LAPORAN PRAKTIKUM

PERTEMUAN 6

Pengenalan C++ : Double Linked List Bagian 01

Nama :

Ahmad Uffi Lestari Ma’ruf(2311104033)

Dosen :

YUDHA ISLAMI SULISTYA, S.Kom., M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. **Tugas Pendahuluan**

Soal satu:

Kode ini membuat struktur linked list ganda dengan operasi penambahan di awal (`insertFirst`) dan di akhir (`insertLast`). Setiap `Node` memiliki data dan dua pointer (`prev` dan `next`) yang menghubungkan antar-node, memungkinkan traversal dua arah.

Pada fungsi `main`, tiga elemen ditambahkan berdasarkan input pengguna: dua elemen pertama dimasukkan di awal, dan elemen ketiga ditambahkan di akhir. Fungsi `displayList` kemudian menampilkan data dari tiap node secara berurutan dengan simbol "<->" di antara node untuk menunjukkan arah dua sisi dari list.

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

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

void insertFirst(Node*& head, int data) {
  Node* newNode = new Node();
  newNode->data = data;
  newNode->prev = nullptr;
  newNode->next = head;
  if(head != nullptr) {
    head->prev = newNode;
  }
  head = newNode;
}

void insertLast(Node*& head, 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 displayList(Node* head) {
  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() {
  Node* head = nullptr;

  int firstElm, secondElm, thirdElm;
  cout << "Masukkan elemen pertama = ";
  cin >> firstElm;
  insertFirst(head, firstElm);

  cout << "Masukkan elemen kedua di awal = ";
  cin >> secondElm;
  insertFirst(head, secondElm);

  cout << "Masukkan elemen ketiga di akhir = ";
  cin >> thirdElm;
  insertLast(head, thirdElm);

  displayList(head);

  return 0;
}
```

output yang dihasilkan:

```
e   --interpreter=mi
Masukkan elemen pertama = 23
Masukkan elemen kedua di awal = 21
Masukkan elemen ketiga di akhir = 43
DAFTAR ANGGOTA LIST: 21 <-> 23 <-> 43
PS E:\ITTP\Modul_Ittp\semester_3\praktiku
```

Soal 2 :

Kode ini membentuk linked list ganda (doubly linked list) dengan struktur `Node` yang memiliki pointer ke node sebelumnya dan berikutnya. Fungsi `insertFirst` menambah node di awal list, `insertLast` menambah di akhir, sedangkan `deleteFirst` dan `deleteLast` menghapus node pertama dan terakhir dari list.

Dalam `main`, program meminta pengguna memasukkan tiga angka untuk ditambahkan ke dalam list. Setelah menambahkan angka pertama di awal dan dua angka berikutnya di akhir, program menghapus elemen pertama dan terakhir, lalu menampilkan hasil list yang tersisa atau pesan jika list kosong.

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

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

void insertFirst(Node*& head, int data) {
  Node* newNode = new Node();
  newNode->data = data;
  newNode->prev = nullptr;
  newNode->next = head;
  if (head != nullptr) {
    head->prev = newNode;
  }
  head = newNode;
}

void insertLast(Node*& head, 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;
}
```

```cpp
void deleteFirst(Node*& head) {
  if (head == nullptr) {
    cout << "List kosong, tidak ada elemen yang bisa dihapus." << endl;
    return;
  }

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

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

  delete temp;
}

void deleteLast(Node*& head) {
  if (head == nullptr) {
    cout << "List kosong, tidak ada elemen yang bisa dihapus." << endl;
    return;
  }

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

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

  temp->prev->next = nullptr;
  delete temp;
}

void displayList(Node* head) {
  if (head == nullptr) {
    cout << "DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: List kosong." <<
endl;return;
  }

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

int main() {
  Node* head = nullptr;

  int firstElm, secondElm, thirdElm;
  cout << "Masukkan elemen pertama = ";
  cin >> firstElm;
  insertFirst(head, firstElm);

  cout << "Masukkan elemen kedua di akhir = ";
  cin >> secondElm;
  insertLast(head, secondElm);

  cout << "Masukkan elemen ketiga di akhir = ";
  cin >> thirdElm;
  insertLast(head, thirdElm);

  deleteFirst(head);
  deleteLast(head);
  displayList(head);
}
```

