

LAPORAN PRAKTIKUM
MODUL V
“SINGLE LINKED LIST BAGIAN 2”



Nama :

Mei Sari Mantiantini(2311104012)

Dosen :

YUDHA ISLAMI S, S.kom, M.cs

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

Tugas 1

```
05_Single_Linked_List_Bagian_2 > TP > G+ tugas1.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10 // Function to create a new node
11 Node* newNode(int data) {
12     Node* node = new Node;
13     node->data = data;
14     node->next = nullptr;
15     return node;
16 }
17
18 // Function to insert a new node at the beginning of the list
19 Node* insertFirst_2311104012(Node* head, int data) {
20     Node* new_node = newNode(data);
21     new_node->next = head;
22     return new_node;
23 }
24
25 // Function to search for an element in the linked list
26 Node* searchElement_2311104012(Node* head, int data) {
27     int position = 1;
28     Node* current = head;
29
30     while (current != nullptr) {
31         if (current->data == data) {
32             cout << "Elemen " << data << " ditemukan pada posisi " << position << endl;
33             return current;
34         }
35         current = current->next;
36         position++;
37     }
38     cout << "Elemen " << data << " tidak ditemukan" << endl;
39     return nullptr;
40 }
41
42 int main() {
43     Node* head = nullptr;
44
45     // Input elements from the user
46     cout << "Masukkan 6 elemen integer: " << endl;
47     for (int i = 0; i < 6; i++) {
48         int data;
49         cin >> data;
50         head = insertFirst_2311104012(head, data);
51     }
52
53     // Search for an element
54     int search_data;
55     cout << "Masukkan elemen yang ingin dicari: ";
56     cin >> search_data;
57
58     searchElement_2311104012(head, search_data);
59
60     return 0;
61 }
```

Output:

```
Masukkan 6 elemen integer:
1 2 3 4 5 6
Masukkan elemen yang ingin dicari: 3
Elemen 3 ditemukan pada posisi 4
PS D:\Praktikum_05\05_Single_Linked_List_Bagian_2\TP> |
```

Penjelasan:

Kode diatas mencakup beberapa fungsi, yaitu pembuatan node baru, penambahan node di awal linked list, dan pencarian elemen di dalam linked list. Fungsi newNode digunakan untuk membuat node baru dengan nilai data tertentu dan pointer next yang diinisialisasi ke nullptr. Fungsi insertFirst_2311104012 menambahkan node baru di awal list, menjadikan node baru tersebut sebagai head yang baru. Dalam fungsi searchElement_2311104012, kode mencari elemen dalam linked list dengan nilai yang diinginkan dan menampilkan posisinya jika ditemukan; jika tidak ditemukan, akan ditampilkan pesan bahwa elemen tersebut tidak ada dalam list. Di dalam main(), pengguna diminta memasukkan 6 elemen integer yang akan dimasukkan ke linked list, lalu memasukkan nilai elemen yang ingin dicari di dalam list.

Tugas 2

```
05_Single_Linked_List_Bagian_2 > TP > G+ tugas2.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  class Tugas2 {
10 private:
11     Node* head;
12
13 public:
14     Tugas2() {
15         head = nullptr;
16     }
17
18     void insert(int value) {
19         Node* newNode = new Node();
20         newNode->data = value;
21         newNode->next = nullptr;
22
23         if (head == nullptr) {
24             head = newNode;
25         } else {
26             Node* temp = head;
27             while (temp->next != nullptr) {
28                 temp = temp->next;
29             }
30             temp->next = newNode;
31         }
32     }
33
34     void bubbleSort() {
35         if (head == nullptr) return;
36
37         bool swapped;
38         do {
39             swapped = false;
```

```

40     Node* current = head;
41
42     while (current->next != nullptr) {
43         if (current->data > current->next->data) {
44             // Tukar data
45             int temp = current->data;
46             current->data = current->next->data;
47             current->next->data = temp;
48             swapped = true;
49         }
50         current = current->next;
51     }
52 } while (swapped);
53 }
54
55 void display() {
56     Node* temp = head;
57     while (temp != nullptr) {
58         cout << temp->data << " ";
59         temp = temp->next;
60     }
61     cout << endl;
62 }
63 };
64
65 int main() {
66     Tugas2 list;
67
68     // Insert contoh data
69     list.insert(5);
70     list.insert(2);
71     list.insert(9);
72     list.insert(1);
73     list.insert(7);
74
75     cout << "Sebelum sorting: ";
76     list.display();
77
78     list.bubbleSort();
79
80     cout << "Setelah sorting: ";
81     list.display();
82
83     return 0;
84 }

