

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL IV
LINKED LIST CIRCULAR DAN
NON CIRCULAR**



Disusun Oleh :

Nia Novela Ariandini
2311102057

Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. Dasar Teori

Linked list dapat dibagi menjadi dua jenis, yaitu linked list circular dan non-circular.

1. Linked list non-circular

Pada linked list non-circular, node terakhir tidak menunjuk pada simpul pertama. Artinya, linked list ini memiliki simpul terakhir yang menunjuk ke null atau tidak menunjuk ke simpul mana pun. Dalam linked list non-circular, traversal atau penelusuran dari awal hingga akhir dapat dilakukan dengan mudah dengan mengikuti alamat setiap node.

2. Linked list circular

Pada linked list circular, simpul terakhir menunjuk pada simpul pertama. Artinya, linked list ini membentuk sebuah lingkaran, dan traversal dapat dimulai dari mana saja dalam linked list ini. Linked list circular memiliki kelebihan yaitu memungkinkan untuk melakukan traversal dari mana saja dan tidak memerlukan operasi tambahan untuk mengembalikan pointer ke simpul awal.

Dalam implementasinya, linked list circular dan non-circular memiliki beberapa perbedaan. Salah satunya adalah pada operasi penambahan atau penghapusan node, dimana linked list circular memerlukan operasi tambahan untuk mengubah pointer pada simpul terakhir agar menunjuk pada simpul pertama. Sedangkan linked list non-circular tidak memerlukan operasi tersebut karena simpul terakhirnya tidak menunjuk ke simpul pertama.

B. Guided

Guided 1

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

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

Node *head;
Node *tail;

void init() {
    head = NULL;
    tail = NULL;
}

bool isEmpty() {
    return head == NULL;
}

void insertDepan(int nilai) {
    Node *baru = new Node;
    baru->data = nilai;
    baru->next = NULL;
    if (isEmpty()) {
        head = tail = baru;
    } else {
        baru->next = head;
        head = baru;
    }
}

void insertBelakang(int nilai) {
    Node *baru = new Node;
    baru->data = nilai;
    baru->next = NULL;
    if (isEmpty()) {
        head = tail = baru;
    }
```

```

    } else {
        tail->next = baru;
        tail = baru;
    }
}

int hitungList() {
    Node *hitung = head;
    int jumlah = 0;
    while (hitung != NULL) {
        jumlah++;
        hitung = hitung->next;
    }
    return jumlah;
}

void insertTengah(int data, int posisi) {
    if (posisi < 1 || posisi > hitungList()) {
        cout << "Posisi diluar jangkauan" << endl;
    } else if (posisi == 1) {
        cout << "Posisi bukan posisi tengah" << endl;
    } else {
        Node *baru = new Node();
        baru->data = data;
        Node *bantu = head;
        int nomor = 1;
        while (nomor < posisi - 1) {
            bantu = bantu->next;
            nomor++;
        }
        baru->next = bantu->next;
        bantu->next = baru;
    }
}

void hapusDepan() {
    if (!isEmpty()) {
        Node *hapus = head;
        if (head->next != NULL) {
            head = head->next;
        } else {
            head = tail = NULL;
        }
    }
}

```

```

        delete hapus;
    } else {
        cout << "List kosong!" << endl;
    }
}

void hapusBelakang() {
    if (!isEmpty()) {
        Node *hapus = tail;
        if (head != tail) {
            Node *bantu = head;
            while (bantu->next != tail) {
                bantu = bantu->next;
            }
            tail = bantu;
            tail->next = NULL;
        } else {
            head = tail = NULL;
        }
        delete hapus;
    } else {
        cout << "List kosong!" << endl;
    }
}

void hapusTengah(int posisi) {
    if (posisi < 1 || posisi > hitungList()) {
        cout << "Posisi di luar jangkauan" << endl;
    } else if (posisi == 1) {
        cout << "Posisi bukan posisi tengah" << endl;
    } else {
        Node *bantu = head;
        Node *hapus;
        Node *sebelum = NULL;
        int nomor = 1;
        while (nomor < posisi) {
            sebelum = bantu;
            bantu = bantu->next;
            nomor++;
        }
        hapus = bantu;
        if (sebelum != NULL) {
            sebelum->next = bantu->next;

```

```

        } else {
            head = bantu->next;
        }
        delete hapus;
    }
}

void ubahDepan(int data) {
    if (!isEmpty()) {
        head->data = data;
    } else {
        cout << "List masih kosong!" << endl;
    }
}

void ubahTengah(int data, int posisi) {
    if (!isEmpty()) {
        if (posisi < 1 || posisi > hitungList()) {
            cout << "Posisi di luar jangkauan" << endl;
        } else if (posisi == 1) {
            cout << "Posisi bukan posisi tengah" << endl;
        } else {
            Node *bantu = head;
            int nomor = 1;
            while (nomor < posisi) {
                bantu = bantu->next;
                nomor++;
            }
            bantu->data = data;
        }
    } else {
        cout << "List masih kosong!" << endl;
    }
}

void ubahBelakang(int data) {
    if (!isEmpty()) {
        tail->data = data;
    } else {
        cout << "List masih kosong!" << endl;
    }
}

```

