

LAPORAN PRAKTIKUM
Tugas Pendahuluan Modul 6
“Double Linked List – Bagian 1”



Disusun Oleh:
Muhammad Atsal Rizandri - 21104057
Struktur Data SE07-01 / SE05-02

Dosen :
Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024

1. Menambahkan elemen di awal dan di akhir double linked list. Pada program ini memberikan pengguna akses untuk memasukkan atau menambahkan elemen ke dalam double linked list di awal dan akhir list. Fungsi yang di implementasikan yaitu fungsi “insertFirst” dan “insertLast” untuk menambahkan elemen masingmasing di awal dan akhir list. Setelah itu list yang sudah berisikan elemen dari pengguna ditampilkan dari depan ke belakang.

Kode:

```
06_Double_Linked_List_Bagian_1 > TP > SOAL_01.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8      Node *prev;
9  };
10
11  typedef Node *List;
12
13  // Fungsi menambahkan elemen di awal list
14  void insertFirst_21104057(List &L, int value)
15  {
16      Node *newNode = new Node;
17      newNode->data = value;
18      newNode->next = L;
19      newNode->prev = nullptr;
20
21      if (L != nullptr)
22      {
23          L->prev = newNode;
24      }
25
26      L = newNode;
27      cout << "Elemen " << value << " ditambahkan di awal list." << endl;
28  }
29
30  // Fungsi menambahkan elemen di akhir list
31  void insertLast_21104057(List &L, int value)
32  {
33      Node *newNode = new Node;
34      newNode->data = value;
35      newNode->next = nullptr;
36
37      if (L == nullptr)
38      {
39          newNode->prev = nullptr;
40          L = newNode;
41          cout << "Elemen " << value << " ditambahkan sebagai elemen pertama di list." << endl;
42          return;
43      }
44
45      Node *temp = L;
46      while (temp->next != nullptr)
47      {
48          temp = temp->next;
49      }
50
51      temp->next = newNode;
52      newNode->prev = temp;
53      cout << "Elemen " << value << " ditambahkan di akhir list." << endl;
54  }
```

```

void printList_21104057(List L)
{
    cout << "DAFTAR ANGGOTA LIST: ";
    Node *temp = L;
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->next != nullptr)
        {
            cout << " < - > ";
        }
        temp = temp->next;
    }
    cout << endl;
}

int main()
{
    List L = nullptr;

    int value;

    cout << "Input elemen kesatu -> ";
    cin >> value;
    insertFirst_21104057(L, value);

    cout << "Input elemen kedua -> ";
    cin >> value;
    insertFirst_21104057(L, value);

    cout << "Input elemen ketiga -> ";
    cin >> value;
    insertLast_21104057(L, value);

    printList_21104057(L);

    return 0;
}

```

- Output

```

Input elemen kesatu -> 21
Elemen 21 ditambahkan di awal list.
Input elemen kedua -> 8
Elemen 8 ditambahkan di awal list.
Input elemen ketiga -> 30
Elemen 30 ditambahkan di akhir list.
DAFTAR ANGGOTA LIST: 8 < - > 21 < - > 30
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```

2. Menghapus elemen di awal dan akhir double linked list. Pada program ini membuat program untuk memberikan pengguna akses untuk menghapus elemen pertama dan elemen terakhir dalam list. Fungsi yang diimplementasikan pada program ini yaitu fungsi “deleteFirst” dan “deleteLast” untuk menghapus elemen pertama dan elemen terakhir. Setelah itu, program menampilkan hasil list yang telah dihapus berdasarkan fungsi yang digunakan dan inputan.

- Kode

```
06_Double_Linked_List_Bagian_1 > TP > SOAL_02.cpp > InsertLast_21104057(List &, int)
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8      Node *prev;
9  };
10
11  typedef Node *List;
12
13  // Fungsi menambahkan elemen di akhir list
14  void insertLast_21104057(List &L, int value)
15  {
16      Node *newNode = new Node;
17      newNode->data = value;
18      newNode->next = nullptr;
19
20      if (L == nullptr)
21      {
22          newNode->prev = nullptr;
23          L = newNode;
24          cout << "Elemen " << value << " ditambahkan sebagai elemen pertama di list." << endl;
25          return;
26      }
27
28      Node *temp = L;
29      while (temp->next != nullptr)
30      {
31          temp = temp->next;
32      }
33
34      temp->next = newNode;
35      newNode->prev = temp;
36      cout << "Elemen " << value << " ditambahkan di akhir list." << endl;
37  }
38
39  // Fungsi menghapus elemen pertama dari list
40  void deleteFirst_21104057(List &L)
41  {
42      Node *temp = L;
43      L = L->next;
44
45      if (L != nullptr)
46      {
47          L->prev = nullptr;
48      }
49
50      delete temp;
51      cout << "Elemen pertama telah dihapus." << endl;
52  }
```

