

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

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



**Disusun Oleh :**

**NAMA : ANISA YASAROH**

**NIM : 2311102063**

**Dosen :**

**WAHYU ANDI SAPUTRA, S.Pd., M.Eng.**

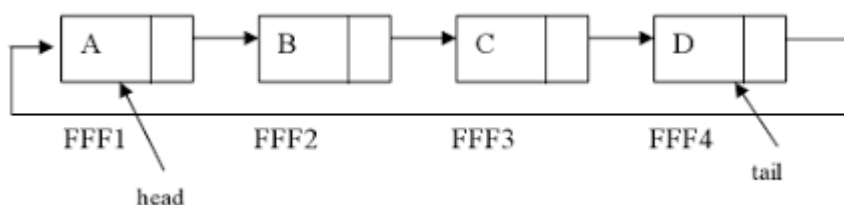
**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

## A. DASAR TEORI

Linked list merupakan struktur data linier yang terdiri dari sejumlah simpul (node) yang saling terhubung melalui pointer atau referensi. Setiap simpul dalam linked list menyimpan data dan memiliki sebuah pointer yang menunjuk ke simpul berikutnya dalam urutan linier. Linked list dapat dibagi menjadi dua jenis, yaitu :

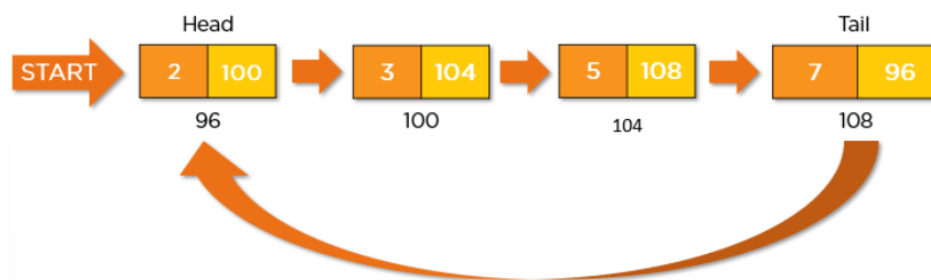
### 1. Linked List Non Circular

Linked list non circular adalah linked list dengan node pertama (head) dan node terakhir (tail) tidak saling terhubung. Linked list non circular merupakan field pointernya hanya satu buah saja dan satu arah serta pada akhir node, pointernya menunjuk NULL yang akan digunakan sebagai kondisi berhenti pada saat pembacaan isi linked list. Linked list non circular dapat digambarkan sebagai berikut.



### 2. Linked list Circular

Linked list circular adalah suatu linked list dimana tail (node terakhir) menuju ke head (node pertama). Jadi tidak ada pointer yang menunjuk NULL. Pada linked list circular, jika ingin menambah atau menghapus sebuah simpul, tidak perlu memperbarui pointer pada simpul sebelumnya karena simpul terakhir selalu menunjuk kembali ke simpul pertama. Linked list circular dapat digambarkan sebagai berikut.



## 1. GUIDED

### 1. Guided 1 – Linked List Non Circular

#### Source Code

```
#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;
}

```

### Screenshoot Output

The screenshot shows a terminal window on the left and a Notepad++ window on the right. The terminal window displays the output of the program, which is a linked list. The output is as follows:

```

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:\Struktur Data smt 2>

```

The Notepad++ window, titled 'Nama AN', shows the following text:

```

Nama : ANISA YASAROH
NIM  : 2311102063

```

The status bar at the bottom of the Notepad++ window indicates 'Ln 2, Col 18 | 38 characters | 100% | Window | UTF-8'.

