LAPORAN PRAKTIKUM

Modul 5

Single Linked List Bagian Kedua

Disusun Oleh:

Hafizh Dwi Andhika Faruq -2311104013

S1SE-07-01

Dosen :

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. Code

1. Terdapat node yang berisi data yang bertipe integer dan pointer next. tambahNode digunakan untuk menambahkan elemen ke akhir list. Kemudian, searchElement digunakan untuk mencari elemen yang ada di dalam list dan juga menampilkan posisi serta alamatnya jika ditemukan.

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

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

void tambahNode_2311104013(Node*& head, int value) {
    Node* node = new Node();
    node->data = value;
    node->next = NULL;

    if (head == NULL) {
        head = node;
    } else {
        Node* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = node;
    }
}

void searchElement_2311104013(Node* head, int i) {
    Node* current = head;
    int position = 1;

    while (current != NULL && current->data != i) {
        current = current->next;
        position++;
    }

    if (current != NULL && current->data == i) {
        cout << "Elemen ditemukan pada alamat: " << current << " dan posisi: urutan ke- " << position << endl;
    } else {
        cout << "Elemen dengan nilai: " << i << " tidak ada dalam list" << endl;
    }
}

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

    for (int i = 0; i < 6; i++) {
        cout << "Masukan elemen ke-" << i+1 << ": ";
        cin >> value;
        tambahNode_2311104013(head, value);
    }

    int cariElemen;
    cout << "Masukan elemen yang ingin dicari: ";
    cin >> cariElemen;
    searchElement_2311104013(head, cariElemen);

    return 0;
}
```

2. Insert digunakan untuk menambahkan elemen ke akhir list. Kemudian bubbleSortList digunakan untuk mengurutkan elemen di dalam list dengan algoritma bubble sort.

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

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

void insert_2311104013(Node*& head, int value) {
    Node* node = new Node();
    node->data = value;
    node->next = NULL;

    if (head == NULL) {
        head = node;
    } else {
        Node* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = node;
    }
}

void bubbleSortList_2311104013(Node* head) {
    bool swapped;
    Node* current;
    Node* lastSorted = NULL;

    if (head == NULL) {
        return;
    }

    do {
        swapped = false;
        current = head;

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

void displayList_2311104013(Node* head) {
    Node* temp = head;
    while (temp != NULL) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

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

    for (int i = 0; i < 5; i++) {
        cout << "Elemen ke-" << i+1 << ": ";
        cin >> value;
        insert_2311104013(head, value);
    }

    cout << "List sebelum diurutkan: ";
    displayList_2311104013(head);

    bubbleSortList_2311104013(head);

    cout << "List setelah diurutkan: ";
    displayList_2311104013(head);

    return 0;
}
```

3. insertSorted ada untuk menempatkan elemen baru pada posisi yang tepat pada linked list, sehingga list akan selalu terurut. displayList digunakan untuk menampilkan seluruh elemen yang ada di dalam list.

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

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

void insertSorted_2311104013(Node*& head, Node* node) {
    Node* Q = head;
    Node* prev = NULL;
    bool found = false;

    while (Q != NULL&& !found) {
        if (Q->data > node->data) {
            found = true;
        } else {
            prev = Q;
            Q = Q->next;
        }
    }

    if (prev == NULL) {
        node->next = head;
        head = node;
    } else if (Q == NULL) {
        prev->next = node;
        node->next = NULL;
    } else {
        prev->next = node;
        node->next = Q;
    }
}

void displayList_2311104013(Node* head) {
    Node* temp = head;
    while (temp != NULL) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

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

    for (int i = 0; i < 4; i++) {
        cout << "Elemen ke-" << i+1 << ": ";
        cin >> value;

        Node* node = new Node();
        node->data = value;
        node->next = NULL;


        insertSorted_2311104013(head, node);
    }

    cout << "List setelah diurutkan: ";
    displayList_2311104013(head);

    return 0;
}
```