

LAPORAN PRAKTIKUM
Modul 6
“Double Linked List (Bagian 1)”



Disusun Oleh:

Haikal Fadhilah Mufid (2311104027)

S1SE-07-1

Dosen :

Yudha Islami, S.KOM., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO
2024

1. kode ini digunakan untuk menambahkan elemen integer secara random dari user, dan ini nanti akan mengurutkan sendiri dari yang terkecil ke yang terbesar menggunakan kodei insertsorted.kode ini menggunakan double linked list untuk menyimpan elemen lebih dari satu, atau secara berturut turut.

```
#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* prev;
    Node* next;
};

Node* head = nullptr;

void insertSorted_231114027(int data) {
    Node* newNode = new Node();
    newNode->data = data;

    if (head == nullptr) {
        newNode->prev = nullptr;
        newNode->next = nullptr;
        head = newNode;
        return;
    }

    if (data <= head->data) {
        newNode->next = head;
        newNode->prev = nullptr;
        head->prev = newNode;
        head = newNode;
        return;
    }

    Node* temp = head;
    while (temp->next != nullptr && temp->next->data < data) {
        temp = temp->next;
    }

    newNode->next = temp->next;
    newNode->prev = temp;

    if (temp->next != nullptr) {
        temp->next->prev = newNode;
    }
    temp->next = newNode;
}
```

```

#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* prev;
    Node* next;
};

Node* head = nullptr;

void insertSorted_231114027(int data) {
    Node* newNode = new Node();
    newNode->data = data;

    if (head == nullptr) {
        newNode->prev = nullptr;
        newNode->next = nullptr;
        head = newNode;
        return;
    }

    if (data <= head->data) {
        newNode->next = head;
        newNode->prev = nullptr;
        head->prev = newNode;
        head = newNode;
        return;
    }

    Node* temp = head;
    while (temp->next != nullptr && temp->next->data < data) {
        temp = temp->next;
    }

    newNode->next = temp->next;
    newNode->prev = temp;

    if (temp->next != nullptr) {
        temp->next->prev = newNode;
    }
    temp->next = newNode;
}

```

Dengan Output sebagai berikut :

```

Masukan Elemen pertama : 10
Masukan Elemen kedua di awal : 5
Masukan Elemen ketiga di akhir : 20
Daftar Anggota List : 5 <-> 10 <-> 20

```

2. kode ini masih sama seperti sebelumnya, yaitu menggunakan DLL, namun di kode ini program menggunakan kode delete untuk mengurangi elemen pertama dan terakhir, jadi yang di outputkan hanyalah kode yang kedua.

```
#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* prev;
    Node* next;
};

Node* head = nullptr;

void insertLast_2311104027 (int data) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->next = nullptr;

    if (head == nullptr) {
        newNode->prev = nullptr;
        head = newNode;
        return;
    }

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

    temp->next = newNode;
    newNode->prev = temp;
}

void deleteFirst() {
    if (head == nullptr) {
        cout << "List kosong, tidak ada yang dihapus." << endl;
        return;
    }

    Node* temp = head;
    head = head->next;

    if (head != nullptr) {
        head->prev = nullptr;
    }
}
```



```

    delete temp;
}

void deleteLast() {
    if (head == nullptr) {
        cout << "List kosong, tidak ada yang dihapus" << endl;
        return;
    }

    Node* temp = head;

    if (temp->next == nullptr) {
        head = nullptr;
        delete temp;
        return;
    }
    while (temp->next != nullptr) {
        temp = temp->next;
    }
    temp->prev->next = nullptr;
    delete temp;
}

void displayList() {
    Node* temp = head;
    cout << "Daftar Anggota List : ";
    while (temp != nullptr) {
        cout << temp->data;
        if (temp->next != nullptr) {
            cout << " <-> ";
        }
        temp = temp->next;
    }
    cout << endl;
}

int main() {
    int data;

    cout << "Masukan Elemen Pertama : ";
    cin >> data;
    insertLast_2311104027(data);
}

```

```
cout << "Masukan Elemen Kedua di akhir : ";  
cin >> data;  
insertLast_2311104027(data);  
  
cout << "Masukan Elemen Ketiga di akhir : ";  
cin >> data;  
insertLast_2311104027(data);  
  
deleteFirst();  
  
deleteLast();  
  
displayList();  
  
return 0;
```

Dengan Output :

```
Masukan Elemen Pertama : 10  
Masukan Elemen Kedua di akhir : 15  
Masukan Elemen Ketiga di akhir : 20  
Daftar Anggota List : 15
```

3. Kode ini menggunakan DLL lagi seperti sebelumnya, dan kode ini digunakan untuk memasukan 4 elemen angka dan ditampilkan dalam dua arah (dari depan belakang, dan dari belakang ke depan)

```
#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* prev;
    Node* next;
};

Node* head = nullptr;
Node* tail = nullptr;

void insertLast_2311104027(int data) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->next = nullptr;

    if (head == nullptr) {
        newNode->prev = nullptr;
        head = newNode;
        tail = newNode;
        return;
    }

    tail->next = newNode;
    newNode->prev = tail;
    tail = newNode;
}

void displayFromFront() {
    Node* temp = head;
    cout << "Daftar Elemen dari depan ke belakang : ";
    while (temp != nullptr) {
        cout << temp->data;
        if (temp->next != nullptr) {
            cout << " <--> ";
        }
        temp = temp->next;
    }
    cout << endl;
}
```



```

void displayFromBack () {
    Node* temp = tail;
    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() {
    int data;

    for (int i = 1; i <= 4; i++) {
        cout << "Masukkan elemen ke-" << i << ": ";
        cin >> data;
        insertLast_2311104027(data);
    }

    displayFromFront();

    displayFromBack();

    return 0;
}

```

Dengan output sebagai berikut :

```

Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Daftar Elemen dari depan ke belakang : 1 <--> 2 <--> 3 <--> 4
Daftar Elemen dari belakang ke depan: 4 <-> 3 <-> 2 <-> 1

```