```

void clearList() {
    Node *bantu = head;
    Node *hapus;
    while (bantu != NULL) {
        hapus = bantu;
        bantu = bantu->next;
        delete hapus;
    }
    head = tail = NULL;
    cout << "List berhasil terhapus!" << endl;
}

void tampil() {
    Node *bantu = head;
    if (!isEmpty()) {
        while (bantu != NULL) {
            cout << bantu->data << " ";
            bantu = bantu->next;
        }
        cout << endl;
    } else {
        cout << "List masih kosong!" << endl;
    }
}

int main() {
    init();
    insertDepan(3);
    tampil();
    insertBelakang(5);
    tampil();
    insertDepan(2);
    tampil();
    insertDepan(1);
    tampil();
    hapusDepan();
    tampil();
    hapusBelakang();
    tampil();
    insertTengah(7, 2);
    tampil();
    hapusTengah(2);
    tampil();
}

```

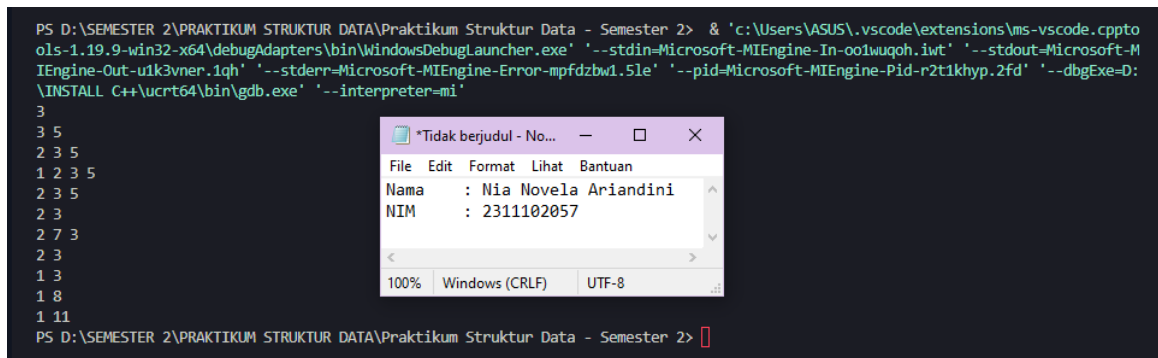
```

    ubahDepan(1);
    tampil();
    ubahBelakang(8);
    tampil();
    ubahTengah(11, 2);
    tampil();

    return 0;
}

```

Screenshots Output :



```

PS D:\SEMESTER 2\PRAKTIKUM STRUKTUR DATA\Praktikum Struktur Data - Semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.9-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-oo1wuqoh.iwt' '--stdout=Microsoft-MIEngine-Out-u1k3vner.1qh' '--stderr=Microsoft-MIEngine-Error-mpfdzbw1.51e' '--pid=Microsoft-MIEngine-Pid-r2t1khyp.2fd' '--dbgExe=D:\INSTALL C++\ucrt64\bin\gdb.exe' '--interpreter=mi'
3
3 5
2 3 5
1 2 3 5
2 3 5
2 3
2 7 3
2 3
1 3
1 8
1 11
PS D:\SEMESTER 2\PRAKTIKUM STRUKTUR DATA\Praktikum Struktur Data - Semester 2>

```

Deskripsi :

Script di atas adalah implementasi dari sebuah linked list non-circular dalam bahasa pemrograman C++. Program ini memiliki fungsi-fungsi dasar seperti penambahan elemen di depan, di belakang, atau di tengah linked list, penghapusan elemen di depan, di belakang, atau di tengah linked list, serta operasi pengubahan data pada elemen-elemen linked list. Setiap operasi dilakukan melalui fungsi-fungsi yang sesuai, dan hasilnya ditampilkan secara berurutan setelah setiap operasi dilakukan dalam fungsi main().

Guided 2

```

#include <iostream>
using namespace std;

struct Node {
    string data;
    Node *next;
};

Node *head, *tail, *baru, *bantu, *hapus;

void init() {

```



```

        head = NULL;
        tail = head;
    }

    int isEmpty() {
        return head == NULL;
    }

    void buatNode(string data) {
        baru = new Node;
        baru->data = data;
        baru->next = NULL;
    }

    int hitungList() {
        bantu = head;
        int jumlah = 0;
        while (bantu != NULL) {
            jumlah++;
            bantu = bantu->next;
        }
        return jumlah;
    }

    void insertDepan(string data) {
        buatNode(data);
        if (isEmpty()) {
            head = baru;
            tail = head;
            baru->next = head;
        } else {
            while (tail->next != head) {
                tail = tail->next;
            }
            baru->next = head;
            head = baru;
            tail->next = head;
        }
    }

    void insertBelakang(string data) {
        buatNode(data);
        if (isEmpty()) {
            head = baru;

```

```

        tail = head;
        baru->next = head;
    } else {
        while (tail->next != head) {
            tail = tail->next;
        }
        tail->next = baru;
        baru->next = head;
    }
}

void insertTengah(string data, int posisi) {
    if (isEmpty()) {
        head = baru;
        tail = head;
        baru->next = head;
    } else {
        baru->data = data;
        int nomor = 1;
        bantu = head;
        while (nomor < posisi - 1) {
            bantu = bantu->next;
            nomor++;
        }
        baru->next = bantu->next;
        bantu->next = baru;
    }
}

void hapusDepan() {
    if (!isEmpty()) {
        hapus = head;
        tail = head;
        if (hapus->next == head) {
            head = NULL;
            tail = NULL;
            delete hapus;
        } else {
            while (tail->next != hapus) {
                tail = tail->next;
            }
            head = head->next;
            tail->next = head;
            hapus->next = NULL;
        }
    }
}

```

```

        delete hapus;
    }
} else {
    cout << "List masih kosong!" << endl;
}
}

void hapusBelakang() {
    if (!isEmpty()) {
        hapus = head;
        tail = head;
        if (hapus->next == head) {
            head = NULL;
            tail = NULL;
            delete hapus;
        } else {
            while (hapus->next != head) {
                hapus = hapus->next;
            }
            while (tail->next != hapus) {
                tail = tail->next;
            }
            tail->next = head;
            hapus->next = NULL;
            delete hapus;
        }
    } else {
        cout << "List masih kosong!" << endl;
    }
}

void hapusTengah(int posisi) {
    if (!isEmpty()) {
        int nomor = 1;
        bantu = head;
        while (nomor < posisi - 1) {
            bantu = bantu->next;
            nomor++;
        }
        hapus = bantu->next;
        bantu->next = hapus->next;
        delete hapus;
    } else {
        cout << "List masih kosong!" << endl;
    }
}

```