```

Output:

```

Sebelum sorting: 5 2 9 1 7
Setelah sorting: 1 2 5 7 9
PS D:\Praktikum_05\05_Single_Linked_List_Bagian_2\TP>

```

Penjelasan:

Kode di atas adalah implementasi dari linked list dengan penambahan fungsi pengurutan menggunakan algoritma Bubble Sort. Struktur Node menyimpan data integer dan pointer next yang menunjuk ke node berikutnya. Kelas Tugas2 mengelola linked list dengan tiga fungsi utama: insert, bubbleSort, dan display. Fungsi insert menambahkan node baru di akhir list, sedangkan bubbleSort mengurutkan data dalam

list secara ascending menggunakan algoritma Bubble Sort. Dalam Bubble Sort ini, dua elemen bertetangga ditukar jika elemen pertama lebih besar dari yang berikutnya, dan proses ini diulang sampai tidak ada lagi pertukaran yang diperlukan. Fungsi display digunakan untuk menampilkan semua elemen dalam linked list. Pada main, beberapa data contoh dimasukkan ke dalam list, kemudian data ditampilkan sebelum dan setelah proses pengurutan.

Tugas 3

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  Node* insertSorted_2311104012(Node* head, int data) {
10     Node* newNode = new Node;
11     newNode->data = data;
12     newNode->next = nullptr;
13
14     if (head == nullptr || data < head->data) {
15         newNode->next = head;
16         return newNode;
17     }
18
19     Node* current = head;
20     Node* previous = nullptr;
21
22     while (current != nullptr && data >= current->data) {
23         previous = current;
24         current = current->next;
25     }
26
27     newNode->next = current;
28     previous->next = newNode;
29     return head;
30 }
31
32 void printList(Node* head) {
33     Node* current = head;
34     while (current != nullptr) {
35         cout << current->data << " ";
36         current = current->next;
37     }
38     cout << endl;
39 }
```

```

41  int main() {
42      Node* head = nullptr;
43
44      int n;
45      cout << "Masukkan jumlah elemen: ";
46      cin >> n;
47
48      cout << "Masukkan " << n << " elemen integer: ";
49      for (int i = 0; i < n; i++) {
50          int data;
51          cin >> data;
52          head = insertSorted_2311104012(head, data);
53      }
54
55      cout << "List terurut: ";
56      printList(head);
57
58      int newElement;
59      cout << "Masukkan elemen baru: ";
60      cin >> newElement;
61
62      head = insertSorted_2311104012(head, newElement);
63
64      cout << "List terurut setelah menambahkan elemen baru: ";
65      printList(head);
66
67      return 0;
68  }

```

Output:

```

Masukkan jumlah elemen: 5
Masukkan 5 elemen integer: 1 2 3 4 5
List terurut: 1 2 3 4 5
Masukkan elemen baru: 7
List terurut setelah menambahkan elemen baru: 1 2 3 4 5 7
PS D:\Praktikum_05\05_Single_Linked_List_Bagian_2\TP>

```

Penjelasan:

Kode di atas adalah implementasi dari linked list yang mempertahankan elemen-elemen dalam keadaan terurut secara ascending. Fungsi `insertSorted_2311104012` menambahkan elemen baru ke dalam linked list secara terurut. Jika linked list kosong atau data elemen baru lebih kecil dari elemen di head, elemen baru ditempatkan di awal. Jika tidak, fungsi ini akan menelusuri linked list sampai menemukan posisi yang sesuai untuk menempatkan elemen baru, sehingga linked list tetap terurut setelah elemen ditambahkan. Fungsi `printList` digunakan untuk mencetak semua elemen di dalam linked list. Di dalam fungsi `main`, pengguna diminta memasukkan sejumlah elemen yang akan dimasukkan dan ditampilkan dalam bentuk list terurut. Setelah itu, pengguna dapat menambahkan elemen baru, dan hasilnya dicetak lagi untuk menunjukkan linked list setelah penambahan elemen.