

**LAPORAN PRAKTIKUM**  
**PERTEMUAN 6**  
**DOUBLE LINKED LIST (BAGIAN PERTAMA)**



**Nama :**

Salman Alfarisi (2311104036)

**Dosen :**

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

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

## KODE TUGAS PENDAHULUAN 1

```
1 #include <iostream>
2 using namespace std;
3
4 class Node {
5 public:
6     int data;
7     Node* prev;
8     Node* next;
9
10    Node(int data) {
11        this->data = data;
12        prev = nullptr;
13        next = nullptr;
14    }
15 };
16
17 class DLL {
18 public:
19     Node* head;
20     Node* tail;
21
22     DLL() {
23         head = nullptr;
24         tail = nullptr;
25     }
26
27     void insertFirst_2311104036(int data) {
28         Node* p = new Node(data);
29
30         if (head == nullptr) {
31             head = p;
32             tail = p;
33         } else {
34             p->next = head;
35             p->prev = nullptr;
36             head->prev = p;
37             head = p;
38         }
39     }
40
41     void insertLast_2311104036(int data) {
42         Node* p = new Node(data);
43
44         if (head == nullptr) {
45             head = p;
46             tail = p;
47         } else {
48             p->prev = tail;
49             tail->next = p;
50             tail = p;
51         }
52     }
53 }
```

```
54 void inputanPengguna_2311104036() {
55     int dataInputan;
56     Node* current = head;
57
58     for (int i = 0; i < 1; i++) {
59         cout << "Masukkan data yang ingin dimasukkan di awal = ";
60         cin >> dataInputan;
61         insertFirst_2311104036(dataInputan);
62     }
63
64     for (int i = 0; i < 1; i++) {
65         cout << "Masukkan data yang ingin dimasukkan di akhir = ";
66         cin >> dataInputan;
67         insertLast_2311104036(dataInputan);
68     }
69 }
70
71
72 void display_2311104036() {
73     Node* current = head;
74
75     while (current != nullptr) {
76         cout << current->data << "<->";
77         current = current->next;
78     }
79
80     cout << "NULL" << endl;
81 }
82
83
84 int main() {
85     DLL dll;
86
87     dll.insertFirst_2311104036(10);
88     dll.display_2311104036();
89     dll.inputanPengguna_2311104036();
90     dll.display_2311104036();
91
92
93     return 0;
94 }
95
```

## OUTPUT

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Double_Linked_List_Bagian_1\TUGAS PENDAHULUAN\TP_1> cd "c:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Double_Linked_List_Bagian_1\TUGAS PENDAHULUAN\TP_1\" ; if ($?) { g++ tp_1.cpp -o tp_1 } ; if ($?) { .\tp_1 }
10<->NULL
Masukkan data yang ingin dimasukkan di awal = 5
Masukkan data yang ingin dimasukkan di akhir = 20
5<->10<->20<->NULL
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Double_Linked_List_Bagian_1\TUGAS PENDAHULUAN\TP_1>
```

## Penjelasan:

Kode di atas adalah implementasi dari kelas doubly linked list (DLL) yang mendukung operasi dasar seperti memasukkan dan menghapus elemen di awal dan akhir daftar. Setiap fungsi dalam kelas, seperti insertFirst, insertLast, deleteFirst, deleteLast, display, dan inputanPengguna, memiliki akhiran \_2311104036 untuk identifikasi khusus. Fungsi insertFirst\_2311104036 dan insertLast\_2311104036 digunakan untuk menambahkan elemen di awal dan akhir daftar, sementara deleteFirst\_2311104036 dan deleteLast\_2311104036 untuk menghapus elemen dari posisi yang sama. Fungsi display\_2311104036 digunakan untuk menampilkan elemen-elemen dalam linked list, dipisahkan dengan simbol <->, dan diakhiri dengan NULL. Selain itu, inputanPengguna\_2311104036 memungkinkan pengguna memasukkan data secara interaktif untuk ditambahkan di awal dan akhir linked list.

## KODE TUGAS PENDAHULUAN 2