Maka output yang di hasilakan :

```
Masukkan elemen pertama = 21
Masukkan elemen kedua di akhir = 11
Masukkan elemen ketiga di akhir = 15
DAFTAR ANGGOTA LIST SETELAH PENGHAPUSAN: 11
```

Soal 3 :

Kode ini membuat linked list ganda (doubly linked list) dan mengimplementasikan fungsi untuk menambahkan elemen di akhir list serta menampilkan elemen-elemen dari depan ke belakang maupun sebaliknya. Fungsi `insertLast` menambahkan node baru di akhir list, sedangkan `displayForward` menampilkan elemen-elemen list dari awal ke akhir, dan `displayBackward` menampilkan elemen-elemen dari akhir ke awal.

Di fungsi `main`, program meminta pengguna memasukkan empat elemen secara berurutan untuk dimasukkan ke dalam list. Setelah semua elemen ditambahkan, program menampilkan list dari depan ke belakang dan dari belakang ke depan, masing-masing dengan pemisah "<->" di antara elemen-elemen.

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

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

void insertLast(Node*& head, 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;
}
```

```cpp
void displayForward(Node* head) {
  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 displayBackward(Node* head) {
  if (head == nullptr) return;

  Node* temp = head;
  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() {
  Node* head = nullptr;
  int n, data;

  cout << "Masukkan 4 elemen secara berurutan: ";
  for (int i = 0; i < 4; i++) {
    cin >> data;
    insertLast(head, data);
  }

  displayForward(head);
  displayBackward(head);
}
```

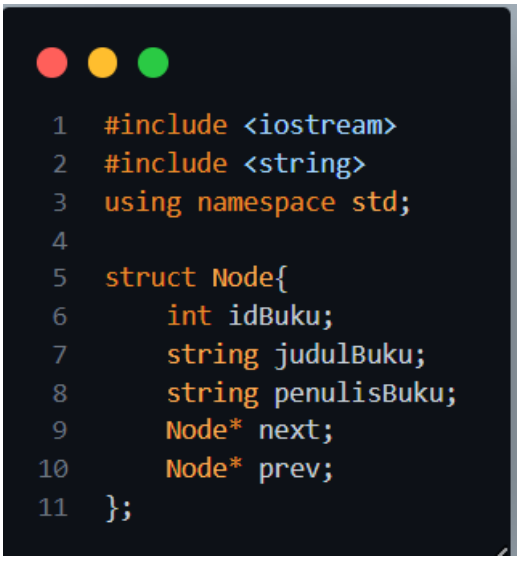
maka output yang dihasilakan adalah :

