

LAPORAN PRAKTIKUM
MODUL VI
“DOUBLE LINKED LIST BAGIAN 1”



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

```
1  #include <iostream>
2  using namespace std;
3
4  // Struktur Node untuk menyimpan data doubly linked list
5  struct Node {
6      int data;
7      Node* prev;
8      Node* next;
9  };
10
11 // Fungsi untuk menambahkan elemen di awal list
12 void insertFirst_2311104012(Node** head, int newData) {
13     Node* newNode = new Node();
14     newNode->data = newData;
15     newNode->prev = nullptr;
16     newNode->next = *head;
17
18     if (*head != nullptr) {
19         (*head)->prev = newNode;
20     }
21
22     *head = newNode;
23 }
24
25 // Fungsi untuk menambahkan elemen di akhir list
26 void insertLast_2311104012(Node** head, int newData) {
27     Node* newNode = new Node();
28     newNode->data = newData;
29     newNode->next = nullptr;
30
31     if (*head == nullptr) {
32         newNode->prev = nullptr;
33         *head = newNode;
34         return;
35     }
36
37     Node* last = *head;
38     while (last->next != nullptr) {
39         last = last->next;
```

```

40     }
41
42     last->next = newNode;
43     newNode->prev = last;
44 }
45
46 // Fungsi untuk menampilkan elemen dari depan ke belakang
47 void printList_2311104012(Node* node) {
48     Node* last = nullptr;
49     cout << "DAFTAR ANGGOTA LIST: ";
50     while (node != nullptr) {
51         cout << node->data;
52         last = node;
53         node = node->next;
54         if (node != nullptr) {
55             cout << " <-> ";
56         }
57     }
58     cout << endl;
59 }
60
61 int main() {
62     Node* head = nullptr;
63
64     // Input berdasarkan contoh pada soal
65     insertFirst_2311104012(&head, 10); // Elemen pertama = 10
66     insertFirst_2311104012(&head, 5);  // Elemen kedua di awal = 5
67     insertLast_2311104012(&head, 20);  // Elemen ketiga di akhir = 20
68
69     // Menampilkan daftar elemen dalam list
70     printList_2311104012(head);
71
72     return 0;
73 }

```

Output:

```

PS D:\Praktikum_06> cd "d:\Praktikum_06\06_Double_Linked_List_Bagian_1\TP\" ; if ($?)
DAFTAR ANGGOTA LIST: 5 <-> 10 <-> 20
PS D:\Praktikum_06\06_Double_Linked_List_Bagian_1\TP>

```

Penjelasan:

Struktur Node memiliki tiga elemen: data, prev (pointer ke node sebelumnya), dan next (pointer ke node berikutnya). Kode ini menyertakan tiga fungsi utama: insertFirst_2311104012, insertLast_2311104012, dan printList_2311104012. Fungsi insertFirst_2311104012 menambahkan node baru di awal list dengan mengatur pointer next dari node baru ke head yang lama dan mengatur prev dari head lama ke node baru. Fungsi insertLast_2311104012 menambahkan node di akhir list, mencari node terakhir, dan mengatur pointer yang sesuai sehingga node baru terhubung dengan benar di akhir. Fungsi printList_2311104012 mencetak semua elemen dari awal ke akhir, menampilkan daftar elemen yang dipisahkan dengan "<->" untuk menunjukkan hubungan dua arah. Di dalam main, tiga elemen ditambahkan untuk diuji: dua di awal (10 dan 5) dan satu di akhir (20), kemudian elemen-elemen tersebut ditampilkan.

