

**LAPORAN PRAKTIKUM**  
**MODUL V**  
**“SINGLE LINKED LIST BAGIAN 2”**



**Nama :**

Reza Irawan(2311104035)

**Dosen :**

YUDHA ISLAMI S, S.kom, M.cs

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## Tugas 1

```
#include <iostream>

using namespace std;

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

// Function to create a new node
Node* newNode(int data) {
    Node* node = new Node;
    node->data = data;
    node->next = nullptr;
    return node;
}

// Function to insert a new node at the beginning of the list
Node* insertFirst_2311104035(Node* head, int data) {
    Node* new_node = newNode(data);
    new_node->next = head;
    return new_node;
}

// Function to search for an element in the linked list
Node* searchElement_2311104035(Node* head, int data) {
    int position = 1;
    Node* current = head;
    while (current != nullptr) {
        if (current->data == data) {
            cout << "Elemen " << data << " ditemukan pada posisi " << position << endl;
            return current;
        }
        current = current->next;
        position++;
    }
    cout << "Elemen " << data << " tidak ditemukan" << endl;
    return nullptr;
}

int main() {
    Node* head = nullptr;

    // Input elements from the user
    cout << "Masukkan 6 elemen integer: " << endl;
    for (int i = 0; i < 6; i++) {
        int data;
        cin >> data;
        head = insertFirst_2311104035(head, data);
    }

    // Search for an element
    int search_data;
    cout << "Masukkan elemen yang ingin dicari: ";
    cin >> search_data;
    searchElement_2311104035(head, search_data);

    return 0;
}
```

Output:

```
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2> cd "d:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2"
pp -o soal01 } ; if ($?) { .\soal01 }
Masukkan 6 elemen integer:
2
3
4
5
6
7
Masukkan elemen yang ingin dicari: 7
Elemen 7 ditemukan pada posisi 1
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2\TP>
```

Penjelasan:

Kode di atas melibatkan beberapa fungsi seperti pembuatan node baru, penambahan node di awal linked list, dan pencarian elemen di dalam linked list. Fungsi newNode digunakan untuk membuat node baru dengan nilai data tertentu dan pointer next yang diinisialisasi ke nullptr. Fungsi insertFirst\_2311104035 menambahkan node baru di awal list, menjadikan node baru tersebut sebagai head yang baru. Pada fungsi searchElement\_2311104035, program mencari nilai elemen dalam linked list dan menampilkan posisinya jika elemen tersebut ditemukan; jika tidak, akan ditampilkan pesan bahwa elemen tersebut tidak ada dalam list.

Di dalam fungsi main(), pengguna diminta untuk memasukkan 6 elemen integer yang akan dimasukkan ke dalam linked list, kemudian menginput nilai elemen yang ingin dicari dalam list.

## Tugas 2

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

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

class Tugas2 {
private:
    Node* head;

public:
    Tugas2() {
        head = nullptr;
    }

    void insert(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 bubbleSort() {
        if (head == nullptr) return;

        bool swapped;
        do {
            swapped = false;
            Node* current = head;

            while (current->next != nullptr) {
                if (current->data > current->next->data) {
                    // Tukar data
                    int temp = current->data;
                    current->data = current->next->data;
                    current->next->data = temp;
                    swapped = true;
                }
                current = current->next;
            }
        } while (swapped);
    }

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

int main() {
    Tugas2 list;

    // Insert contoh data
    list.insert(7);
    list.insert(9);
    list.insert(2);
    list.insert(4);
    list.insert(6);

    cout << "Sebelum sorting: ";
    list.display();

    list.bubbleSort();

    cout << "Setelah sorting: ";
    list.display();

    return 0;
}
```

Output:

```
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2> cd "d:\STRU
pp -o soal02 } ; if ($?) { .\soal02 }
Sebelum sorting: 7 9 2 4 6
Setelah sorting: 2 4 6 7 9
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2\TP>
```

Penjelasan:

Implementasi kode di atas merupakan contoh dari penggunaan linked list yang memiliki tambahan fungsi pengurutan dengan menggunakan algoritma Bubble Sort. Struktur Node menyimpan data bilangan bulat dan pointer next yang menuju node selanjutnya. Kelas Tugas2 bertanggung jawab untuk mengelola linked list dengan tiga fungsi utama, yaitu insert, bubbleSort, dan display. Fungsi insert bertujuan untuk menambahkan node baru di bagian akhir list, sementara bubbleSort digunakan untuk mengurutkan data dalam list secara ascending dengan menggunakan algoritma Bubble Sort. Dalam algoritma Bubble Sort, pertukaran dilakukan antara dua elemen tetangga jika elemen pertama lebih besar daripada elemen kedua. Proses ini diulang sampai tidak ada pertukaran yang diperlukan. Fungsi display digunakan untuk menampilkan semua elemen dalam linked list. Pada awalnya, beberapa contoh data dimasukkan ke dalam daftar, kemudian data ditampilkan sebelum dan sesudah proses pengurutan.

### Tugas 3

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

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

Node* insertSorted_2311104035(Node* head, int data) {
    Node* newNode = new Node;
    newNode->data = data;
    newNode->next = nullptr;

    if (head == nullptr || data < head->data) {
        newNode->next = head;
        return newNode;
    }

    Node* current = head;
    Node* previous = nullptr;

    while (current != nullptr && data >= current->data) {
        previous = current;
        current = current->next;
    }

    newNode->next = current;
    previous->next = newNode;
    return head;
}

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

int main() {
    Node* head = nullptr;

    int n;
    cout << "Masukkan jumlah elemen: ";
    cin >> n;

    cout << "Masukkan " << n << " elemen integer: ";
    for (int i = 0; i < n; i++) {
        int data;
        cin >> data;
        head = insertSorted_2311104035(head, data);
    }

    cout << "List terurut: ";
    printList(head);

    int newElement;
    cout << "Masukkan elemen baru: ";
    cin >> newElement;

    head = insertSorted_2311104035(head, newElement);

    cout << "List terurut setelah menambahkan elemen baru: ";
    printList(head);

    return 0;
}
```

Output:

```
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2> cd "d:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2"
pp -o soal03 } ; if ($?) { .\soal03 }
Masukkan jumlah elemen: 5
Masukkan 5 elemen integer: 1 2 3 4 5
List terurut: 1 2 3 4 5
Masukkan elemen baru: 10
List terurut setelah menambahkan elemen baru: 1 2 3 4 5 10
PS D:\STRUKTUR DATA\05_Single_Lingked_List_Bagian_2\TP> |
```

Penjelasan:

Implementasi kode di atas adalah dari linked list yang menyimpan elemen-elemen secara terurut secara meningkat. InsertSorted\_2311104035 Jika linked list kosong atau data elemen baru lebih kecil dari elemen di head, elemen baru ditempatkan di awal. Jika tidak, fungsi ini akan mencari posisi yang tepat dalam linked list untuk menempatkan elemen baru, sehingga linked list tetap terurut setelah elemen ditambahkan. Fungsi printList digunakan untuk mencetak semua elemen dalam linked list. Dalam fungsi utama, pengguna diminta untuk memasukkan sejumlah elemen yang akan diurutkan dan ditampilkan dalam bentuk daftar. Setelah itu, pengguna dapat menambahkan elemen baru, dan hasilnya dicetak lagi untuk menunjukkan linked list setelah penambahan elemen.

