LAPORAN PRAKTIKUM

Modul 5

“Single Linked List (Bagian II)”

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. Mencari elemen tertentu dalam SLL

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

void add(Node*& head, int v) {
    if (!head) {
        head = new Node(v);
        return;
    }
    Node* curr = head;
    while (curr->next) curr = curr->next;
    curr->next = new Node(v);
}

void find(Node* head, int v) {
    Node* curr = head;
    int pos = 1;

    while (curr) {
        if (curr->val == v) {
            std::cout << v << " found at address " << curr << ",
position " << pos << "\n";
            return;
        }
        curr = curr->next;
        pos++;
    }
    std::cout << v << " not found\n";
}

int main() {
    Node* head = nullptr;
    std::cout << "Enter 6 elements:\n";

    for (int i = 0; i < 6; i++) {
        int v;
        std::cout << i+1 << ": ";
        std::cin >> v;
        add(head, v);
    }

    int target;
    std::cout << "Search for: ";
    std::cin >> target;
    find(head, target);
}
```

```
Enter 6 elements:
1: 2
2: 7
3: 5
4: 8
5: 9
6: 3
Search for: 8
8 found at address 0x55555556b730, position 4
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-nwy5gic2.nyz" 1>"/tmp/Microsoft-MIEngine-Out-dr34p1ui.kca"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```

2. Mengurutkan list menggunakan Bubble Sort

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

void add(Node*& head, int v) {
    if (!head) {
        head = new Node(v);
        return;
    }
    Node* curr = head;
    while (curr->next) curr = curr->next;
    curr->next = new Node(v);
}

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

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

    Node* last = nullptr;
    bool swapped;

    do {
        swapped = false;
        Node* curr = head;

        while (curr->next != last) {
            if (curr->val > curr->next->val) {
                std::swap(curr->val, curr->next->val);
                swapped = true;
            }
            curr = curr->next;
        }
        last = curr;
    } while (swapped);
}
```

```cpp
int main() {
    Node* head = nullptr;
    std::cout << "Enter 5 elements:\n";

    for (int i = 0; i < 5; i++) {
        int v;
        std::cout << i+1 << ": ";
        std::cin >> v;
        add(head, v);
    }

    std::cout << "Before sort: ";
    print(head);
    sort(head);
    std::cout << "After sort: ";
    print(head);
}
```

```
Enter 5 elements:
1: 4
2: 1
3: 9
4: 6
5: 0
Before sort: 4 1 9 6 0
After sort: 0 1 4 6 9
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-zzium24g.b5f" 1>"/tmp/Microsoft-MIEngine-Out-hqnd5ys4.4am"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```

3. Menambahkan elemen secara terurut

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

void insert(Node*& head, int v) {
    Node* node = new Node(v);

    if (!head || head->val >= v) {
        node->next = head;
        head = node;
        return;
    }

    Node* curr = head;
    while (curr->next && curr->next->val < v) {
        curr = curr->next;
    }
    node->next = curr->next;
    curr->next = node;
}

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

```
int main() {
    Node* head = nullptr;
    std::cout << "Enter 4 elements in order:\n";

    for (int i = 0; i < 4; i++) {
        int v;
        std::cout << i+1 << ": ";
        std::cin >> v;
        insert(head, v);
    }

    std::cout << "Current list: ";
    print(head);

    std::cout << "Enter new element: ";
    int v;
    std::cin >> v;
    insert(head, v);

    std::cout << "Updated list: ";
    print(head);
}
```

```
Enter 4 elements in order:
1: 4
2: 0
3: 8
4: 1
Current list: 0 1 4 8
Enter new element: 5
Updated list: 0 1 4 5 8
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEn
gine-In-qahddkmt.2fe" 1>"/tmp/Microsoft-MIEngine-Out-4bihceml.p1n"
@slashedzer0 ➜/workspaces/STD_Doni_Wicaksono_21104062 (main) $
```