

LAPORAN PRAKTIKUM
PERTEMUAN 5
SINGLE LINKED LIST (BAGIAN KEDUA)



Nama :

GAHA ZIDANE NURRAIHAN (2311104038)

Dosen :

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

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

I. TP

1. Mencari Elemen Tertentu dalam SLL

berikut adalah subprogramnya :

```
TP > Soal1.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8  };
9
10 class LinkedList
11 {
12 public:
13     Node *head;
14
15     LinkedList()
16     {
17         head = nullptr;
18     }
19
20     // Menambahkan elemen baru di akhir list
21     void append(int new_data)
22     {
23         Node *new_node = new Node();
24         new_node->data = new_data;
25         new_node->next = nullptr;
26
27         if (head == nullptr)
28         {
29             head = new_node;
30             return;
31         }
32
33         Node *last = head;
34         while (last->next != nullptr)
35         {
36             last = last->next;
37         }
```

kode lanjutannya

```
38         last->next = new_node;
39     }
40
41     // Fungsi untuk mencari elemen dalam list
42     Node *searchElement(int i, int &position)
43     {
44         Node *current = head;
45         position = 1;
46
47         // Mencari elemen selama current tidak null
48         while (current != nullptr)
49         {
50             if (current->data == i)
51             {
52                 return current;
53             }
54             current = current->next;
55             position++;
56         }
57         return nullptr;
58     }
59 };
60
61 int main()
62 {
63     LinkedList list;
64     int element, position;
65
66     // Memasukkan 6 elemen ke dalam list
67     cout << "Masukkan 6 elemen integer ke dalam list:\n";
68     for (int j = 0; j < 6; j++)
69     {
```

```

70     cout << "Elemen ke-" << j + 1 << ": ";
71     cin >> element;
72     list.append(element);
73 }
74
75 // Meminta pengguna untuk memasukkan elemen yang akan dicari
76 cout << "\nMasukkan nilai yang ingin dicari: ";
77 cin >> element;
78
79 // Mencari elemen
80 Node *result = list.searchElement(element, position);
81
82 if (result != nullptr)
83 {
84     cout << "Elemen ditemukan pada alamat: " << result
85         << ", posisi: urutan ke-" << position << " dalam list.\n";
86 }
87 else
88 {
89     cout << "Elemen " << element << " tidak ditemukan dalam list.\n";
90 }
91
92 return 0;
93 }

```

kode outputnya yang dikeluarkan adalah sebagai berikut dan kode tersebut adalah alamat yang disimpan di memori yang persis kita inginkan.

```

PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2> cd 'c:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output'
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output> & .\Soal1.exe
Masukkan 6 elemen integer ke dalam list:
Elemen ke-1: 20
Elemen ke-2: 12
Elemen ke-3: 70
Elemen ke-4: 80
Elemen ke-5: 90
Elemen ke-6: 14

Masukkan nilai yang ingin dicari: 12
Elemen ditemukan pada alamat: 0x29b0db4ff30, posisi: urutan ke-2 dalam list.
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output>

```

2. Mengurutkan List Menggunakan Bubble Sort :

```

TP > C:\soal2.cpp > ...
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;

```

Kode lanjutannya:

```

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
72         tambahElemen(head, value);
73     }
74
75     cout << "\nList sebelum diurutkan: ";
76     tampilkanList(head);
77
78     bubbleSortList(head);
79
80     cout << "List setelah diurutkan: ";
81     tampilkanList(head);
82
83     return 0;
84 }

```

Kode output yang dikeluarkan adalah seperti ini:

```

PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL  PORTS
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2> cd 'c:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output'
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output> & .\soal2.exe
Masukkan 5 elemen integer ke dalam list:
Elemen ke-1: 45
Elemen ke-2: 55
Elemen ke-3: 67
Elemen ke-4: 78
Elemen ke-5: 34

List sebelum diurutkan: 45 55 67 78 34
List setelah diurutkan: 34 45 55 67 78
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output>

```

3. Menambahkan Elemen Secara Terurut

```
TP > soal3.cpp > Node > data
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     }
```

Kode lanjutannya

```
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 }
```

kode outputnya adalah:

```
Elemen ke-1: 23
Elemen ke-2: 12
Elemen ke-3: 45
Elemen ke-4: 67
```

```
List setelah memasukkan 4 elemen: 12 23 45 67
```

```
Masukkan elemen tambahan yang ingin ditambahkan secara terurut: 67
```

```
list setelah memasukkan elemen tambahan: 12 23 45 67 67
```

```
PS C:\Users\VICTUS\OneDrive\LAPRAK STD\STD_Ghaza_Zidane_Nurraihan_2311104038\05_Single_lingked_List_Bagian 2\TP\output>
```