LAPORAN PRAKTIKUM

Modul 6

“Double Linked List (Bagian I)”

Disusun Oleh:

Doni Wicaksono - 21104062

SE-05-02

Dosen:

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

1. Menambahkan elemen di awal dan akhir DLL

```cpp
#include <iostream>
struct Node {
    int val;
    Node *prev, *next;
    Node(int v) : val(v), prev(nullptr), next(nullptr) {}
};

void addFirst(Node*& head, int v) {
    Node* node = new Node(v);
    node->next = head;
    if (head) head->prev = node;
    head = node;
}

void addLast(Node*& head, int v) {
    if (!head) {
        head = new Node(v);
        return;
    }

    Node* curr = head;
    while (curr->next) curr = curr->next;

    Node* node = new Node(v);
    curr->next = node;
    node->prev = curr;
}

void print(Node* head) {
    std::cout << "List: ";
    while (head) {
        std::cout << head->val;
        if (head->next) std::cout << " <-> ";
        head = head->next;
    }
    std::cout << "\n";
}
```

```cpp
int main() {
    Node* head = nullptr;
    int v;

    std::cout << "First element: ";
    std::cin >> v;
    addLast(head, v);
    print(head);

    std::cout << "Second element (front): ";
    std::cin >> v;
    addFirst(head, v);
    print(head);

    std::cout << "Third element (back): ";
    std::cin >> v;
    addLast(head, v);
    print(head);
}
```

```
First element: 23
List: 23
Second element (front): 11
List: 11 <-> 23
Third element (back): 8
List: 11 <-> 23 <-> 8
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-dagxqknx.shi" 1>"/tmp/Microsoft-MIEngine-Out-2jfq42de.s0g"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```

2. Menghapus elemen di awal dan akhir DLL

```cpp
#include <iostream>
struct Node {
    int val;
    Node *prev, *next;
    Node(int v) : val(v), prev(nullptr), next(nullptr) {}
};

void add(Node*& head, int v) {
    if (!head) {
        head = new Node(v);
        return;
    }

    Node* curr = head;
    while (curr->next) curr = curr->next;

    Node* node = new Node(v);
    curr->next = node;
    node->prev = curr;
}
```

```cpp
void popFirst(Node*& head) {
    if (!head) {
        std::cout << "Empty list\n";
        return;
    }

    Node* temp = head;
    head = head->next;
    if (head) head->prev = nullptr;
    delete temp;
}

void popLast(Node*& head) {
    if (!head) {
        std::cout << "Empty list\n";
        return;
    }

    if (!head->next) {
        delete head;
        head = nullptr;
        return;
    }

    Node* curr = head;
    while (curr->next) curr = curr->next;
    curr->prev->next = nullptr;
    delete curr;
}

void print(Node* head) {
    std::cout << "List: ";
    while (head) {
        std::cout << head->val;
        if (head->next) std::cout << " <-> ";
        head = head->next;
    }
    std::cout << "\n";
}
```

```cpp
int main() {
    Node* head = nullptr;
    int v;

    std::cout << "First element: ";
    std::cin >> v;
    add(head, v);
    print(head);

    std::cout << "Second element: ";
    std::cin >> v;
    add(head, v);
    print(head);

    std::cout << "Third element: ";
    std::cin >> v;
    add(head, v);
    print(head);

    std::cout << "=================\n";
    std::cout << "Removing first and last:\n";
    popFirst(head);
    popLast(head);
    print(head);
}
```

```
First element: 77
List: 77
Second element: 28
List: 77 <-> 28
Third element: 19
List: 77 <-> 28 <-> 19
=================
Removing first and last:
List: 28
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-gplgjos1.we1" 1>"/tmp/Microsoft-MIEngine-Out-sleu0xbt.umn"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```

3. Menampilkan elemen dari depan ke belakang dan sebaliknya

```cpp
#include <iostream>
struct Node {
    int val;
    Node *prev, *next;
    Node(int v) : val(v), prev(nullptr), next(nullptr) {}
};

void add(Node*& head, int v) {
    if (!head) {
        head = new Node(v);
        return;
    }

    Node* curr = head;
    while (curr->next) curr = curr->next;

    Node* node = new Node(v);
    curr->next = node;
    node->prev = curr;
}

void printFwd(Node* head) {
    std::cout << "Forward: ";
    while (head) {
        std::cout << head->val;
        if (head->next) std::cout << " <-> ";
        head = head->next;
    }
    std::cout << "\n";
}

void printRev(Node* head) {
    if (!head) return;

    Node* curr = head;
    while (curr->next) curr = curr->next;

    std::cout << "Reverse: ";
    while (curr) {
        std::cout << curr->val;
        if (curr->prev) std::cout << " <-> ";
        curr = curr->prev;
    }
    std::cout << "\n";
}
```

```cpp
int main() {
    Node* head = nullptr;
    int v;

    for (int i = 1; i <= 4; i++) {
        std::cout << "Element " << i << ": ";
        std::cin >> v;
        add(head, v);
    }

    printFwd(head);
    printRev(head);
}
```

```
Element 1: 1
Element 2: 3
Element 3: 3
Element 4: 7
Forward: 1 <-> 3 <-> 3 <-> 7
Reverse: 7 <-> 3 <-> 3 <-> 1
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-xehw4olz.ieg" 1>"/tmp/Microsoft-MIEngine-Out-g1wfmmh3.tgu"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```