### Deskripsi Program

Program diatas yaitu terdapat struktur 'node' yang menyimpan data dan pointer ke simpul berikutnya. Terdapat juga dua pointer, 'head' dan 'tail', yang menunjukkan awal dan akhir dari linked list. Program menyediakan fungsi-fungsi

dasar seperti 'insertDepan', 'insertBelakang', 'insertTengah', 'hapusDepan', 'hapusBelakang', 'hapusTengah', 'ubahDepan', 'ubahBelakang', 'ubahTengah', 'hitungList', 'isEmpty', 'clearList', dan 'tampil' untuk mengoperasikan linked list tersebut. Fungsi-fungsi tersebut dipanggil dalam 'main()' untuk melakukan operasi-operasi dasar pada linked list seperti penyisipan, penghapusan, pengubahan elemen, serta menampilkan isi linked list setiap kali terjadi perubahan,

## 2. Guided 2 – Linked List Circular

### Source Code

```
#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();
}

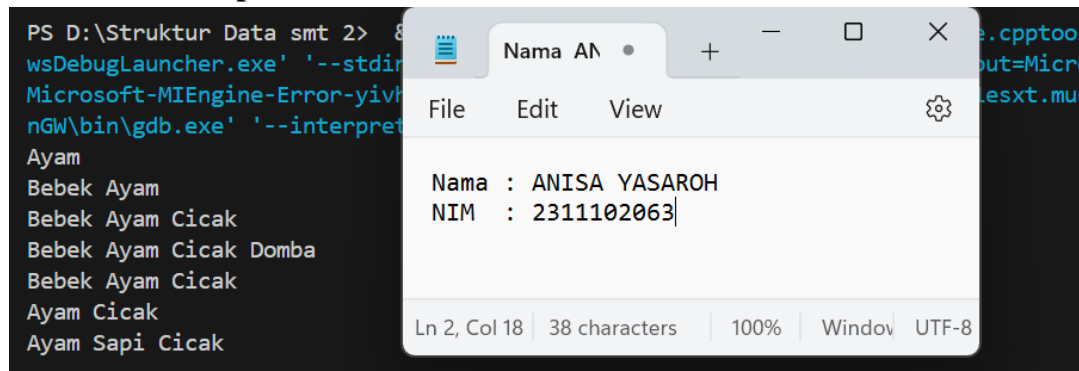
```

```

insertBelakang("Domba");
tampil();
hapusBelakang();
tampil();
hapusDepan();
tampil();
insertTengah("Sapi", 2);
tampil();
hapusTengah(2);
tampil();
return 0;
}

```

### Screenshoot Output



### Deskripsi Program

Program diatas yaitu terdapat struktur 'node' yang memiliki dua anggota,, yaitu 'data' yang menyimpan string, dan 'next' yang merupakan pointer ke simpul berikutnya. Terdapat juga beberapa pointer seperti 'head', 'tail', 'baru', 'bantu', dan 'hapus', yang digunakan untuk mengantur operasi pada linked list. Program ini menyediakan fungsi-fungsi dasar seperti 'init', 'isEmpty', 'butNode', 'hitungList', 'insertDepan', 'insertBelakang', 'insertTengah', 'hapusDepan', 'hapusBelakang', 'hapusTengah', 'clearList', dan 'tampil', yang digunakan untuk mengelola circular linked list, seperti menyisipkan, menghapus, dan menampilkan elemen-elemen dalam linked list. Dalam 'main()', fungsi-fungsi tersebut dipanggil untuk melakukan operasi-operasi tersebut dan menampilkan hasilnya.

## 2. UNGUIDED

Buatlah program menu Linked List Non Circular untuk menyimpan **Nama dan NIM mahasiswa**, dengan menggunakan *input* dari user.

