LAPORAN PRAKTIKUM
PERTEMUAN 5
MODUL 5 STD

Nama :

Farrel I’zaz Yuwono (2311104014)

Dosen :

YUDHA ISLAMI SULISTYA

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

Tugas Pendahuluan

1.

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

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

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

    void append_2311104014(int data) {
        Node* newNode = new Node();
        newNode->data = data;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newNode;
        }
    }

    void searchElement_2311104014(int value) {
        Node* current = head;
        int position = 1;
        while (current != nullptr) {
            if (current->data == value) {
                cout << "Elemen ditemukan di alamat node " << current << " dan urutan ke-" << position <<
endl;
                return;
            }
            current = current->next;
            position++;
        }
        cout << "Elemen tidak ditemukan dalam list" << endl;
    }
};

int main() {
    SinglyLinkedList sll;
    int elemen;
    for (int i = 0; i < 6; i++) {
        cout << "Masukkan elemen ke-" << i+1 << ": ";
        cin >> elemen;
        sll.append_2311104014(elemen);
    }

    int nilai_cari;
    cout << "Masukkan elemen yang ingin dicari: ";
    cin >> nilai_cari;
    sll.searchElement_2311104014(nilai_cari);

    return 0;
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output> & .\'TP_Soal_1.exe'
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen ke-5: 5
Masukkan elemen ke-6: 6
Masukkan elemen yang ingin dicari: 4
Elemen ditemukan di alamat node 0x1dd0f6d2a20 dan urutan ke-4
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output>
```

2.

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

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

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

    void append_2311104014(int data) {
        Node* newNode = new Node();
        newNode->data = data;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newNode;
        }
    }

    void bubbleSort_2311104014() {
        bool swapped;
        Node* current;
        Node* lastPtr = nullptr;

        if (head == nullptr)
            return;

        do {
            swapped = false;
            current = head;

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

    void display_2311104014() {
        Node* current = head;
        while (current != nullptr) {
            cout << current->data << " ";
            current = current->next;
        }
        cout << endl;
    }
};

int main() {
    SinglyLinkedList sll;
    int elemen;
    for (int i = 0; i < 5; i++) {
        cout << "Masukkan elemen ke-" << i+1 << ": ";
        cin >> elemen;
        sll.append_2311104014(elemen);
    }

    cout << "Sebelum diurutkan: ";
    sll.display_2311104014();

    sll.bubbleSort_2311104014();

    cout << "Setelah diurutkan: ";
    sll.display_2311104014();

    return 0;
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output> cd 'd:\kuliah\strukt
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output> & .\'TP_Soal_2.exe'
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen ke-5: 5
Sebelum diurutkan: 1 2 3 4 5
Setelah diurutkan: 1 2 3 4 5
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output>
```

3.

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

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

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

    void append_2311104014(int data) {
        Node* newNode = new Node();
        newNode->data = data;
        newNode->next = nullptr;
        if (head == nullptr) {
            head = newNode;
        } else {
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newNode;
        }
    }

    void insertSorted_2311104014(int data) {
        Node* newNode = new Node();
        newNode->data = data;

        if (head == nullptr || head->data >= newNode->data) {
            newNode->next = head;
            head = newNode;
        } else {
            Node* current = head;
            while (current->next != nullptr && current->next->data < newNode->data) {
                current = current->next;
            }
            newNode->next = current->next;
            current->next = newNode;
        }
    }

    void display_2311104014() {
        Node* current = head;
        while (current != nullptr) {
            cout << current->data << " ";
            current = current->next;
        }
        cout << endl;
    }
};

int main() {
    SinglyLinkedList sll;
    int elemen;
    for (int i = 0; i < 4; i++) {
        cout << "Masukkan elemen ke-" << i+1 << ": ";
        cin >> elemen;
        sll.append_2311104014(elemen);
    }

    int tambahan;
    cout << "Masukkan elemen tambahan yang ingin disisipkan secara terurut: ";
    cin >> tambahan;
    sll.insertSorted_2311104014(tambahan);

    cout << "List setelah elemen baru dimasukkan: ";
    sll.display_2311104014();