```

// Fungsi menghapus elemen terakhir dari list
void deleteLast_21104057(List &L)
{
    if (L->next == nullptr)
    {
        delete L;
        L = nullptr;
        cout << "Elemen terakhir telah dihapus." << endl;
        return;
    }

    Node *temp = L;
    while (temp->next != nullptr)
    {
        temp = temp->next;
    }

    temp->prev->next = nullptr;
    delete temp;
    cout << "Elemen terakhir telah dihapus." << endl;
}

void printList_21104057(List L)
{
    cout << "DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: ";
    Node *temp = L;
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->next != nullptr)
        {
            cout << " <-> ";
        }
        temp = temp->next;
    }
    cout << endl;
}

int main()
{
    List L = nullptr;
    int value;

    cout << "Input elemen kesatu -> ";
    cin >> value;
    insertLast_21104057(L, value);

    cout << "Input elemen kedua -> ";
    cin >> value;
    insertLast_21104057(L, value);

    cout << "Input elemen ketiga -> ";
    cin >> value;
    insertLast_21104057(L, value);

    deleteFirst_21104057(L);
    deleteLast_21104057(L);

    printList_21104057(L);
    return 0;
}

```

- Output

```

Input elemen kesatu -> 16
Elemen 16 ditambahkan sebagai elemen pertama di list.
Input elemen kedua -> 25
Elemen 25 ditambahkan di akhir list.
Input elemen ketiga -> 28
Elemen 28 ditambahkan di akhir list.
Elemen pertama telah dihapus.
Elemen terakhir telah dihapus.
DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 25
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliah\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```

3. Menampilkan elemen dari depan ke belakang dan sebaliknya dalam double linked list. Program ini memberikan pengguna untuk memasukkan beberapa elemen ke dalam double linked list, setelah dimasukkan, program menampilkan seluruh elemen di dalam list dari depan ke belakang, begitu juga sebaliknya.

- Kode

```
06_Double_Linked_List_Bagian_1 > TP > SOAL_03.cpp > insertLast_21104057(List &, int)
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8      Node *prev;
9  };
10
11  typedef Node *List;
12
13  void insertLast_21104057(List &L, int value)
14  {
15      Node *newNode = new Node;
16      newNode->data = value;
17      newNode->next = nullptr;
18
19      if (L == nullptr)
20      {
21          newNode->prev = nullptr;
22          L = newNode;
23          cout << "Elemen " << value << " ditambahkan sebagai elemen pertama di list." << endl;
24          return;
25      }
26
27      Node *temp = L;
28      while (temp->next != nullptr)
29      {
30          temp = temp->next;
31      }
32
33      temp->next = newNode;
34      newNode->prev = temp;
35      cout << "Elemen " << value << " ditambahkan di akhir list." << endl;
36  }
```

```

void printDepan_21104057(List L)
{
    cout << "Daftar elemen dari depan ke belakang: ";
    Node *temp = L;
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->next != nullptr)
        {
            cout << " < - > ";
        }
        temp = temp->next;
    }
    cout << endl;
}

void printBelakang_21104057(List L)
{
    Node *temp = L;
    while (temp->next != nullptr)
    {
        temp = temp->next;
    }

    cout << "Daftar elemen dari belakang ke depan: ";
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->prev != nullptr)
        {
            cout << " < - > ";
        }
        temp = temp->prev;
    }
    cout << endl;
}

int main()
{
    List L = nullptr;
    int value;

    for (int a = 1; a <= 4; a++)
    {
        cout << "Input elemen ke-" << a << " : ";
        cin >> value;
        insertLast_21104057(L, value);
    }

    printDepan_21104057(L);
    printBelakang_21104057(L);

    return 0;
}

```

- Output

```

Input elemen ke-1 : 34
Elemen 34 ditambahkan sebagai elemen pertama di list.
Input elemen ke-2 : 46
Elemen 46 ditambahkan di akhir list.
Input elemen ke-3 : 59
Elemen 59 ditambahkan di akhir list.
Input elemen ke-4 : 71
Elemen 71 ditambahkan di akhir list.
Daftar elemen dari depan ke belakang: 34 < - > 46 < - > 59 < - > 71
Daftar elemen dari belakang ke depan: 71 < - > 59 < - > 46 < - > 34
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```