```

    }
}

void clearList() {
    if (head != NULL) {
        hapus = head->next;
        while (hapus != head) {
            bantu = hapus->next;
            delete hapus;
            hapus = bantu;
        }
        delete head;
        head = NULL;
    }
    cout << "List berhasil terhapus!" << endl;
}

void tampil() {
    if (!isEmpty()) {
        tail = head;
        do {
            cout << tail->data << " ";
            tail = tail->next;
        } while (tail != head);
        cout << endl;
    } else {
        cout << "List masih kosong!" << endl;
    }
}

int main() {
    init();
    insertDepan("Ayam");
    tampil();
    insertDepan("Bebek");
    tampil();
    insertBelakang("Cicak");
    tampil();
    insertBelakang("Domba");
    tampil();
    hapusBelakang();
    tampil();
    hapusDepan();
    tampil();
}

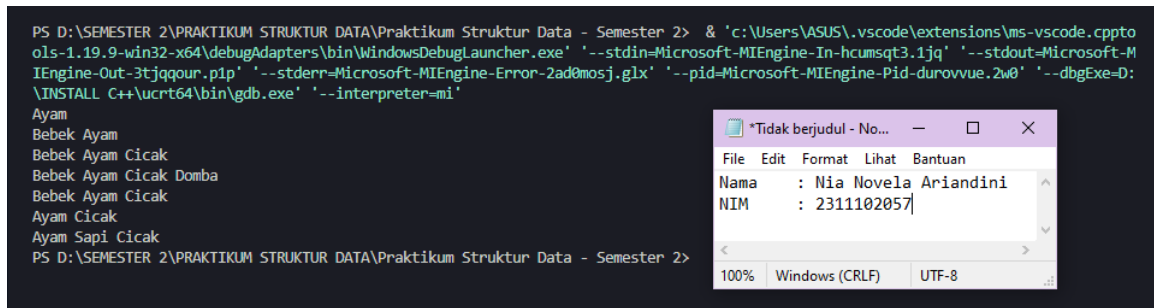
```

```

insertTengah("Sapi", 2);
tampil();
hapusTengah(2);
tampil();
return 0;
}

```

Screenshots Output :



Deskripsi :

Script di atas adalah implementasi dari sebuah linked list circular dalam bahasa pemrograman C++. Program ini memiliki fungsi-fungsi dasar seperti penambahan elemen di depan, di belakang, atau di tengah linked list, penghapusan elemen di depan, di belakang, atau di tengah linked list, serta operasi penghapusan seluruh isi linked list. Setiap operasi dilakukan melalui fungsi-fungsi yang sesuai, dan hasilnya ditampilkan secara berurutan setelah setiap operasi dilakukan dalam fungsi main().

C. Unguided/Tugas

Unguided 1

1. Buatlah menu untuk menambahkan, mengubah, menghapus, dan melihat Nama dan NIM mahasiswa, berikut contoh tampilan output dari nomor 1 :

• Tampilan Menu:

```

PROGRAM SINGLE LINKED LIST NON-CIRCULAR

1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. TAMPILKAN
0. KELUAR

Pilih Operasi :

```

• Tampilan Operasi Tambah:

```

-Tambah Depan

Masukkan Nama :
Masukkan NIM :

Data telah ditambahkan

```

```

-Tambah Tengah

Masukkan Nama :
Masukkan NIM :
Masukkan Posisi :

Data telah ditambahkan

```

-Hapus Belakang

Data (nama mahasiswa yang dihapus) berhasil dihapus

-Hapus Tengah

Masukkan posisi :

Data (nama mahasiswa yang dihapus) berhasil dihapus

• Tampilan Operasi Ubah:

-Ubah Belakang

Masukkan nama :
Masukkan NIM :

Data (nama lama) telah diganti dengan data (nama baru)

-Ubah Belakang

Masukkan nama :
Masukkan NIM :
Masukkan posisi :

Data (nama lama) telah diganti dengan data (nama baru)

• Tampilan Operasi Tampil Data:

DATA MAHASISWA

NAMA NIM
Nama1 NIM1
Nama2 NIM2

***Buat tampilan output sebagus dan secantik mungkin sesuai kreatifitas anda masing-masing, jangan terpaku pada contoh output yang diberikan**

Source Code :

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

struct identitasMhs_057 {
    string nama;
    string nim;
    identitasMhs_057* next;
};

class LinkedList {
private:
    identitasMhs_057* head;

public:
    LinkedList() {
        head = NULL;
    }

    void addFirst(string nama, string nim) {
        identitasMhs_057* newNode = new identitasMhs_057();
        newNode->nama = nama;
        newNode->nim = nim;
```

```

        newNode->next = head;
        head = newNode;
        cout << endl << "Data telah ditambahkan"<<endl;
    }

void addLast(string nama, string nim) {
    identitasMhs_057* newNode = new identitasMhs_057();
    newNode->nama = nama;
    newNode->nim = nim;
    newNode->next = NULL;
    if (head == NULL) {
        head = newNode;
    } else {
        identitasMhs_057* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
    cout << endl << "Data telah ditambahkan"<<endl;
}

void addMiddle(string nama, string nim, int posisi) {
    if (posisi < 1) {
        cout << endl << "Posisi tidak valid"<<endl;
        return;
    }
    identitasMhs_057* newNode = new identitasMhs_057();
    newNode->nama = nama;
    newNode->nim = nim;
    if (posisi == 1) {
        newNode->next = head;
        head = newNode;
    } else {
        identitasMhs_057* temp = head;
        for (int i = 1; i < posisi - 1; i++) {
            if (temp == NULL) {
                cout << endl << "Posisi tidak
valid"<<endl;
                return;
            }
            temp = temp->next;
        }
    }
}

