

LAPORAN PRAKTIKUM
Modul 05
“SINGLE LINKED LIST (BAGIAN KEDUA)”



Disusun Oleh:
M.Faris (2311104017)
SE-07-01

Dosen :
Yudha Islami Sulistiya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO
2024

A. Soal Tugas Pendahuluan

1. Buatlah program yang memungkinkan pengguna memasukkan 6 elemen integer ke dalam list. Implementasikan function searchElement untuk mencari apakah sebuah nilai tertentu ada dalam list

Code :

```
#include <iostream>
using namespace std;
struct Node {
    int data;
    Node *next;
};
class LinkedList{
public:
    Node *head;
    LinkedList(){
        head = nullptr;
    }
    void tambahElemen(int nilai){
        Node *newNode = new Node();
        newNode->data = nilai;
        newNode->next = nullptr;
        if(head == nullptr){
            head = newNode;
        } else {
            Node *temp = head;
            while (temp->next != nullptr){
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }

    void searchElement(int nilai){
        Node *current = head;
        int position = 1;
        bool found = false;
        while (current != nullptr){
            if (current->data == nilai){
                found = true;
                cout << "Elemen " << nilai << "ditemukan di alamat " << current << " urutan ke: " << position << endl;
                break;
            }
            current = current->next;
            position++;
        }
        if (!found) {
            cout << "Elemen " << nilai << "tidak ditemukan di list\n";
        }
    }

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

```

int main(){
    LinkedList list;
    int nilaicari;

    cout << "Masukan 6 elemen int :\n";
    for (int i = 0; i < 6; i++){
        int elemen;
        cout << "Elemen ke: " << i + 1 << ": ";
        cin >> elemen;
        list.tambahElemen(elemen);
    }

    cout << "\nelemen dalam list:\n";
    list.tampilkanList();

    cout << "Masukan nilai yang dicari: ";
    cin >> nilaicari;

    list.searchElement(nilaicari);

    return 0;
}

```

Output:

```

Masukan 6 elemen int :
Elemen ke: 1: 10
Elemen ke: 2: 5
Elemen ke: 3: 15
Elemen ke: 4: 20
Elemen ke: 5: 30
Elemen ke: 6: 25

elemen dalam list:
10->5->15->20->30->25->Masukan nilai yang dicari: 15
Elemen 15ditemukan di alamat 0x2357a090c50 urutan ke: 3
Press any key to continue

```

Penjelasan :

Jadi code tersebut adalah linked list yang mengimplementasikan SearchElement untuk mencari nilai tertentu beserta posisi dan alamat memorinya. Di program tersebut juga ada method tambahElemen untuk menambah elemen di akhir list dan tampilkan list untuk menampilkan seluruh elemen.

Pada fungsi main program meminta user untuk memasukan 6 elemen integer untuk dimasukan ke dalam linked list, lalu menampilkan semua elemen didalamnya. Kemudian user dapat mencari nilai tertentu dalam list

2. Buatlah program yang mengizinkan pengguna memasukkan 5 elemen integer ke dalam list. Implementasikan procedure bubbleSortList untuk mengurutkan elemen-elemen dalam list dari nilai terkecil ke terbesar

Code:

```

#include <iostream>
using namespace std;
struct Node {
    int data;
    Node *next;
};
class LinkedList{
public:
    Node *head;

    LinkedList(){
        head = nullptr;
    }
    void tambahElemen(int nilai){
        Node *newNode = new Node();
        newNode->data = nilai;
        newNode->next = nullptr;
        if(head == nullptr){
            head = newNode;
        } else {
            Node *temp = head;
            while (temp->next != nullptr){
                temp = temp->next;
            }
            temp->next = newNode;
        }
    }
    void tampilkanList(){
        Node *temp = head;
        while (temp != nullptr){
            cout << temp -> data << "->";
            temp = temp-> next;
        }
        cout << "NULL\n";
    }
}

```

```

    void bubbleSortList(){
        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){
                    swap(current->data, current->next->data);
                    swapped = true;
                }
                current = current->next;
            }
            lastptr = current;
        } while (swapped);
    }
};

```