### 1. Unguided 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

• Tampilan Operasi Hapus:

-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

- 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

## Source Code

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

struct dataMahasiswa63 {
    string nama;
    string nim;
    dataMahasiswa63* next;
};

class LinkedList {
private:
    dataMahasiswa63* head;

public:
    LinkedList() {
        head = NULL;
    }

    void addFirst(string nama, string nim) {
        dataMahasiswa63* newNode = new dataMahasiswa63();
        newNode->nama = nama;
        newNode->nim = nim;
        newNode->next = head;
        head = newNode;
        cout << endl << "Data telah ditambahkan."<< endl;
    }

    void addLast(string nama, string nim) {
        dataMahasiswa63* newNode = new dataMahasiswa63();
        newNode->nama = nama;
        newNode->nim = nim;
        newNode->next = NULL;
        if (head == NULL) {
            head = newNode;
        } else {
            dataMahasiswa63* 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;
    }
    dataMahasiswa63* newNode = new dataMahasiswa63();
    newNode->nama = nama;
    newNode->nim = nim;
    if (posisi == 1) {
        newNode->next = head;
        head = newNode;
    } else {
        dataMahasiswa63* 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;
    }
    dataMahasiswa63* 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;
        }
        dataMahasiswa63* 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;
        }
        dataMahasiswa63* 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;
    }
    dataMahasiswa63* 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;
    }
    dataMahasiswa63* 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;
    }
    dataMahasiswa63* 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 << "===== "<<
endl;
    cout << "          DATA MAHASISWA          "<<
endl;
    cout << "===== "<<
endl;
    cout << setw(20) << left << "NAMA" << setw(20) <<
left << "NIM" << endl;
    dataMahasiswa63* 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) {
        dataMahasiswa63* 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;

```

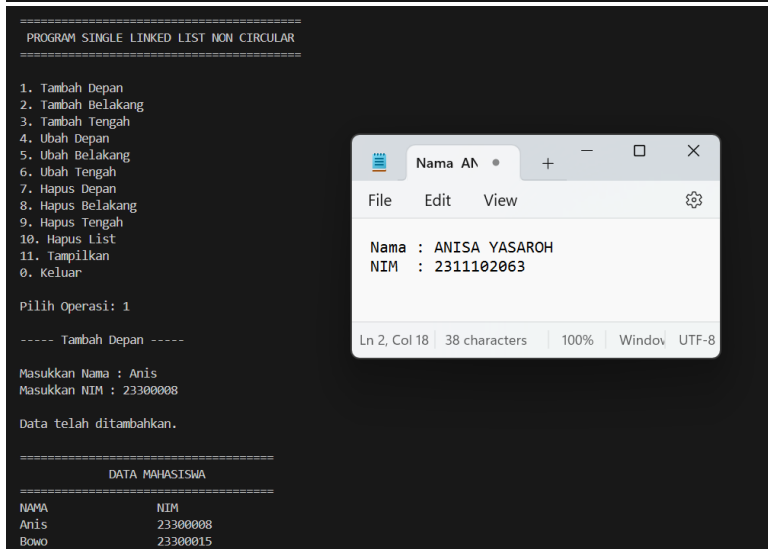
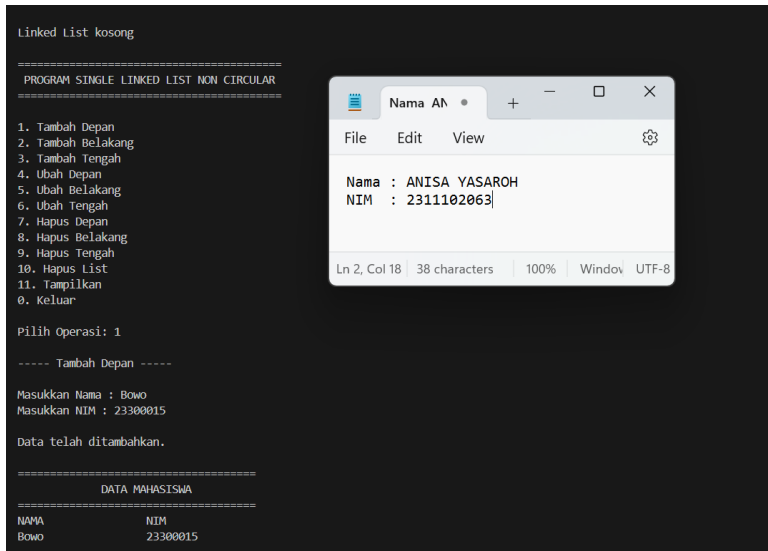
```

