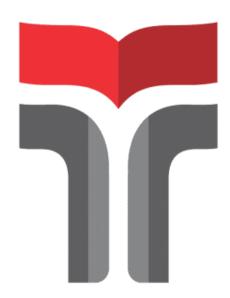
LAPORAN PRAKTIKUM Modul 5 Single Linked List Bagian Kedua



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.

```
• • •
     int data:
    Node* next;
    Node* node = new Node();
node→data = value;
node→next = NULL;
     if (head = NULL) {
          Node* temp = head;
          while (temp\rightarrownext \neq NULL) {
             temp = temp→next;
          temp \rightarrow next = node;
void searchElement_2311104013(Node* head, int i) {
     Node* current = head;
     while (current \neq NULL && current\rightarrowdata \neq i) {
          current = current→next;
position++;
    if (current \neq NULL 65 current\rightarrowdata = i) { cout \ll "Elemen ditemukan pada alamat: " \ll current \ll " dan posisi: urutan ke- " \ll position \ll endl;
          cout << "Elemen dengan nilai: " << i << " tidak ada dalam list" << endl;</pre>
     Node* head = nullptr;
     for (int i = 0; i < 6; i++) { cout << "Masukan elemen ke-" << i+1 << ": ";
          cin >> value;
tambahNode_2311104013(head, value);
     int cariElemen;
                "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.

```
. .
       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 \rightarrow next = node;
       bool swapped;
       Node* current;
Node* lastSorted = NULL;
              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);
       d displayList_231104013(Node* r
Node* temp = head;
while (temp ≠ NULL) {
   cout ≪ temp→data ≪ " ";
   temp = temp→next;
        cout << endl:
int main() {
   Node* head = NULL;
       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);</pre>
       bubbleSortList_2311104013(head);
        displayList_2311104013(head);
```

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.

```
. . .
#include <iostream>
using namespace std;
struct Node {
      int data;
      Node* next;
      Node* Q = head;
     Node* prev = NULL;
bool found = false;
      while (Q \neq NULL&& !found) {
            if (Q→data > node→data) {
                  found = true;
            } else {
    prev = Q;
    Q = Q→next;
            node→next = head;
      head = node;
} else if (Q = NULL) {
prev→next = node;
node→next = NULL;
      } else {
           prev→next = node;
node→next = Q;
     Node* temp = head;
while (temp ≠ NULL) {
  cout << temp→data << " ";
  temp = temp→next;
      cout << endl;</pre>
int main() {
   Node* head = NULL;
   int value;
      for (int i = 0; i < 4; i++) {
    cout << "Elemen ke-" << i+1 << ": ";
            Node* node = new Node();
node→data = value;
node→next = NULL;
            insertSorted_2311104013(head, node);
      displayList_2311104013(head);
      return 0;
                                                                                     snappify.com
```