```

int main(){
    LinkedList List;
    int elemen;

    cout << "Masukan 5 elemen int ke list:\n";
    for (int i = 0; i < 5; i++){
        cout << "Elemen ke-" << i + 1 << " ";
        cin >> elemen;
        List.tambahElemen(elemen);
    }

    cout << "list sebelum diurutkan:\n";
    List.tampilkanList();

    List.bubbleSortList();

    cout << "list setelah diurutkan: \n";
    List.tampilkanList();

    return 0;
}

```

Output:

```

Masukan 5 elemen int ke list:
Elemen ke-1:10
Elemen ke-2:5
Elemen ke-3:20
Elemen ke-4:30
Elemen ke-5:25
list sebelum diurutkan:
10->5->20->30->25->NULL
list setelah diurutkan:
5->10->20->25->30->NULL
Press any key to continue

```

Penjelasan :

Jadi code tersebut adalah linked list yang mengimplementasikan procedure Bubble sort untuk mengurutkan elemen dalam list secara ascending menggunakan bubble sort. Diprogram tersebut juga ada method tambahElemen untuk menambah elemen baru diakhir list dan tampilkan list untuk menampilkan semua elemen

Pada fungsi main program meminta user untuk memasukan 5 elemen integer yang akan disimpan di linked list lalu menampilkan isi list sebelum dan sesudah pengurutan.

3. Buatlah program yang mengizinkan pengguna memasukkan 4 elemen integer ke dalam list secara manual. Kemudian, minta pengguna memasukkan elemen tambahan yang harus ditempatkan di posisi yang sesuai sehingga list tetap terurut

Code :

```

dal_n03.cpp > ...
#include <iostream>
using namespace std;

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

class LinkedList{
public:
    Node * head;

    LinkedList(){
        head = nullptr;
    }

    void insertSorted(int nilai){
        Node *newNode = new Node();
        newNode->data = nilai;
        newNode->next = nullptr;

        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 tampilkanList(){
        Node *temp = head;
        while(temp != nullptr){
            cout << temp->data << "->";
            temp = temp->next;
        }
        cout << "NULL\n";
    }
};

```

```

int main() {
    LinkedList list;
    int elemen;
    cout << "Masukkan 4 elemen int ke list secaraurut:\n";
    for (int i = 0; i < 4; i++) {
        cout << "Elemen ke-" << i + 1 << ": ";
        cin >> elemen;
        list.insertSorted(elemen);
    }
    cout << "\nList setelah memasukkan 4 elemen:\n";
    list.tampilkanList();

    cout << "\nMasukkan elemen int yang ingin ditambahkan: ";
    cin >> elemen;
    list.insertSorted(elemen);

    cout << "\nList setelah menambahkan elemen baru:\n";
    list.tampilkanList();

    return 0;
}

```

Output:

```
Masukkan 4 elemen int ke list secara rurut:
Elemen ke-1: 10
Elemen ke-2: 5
Elemen ke-3: 20
Elemen ke-4: 15

List setelah memasukkan 4 elemen:
5->10->15->20->NULL

Masukkan elemen int yang ingin ditambahkan: 12

List setelah menambahkan elemen baru:
5->10->12->15->20->NULL

Press any key to continue
```

Penjelasan:

Jadi code tersebut adalah linked list yang mengimplementasikan procedure insertSorted untuk menambahkan elemen baru ke dalam list yang sudah terurut

Method insertSorted fungsinya untuk menyisipkan elemen baru dalam urutan terurut jika head kosong atau elemen baru lebih kecil atau sama dengan head , elemen akan dimasukan di depan list. Jika tidak program akan mencari posisi yang sesuai di list agar urutannya tetap ascending.

Pada fungsi main program akan meminta pengguna untuk memasukan 4 elemen integer yang langsung diututkan dalam list. Setelah itu, elemen baru dapat ditambahkan dan list akan tetap urut saat menampilkan hasilnya.