

LAPORAN PRAKTIKUM
STRUKTUR DATA
SINGLE LINKED LIST BAGIAN 2



Nama :

Muhammad Mahrus Ali (2311104006)

Dosen :

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

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

A. Soal TP

1. Latsol_01

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

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

void insertLast(Node*& head, 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 searchElement(Node* head, int value) {
    Node* temp = head;
    int position = 1;
    bool found = false;

    while (temp != nullptr) {
        if (temp->data == value) {
            cout << "Elemen ditemukan pada urutan ke-" << position << endl;
            cout << "Alamat elemen: " << temp << endl;
            found = true;
            break;
        }
        temp = temp->next;
        position++;
    }

    if (!found) {
        cout << "Elemen " << value << " tidak ada dalam list." << endl;
    }
}

int main() {
    Node* head = nullptr;
    int value;

    cout << "Masukkan 6 elemen integer ke dalam list: " << endl;
    for (int i = 0; i < 6; i++) {
        int element;
        cout << "Elemen ke-" << (i + 1) << ": ";
        cin >> element;
        insertLast(head, element);
    }

    cout << "Masukkan nilai yang ingin dicari: ";
    cin >> value;

    searchElement(head, value);

    return 0;
}
```

Output dari kode diatas :

```
Masukkan 6 elemen integer ke dalam list:
Elemen ke-1: 2
Elemen ke-2: 1
Elemen ke-3: 5
Elemen ke-4: 3
Elemen ke-5: 2
Elemen ke-6: 1
Masukkan nilai yang ingin dicari: 3
Elemen ditemukan pada urutan ke-4
Alamat elemen: 0x1fd4249e690
```

2. Latsol_02

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

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

void insertLast(Node*& head, 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 displayList(Node* head) {
    Node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}
```

```

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

    bool swapped;
    Node* current;
    Node* lastPtr = nullptr;

    do {
        swapped = false;
        current = head;

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

int main() {
    Node* head = nullptr;

    cout << "Masukkan 5 elemen integer ke dalam list: " << endl;
    for (int i = 0; i < 5; i++) {
        int element;
        cout << "Elemen ke-" << (i + 1) << ": ";
        cin >> element;
        insertLast(head, element);
    }

    cout << "List sebelum diurutkan: ";
    displayList(head);

    bubbleSortList(head);

    cout << "List setelah diurutkan: ";
    displayList(head);

    return 0;
}

```

Outputnya :

```

Masukkan 5 elemen integer ke dalam list:
Elemen ke-1: 2
Elemen ke-2: 3
Elemen ke-3: 1
Elemen ke-4: 4
Elemen ke-5: 0
List sebelum diurutkan: 2 3 1 4 0
List setelah diurutkan: 0 1 2 3 4

```

3. Latsol_03

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

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

void insertSorted(Node*& head, int value) {
    Node* newNode = new Node();
    newNode->data = value;
    newNode->next = nullptr;

    if (head == nullptr || head->data >= value) {
        newNode->next = head;
        head = newNode;
    } else {
        Node* temp = head;

        while (temp->next != nullptr && temp->next->data < value) {
            temp = temp->next;
        }
        newNode->next = temp->next;
        temp->next = newNode;
    }
}

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

int main() {
    Node* head = nullptr;
    int element;

    cout << "Masukkan 4 elemen integer secara terurut: " << endl;
    for (int i = 0; i < 4; i++) {
        cout << "Elemen ke-" << (i + 1) << ": ";
        cin >> element;
        insertSorted(head, element);
    }

    cout << "List sebelum menambahkan elemen baru: ";
    displayList(head);

    cout << "Masukkan elemen tambahan untuk dimasukkan: ";
    cin >> element;

    insertSorted(head, element);
    cout << "List setelah menambahkan elemen baru: ";
    displayList(head);
}
```

Output code :

```
Masukkan 4 elemen integer secara terurut:  
Elemen ke-1: 4  
Elemen ke-2: 6  
Elemen ke-3: 7  
Elemen ke-4: 89  
List sebelum menambahkan elemen baru: 4 6 7 89  
Masukkan elemen tambahan untuk dimasukkan: 5  
List setelah menambahkan elemen baru: 4 5 6 7 89
```

B. UNGUIDED

1. Latihan_01

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

struct Mahasiswa {
    int nim;
    string nama;
};

struct Node{
    Mahasiswa info;
    Node* next;
};

void tambahMahasiswa(Node*& head, int nim, string nama) {
    Node* newNode = new Node();
    newNode->info.nim = nim;
    newNode->info.nama = nama;
    newNode->next = nullptr;

    if(!head) {
        head = newNode;
    } else {
        Node* temp = head;
        while(temp -> next) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
    cout << "Mahasiswa dengan NIM " << nim << " sukses ditambahkan" << endl;
}

void cariMahasiswa(Node* head, int nim) {
    Node* temp = head;
    while(temp) {
        if(temp->info.nim == nim) {
            cout << "Mahasiswa ditemukan: " << temp->info.nama << endl;
            return;
        }
        temp = temp->next;
    }
    cout << "Mahasiswa dengan NIM" << nim << "tidak ditemukan." << endl;
}
```

```

void tampilkanSemuaMahasiswa(Node* head) {
    if(!head) {
        cout << "Daftar Mahasiswa kosong." << endl;
        return;
    }
    Node* temp = head;
    while (temp) {
        cout << "NIM: " << temp->info.nim << ", Nama: " << temp->info.nama << endl;
        temp = temp -> next;
    }
}

int main() {
    Node* head = nullptr;
    int pilihan, nim;
    string nama;

    do {
        cout << "\nMenu:\n";
        cout << "1. Tambah Mahasiswa\n";
        cout << "2. Cari Mahasiswa\n";
        cout << "3. Tampilkan Semua Mahasiswa\n";
        cout << "4. Keluar\n";
        cout << "Pilih Opsi: ";
        cin >> pilihan;
        switch (pilihan) {
            case 1:
                cout << "Masukkan NIM Mahasiswa: ";
                cin >> nim;
                cin.ignore();
                cout << "Masukkan Nama Mahasiswa: ";
                getline(cin, nama);
                tambahMahasiswa(head, nim, nama);
                break;

            case 2:
                cout << "Masukkan NIM yang dicari: ";
                cin >> nim;
                cariMahasiswa(head, nim);
                break;

            case 3:
                tampilkanSemuaMahasiswa(head);
                break;

            case 4:
                cout << "Keluar dari program." << endl;
                break;

            default:
                cout << "Opsi tidak valid." << endl;
        }
    } while (pilihan != 4);
    return 0;
}

```


Output dari kodingan diatas :

- Tambah mahasiswa

```
Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih Opsi: 1
Masukkan NIM: 23
Masukkan Nama: rifki
Mahasiswa dengan NIM 23 ditambahkan
```

- Cari mahasiswa

```
Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih Opsi: 2
Masukkan NIM yang dicari: 23
Mahasiswa ditemukan: rifki
```

Penjelasan Kode :

Program di atas merupakan program sederhana yang memiliki fitur tambah mahasiswa, cari mahasiswa, dan menampilkan seluruh mahasiswa, didalamnya terdapat struct mahasiswa untuk menyimpan informasi data mahasiswa, ada fungsi tambah mahasiswa, cari mahasiswa dan tampilkan seluruh mahasiswa.