        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;
}

```

## Screenshoot 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

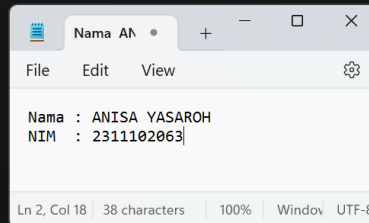
Pilih Operasi: 1

----- Tambah Depan -----

Masukkan Nama : Denis
Masukkan NIM : 23300005

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA          NIM
Denis          23300005
Anis           23300008
Bowo           23300015
```



```
=====
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

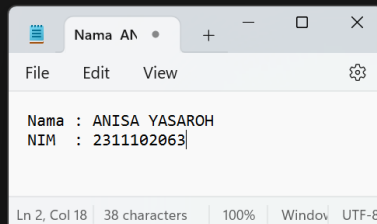
Pilih Operasi: 1

----- Tambah Depan -----

Masukkan Nama : Farrel
Masukkan NIM : 23300003

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA          NIM
Farrel         23300003
Denis          23300005
Anis           23300008
Bowo           23300015
```



```
=====
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

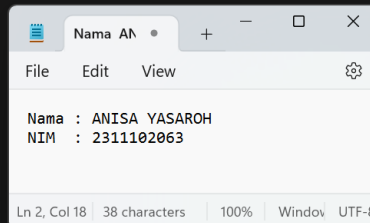
Pilih Operasi: 1

----- Tambah Depan -----

Masukkan Nama : Jawad
Masukkan NIM : 23300001

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad          23300001
Farrel         23300003
Denis          23300005
Anis           23300008
Bowo           23300015
```



```
=====
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

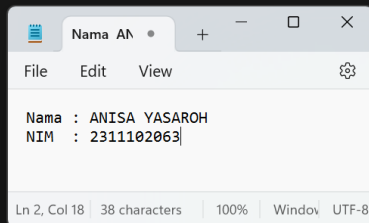
Pilih Operasi: 3

----- Tambah Tengah -----

Masukkan Nama : Anisa
Masukkan NIM : 2311102063
Masukkan Posisi : 2

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad          23300001
Anisa          2311102063
Farrel         23300003
Denis          23300005
Anis           23300008
Bowo           23300015
```



```
=====
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

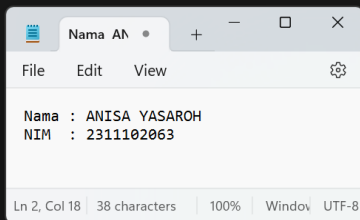
Pilih Operasi: 2

----- Tambah Belakang -----

Masukkan Nama : Gahar
Masukkan NIM : 23300040

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad          23300001
Anisa          2311102063
Farrel         23300003
Denis          23300005
Anis           23300008
Bowo           23300015
Gahar          23300040
```





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: 2

----- Tambah Belakang -----

Masukkan Nama : Udin  
Masukkan NIM : 23300048

Data telah ditambahkan

DATA MAHASISWA	
NAMA	NIM
Jawad	23300001
Anisa	2311102063
Farrel	23300003
Denis	23300005
Anis	23300008
Bowo	23300015
Gahar	23300040
Udin	23300048

Nama AN

File Edit View

Nama : ANISA YASAROH  
NIM : 2311102063

Ln 2, Col 18 | 38 characters | 100% | Window UTF-8

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: 2

----- Tambah Belakang -----

Masukkan Nama : Budi  
Masukkan NIM : 23300099

Data telah ditambahkan

DATA MAHASISWA	
NAMA	NIM
Jawad	23300001
Anisa	2311102063
Farrel	23300003
Denis	23300005
Anis	23300008
Bowo	23300015
Gahar	23300040
Udin	23300048
Budi	23300099

