

LAPORAN PRAKTIKUM
MODUL 5
SINGLE LINKED LIST BAGIAN 2



Nama :

Fajar Budiawan (2311104039)

Dosen :

Yudha Islami Sulistya S.Kom,
M.Cs

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

TUGAS PENDAHULUAN

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

```
1  #include <iostream>
2  using namespace std;
3
4  // Struktur node untuk menyimpan data integer
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 // Fungsi untuk menambahkan elemen ke dalam linked list
11 void tambahElemen(Node*& head, int data) {
12     Node* newNode = new Node();
13     newNode->data = data;
14     newNode->next = nullptr;
15
16     if (head == nullptr) {
17         head = newNode;
18     } else {
19         Node* temp = head;
20         while (temp->next != nullptr) {
21             temp = temp->next;
22         }
23         temp->next = newNode;
24     }
25 }
26
27 // Fungsi untuk mencari elemen dalam linked list
28 void searchElement(Node* head, int i) {
29     Node* current = head;
30     int position = 1;
31     bool found = false;
32
33     while (current != nullptr) {
34         if (current->data == i) {
35             cout << "Elemen " << i << " ditemukan pada alamat " << current << " di posisi ke-" << position << endl;
36             found = true;
37             break;
38         }
39         current = current->next;
40         position++;
41     }
42
43     if (!found) {
44         cout << "Elemen " << i << " tidak ada dalam list." << endl;
45     }
46 }
47
48 int main() {
49     Node* head = nullptr;
50     int value;
51
52     cout << "Masukkan 6 elemen integer ke dalam list:\n";
53     for (int i = 0; i < 6; ++i) {
54         cout << "Elemen ke-" << i + 1 << ": ";
55         cin >> value;
56         tambahElemen(head, value);
57     }
58
59     cout << "Masukkan nilai yang ingin dicari: ";
60     cin >> value;
61     searchElement(head, value);
62
63     return 0;
64 }
65
```

Output :

```
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5> cd 'd:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output'
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output> & .\program.exe
Masukkan 6 elemen integer ke dalam list:
Elemen ke-1: 3
Elemen ke-2: 4
Elemen ke-3: 5
Elemen ke-4: 6
Elemen ke-5: 1
Elemen ke-6: 2
Masukkan nilai yang ingin dicari: 4
Elemen 4 ditemukan pada alamat 0xfe6f28 di posisi ke-2
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output>
```

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

```

1  #include <iostream>
2  using namespace std;
3
4  // Struktur node untuk menyimpan data integer
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 // Fungsi untuk menambahkan elemen ke dalam linked list
11 void tambahElemen(Node*& head, int data) {
12     Node* newNode = new Node();
13     newNode->data = data;
14     newNode->next = nullptr;
15
16     if (head == nullptr) {
17         head = newNode;
18     } else {
19         Node* temp = head;
20         while (temp->next != nullptr) {
21             temp = temp->next;
22         }
23         temp->next = newNode;
24     }
25 }
26
27 // Fungsi untuk mengurutkan linked list menggunakan Bubble Sort
28 void bubbleSortList(Node* head) {
29     if (head == nullptr) return;
30
31     bool swapped;
32     Node* current;
33     Node* lastSorted = nullptr; // Menandai bagian akhir list yang sudah diurutkan
34
35     do {
36         swapped = false;
37         current = head;
38
39         while (current->next != lastSorted) {
40             if (current->data > current->next->data) {
41                 // Pertukaran data antar node
42                 int temp = current->data;
43                 current->data = current->next->data;
44                 current->next->data = temp;
45                 swapped = true;
46             }
47             current = current->next;
48         }
49         lastSorted = current; // Memperbarui posisi terakhir yang sudah diurutkan
50     } while (swapped);
51 }
52
53 // Fungsi untuk menampilkan elemen-elemen dalam linked list
54 void tampilkanList(Node* head) {
55     Node* temp = head;
56     while (temp != nullptr) {
57         cout << temp->data << " ";
58         temp = temp->next;
59     }
60     cout << endl;
61 }
62
63 int main() {
64     Node* head = nullptr;
65     int value;
66
67     cout << "Masukkan 5 elemen integer ke dalam list:\n";
68     for (int i = 0; i < 5; ++i) {
69         cout << "Elemen ke-" << i + 1 << ": ";
70         cin >> value;
71         tambahElemen(head, value);
72     }
73
74     cout << "\nList sebelum diurutkan: ";
75     tampilkanList(head);
76
77     bubbleSortList(head);
78
79     cout << "List setelah diurutkan: ";
80     tampilkanList(head);
81
82     return 0;
83 }
84

```

Output:

```
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5> cd 'd:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output'
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output>
Masukkan 5 elemen integer ke dalam list:
Elemen ke-1: 23
Elemen ke-2: 43
Elemen ke-3: 79
Elemen ke-4: 33
Elemen ke-5: 22

List sebelum diurutkan: 23 43 79 33 22
List setelah diurutkan: 22 23 33 43 79
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output>
```

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.

```

1  #include <iostream>
2  using namespace std;
3
4  // Struktur node untuk menyimpan data integer
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 // Fungsi untuk menambahkan elemen secara terurut dalam linked list
11 void insertSorted(Node*& head, int value) {
12     Node* newNode = new Node();
13     newNode->data = value;
14     newNode->next = nullptr;
15
16     // Jika list kosong atau nilai elemen baru lebih kecil dari head, masukkan di awal
17     if (head == nullptr || head->data >= value) {
18         newNode->next = head;
19         head = newNode;
20     } else {
21         Node* current = head;
22         // Menelusuri list untuk menemukan posisi yang tepat
23         while (current->next != nullptr && current->next->data < value) {
24             current = current->next;
25         }
26         // Menyisipkan elemen baru di antara current dan current->next
27         newNode->next = current->next;
28         current->next = newNode;
29     }
30 }
31
32 // Fungsi untuk menampilkan elemen-elemen dalam linked list
33 void tampilkanList(Node* head) {
34     Node* temp = head;
35     while (temp != nullptr) {
36         cout << temp->data << " ";
37         temp = temp->next;
38     }
39     cout << endl;
40 }
41
42 int main() {
43     Node* head = nullptr;
44     int value;
45
46     cout << "Masukkan 4 elemen integer secara terurut ke dalam list:\n";
47     for (int i = 0; i < 4; ++i) {
48         cout << "Elemen ke-" << i + 1 << ": ";
49         cin >> value;
50         insertSorted(head, value);
51     }
52
53     cout << "\nList setelah memasukkan 4 elemen: ";
54     tampilkanList(head);
55
56     // Memasukkan elemen tambahan secara terurut
57     cout << "\nMasukkan elemen tambahan yang ingin ditambahkan secara terurut: ";
58     cin >> value;
59     insertSorted(head, value);
60
61     cout << "List setelah memasukkan elemen tambahan: ";
62     tampilkanList(head);
63
64     return 0;
65 }
66

```

Output:

```
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5> cd 'd:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output'
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output> &
Masukkan 4 elemen integer secara terurut ke dalam list:
Elemen ke-1: 65
Elemen ke-2: 43
Elemen ke-3: 86
Elemen ke-4: 27

List setelah memasukkan 4 elemen: 27 43 65 86

Masukkan elemen tambahan yang ingin ditambahkan secara terurut: 98
List setelah memasukkan elemen tambahan: 27 43 65 86 98
PS D:\Praktikum Struktur Data\05_Pengenalan_CPP_Bagian_5\TP\output>
```