Tugas 2

```
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;
```

```

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_06\06_Double_Linked_List_Bagian_1\TP>

```

Penjelasan:

Kode di atas adalah implementasi dari linked list yang menyertakan fungsi pengurutan menggunakan algoritma Bubble Sort dalam kelas Tugas2. Struktur Node menyimpan data integer dan pointer next yang menunjuk ke node berikutnya. Kelas Tugas2

memiliki atribut head yang menunjuk ke node pertama dalam list. Metode insert menambahkan node baru di akhir linked list, sementara bubbleSort mengurutkan elemen-elemen dalam list secara ascending menggunakan algoritma Bubble Sort. Dalam algoritma ini, setiap elemen diperiksa berurutan, dan jika elemen saat ini lebih besar dari elemen berikutnya, keduanya ditukar. Proses ini diulang sampai tidak ada lagi pertukaran yang diperlukan, yang berarti list telah terurut. Metode display mencetak semua elemen dalam linked list. Pada fungsi main, beberapa data contoh ditambahkan ke list, lalu ditampilkan sebelum dan setelah proses pengurutan.

Tugas 3

```
1  #include <iostream>
2
3  using namespace std;
4
5  struct Node {
6      int data;
7      Node* prev;
8      Node* next;
9  };
10
11 // Fungsi untuk membuat node baru
12 Node* newNode(int data) {
13     Node* node = new Node;
14     node->data = data;
15     node->prev = NULL;
16     node->next = NULL;
17     return node;
18 }
19
20 // Fungsi untuk menambahkan node ke awal list
21 Node* insertFirst_2311104012(Node* head, int data) {
22     Node* node = newNode(data);
23     if (head == NULL) {
24         return node;
25     }
26     node->next = head;
27     head->prev = node;
28     return node;
29 }
30
31 // Fungsi untuk menampilkan elemen dari depan ke belakang
32 void printForward_2311104012(Node* head) {
33     Node* current = head;
34     while (current != NULL) {
35         cout << current->data << " ";
36         current = current->next;
37     }
38     cout << endl;
39 }
```

```

41 // Fungsi untuk menampilkan elemen dari belakang ke depan
42 void printBackward_2311104012(Node* head) {
43     Node* current = head;
44     while (current->next != NULL) {
45         current = current->next;
46     }
47     while (current != NULL) {
48         cout << current->data << " ";
49         current = current->prev;
50     }
51     cout << endl;
52 }
53
54 int main() {
55     Node* head = NULL;
56     int n = 4;
57     int data;
58     cout << "Masukkan " << n << " elemen secara berurutan: ";
59     for (int i = 0; i < n; i++) {
60         cin >> data;
61         head = insertFirst_2311104012(head, data);
62     }
63     cout << "Daftar elemen dari depan ke belakang: ";
64     printForward_2311104012(head);
65     cout << "Daftar elemen dari belakang ke depan: ";
66     printBackward_2311104012(head);
67     return 0;
68 }

```

Output:

```

Masukkan 4 elemen secara berurutan: 1 6 8 10
Daftar elemen dari depan ke belakang: 10 8 6 1
Daftar elemen dari belakang ke depan: 1 6 8 10
PS D:\Praktikum_06\06_Double_Linked_List_Bagian_1\TP>

```

Penjelasan:

Kode di atas adalah implementasi dari doubly linked list (DLL) yang memiliki fungsi untuk menambahkan elemen di awal list serta menampilkan elemen dari kedua arah, yaitu dari depan ke belakang dan dari belakang ke depan. Struktur Node terdiri dari data integer, pointer prev yang menunjuk ke node sebelumnya, dan pointer next yang menunjuk ke node berikutnya. Fungsi newNode digunakan untuk membuat node baru dengan data yang diberikan. Fungsi insertFirst_2311104012 menambahkan node di awal linked list. Jika linked list kosong, node baru menjadi head; jika tidak, node baru ditambahkan di depan dan dihubungkan dengan node sebelumnya. Fungsi printForward_2311104012 mencetak elemen dari depan ke belakang, sedangkan printBackward_2311104012 mencetak elemen dari belakang ke depan dengan menavigasi ke node terakhir terlebih dahulu, kemudian mencetak dari node terakhir ke node pertama. Pada fungsi main, program meminta pengguna memasukkan empat elemen yang kemudian ditambahkan ke list, dan hasilnya ditampilkan dalam kedua arah.