```
Masukkan 4 elemen secara berurutan: 3
3
4
5
Daftar elemen dari depan ke belakang: 3 <-> 3 <-> 4 <-> 5
Daftar elemen dari belakang ke depan: 5 <-> 4 <-> 3 <-> 3
PS F:\ITTP\Modul Ittp\semester 3\praktikum std\STD Ahmad Ufi
```

2. Unguided

Kode ini membuat program manajemen daftar buku menggunakan linked list ganda (doubly linked list). Struktur `Node` menyimpan data buku, seperti `idBuku`, `judulBuku`, dan `penulisBuku`, serta pointer `next` dan `prev` untuk menghubungkan node. Kelas `DoubleLinkedList` memiliki fungsi `tambahData` untuk menambahkan buku baru di akhir list, `tampilDariAwal` untuk menampilkan buku dari awal hingga akhir, dan `tampilDariAkhir` untuk menampilkan buku dari akhir ke awal.

Dalam `main`, program menampilkan menu pilihan untuk pengguna. Pengguna dapat menambah buku baru dengan memasukkan ID, judul, dan nama penulis, atau memilih untuk melihat daftar buku dari awal atau akhir. Program terus menjalankan menu hingga pengguna memilih opsi keluar.



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

struct Node{
    int idBuku;
    string judulBuku;
    string penulisBuku;
    Node* next;
    Node* prev;
};
```

```
class DoubleLinkedList{
    private:
        Node* head;
        Node* tail;

    public:
        // constructor
        DoubleLinkedList(){
            head = NULL;
            tail = NULL;
        }

        // fungsi menambahkan data
        void tambahData(int idBuku, string judulBuku, string penulisBuku)
{           Node* newNode = new Node();
            newNode->idBuku = idBuku;
            newNode->judulBuku = judulBuku;
            newNode->penulisBuku = penulisBuku;
            newNode->next = NULL;
            newNode->prev = NULL;

            if(head == NULL){
                head = newNode;
                tail = newNode;
            } else {
                tail->next = newNode;
                newNode->prev = tail;
                tail = newNode;
            }

            cout << "Data berhasil ditambahkan" << endl;
        }

        // Tampilkan dari awal
        void tampilDariAwal(){
            if(head == NULL){
                cout << "Data kosong" << endl;
                return;
            }

            cout << "\nDaftar Buku dari awal:" << endl;
            Node* current = head;
            while(current != NULL){
                cout << "ID Buku: " << current->idBuku << endl;
                cout << "Judul Buku: " << current->judulBuku << endl;
                cout << "Penulis Buku: " << current->penulisBuku << endl;
                cout << "----------------------" << endl;
                current = current->next;
            }
        }

        // Tampilkan dari akhir
        void tampilDariAkhir(){
            if (tail == NULL){
                cout << "Data kosong" << endl;
                return;
            }

            cout << "\nDaftar Buku dari akhir:" << endl;
            Node* current = tail;
            while (current != NULL){
                cout << "ID Buku: " << current->idBuku << endl;
                cout << "Judul Buku: " << current->judulBuku << endl;
                cout << "Penulis Buku: " << current->penulisBuku << endl;
                cout << "----------------------" << endl;
                current = current->prev;
            }
        }
};
```

```cpp
int main(){
    DoubleLinkedList daftarBuku;
    int pilihan, id;
    string judul, penulis;

    do {
        cout << "\nMenu:\n";
        cout << "1. Tambah Buku\n";
        cout << "2. Tampilkan Buku dari awal\n";
        cout << "3. Tampilkan Buku dari akhir\n";
        cout << "4. Keluar\n";
        cout << "Pilihan: ";
        cin >> pilihan;
        cin.ignore();  // Membersihkan newline dari input sebelumnya

        switch (pilihan) {
            case 1:
                cout << "Masukkan ID Buku: ";
                cin >> id;
                cin.ignore();  // Mengabaikan newline setelah ID

                cout << "Masukkan Judul Buku: ";
                getline(cin, judul);

                cout << "Masukkan Penulis Buku: ";
                getline(cin, penulis);

                daftarBuku.tambahData(id, judul, penulis);
                break;

            case 2:
                daftarBuku.tampilDariAwal();
                break;

            case 3:
                daftarBuku.tampilDariAkhir();
                break;

            case 4:
                cout << "Keluar dari program." << endl;
                break;

            default:
                cout << "Pilihan tidak valid, coba lagi." << endl;
                break;
        }

    } while (pilihan != 4);

    return 0;
}
```

maka hasil outputnya :

```
2. Tampilkan Buku dari awal
3. Tampilkan Buku dari akhir
4. Keluar
Pilihan: 1
Masukkan ID Buku: 2
Masukkan Judul Buku: sedikitLagi
Masukkan Penulis Buku: ahmadUffi
Data berhasil ditambahkan

Menu:
1. Tambah Buku
2. Tampilkan Buku dari awal
3. Tampilkan Buku dari akhir
4. Keluar
Pilihan: 1
Masukkan ID Buku: 4
Masukkan Judul Buku: neverGiveup
Masukkan Penulis Buku: Intan Azizah
Data berhasil ditambahkan

Menu:
1. Tambah Buku
2. Tampilkan Buku dari awal
3. Tampilkan Buku dari akhir
4. Keluar
Pilihan: 2

Daftar Buku dari awal:
ID Buku: 2
Judul Buku: sedikitLagi
Penulis Buku: ahmadUffi
----------------------
ID Buku: 4
Judul Buku: neverGiveup
Penulis Buku: Intan Azizah
----------------------
```