```

```

        newNode->next = temp->next;
        temp->next = newNode;
    }
    cout << endl << "Data telah ditambahkan"<<endl;
}

void changeFirst(string nama, string nim) {
    if (head == NULL) {
        cout << endl << "Linked List kosong"<<endl;
        return;
    }
    head->nama = nama;
    head->nim = nim;
    cout << endl << "Data di depan telah diubah"<<endl;
}

void changeLast(string nama, string nim) {
    if (head == NULL) {
        cout << endl << "Linked List kosong"<<endl;
        return;
    }
    identitasMhs_057* temp = head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->nama = nama;
    temp->nim = nim;
    cout << endl << "Data di belakang telah diubah"<<endl;
}

void changeMiddle(string nama, string nim, int posisi) {
    if (posisi < 1) {
        cout << endl << "Posisi tidak valid"<<endl;
        return;
    }
    if (head == NULL) {
        cout << endl << "Linked List kosong"<<endl;
        return;
    }
    identitasMhs_057* temp = head;
    for (int i = 1; i < posisi; i++) {
        if (temp == NULL) {
            cout << endl << "Posisi tidak valid"<<endl;

```



```

        return;
    }
    temp = temp->next;
}
temp->nama = nama;
temp->nim = nim;
cout << endl << "Data di posisi " << posisi << " telah
diubah"<<endl;
}

void deleteFirst() {
    if (head == NULL) {
        cout << endl << "Linked List kosong"<<endl;
        return;
    }
    identitasMhs_057* temp = head;
    head = head->next;
    delete temp;
    cout << endl << "Data di depan telah dihapus"<<endl;
}

void deleteLast() {
    if (head == NULL) {
        cout << endl << "Linked List kosong"<<endl;
        return;
    }
    if (head->next == NULL) {
        delete head;
        head = NULL;
        cout << endl << "Data di belakang telah
dihapus"<<endl;
        return;
    }
    identitasMhs_057* temp = head;
    while (temp->next->next != NULL) {
        temp = temp->next;
    }
    delete temp->next;
    temp->next = NULL;
    cout << endl << "Data di belakang telah
dihapus"<<endl;
}

void deleteMiddle(int posisi) {

```

```

        if (posisi < 1) {
            cout << endl << "Posisi tidak valid"<<endl;
            return;
        }
        if (head == NULL) {
            cout << endl << "Linked List kosong"<<endl;
            return;
        }
        identitasMhs_057* temp = head;
        if (posisi == 1) {
            head = head->next;
            delete temp;
            cout << endl << "Data di posisi " << posisi << "
telah dihapus"<<endl;
            return;
        }
        for (int i = 1; i < posisi - 1; i++) {
            if (temp == NULL || temp->next == NULL) {
                cout << endl << "Posisi tidak valid"<<endl;
                return;
            }
            temp = temp->next;
        }
        identitasMhs_057* nextNode = temp->next->next;
        delete temp->next;
        temp->next = nextNode;
        cout << endl << "Data di posisi " << posisi << " telah
dihapus"<<endl;
    }

    void print() {
        if (head == NULL) {
            cout << endl << "Linked List kosong"<<endl;
            return;
        }
        cout << "===== DATA MAHASISWA ====="<<
endl;
        cout << "===== "<<
endl;
        cout << setw(20) << left << "NAMA" << setw(20) << left
<< "NIM" << endl;
        identitasMhs_057* temp = head;
        while (temp != NULL) {
            cout << setw(20) << left << temp->nama << setw(20)
<< left << temp->nim << endl;

```

```

        temp = temp->next;
    }
}

void deleteAll() {
    while (head != NULL) {
        identitasMhs_057* temp = head;
        head = head->next;
        delete temp;
    }
    cout << endl << "Semua data mahasiswa telah
dihapus"<<endl;
}
};

int main() {
    LinkedList linkedList;
    int pilihan;
    string nama, nim;
    int posisi;
    while (true) {
        cout << endl;
        linkedList.print();
        cout << endl;
        cout << "===== " <<
endl;
        cout << "PROGRAM SINGLE LINKED LIST NON-
CIRCULAR"<<endl;
        cout << "===== " <<
endl;
        cout << endl ;
        cout << "1. Tambah Depan"<<endl;
        cout << "2. Tambah Belakang"<<endl;
        cout << "3. Tambah Tengah"<<endl;
        cout << "4. Ubah Depan"<<endl;
        cout << "5. Ubah Belakang"<<endl;
        cout << "6. Ubah Tengah"<<endl;
        cout << "7. Hapus Depan"<<endl;
        cout << "8. Hapus Belakang"<<endl;
        cout << "9. Hapus Tengah"<<endl;
        cout << "10. Hapus List"<<endl;
        cout << "11. Tampilkan"<<endl;
        cout << "0. Keluar"<<endl;

        cout << endl;

