

LAPORAN PRAKTIKUM
PERTEMUAN 1
SINGLE LINKED LIST BAGIAN 1



Nama :

Ardhian Dwi Saputra (2311104040)

Dosen :

Yudha Islami Sulistya

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

1. Program C++ yang Anda berikan adalah implementasi dari doubly linked list (daftar terhubung ganda). Dalam struktur ini, setiap node memiliki dua pointer: satu untuk node sebelumnya (prev) dan satu untuk node berikutnya (next).

Source code :

D: > SEMS 3 > Cpp std > modul6 > TP >  Tp1.cpp >  DoublyLinkedList >  head

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* prev;
7      Node* next;
8  };
9
10 struct DoublyLinkedList {
11     Node* head;
12     Node* tail;
13 };
14
15
16 Node* createNode(int data) {
17     Node* newNode = new Node();
18     newNode->data = data;
19     newNode->prev = newNode->next = nullptr;
20     return newNode;
21 }
22
23 void initList(DoublyLinkedList &dll) {
24     dll.head = dll.tail = nullptr;
25 }
26
27
```

```
28 void insertFirst(DoublyLinkedList &dll, int data) {
29     Node* newNode = createNode(data);
30     if (dll.head == nullptr) {
31         dll.head = dll.tail = newNode;
32     } else {
33         newNode->next = dll.head;
34         dll.head->prev = newNode;
35         dll.head = newNode;
36     }
37 }
38
39
```

```

40 void insertLast(DoublyLinkedList &dll, int data) {
41     Node* newNode = createNode(data);
42     if (dll.tail == nullptr) {
43         dll.head = dll.tail = newNode;
44     } else {
45         newNode->prev = dll.tail;
46         dll.tail->next = newNode;
47         dll.tail = newNode;
48     }
49 }
50
51
52 void displayForward(DoublyLinkedList dll) {
53     Node* temp = dll.head;
54     while (temp != nullptr) {
55         cout << temp->data;
56         if (temp->next != nullptr) {
57             cout << " <-> ";
58         }
59         temp = temp->next;
60     }
61     cout << endl;
62 }
63

```

```

64 int main() {
65     DoublyLinkedList dll;
66     initList(dll);
67
68
69     insertFirst(dll, 10);
70     insertFirst(dll, 5);
71     insertLast(dll, 20);
72
73     cout << "DAFTAR ANGGOTA LIST: ";
74     displayForward(dll);
75
76     return 0;
77 }

```

Output :

```

PS D:\SEMS 3\C++\modul5> & "c:\Users\lenovo\.vscode\extensions\ms-vscode.cpptools-1.22.10-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe"
"--stdin=Microsoft-MIEngine-In-neaaps3.zyi" "--stdout=Microsoft-MIEngine-Out-wndpbma0.sx4" "--stderr=Microsoft-MIEngine-Error-t3lkvcvq.iys"
"--pid=Microsoft-MIEngine-Pid-t1phtfql.zn1" "--dbgExe=C:\msys64\ucrt64\bin\gdb.exe" "--interpreter=mi"
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20

```

2. Kode ini memberikan implementasi dasar dari doubly linked list dengan kemampuan untuk menambah dan menghapus elemen dari kedua ujung, serta menampilkan isi daftar.

Source code:

```
D: > SEMS 3 > Cpp std > modul6 > TP > Tp2.cpp > Node > data
1  #include <iostream>
2
3  using namespace std;
4
5  struct Node {
6      int data;
7      Node* prev;
8      Node* next;
9  };
10
11  Node* head = nullptr;
12
13  void insertLast_2311104040 (int data) {
14      Node* newNode = new Node();
15      newNode->data = data;
16      newNode->next = nullptr;
17
18      if (head == nullptr) {
19          newNode->prev = nullptr;
20          head = newNode;
21          return;
22      }
23
24      Node* temp = head;
25      while (temp->next != nullptr) {
26          temp = temp->next;
27      }
28
29      temp->next = newNode;
30      newNode->prev = temp;
31  }
```