```
1 #include <iostream>
2 using namespace std;
3
4 class Node {
5 public:
6     int data;
7     Node* prev;
8     Node* next;
9
10    Node(int data) {
11        this->data = data;
12        prev = nullptr;
13        next = nullptr;
14    }
15 };
16
17 class DLL {
18 public:
19     Node* head;
20     Node* tail;
21
22    DLL() {
23        head = nullptr;
24        tail = nullptr;
25    }
26
27    void insertFirst_2311104036(int data) {
28        Node* p = new Node(data);
29
30        if (head == nullptr) {
31            head = p;
32            tail = p;
33        } else {
34            p->next = head;
35            p->prev = nullptr;
36            head->prev = p;
37            head = p;
38        }
39    }
40
41    void insertLast_2311104036(int data) {
42        Node* p = new Node(data);
43
44        if (head == nullptr) {
45            head = p;
46            tail = p;
47        } else {
48            p->prev = tail;
49            tail->next = p;
50            tail = p;
51        }
52    }
53
54    void inputanPengguna_2311104036() {
55        int dataInputan;
56        Node* current = head;
57
58        for (int i = 0; i < 1; i++) {
59            cout << "Masukkan data yang ingin dimasukkan di awal = ";
60            cin >> dataInputan;
61            insertFirst_2311104036(dataInputan);
62        }
63
64        for (int i = 0; i < 1; i++) {
65            cout << "Masukkan data yang ingin dimasukkan di akhir = ";
66            cin >> dataInputan;
67            insertLast_2311104036(dataInputan);
68        }
69    }
70 }
```

```
71 void deleteFirst_2311104036() {
72     Node* temp = head;
73     head = head->next;
74     head->prev = nullptr;
75     delete temp;
76 }
77
78 void deleteLast_2311104036() {
79     Node* temp = tail;
80     tail = tail->prev;
81     tail->next = nullptr;
82     delete temp;
83 }
84
85 void display_2311104036() {
86     Node* current = head;
87
88     while (current != nullptr) {
89         cout << current->data << "<->";
90         current = current->next;
91     }
92
93     cout << "NULL" << endl;
94 }
95
96
97 int main() {
98     DLL dll;
99
100    dll.insertFirst_2311104036(10);
101    dll.display_2311104036();
102    dll.inputanPengguna_2311104036();
103    dll.display_2311104036();
104    dll.deleteFirst_2311104036();
105    dll.deleteLast_2311104036();
106    dll.display_2311104036();
107
108
109    return 0;
110 }
111
```

## OUTPUT

```
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Doub
6_Double_Linked_List_Bagian_1\TUGAS PENDAHULUAN\TP_2.cpp\" ;
10<->NULL
Masukkan data yang ingin dimasukkan di awal = 15
Masukkan data yang ingin dimasukkan di akhir = 20
15<->10<->20<->NULL
10<->NULL
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Doub
```

### Penjelasan:

Kode di atas adalah implementasi dari kelas doubly linked list (DLL) yang memiliki fungsi-fungsi dasar untuk menambahkan dan menghapus elemen di awal dan akhir daftar. Fungsi `insertFirst_2311104036` menambahkan elemen baru di awal daftar, sementara `insertLast_2311104036` menambahkannya di akhir. Fungsi `inputanPengguna_2311104036` memungkinkan pengguna memasukkan data secara interaktif untuk ditambahkan ke awal dan akhir linked list. `deleteFirst_2311104036` dan `deleteLast_2311104036` berfungsi untuk menghapus elemen pertama dan terakhir dalam daftar. Fungsi `display_2311104036` menampilkan semua elemen dalam linked list dari head hingga tail, dipisahkan dengan `<->` dan diakhiri dengan `NULL` untuk menunjukkan akhir daftar. Program utama (main) menunjukkan contoh penggunaan fungsi-fungsi ini, dimulai dengan menambahkan elemen ke awal dan akhir daftar, menghapus elemen, dan menampilkan daftar setelah setiap operasi.

## KODE PENDAHULUAN 3

