

Tugas 2

Linked List Kepala Ekor

1. Source Code

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

struct Buah {
    string nama_buah;
    int harga;
};

struct Node {
    Buah buah;
    Node * next;
};

Node* head, *tail;

Node* inisialisasi() {
    Buah h_buah, t_buah;
    h_buah.nama_buah = "KOSONG-BAWAH";
    h_buah.harga = -99999;

    t_buah.nama_buah = "KOSONG-ATAS";
    t_buah.harga = 99999;

    head = new Node;
    tail = new Node;
    head->buah = h_buah;
    head->next = tail;
    tail->buah = t_buah;
    tail->next = NULL;

    return head;
}

Node* sisipDepan(Buah buah) {

    // new keyword sama kayak memalloc versi modern nya (versi c++)
    Node* newNode = new Node;
    newNode->buah = buah;
    newNode->next = head->next;

    head->next = newNode;

    return head;
}

Node* sisipBelakang(Buah buah) {
    Node* sblm_tail = head;
    while (sblm_tail->next != tail) {
        sblm_tail = sblm_tail->next;
    }

    Node *newNode = new Node;
    newNode->next = tail;
```

```

        newNode->buah = buah;

        sbml_tail->next = newNode;

        return head;
    }

    // Sisip tengah menggunakan asumsi jika data selaluurut
    // Sehingga node baru (n) > n-1
    Node* sisipTengah( Buah buah) {
        Node *prevPos = head;
        while (prevPos->next->buah.harga < buah.harga || prevPos->next ==
NULL){
            prevPos = prevPos->next;
        }

        Node *newNode = new Node;
        newNode->buah = buah;
        newNode->next = prevPos->next;
        prevPos->next = newNode;

        return newNode;
    }

    void hapusNode(Node *node, string buah) {
        if (node->buah.nama_buah == buah) {
            if (node->next != NULL){
                head = node->next;
            }
            delete node;
            return;
        }
        while (node->next->buah.nama_buah != buah) {
            if (node->next == NULL) {
                cout << "BUAH TidAK ADA!" << endl;
                return;
            }
            node = node->next;
        }
        Node* belakang = node;
        Node* depan = node->next->next;
        Node* hapus = node->next;

        belakang->next = depan;
        delete hapus;
    }

    void tampilNode() {
        if (head->next == tail) {
            cout << "KOSONG! KAYAK HATIKU!" << endl;
            return;
        }
        Node* node = head->next;
        do {
            cout << "Nama Buah: " << node->buah.nama_buah << endl;
            cout << "Harga Buah: " << node->buah.harga << endl;
            cout << setw(50) << setfill('-') << " " << endl;
            node = node->next;
        }
        while (node != tail);
    }
}

```

```

// Tampilan tampilan
void view_utama(), view_insert_option(), view_sisiptengah(), view_hapus(),
view_tampilkan();

// fungsi utilitas
void lots_of_eq() {
    // fungsi untuk =====
    cout << setw(51) << setfill('=') << " " << endl;
}

void njir_bisa_tengah(string data) {
    // fungsi untuk menampilkan data ditengah
    int size = data.size();
    int left_space = (51-size) / 2;
    int right_space = 51-size-left_space;

    cout << setw(left_space + size) << right << setfill(' ') << data <<
left << setw(right_space) << setfill(' ') << right << endl;
}

// Node* sisipApapunSaatKosong(Buah buah) {
//     // menginisialisasi jika belum ada data
//     return inisialisasi();
// }

// Node* sisipJikaNodeSudahTerinisialisasi(Buah buah, Node* (*func)(Node*,
Buah)) {
//     // Utility untuk menjalankan fungsi (argumen 2) jika node sudah
terisi
//     if (head == NULL){
//         return sisipApapunSaatKosong(buah);
//     }

//     return func(head, buah);
// }

int main(){
    inisialisasi();
    view_utama();
}

void view_utama() {
    int choice;
    do {
        cout << setw(51) << setfill('=') << " " << endl;
        cout << setw((51-16)/2 + 16) << right << setfill(' ') << "Linked
List Buah" << left << setw((51-16)/2) << setfill(' ') << right << endl;
        cout << setw(51) << setfill('=') << " " << endl;
        cout << "1. Tambah Data" << endl;
        cout << "2. Hapus Data" << endl;
        cout << "3. Lihat Data" << endl;
        cout << "4. Exit" << endl;
        cout << "Pilih menu (1-4): "; cin >> choice;

        if (choice < 1 || choice > 4) {
            cout << "PILIH YANG BENER!" << endl;
            continue;
        }

        switch (choice)
        {

```

```

        case 1:
            view_insert_option();
            break;
        case 2:
            view_hapus();
        case 3:
            view_tampilkan();
        default:
            cout << "PILIH YANG BENER!" << endl;
            break;
    }
    } while (choice != 4);
}

void view_insert_option() {
    lots_of_eq();
    njir_bisa_tengah("TAMBAH DATA");
    lots_of_eq();