Nama AN

File Edit View

Nama : ANISA YASAROH  
NIM : 2311102063

Ln 2, Col 18 | 38 characters | 100% | Window UTF-8

```
=====
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

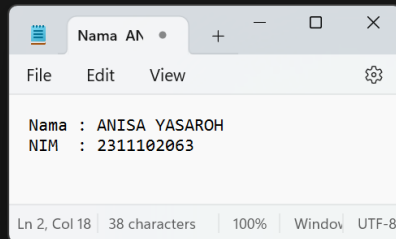
Pilih Operasi: 3

----- Tambah Tengah -----

Masukkan Nama : Uco
Masukkan NIM : 23300050
Masukkan Posisi : 9

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA      NIM
Jawad     23300001
Anisa     2311102063
Farrel    23300003
Denis     23300005
Anis      23300008
Bowo      23300015
Gahar     23300040
Udin      23300048
Uco       23300050
Budi      23300099
```



3. Lakukan perintah berikut:
- a) Tambahkan data berikut diantara Farrel dan Denis:  
**Wati 2330004**
  - b) Hapus data Denis
  - c) Tambahkan data berikut di awal:  
**Owi 2330000**
  - d) Tambahkan data berikut di akhir:  
**David 23300100**
  - e) Ubah data Udin menjadi data berikut:  
**Idin 23300045**
  - f) Ubah data terakhir menjadi berikut:  
**Lucy 23300101**
  - g) Hapus data awal
  - h) Ubah data awal menjadi berikut:  
**Bagas 2330002**
  - i) Hapus data akhir
  - j) Tampilkan seluruh data

## Screenshot Output

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

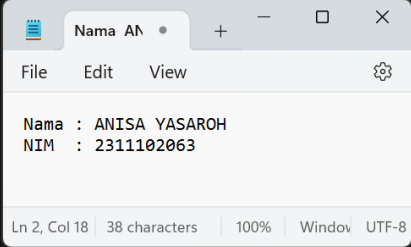
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 3

----- Tambah Tengah -----

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

Data telah ditambahkan.
```



```
=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad         23300001
Anisa         2311102063
Farrel        23300003
Wati          23300004
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: 9

----- Hapus Tengah -----

Masukkan Posisi : 5

Data di posisi 5 telah dihapus.

=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad         23300001
Anisa         2311102063
Farrel        23300003
Wati          23300004
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

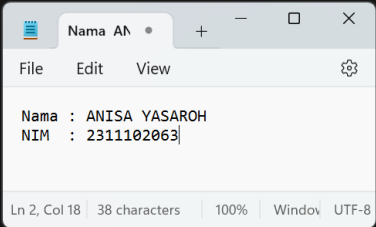
Pilih Operasi: 1

----- Tambah Depan -----

Masukkan Nama : Owi
Masukkan NIM : 2330000

Data telah ditambahkan.

=====
DATA MAHASISWA
=====
NAMA      NIM
Owi        2330000
Jawad      23300001
Anisa      2311102063
Farrel     23300003
Wati       23300004
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