```
33  void deleteFirst() {
34      if (head == nullptr) {
35          cout << "List kosong, tidak ada yang dihapus." << endl;
36          return;
37      }
38
39      Node* temp = head;
40      head = head->next;
41
42      if (head != nullptr) {
43          head->prev = nullptr;
44      }
45
46      delete temp;
47  }
48
49  void deleteLast() {
50      if (head == nullptr) {
51          cout << "List kosong, tidak ada yang dihapus" << endl;
52          return;
53      }
54
55      Node* temp = head;
56
57      if (temp->next == nullptr) {
58          head = nullptr;
59          delete temp;
60          return;
61      }
62      while (temp->next != nullptr) {
63          temp = temp->next;
64      }
65      temp->prev->next = nullptr;
66      delete temp;
67  }
```

```

69 void displayList() {
70     Node* temp = head;
71     cout << "Daftar Anggota List : ";
72     while (temp != nullptr) {
73         cout << temp->data;
74         if (temp->next != nullptr) {
75             cout << " <-> ";
76         }
77         temp = temp->next;
78     }
79     cout << endl;
80 }
81
82 int main() {
83     int data;
84
85     cout << "Masukan Elemen Pertama : ";
86     cin >> data;
87     insertLast_2311104040(data);
88
89     cout << "Masukan Elemen Kedua di akhir : ";
90     cin >> data;
91     insertLast_2311104040(data);
92
93     cout << "Masukan Elemen Ketiga di akhir : ";
94     cin >> data;
95     insertLast_2311104040(data);
96
97     deleteFirst();
98
99     deletelast();
100
101     displayList();
102
103     return 0;
104 }

```

Outputnya:

```

PS D:\SEMS 3\Cpp std\modul5> & 'c:\Users\lenovo\.vscode\ex
hr4j3e4m.0w5' '--stdout=Microsoft-MIEngine-Out-pq54yx23.nb0
\bin\gdb.exe' '--interpreter=mi'
Masukan Elemen Pertama : 23
Masukan Elemen Kedua di akhir : 41
Masukan Elemen Ketiga di akhir : 77
Daftar Anggota List : 41
PS D:\SEMS 3\Cpp std\modul5> 

```

3. Kode ini memberikan implementasi dasar dari doubly linked list dengan kemampuan untuk menambah dan menghapus elemen dari kedua ujung, serta menampilkan isi daftar.

```

D: > SEMS 3 > Cpp std > modul6 > TP > Tp3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* prev;
7      Node* next;
8  };
9
10 void insertLast_2311104040(Node*& head, int data) {
11     Node* newNode = new Node();
12     newNode->data = data;
13     newNode->next = nullptr;
14
15     if (head == nullptr) {
16         newNode->prev = nullptr;
17         head = newNode;
18         return;
19     }
20
21     Node* temp = head;
22     while (temp->next != nullptr) {
23         temp = temp->next;
24     }
25
26     temp->next = newNode;
27     newNode->prev = temp;
28 }
29
30 void displayForward(Node* head) {
31     Node* temp = head;
32     cout << "Daftar elemen dari depan ke belakang: ";
33     while (temp != nullptr) {
34         cout << temp->data;
35         if (temp->next != nullptr) {
36             cout << " <-> ";
37         }
38         temp = temp->next;
39     }
40     cout << endl;
41 }

```

```

43 void displayBackward(Node* head) {
44     if (head == nullptr) return;
45
46     Node* temp = head;
47     while (temp->next != nullptr) {
48         temp = temp->next;
49     }
50
51     cout << "Daftar elemen dari belakang ke depan: ";
52     while (temp != nullptr) {
53         cout << temp->data;
54         if (temp->prev != nullptr) {
55             cout << " <-> ";
56         }
57         temp = temp->prev;
58     }
59     cout << endl;
60 }
61
62 int main() {
63     Node* head = nullptr;
64     int n, data;
65
66     cout << "Masukkan 4 elemen secara berurutan: ";
67     for (int i = 0; i < 4; i++) {
68         cin >> data;
69         insertLast_2311104040(head, data);
70     }
71
72     displayForward(head);
73     displayBackward(head);
74 }

```

Outputnya sebagai berikut :

```
PS D:\SEMS 3\Cpp std\modul5> & 'c:\Users\lenovo\.vscode\extensions\ms-  
gine-In-caivrg0b.gk1' '--stdout=Microsoft-MIEngine-Out-runggi01.30k' '-  
C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'  
Masukkan 4 elemen secara berurutan: 1  
2  
3  
4  
Daftar elemen dari depan ke belakang: 1 <-> 2 <-> 3 <-> 4  
Daftar elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1  
PS D:\SEMS 3\Cpp std\modul5> 
```