

LAPORAN PRAKTIKUM
PERTEMUAN 1
SINGLE LINKED LIST BAGIAN 2



Nama :

Ardhian Dwi Saputra (2311104040)

Dosen :

Yudha Islami Sulistya

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

1. Program ini adalah contoh sederhana dari penggunaan linked list dalam C++. Ini menunjukkan bagaimana cara menambahkan elemen ke dalam list dan mencari elemen tertentu. Penggunaan linked list memungkinkan penambahan elemen secara dinamis, yang lebih fleksibel dibandingkan dengan array statis.

Source code :

```
Latihan01.cpp > createNode(int)
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  Node* createNode(int data) {
10     Node* newNode = new Node();
11     newNode->data = data;
12     newNode->next = nullptr;
13     return newNode;
14 }
15
16 void insertNode(Node*& head, int data) {
17     Node* newNode = createNode(data);
18     if (head == nullptr) {
19         head = newNode;
20     } else {
21         Node* temp = head;
22         while (temp->next != nullptr) {
23             temp = temp->next;
24         }
25         temp->next = newNode;
26     }
27 }
28
```

```

28
29 void searchElement(Node* head, int target) {
30     Node* current = head;
31     int position = 1;
32     bool found = false;
33
34     while (current != nullptr) {
35         if (current->data == target) {
36             cout << "Element " << target << " Ditemukan di Posisi " << position << " Dengan alamat " << current << endl;
37             found = true;
38             break;
39         }
40
41         current = current->next; position++;
42     }
43     if (!found) {
44         cout << "Elemen " << target << " Tidak ditemukan di dalam list." << endl;
45     }
46 }
47
48 int main () {
49     Node* head = nullptr;
50     int element;
51
52     cout << "Masukan 6 Elemen untuk List: " << endl;
53     for (int i = 0; i < 6; i++) {
54         cin >> element;
55         insertNode(head, element);
56     }
57
58     cout << "masukan elemen yang dicari: ";
59     cin >> element ;
60     searchElement(head, element);
61
62     return 0;
63 }

```

Output :

Jika elemen ditemukan maka outputnya akan seperti ini :

```

Masukan 6 Elemen untuk List:
1
2
3
4
5
6
masukan elemen yang dicari: 5
Element 5 Ditemukan di Posisi 5 Dengan alamat 0x799830
PS D:\SEMS 3\Cpp std\modul5> 

```

Jika elemen tidak ditemukan maka outputnya seperti ini:

```
Masukan 6 Elemen untuk List:
1
2
3
4
4
5
masukan elemen yang dicari: 6
Elemen 6 Tidak ditemukan di dalam list.
PS D:\SEMS 3\Cpp std\modul5> 
```

2. Program ini adalah contoh sederhana dari penggunaan linked list dalam C++. Ini menunjukkan cara menambahkan elemen ke dalam list, menampilkan isi list, dan mengurutkan list menggunakan Bubble Sort. Program ini berguna untuk memahami konsep dasar dari struktur data linked list dan algoritma pengurutan.
Source code:

Latihan02.cpp > main()

```
1  #include <iostream>
2  using namespace std;
3
4
5  struct Node {
6      int data;
7      Node* next;
8  };
9
10
11 void insertLast(Node*& head, int value) {
12     Node* newNode = new Node();
13     newNode->data = value;
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 }
```

```
27 void displayList(Node* head) {
28     Node* temp = head;
29     while (temp != nullptr) {
30         cout << temp->data << " ";
31         temp = temp->next;
32     }
33     cout << endl;
34 }
35
36 void bubbleSortList(Node* head) {
37     if (head == nullptr) return;
38
39     bool swapped;
40     Node* current;
41     Node* lastPtr = nullptr;
42
43     do {
44         swapped = false;
45         current = head;
46
47         while (current->next != lastPtr) {
48             if (current->data > current->next->data) {
```

```

50         int temp = current->data;
51         current->data = current->next->data;
52         current->next->data = temp;
53         swapped = true;
54     }
55     current = current->next;
56 }
57 lastPtr = current;
58 } while (swapped);
59 }
60
61 int main() {
62     Node* head = nullptr;
63
64     cout << "Masukkan 5 elemen integer ke dalam list: " << endl;
65     for (int i = 0; i < 5; i++) {
66         int element;
67         cout << "Elemen ke-" << (i + 1) << ": ";
68         cin >> element;
69         insertLast(head, element);
70     }
71
72     cout << "List sebelum diurutkan: ";
73     displayList(head);
74
75     bubbleSortList(head);
76
77     cout << "List setelah diurutkan: ";
78     displayList(head);
79
80     return 0;
81 }
82

```

Outputnya:

```

PS D:\SEMS 3\Cpp std\modul5> & 'c:\Users\lenovo\.vscode\extensions\ms-vscode.cpptools-1.22.10-win32-x64\debugAdapters\bin\
WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2eyvsntl.aji' '--stdout=Microsoft-MIEngine-Out-qnpovdqa.dqh' '--st
denr=Microsoft-MIEngine-Error-1bjirhea.v2a' '--pid=Microsoft-MIEngine-Pid-mgof5ofm.eqy' '--dbgExe=C:\msys64\ucrt64\bin\gdb.
exe' '--interpreter=mi'
Masukkan 5 elemen integer ke dalam list:
Elemen ke-1: 4
Elemen ke-2: 2
Elemen ke-3: 1
Elemen ke-4: 5
Elemen ke-5: 3
List sebelum diurutkan: 4 2 1 5 3
List setelah diurutkan: 1 2 3 4 5

```

3. Program ini adalah contoh penggunaan linked list dengan penambahan elemen yang terurut. Ini berguna untuk memahami cara menyisipkan elemen ke dalam struktur data sambil mempertahankan urutan. Program ini juga menunjukkan dasar-dasar manipulasi linked list dalam C++.

G+ Latihan03.cpp > ...

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  void insertSorted(Node*& head, int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = nullptr;
13
14     if (head == nullptr || head->data >= value) {
15         newNode->next = head;
16         head = newNode;
17     } else {
18         Node* temp = head;
19
20         while (temp->next != nullptr && temp->next->data < value) {
21             temp = temp->next;
22         }
23         newNode->next = temp->next;
24         temp->next = newNode;
25     }
26 }
27
```

```
28 void displayList(Node* head) {
29     Node* temp = head;
30     while (temp != nullptr) {
31         cout << temp->data << " ";
32         temp = temp->next;
33     }
34     cout << endl;
35 }
36
```

```
37 int main() {
38     Node* head = nullptr;
39     int element;
40
41     cout << "Masukkan 4 elemen integer secara terurut: " << endl;
42     for (int i = 0; i < 4; i++) {
43         cout << "Elemen ke-" << (i + 1) << ": ";
44         cin >> element;
45         insertSorted(head, element);
46     }
47
48     cout << "List sebelum menambahkan elemen baru: ";
49     displayList(head);
50
51     cout << "Masukkan elemen tambahan untuk dimasukkan: ";
52     cin >> element;
53
54     insertSorted(head, element);
55     cout << "List setelah menambahkan elemen baru: ";
56     displayList(head);
57 }
```

Outputnya sebagai berikut :

```
Masukkan 4 elemen integer secara terurut:  
Elemen ke-1: 1  
Elemen ke-2: 2  
Elemen ke-3: 3  
Elemen ke-4: 4  
List sebelum menambahkan elemen baru: 1 2 3 4  
Masukkan elemen tambahan untuk dimasukkan: 5  
List setelah menambahkan elemen baru: 1 2 3 4 5  
PS D:\SEMS 3\Cpp std\modul5> 
```