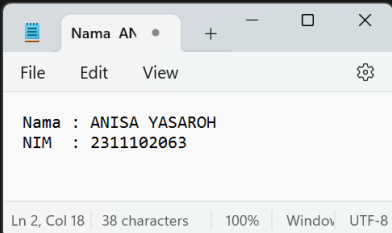
Pilih Operasi: 2

----- Tambah Belakang -----

Masukkan Nama : David
Masukkan NIM : 23300100

Data telah ditambahkan

=====
DATA MAHASISWA
=====
NAMA      NIM
Owi        2330000
Jawad      23300001
Anisa      2311102063
Farrel     23300003
Wati       23300004
Anis       23300008
Bowo       23300015
Gahar      23300040
Udin       23300048
Ucok       23300050
Budi       23300099
David      23300100
```



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

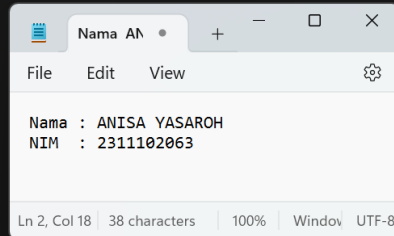
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 6

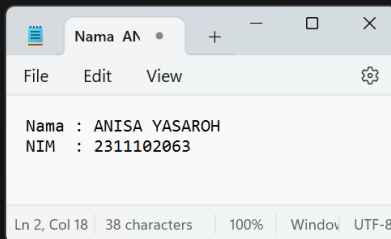
----- Ubah Tengah -----

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

Data di posisi 9 telah diubah.
```



```
=====
DATA MAHASISWA
=====
NAMA      NIM
Owi        2330000
Jawad      23300001
Anisa      2311102063
Farrel     23300003
Wati       23300004
Anis       23300008
Bowo       23300015
Gahar      23300040
Idin       23300045
Ucok       23300050
Budi       23300099
David      23300100
```



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

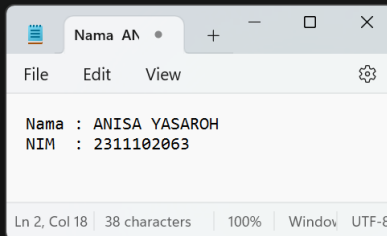
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 5

----- Ubah Belakang -----

Masukkan Nama Baru : Lucy
Masukkan NIM Baru : 23300101

Data di belakang telah diubah.
```



```
=====
DATA MAHASISWA
=====
NAMA      NIM
Owi        2330000
Jawad      23300001
Anisa      2311102063
Farrel     23300003
Wati       23300004
Anis       23300008
Bowo       23300015
Gahar      23300040
Idin       23300045
Ucok       23300050
Budi       23300099
Lucy       23300101
```

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

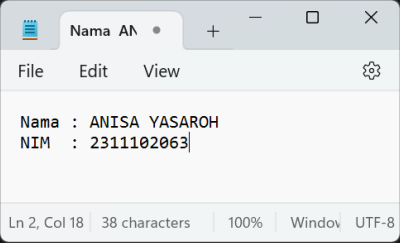
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 7

----- Hapus Depan -----

Data di depan telah dihapus.

=====
DATA MAHASISWA
=====
NAMA          NIM
Jawad          23300001
Anisa           2311102063
Farrel          23300003
Wati            23300004
Anis            23300008
Bowo            23300015
Gahar           23300040
Idin            23300045
Ucok            23300050
Budi            23300099
Lucy            23300101
```



```
=====
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

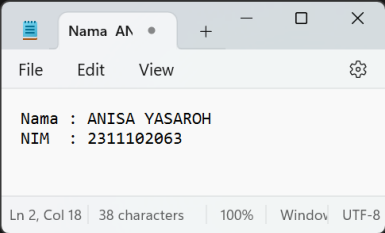
Pilih Operasi: 4

----- Ubah Depan -----

Masukkan Nama Baru : Bagus
Masukkan NIM Baru : 2330002

Data di depan telah diubah

=====
DATA MAHASISWA
=====
NAMA          NIM
Bagas          2330002
Anisa           2311102063
Farrel          23300003
Wati            23300004
Anis            23300008
Bowo            23300015
Gahar           23300040
Idin            23300045
Ucok            23300050
Budi            23300099
Lucy            23300101
```



```
=====
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: 8

----- Hapus Belakang -----

Data di belakang telah dihapus.

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

NAMA	NIM
Bagas	2330002
Anisa	2311102063
Farrel	23300003
Wati	2330004
Anis	23300008
Bowo	23300015
Gahar	23300040
Idin	23300045
Ucok	23300050
Budi	23300099

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

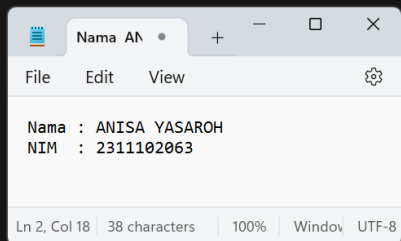
1. Tambah Depan
2. Tambah Belakang
3. Tambah Tengah
4. Ubah Depan
5. Ubah Belakang
6. Ubah Tengah
7. Hapus Depan
8. Hapus Belakang
9. Hapus Tengah
10. Hapus List
11. Tampilkan
0. Keluar

Pilih Operasi: 11

----- Tampilkan -----

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

NAMA	NIM
Bagas	2330002
Anisa	2311102063
Farrel	23300003
Wati	2330004
Anis	23300008
Bowo	23300015
Gahar	23300040
Idin	23300045
Ucok	23300050
Budi	23300099

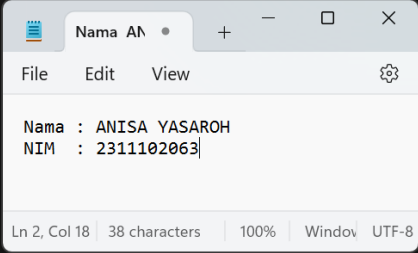