```

```
cout << "Pilih Operasi: ";
cin >> pilihan;

cout << endl;

switch (pilihan) {
    case 1:
        cout << "-Tambah Depan"<<endl;
        cout << endl;
        cout << "Masukkan Nama : ";
        cin >> nama;
        cout << "Masukkan NIM : ";
        cin >> nim;
        linkedList.addFirst(nama, nim);
        break;
    case 2:
        cout << "-Tambah Belakang"<<endl;
        cout << endl;
        cout << "Masukkan Nama : ";
        cin >> nama;
        cout << "Masukkan NIM : ";
        cin >> nim;
        linkedList.addLast(nama, nim);
        break;
    case 3:
        cout << "-Tambah Tengah"<<endl;
        cout << endl;
        cout << "Masukkan Nama : ";
        cin >> nama;
        cout << "Masukkan NIM : ";
        cin >> nim;
        cout << "Masukkan Posisi : ";
        cin >> posisi;
        linkedList.addMiddle(nama, nim, posisi);
        break;
    case 4:
        cout << "-Ubah Depan"<<endl;
        cout << endl;
        cout << "Masukkan Nama Baru : ";
        cin >> nama;
        cout << "Masukkan NIM Baru : ";
        cin >> nim;
        linkedList.changeFirst(nama, nim);
        break;
```

```

case 5:
    cout << "-Ubah Belakang"<<endl;
    cout << endl;
    cout << "Masukkan Nama Baru : ";
    cin >> nama;
    cout << "Masukkan NIM Baru : ";
    cin >> nim;
    linkedList.changeLast(nama, nim);
    break;
case 6:
    cout << "-Ubah Tengah"<<endl;
    cout << endl;
    cout << "Masukkan Nama Baru : ";
    cin >> nama;
    cout << "Masukkan NIM Baru : ";
    cin >> nim;
    cout << "Masukkan Posisi : ";
    cin >> posisi;
    linkedList.changeMiddle(nama, nim, posisi);
    break;
case 7:
    cout << "-Hapus Depan"<<endl;
    cout << endl;
    linkedList.deleteFirst();
    break;
case 8:
    cout << "-Hapus Belakang"<<endl;
    cout << endl;
    linkedList.deleteLast();
    break;
case 9:
    cout << "-Hapus Tengah"<<endl;
    cout << endl;
    cout << "Masukkan Posisi : ";
    cin >> posisi;
    linkedList.deleteMiddle(posisi);
    break;
case 10:
    cout << "-Hapus List"<<endl;
    cout << endl;
    linkedList.deleteAll();
    break;
case 11:
    cout << "-Tampilkan"<<endl;

```

```

        cout << endl;
        linkedList.print();
        break;
    case 0:
        exit(0);
    default:
        cout << "Pilihan tidak valid"<<endl;
    }
}

return 0;
}

```

Screenshots Output :

```

=====
PROGRAM SINGLE LINKED LIST NON-CIRCULAR
=====

1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 1

-Tambah Depan

Masukkan Nama : Nia
Masukkan NIM : 2311102057

Data telah ditambahkan

```

```
=====
PROGRAM SINGLE LINKED LIST NON-CIRCULAR
=====
```

1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

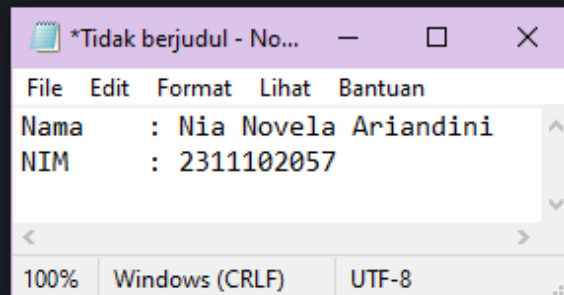
Pilih Operasi: 4

-Ubah Depan

Masukkan Nama Baru : Novel

Masukkan NIM Baru : 2311102057

Data di depan telah diubah



```
=====
PROGRAM SINGLE LINKED LIST NON-CIRCULAR
=====
```

1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 7

-Hapus Depan

Data di depan telah dihapus

Deskripsi :

Program yang telah dibuat pada Unguided1 adalah program menu untuk Linked List Non-Circular yang digunakan untuk menyimpan Nama dan NIM mahasiswa. Program tersebut memungkinkan pengguna untuk melakukan berbagai operasi dasar seperti penambahan data (di depan, di belakang, atau di tengah), penghapusan data (di depan, di belakang, atau di tengah), pengubahan data, dan menampilkan isi linked list. Pengguna dapat memilih operasi yang diinginkan melalui menu yang disediakan hingga mereka memilih untuk keluar dari program. Setiap operasi yang dipilih akan memanggil metode yang sesuai pada kelas LinkedList. Setiap node dalam linked list menyimpan informasi mengenai nama dan NIM mahasiswa serta pointer yang menunjuk ke node berikutnya. Setiap opsi dalam menu memiliki fungsi spesifik dalam mengelola linked list :

2. Tambah Depan : Menambahkan node dengan data baru di bagian depan linked list.
3. Tambah Belakang : Menambahkan node dengan data baru di bagian belakang linked list.
4. Tambah Tengah : Menambahkan node dengan data baru pada posisi tertentu dalam linked list.
5. Hapus Depan : Menghapus node pertama dari linked list.
6. Hapus Belakang : Menghapus node terakhir dari linked list.
7. Hapus Tengah : Menghapus node pada posisi tertentu dalam linked list.
8. Ubah Depan : Mengubah data pada node pertama dalam linked list.
9. Ubah Belakang : Mengubah data pada node terakhir dalam linked list.
10. Ubah Tengah : Mengubah data pada node pada posisi tertentu dalam linked list.
11. Tampilkan : Menampilkan seluruh data dalam linked list.
12. Hapus List : Menghapus seluruh data dalam linked list. Setiap opsi memberikan pengguna kemampuan untuk memanipulasi data dalam linked list sesuai dengan kebutuhan mereka.

2. Setelah membuat menu tersebut, masukkan data sesuai urutan berikut, lalu tampilkan data yang telah dimasukkan. (Gunakan insert depan, belakang atau tengah).

Nama	NIM
Jawad	23300001
[Nama Anda]	[NIM Anda]
Farrel	23300003
Denis	23300005
Anis	23300008
Bowo	23300015
Gahar	23300040
Udin	23300048
Ucok	23300050
Budi	23300099

Screenshots Output :

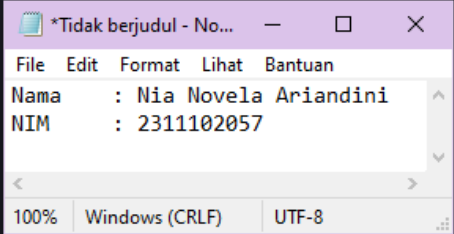
```
===== DATA MAHASISWA =====
=====
NAMA          NIM
Jawad         23300001
Nia           2311102057
Farrel        23300003
Denis         23300005
Anis          23300008
Bowo          23300015
Gahar         23300040
Udin          23300048
Ucok          23300050
Budi          23300099

=====
PROGRAM SINGLE LINKED LIST NON-CIRCULAR
=====

1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 0

PS D:\SEMESTER 2\PRAKTIKUM STRUKTUR DATA\Praktikum Struktur Data - Semester 2> 
```