    return 0;
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output> cd 'd:\kuliah\strukturd
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output> & .\'TP_Soal_3.exe'
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen tambahan yang ingin disisipkan secara terurut: 5
List setelah elemen baru dimasukkan: 1 2 3 4 5
PS D:\kuliah\strukturdat\tugas kuliah\05_Single_Linked_List_Bagian_2\TP\output>
```

Penjelasan Tugas Pendahuluan

1. struct Node: Menyimpan data dan pointer ke node berikutnya.

   Konstruktor: Menginisialisasi list dengan head = nullptr.

   append_2311104014:
   - Menambahkan elemen ke akhir linked list.
   - Jika list kosong, node baru menjadi head.

   searchElement_2311104014:
   - Mencari elemen di dalam linked list.
   - Menampilkan alamat memori dan posisi elemen jika ditemukan.
   - Jika tidak ditemukan, menampilkan pesan "Elemen tidak ditemukan".

   Program Utama (main):

   Meminta pengguna memasukkan 6 elemen dan menyimpannya di list.

   Meminta pengguna memasukkan nilai yang ingin dicari, lalu mencari nilai tersebut dalam list.

2. Penjelasan Singkat Program Singly Linked List dengan Bubble Sort

   Struktur Node Menyimpan data dan pointer ke node berikutnya.

   Fungsi append_2311104014 :
   -Menambahkan elemen baru ke akhir linked list.
   -Jika list kosong, node baru menjadi head.

   Fungsi bubbleSort_2311104014 :
   -Mengurutkan elemen di dalam list menggunakan Bubble Sort.
   -Melakukan pertukaran jika elemen saat ini lebih besar dari elemen berikutnya hingga semua elemen terurut.

   Fungsi display_2311104014 :
   -Menampilkan semua elemen di dalam linked list.

   Program Utama (main) :

   -Meminta pengguna memasukkan 5 elemen.
   -Menampilkan elemen sebelum dan setelah diurutkan dengan Bubble Sort.

3. Penjelasan Super Singkat Singly Linked List dengan Insert Terurut

- Node: Struktur yang menyimpan data dan pointer ke node berikutnya.
- append_2311104014: Menambah elemen di akhir list.
- insertSorted_2311104014:
    - Menyisipkan elemen secara terurut.
    - Jika elemen lebih kecil dari head, ditempatkan di depan.
    - Jika tidak, elemen ditempatkan setelah node dengan nilai lebih kecil.
- display_2311104014: Menampilkan semua elemen di dalam list.

unguided

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

struct Node {
    int NIM;
    string nama;
    Node* next;
};

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

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

        if (head == nullptr) {
            head = newNode;
        } else {
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newNode;
        }
        cout << "Mahasiswa dengan NIM " << nim << " dan nama " << nama << " berhasil ditambahkan." << endl;
    }

    void cariMahasiswa(int nim) {
        Node* current = head;
        while (current != nullptr) {
            if (current->NIM == nim) {
                cout << "Mahasiswa dengan NIM " << nim << " ditemukan. Nama: " << current->nama << endl;
                return;
            }
            current = current->next;
        }
        cout << "Mahasiswa dengan NIM " << nim << " tidak ditemukan." << endl;
    }

    void tampilkanMahasiswa() {
        Node* current = head;
        while (current != nullptr) {
            cout << "NIM: " << current->NIM << ", Nama: " << current->nama << endl;
            current = current->next;
        }
    }
};

int main() {
    SinglyLinkedList list;
    int pilihan, nim;
    string nama;

    while (true) {
        cout << "\nMenu:\n1. Tambah Mahasiswa\n2. Cari Mahasiswa berdasarkan NIM\n3. Tampilkan Semua Mahasiswa\n4. Keluar\n";
        cout << "Pilih opsi: ";
        cin >> pilihan;

        switch (pilihan) {
        case 1:
            cout << "Masukkan NIM: ";
            cin >> nim;
            cout << "Masukkan Nama: ";
            cin.ignore();
            getline(cin, nama);
            list.tambahMahasiswa(nim, nama);
            break;
        case 2:
            cout << "Masukkan NIM yang ingin dicari: ";
            cin >> nim;
            list.cariMahasiswa(nim);
            break;
        case 3:
            list.tampilkanMahasiswa();
            break;
        case 4:
            cout << "Keluar program.\n";
            return 0;
        default:
            cout << "Pilihan tidak valid.\n";
        }
    }

    return 0;
}
```

Unguided

```
Mahasiswa dengan NIM 23 dan nama hah berhasil ditambahkan.

Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa berdasarkan NIM
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih opsi: 1
Masukkan NIM: 32
Masukkan Nama: jir
Mahasiswa dengan NIM 32 dan nama jir berhasil ditambahkan.

Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa berdasarkan NIM
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih opsi: 3
NIM: 23, Nama: hah
NIM: 32, Nama: jir

Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa berdasarkan NIM
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih opsi: 2
Masukkan NIM yang ingin dicari: 32
Mahasiswa dengan NIM 32 ditemukan. Nama: jir

Menu:
1. Tambah Mahasiswa
2. Cari Mahasiswa berdasarkan NIM
3. Tampilkan Semua Mahasiswa
4. Keluar
Pilih opsi:
```

Penjelasan :

Setiap node menyimpan:

- NIM: Nomor Induk Mahasiswa.
- nama: Nama mahasiswa.
- next: Pointer ke node berikutnya.

Konstruktor :

```
SinglyLinkedList() {
   head = nullptr;
}
```

- Inisialisasi awal dengan head bernilai nullptr (list kosong).

Fungsi tambahMahasiswa

- Menambahkan mahasiswa ke akhir list.
- Jika list kosong, node baru menjadi head.
- Jika tidak, elemen baru ditambahkan setelah node terakhir.

cariMahasiswa

- Mencari mahasiswa berdasarkan NIM.
- Jika ditemukan, menampilkan nama mahasiswa.
- Jika tidak, menampilkan pesan bahwa mahasiswa tidak ditemukan.

Menu utama memungkinkan pengguna:

1. Tambah Mahasiswa.
2. Cari Mahasiswa berdasarkan NIM.
3. Tampilkan Semua Mahasiswa.
4. Keluar dari program.