```
=====
PROGRAM SINGLE LINKED LIST NON CIRCULAR
=====

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:\Struktur Data smt 2> 
```



### Deskripsi Program

Program diatas yaitu memiliki struktur data 'dataMahasiswa63' yang menyimpan informasi nama dan NIM mahasiswa, serta pointer 'next' yang menunjukkan ke node selanjutnya dalam linked list. Kelas 'LinkedList' menyediakan berbagai operasi dasar pada linked list, seperti penambahan data di awal, di akhir, atau di tengah linked list, penghapusan data di awal, di akhir, atau di tengah linked list, serta pengubahan data di awal, di akhir, atau di tengah linked list. Fungsi 'main()' merupakan menu utama program, dimana pengguna dapat memilih operasi yang ingin dilakukan terhadap linked list, seperti menambah, mengubah, menghapus, atau menampilkan data mahasiswa. Setiap operasi diimplementasikan dengan memanipulasi pointer-pointer node linked list dan mengatur hubungan antara node-node tersebut sesuai dengan operasi yang diinginkan. Program akan terus berjalan hingga pengguna memilih opsi untuk keluar (pilihan 0).

### 3. KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa :

1. Pada linked list non circular, tail memiliki nilai pointer NULL, sedangkan pada linked list circular, tail memiliki nilai pointer yang menunjuk pada head.
2. Linked list non circular lebih sederhana dan mudah dipahami, sedangkan linked list circular membutuhkan sedikit pemahaman tambahan tentang bagaimana elemen terkait satu sama lain.
3. Pada linked list non circular, jika ingin mengunjungi elemen terakhir, harus memulai dari elemen pertama dan menelusuri seluruh linked list, sedangkan linked list circular bisa langsung melompat dari elemen terakhir ke elemen pertama.



#### 4. REFERENSI

Jierui, C., & Fang, S. (2021, June). STL Performance Experiments and Analysis in undergraduate Data Structure Course. In *2021 IEEE 3rd International Conference on Computer Science and Educational Informatization (CSEI)* (pp. 148-152). IEEE.

Romm, C. (2022). Data Structure. *Organization*, 16(58), 31.

Rutkowski, W. (2020). CS 116-003: Introduction to Computer Science II/C++.

Umann, K., & Porkolab, Z. (2022). Detecting uninitialized variables in C++ with the clang static analyzer. *Acta Cybernetica*, 25(4), 923-940.

Yu, X. (2021). Spacetime C++ Core: *Higher Performance Implementation of Global Object Tracker Programming Model*. University of California, Irvine.