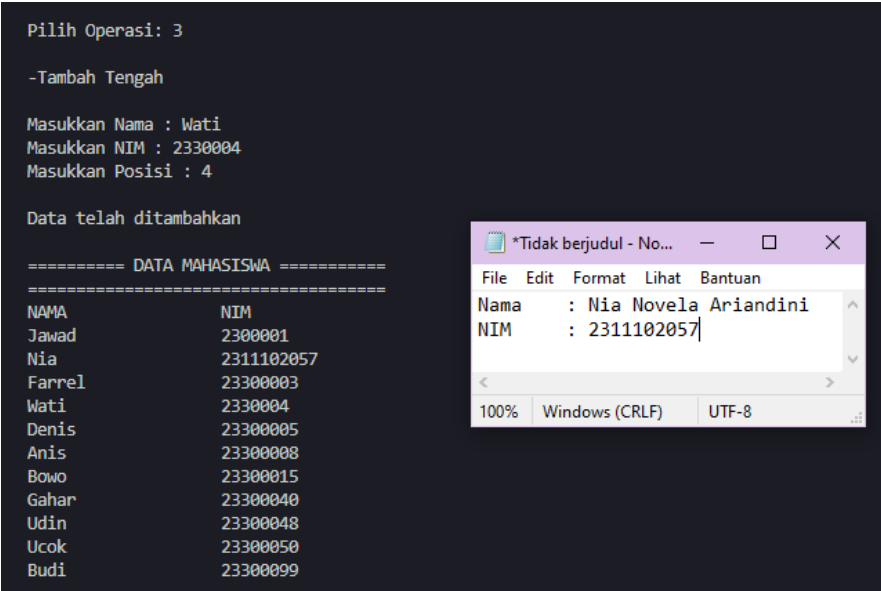
Deskripsi :

Setelah membuat menu, data dimasukkan sesuai urutan yang diminta dari soal Unguided2. Pertama, data dengan urutan pertama dimasukkan menggunakan operasi tambah depan. Kemudian, data urutan selanjutnya dimasukkan menggunakan operasi tambah belakang. Tampilan data akan muncul otomatis ketika setiap data ditambahkan.

3. Lakukan perintah berikut :

a) Tambahkan data berikut diantara Farrel dan Denis:

Wati 2330004



Pilih Operasi: 3

-Tambah Tengah

Masukkan Nama : Wati
Masukkan NIM : 2330004
Masukkan Posisi : 4

Data telah ditambahkan

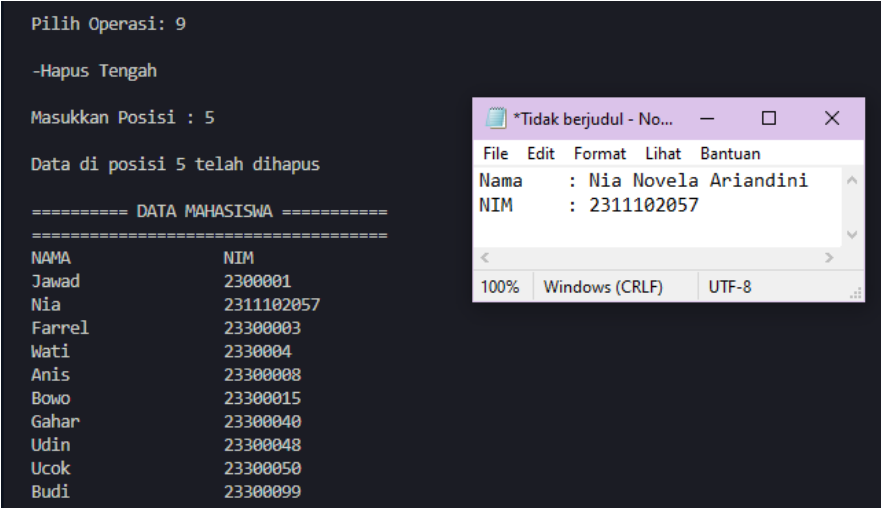
```
===== DATA MAHASISWA =====  
=====
```

NAMA	NIM
Jawad	2300001
Nia	2311102057
Farrel	23300003
Wati	23300004
Denis	23300005
Anis	23300008
Bowo	23300015
Gahar	23300040
Udin	23300048
Ucok	23300050
Budi	23300099

Inset window content:

```
*Tidak berjudul - No...  
File Edit Format Lihat Bantuan  
Nama : Nia Novela Ariandini  
NIM : 2311102057  
100% Windows (CRLF) UTF-8
```

b) Hapus data Denis



Pilih Operasi: 9

-Hapus Tengah

Masukkan Posisi : 5

Data di posisi 5 telah dihapus

```
===== DATA MAHASISWA =====  
=====
```

NAMA	NIM
Jawad	2300001
Nia	2311102057
Farrel	23300003
Wati	23300004
Anis	23300008
Bowo	23300015
Gahar	23300040
Udin	23300048
Ucok	23300050
Budi	23300099

Inset window content:

```
*Tidak berjudul - No...  
File Edit Format Lihat Bantuan  
Nama : Nia Novela Ariandini  
NIM : 2311102057  
100% Windows (CRLF) UTF-8
```

c) Tambahkan data berikut di awal:

Owi 2330000

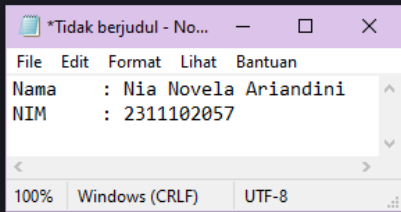
```
Pilih Operasi: 1

-Tambah Depan

Masukkan Nama : Owi
Masukkan NIM : 2330000

Data telah ditambahkan

===== DATA MAHASISWA =====
=====
NAMA          NIM
Owi           2330000
Jawad        2300001
Nia          2311102057
Farrel       23300003
Wati         23300004
Anis         23300008
Bowo         23300015
Gahar        23300040
Udin         23300048
Ucok         23300050
Budi         23300099
```



d) Tambahkan data berikut di akhir:

David 23300100

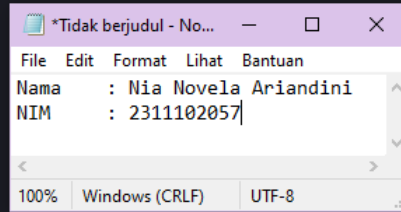
```
Pilih Operasi: 2

-Tambah Belakang

Masukkan Nama : David
Masukkan NIM : 23300100

Data telah ditambahkan

===== DATA MAHASISWA =====
=====
NAMA          NIM
Owi           2330000
Jawad        2300001
Nia          2311102057
Farrel       23300003
Wati         23300004
Anis         23300008
Bowo         23300015
Gahar        23300040
Udin         23300048
Ucok         23300050
Budi         23300099
David        23300100
```



e) Ubah data Udin menjadi data berikut:

Idin 23300045

The terminal window shows the following sequence of operations:

```
Pilih Operasi: 6
-Ubah Tengah

Masukkan Nama Baru : Idin
Masukkan NIM Baru : 23300045
Masukkan Posisi : 9

Data di posisi 9 telah diubah

===== DATA MAHASISWA =====
=====
NAMA          NIM
Owi           2330000
Jawad         2300001
Nia           2311102057
Farrel        23300003
Wati          2330004
Anis          23300008
Bowo          23300015
Gahar         23300040
Idin          23300045
Ucok          23300050
Budi          23300099
David         23300100
```

The text editor window shows the following content:

```
*Tidak berjudul - No...
File Edit Format Lihat Bantuan
Nama : Nia Novela Ariandini
NIM : 2311102057
100% Windows (CRLF) UTF-8
```

f) Ubah data terkahir menjadi berikut:

Lucy 23300101

The terminal window shows the following sequence of operations:

```
Pilih Operasi: 5
-Ubah Belakang

Masukkan Nama Baru : Lucy
Masukkan NIM Baru : 23300101

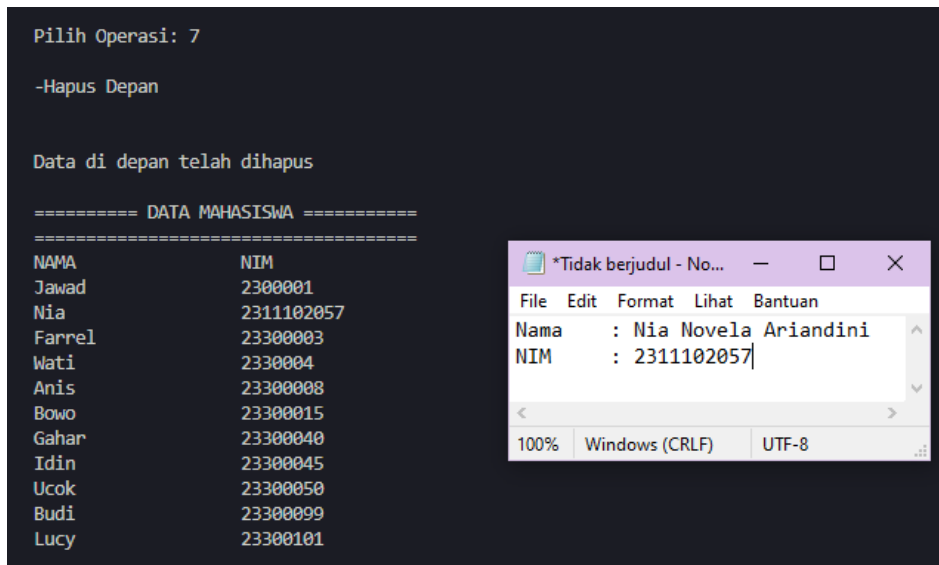
Data di belakang telah diubah

===== DATA MAHASISWA =====
=====
NAMA          NIM
Owi           2330000
Jawad         2300001
Nia           2311102057
Farrel        23300003
Wati          2330004
Anis          23300008
Bowo          23300015
Gahar         23300040
Idin          23300045
Ucok          23300050
Budi          23300099
Lucy          23300101
```

The text editor window shows the following content:

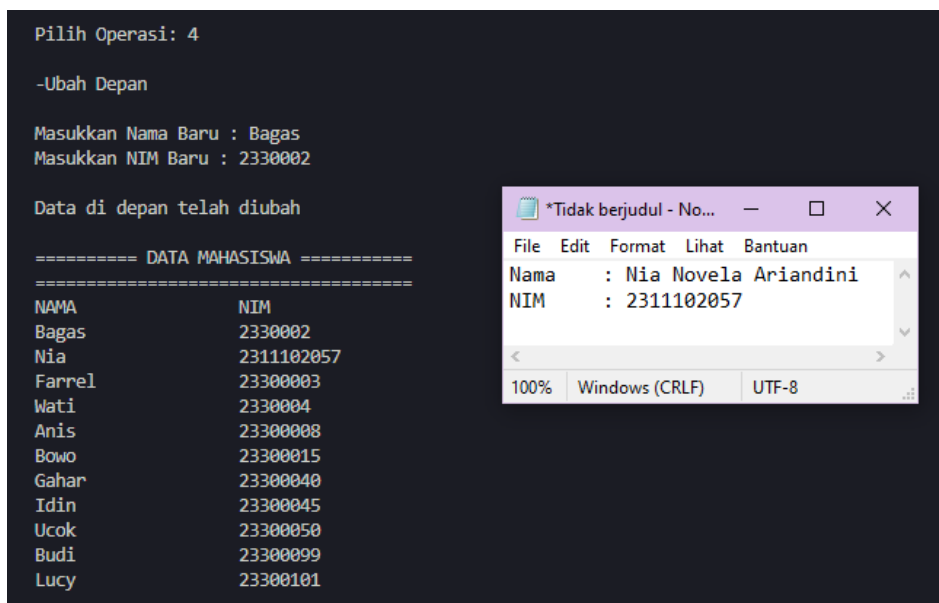
```
*Tidak berjudul - No...
File Edit Format Lihat Bantuan
Nama : Nia Novela Ariandini
NIM : 2311102057
100% Windows (CRLF) UTF-8
```