    Buah buah;
    cin.ignore();
    cout << "Nama buah: "; getline(cin, buah.nama_buah);
    cout << "Harga: "; cin >> buah.harga;

    int menu_sisip_choice;
    do {
        lots_of_eq();
        njir_bisa_tengah("MENU SISIP");
        lots_of_eq();

        cout << "1. SISIP DEPAN" << endl;
        cout << "2. SISIP BELAKANG" << endl;
        cout << "3. SISIP TENGAH" << endl;
        cout << "Pilih (1-3): "; cin >> menu_sisip_choice;
        if (menu_sisip_choice > 3 || menu_sisip_choice < 1){
            cout << "PILIH YANG BENER!" << endl;
            continue;
        }

        switch (menu_sisip_choice)
        {
            case 1:
                sisipDepan(buah);
                break;
            case 2:
                sisipBelakang(buah);
                break;
            case 3:
                sisipTengah(buah);
                break;
            default:
                cout << "PILIH Yang bener!" << endl;
                break;
        }
        cout << "Berhasil tambah data!" << endl;

    } while (menu_sisip_choice > 3 || menu_sisip_choice < 1);
}

void view_hapus() {

```

```

    lots_of_eq();
    njir_bisa_tengah("HAPUS BUAH");
    lots_of_eq();
    string buah;

    cin.ignore();
    cout << "Nama buah: "; getline(cin, buah);
    hapusNode(head, buah);

}

void view_tampilkan() {
    lots_of_eq();
    njir_bisa_tengah("DAFTAR BUAH!");
    lots_of_eq();

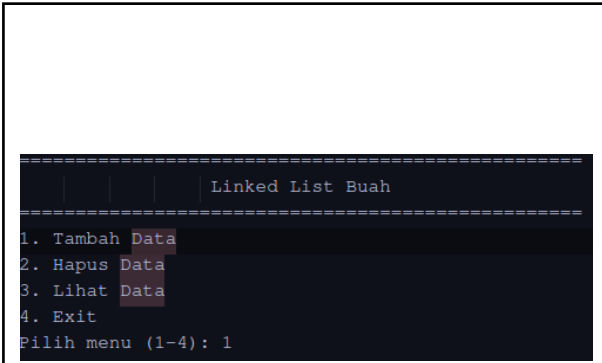
    tampilNode();

    char option;
    do {
        cout << "(q) for quit: "; cin >> option;
    } while(option != 'q');

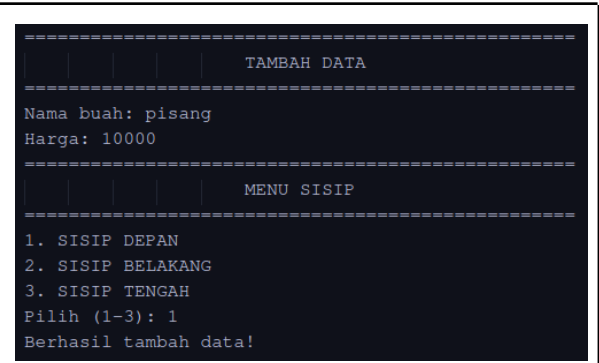
}

```

2. Screenshot



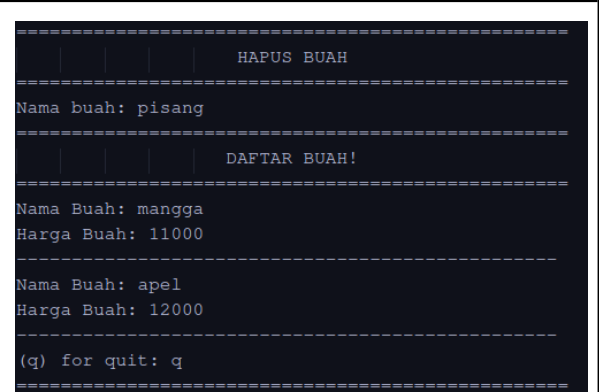
Gambar 1. Awal Menu



Gambar 2. Tambah Data



Gambar 3. Daftar Buah



Gambar 4. Hapus Buah