```
1 #include <iostream>
2 using namespace std;
3
4 class Node {
5 public:
6     int data;
7     Node* prev;
8     Node* next;
9
10    Node(int data) {
11        this->data = data;
12        prev = nullptr;
13        next = nullptr;
14    }
15 };
16
17 class DLL {
18 public:
19     Node* head;
20     Node* tail;
21
22    DLL() {
23        head = nullptr;
24        tail = nullptr;
25    }
26
27    void insertFirst_2311104036(int data) {
28        Node* p = new Node(data);
29
30        if (head == nullptr) {
31            head = p;
32            tail = p;
33        } else {
34            p->next = head;
35            p->prev = nullptr;
36            head->prev = p;
37            head = p;
38        }
39    }
40
41    void insertLast_2311104036(int data) {
42        Node* p = new Node(data);
43
44        if (head == nullptr) {
45            head = p;
46            tail = p;
47        } else {
48            p->prev = tail;
49            tail->next = p;
50            tail = p;
51        }
52    }
53
54    void inputFirstPengguna_2311104036() {
55        int dataInputan;
56        Node* current = head;
57        for (int i = 0; i < 1; i++) {
58            cout << "Masukkan data dipisahkan dengan spasi = ";
59            for (int j = 0; j < 4; j++) {
60                cin >> dataInputan;
61                insertFirst_2311104036(dataInputan);
62            }
63        }
64    }
65 }
```

```
65
66 void deleteFirst_2311104036() {
67     Node* temp = head;
68     head = head->next;
69     if (head != nullptr) { // Cek untuk memastikan head tidak null
70         head->prev = nullptr;
71     }
72     delete temp;
73 }
74
75 void deleteLast_2311104036() {
76     Node* temp = tail;
77     tail = tail->prev;
78     if (tail != nullptr) { // Cek untuk memastikan tail tidak null
79         tail->next = nullptr;
80     }
81     delete temp;
82 }
83
84 void displayAwal_2311104036() {
85     Node* current = head;
86     cout << "Daftar elemen dari belakang ke depan: ";
87     while (current != nullptr) {
88         cout << current->data << "<->";
89         current = current->next;
90     }
91     cout << "NULL" << endl;
92 }
93
94 void displayAkhir_2311104036() {
95     Node* current = tail;
96     cout << "Daftar elemen dari depan ke belakang: ";
97     while (current != nullptr) {
98         cout << current->data << "<->";
99         current = current->prev;
100    }
101    cout << "NULL" << endl;
102 }
103 };
104
105 int main() {
106     DLL dll;
107
108     dll.inputFirstPengguna_2311104036();
109     dll.displayAwal_2311104036();
110     dll.displayAkhir_2311104036();
111
112     return 0;
113 }
114 }
```

## OUTPUT

```
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Double_Link
AN\06_Double_Linked_List_Bagian_1\TUGAS PENDAHULUAN\TP_3.cpp\" ; if
Masukkan data dipisahkan dengan spasi = 1 2 3 4
Daftar elemen dari belakang ke depan 4<->3<->2<->1<->NULL
Daftar elemen dari depan ke belakang 1<->2<->3<->4<->NULL
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\06_Double_Link
```

## **PENJELASAN**

Fungsi `insertFirst_2311104036` dan `insertLast_2311104036` memungkinkan pengguna untuk menambahkan elemen di awal atau akhir linked list, sedangkan `inputFirstPengguna_2311104036` meminta pengguna memasukkan empat angka yang dimasukkan di awal daftar. Fungsi `deleteFirst_2311104036` dan `deleteLast_2311104036` berfungsi untuk menghapus elemen pertama dan terakhir, dengan memastikan bahwa daftar tidak kosong setelah penghapusan. Untuk menampilkan elemen, terdapat dua fungsi: `displayAwal_2311104036`, yang menampilkan elemen dari akhir ke awal, dan `displayAkhir_2311104036`, yang menampilkan elemen dari awal ke akhir, keduanya menggunakan pemisah `<->` dan diakhiri dengan `NULL` untuk menunjukkan akhir daftar. Program utama mengintegrasikan semua fungsi ini untuk menerima input pengguna dan menampilkan daftar dalam kedua arah, menunjukkan fleksibilitas dari struktur data ini.