g) Hapus data awal

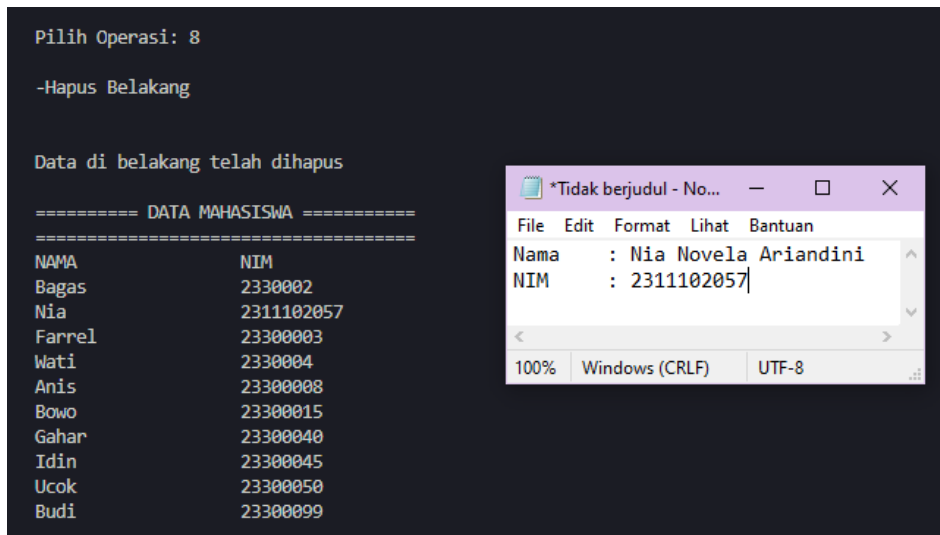


h) Ubah data awal menjadi berikut:

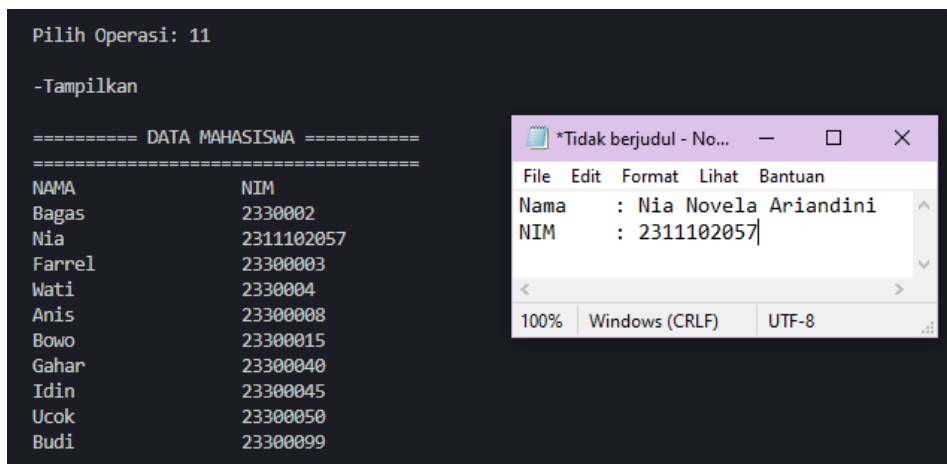
Bagas 2330002



i) Hapus data akhir



j) Tampilkan seluruh data



Deskripsi :

Pada perintah `unguided3`, masih menggunakan kode yang sama dengan `unguided 1` dan `2`, namun ditambahkan beberapa menu baru. Menu-menu yang ditambahkan antara lain untuk menambahkan data di tengah antara Farrel dan Denis, menghapus data Denis, menambahkan data Owi di bawah data yang sudah ada, mengubah data tengah menjadi Idin dari nama Udin, mengubah data terakhir menjadi Lucy, menghapus data awal, mengubah data awal menjadi Bagas, menghapus data terakhir, dan menampilkan seluruh data menggunakan menu tampilkan. Untuk menjalankan perintah-perintah tersebut, digunakan linked list untuk menyimpan data mahasiswa. Linked list adalah struktur data yang terdiri dari simpul-simpul yang terhubung satu sama lain secara berurutan, dimana setiap simpul memiliki dua bagian utama: data (nama dan NIM mahasiswa) dan pointer yang menunjuk ke simpul berikutnya.

D. Kesimpulan

Yang terdapat dalam guided maupun unguided, merupakan implementasi dari struktur data Linked List dalam bahasa pemrograman C++. Program-program tersebut mencakup fitur dasar seperti penambahan, penghapusan, dan pengubahan data dalam Linked List, serta kemampuan untuk menampilkan isi Linked List. Program-program ini juga dilengkapi dengan menu interaktif yang memungkinkan pengguna untuk memilih operasi yang diinginkan terhadap Linked List, seperti menambah, mengubah, atau menghapus data, serta menampilkan isi Linked List. Dengan demikian, program-program tersebut dapat digunakan sebagai contoh implementasi Linked List dalam bahasa pemrograman C++ dan sebagai alat yang berguna dalam pengelolaan dan manipulasi data menggunakan struktur data Linked List.

E. Referensi

[1] Anita, Sindar. STRUKTUR DATA DAN ALGORITMA DENGAN C++ : 2019 [2] Joseph Teguh Santoso., STRUKTUR DATA dan ALGORITMA (Bagian 1). Semarang: Yayasan Prima Agus Teknik, 2021.