
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }

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

```
void deleteNode(int value) {
    if (head == nullptr) {
        cout << "List kosong." << endl;
        return;
    }

    if (head->data == value) {
        Node* temp = head;
        head = head->next;
        delete temp;
        return;
    }

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

    if (temp->next == nullptr) {
        cout << "Node dengan nilai " << value << " tidak ditemukan." << endl;
        return;
    }

    Node* nodeToDelete = temp->next;
    temp->next = nodeToDelete->next;
    delete nodeToDelete;
}

int main() {
    LinkedList list;

    list.insertFront(10);
    list.insertBack(20);
    list.insertFront(5);

    list.deleteNode(10);
    cout << "Linked List setelah penghapusan: ";
    list.printList(); // Output: 5 -> 20

    return 0;
}
```

awalan insert nilai 10, 20 dan 5 sesuai perintah yang ada pada modul, lalu kita akan menghapus node dengan value 10 yang dimana method akan mencari nilai tersebut, bila nilai ada pada list node akan dihapus, apabila tidak maka akan mengeluarkan "nilai tidak ditemukan", dan outputnya akan seperti berikut ini:

```
Linked List setelah penghapusan: 5 -> 20
PS C:\Users\jerry\OneDrive\Documents\Belajar
```

3 Pencarian dan hitung panjang dari linked list, untuk Unguided ini dapat diselesaikan dengan kodingan seperti berikut ini

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

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

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

    void insertFront(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = head;
        head = newNode;
    }

    void insertBack(int value) {
        Node* newNode = new Node();
        newNode->data = value;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* temp = head;
            while (temp->next != nullptr) {
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }

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

```
bool search(int value) {
    Node* temp = head;
    while (temp != nullptr) {
        if (temp->data == value) {
            return true;
        }
        temp = temp->next;
    }
    return false;
}

int countNodes() {
    int count = 0;
    Node* temp = head;
    while (temp != nullptr) {
        count++;
        temp = temp->next;
    }
    return count;
}

int main() {
    LinkedList list;

    list.insertFront(10);
    list.insertBack(20);
    list.insertFront(5);

    if (list.search(20)) {
        cout << "Node Nilai 20 Found" << endl;
    } else {
        cout << "Node Nilai 20 not found" << endl;
    }
    cout << "Panjang SLL adalah: " << list.countNodes() << endl;
    return 0;
}
```

panjang dari linked list kita dengan method countNodes, output dari kodingan kita adalah sebagai berikut ini:

```
Node Nilai 20 Found
Panjang SLL adalah: 3
```

### C. TP modul 4

1 pembuatan single linked list dengan 3 file berbeda (list.h, list.cpp, dan main.cpp)

```
#ifndef LIST_H
#define LIST_H
#include <iostream>
using namespace std;

typedef int infotype;
typedef struct ElmtList *address;

struct ElmtList {
    infotype info;
    address next;
};

struct List {
    address first;
};

void createList(List &L);
address allocate(infotype x);
void deallocate(address P);
void insertFirst(List &L, address P);
void insertLast(List &L, address P);
void printInfo(List L);

#endif
```

```
#include "list.h"

void createList(List &L) {
    L.first = nullptr;
}

address allocate(infotype x) {
    address P = new ElmtList;
    P->info = x;
    P->next = nullptr;
    return P;
}

void deallocate(address P) {
    delete P;
}

void insertFirst(List &L, address P) {
    P->next = L.first;
    L.first = P;
}

void insertLast(List &L, address P) {
    if (L.first == nullptr) {
        insertFirst(L, P);
    } else {
        address last = L.first;
        while (last->next != nullptr) {
            last = last->next;
        }
        last->next = P;
    }
}

void printInfo(List L) {
    address P = L.first;
    while (P != nullptr) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}
```

Untuk awalan kita akan mengetikkan sintaks seperti pada capture disamping dan kita akan memasukkan sintaks tersebut kedalam file pertama kita yakni list.h

Lalu langkah berikutnya adalah kita membuat file baru yakni list.cpp, langkah pertama pada file list.cpp adalah kita akan membuat list kosong terlebih dahulu sesuai perintah dari TP, lalu kita akan membuat elemen dengan fungsi allocate, dengan kodingan yang mengikuti tp, lalu kita akan membuat method untuk meng insertkan elemen yang telah kita buat menggunakan insert first atau last yang dimana kita akan menginputkan nilai sesuai apa yang kita gunakan, lalu kita akan membuat method untuk print list tersebut, serta kita akan mengecek apakah list kita isi atau tidak. Lalu kita akan lanjut ke file ketiga yakni main.cpp yang berisi kodingan seperti berikut:

```
#include "list.h"
int main() {
    List L;
    createList(L);

    int nim[] = {4, 2, 5};

    for (int i = 0; i < 3; i++) {
        address P = allocate(nim[i]);
        insertFirst(L, P);
    }

    cout << "Isi dari Single Linked List: ";
    printInfo(L);

    return 0;
}
```

dalam file main ini kita akan memanggil list.h untuk penggunaan sintaksnya, lalu kita akan membuat penggunaan method dari file list.cpp yang dimana kita akan membuat list L yang akan dimasukan nim 4,2 dan 5 kedalam array nim, lalu kita akan membuat perulangan untuk insert elemen kedalam linked list, dan ahirnya kita akan mengeprint hasilnya dengan output sebagai berikut ini:

```
Isi dari Single Linked List: 5 2 4
PS C:\Users\jerry\OneDrive\Documents
```