LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

MODUL IV
LINKED LIST CIRCULAR DAN NON CIRCULAR

Disusun Oleh :

NAMA : FAISAL KHOIRUDDIN

NIM : 2311102046

Dosen

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

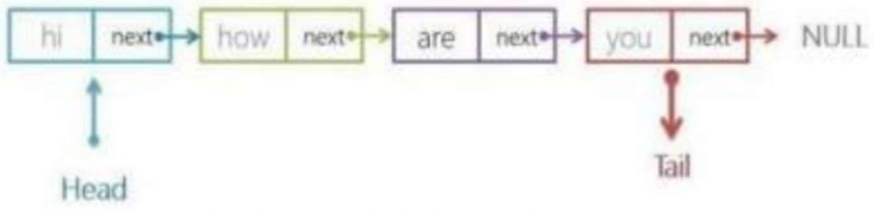
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

A. Dasar Teori

1) Linked List Non Circular

Linked list non circular adalah linked list dengan node pertama (head) dan node terakhir (tail) yang tidak saling terhubung. Pointer terakhir (tail) pada Linked List ini selalu bernilai 'NULL' sebagai pertanda data terakhir dalam list-nya. Linked list non circular dapat digambarkan sebagai berikut.



Gambar 1 *Single Linked List Non Circular*

OPERASI PADA LINKED LIST NON CIRCULAR

1. Deklarasi Simpul (Node)

```
struct node
{
    int data;
    node *next;
};
```

2. Membuat dan Menginisialisasi Pointer Head dan Tail

```
node *head, *tail;
void init()
{
   head = NULL;
   tail = NULL;
};
```

3. Pengecekan Kondisi Linked List

```
bool isEmpty()
{
   if (head == NULL && tail ==
```

```
   NULL) {
      return true;
   }
   else
   {
      return false;
   }
}
```

4. Penambahan Simpul (Node)

```
void insertBelakang(string
dataUser) {
     if (isEmpty() == true)
     {
          node *baru = new node;
          baru->data = dataUser;
          head = baru;
          tail = baru;
          baru->next = NULL;
     }
     else
     {
        node *baru = new node;
        baru->data = dataUser;
        baru->next = NULL;
        tail->next = baru;
        tail = baru;
     }
};
```

5. Penghapusan Simpul (Node)

```
void hapusDepan()
{
    if (isEmpty() == true)
    {
        cout << "List kosong!" <<
    endl; }
    else
    {
       node *helper;
       helper = head;
```

```
        if (head == tail)
        {
            head = NULL;
            tail = NULL;
            delete helper;
        }
        else
           head = head->next;
        helper->next = NULL;
        delete helper;
       }
    }
}
```

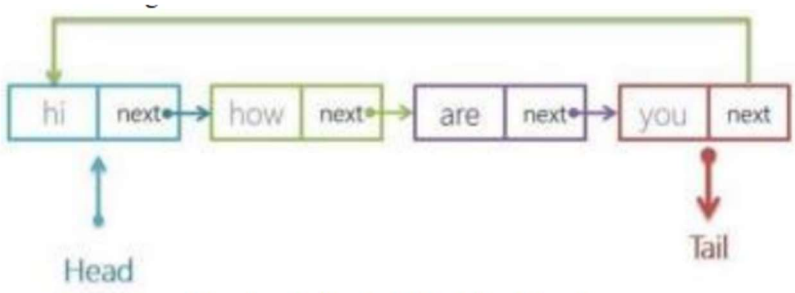
6. Tampil Data Linked List

```
void tampil()
{
   if (isEmpty() == true)
   {
       cout << "List kosong!" << endl;
   }
   else
   {
      node *helper;
      helper = head;
      while (helper != NULL)
      {
      cout << helper->data << ends;
      helper = helper->next;
      }
   }
}
```

2) Linked List Circular

Linked list circular adalah linked list yang tidak memiliki akhir karena node terakhir (tail) tidak bernilai ‘NULL’, tetapi terhubung dengan node pertama (head). Saat menggunakan linked list circular kita membutuhkan dummy node atau node pengecoh yang biasanya dinamakan dengan node current supaya program dapat berhenti menghitung data ketika node current mencapai node

pertama (head). Linked list circular dapat digunakan untuk menyimpan data yang perlu diakses secara berulang, seperti daftar putar lagu, daftar pesan dalam antrian, atau penggunaan memori berulang dalam suatu aplikasi. Linked list circular dapat digambarkan sebagai berikut.



Gambar 2 *Single Linked List Circular*

OPERASI PADA LINKED LIST CIRCULAR

1. Deklarasi Simpul (Node)

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

2. Membuat dan Menginisialisasi Pointer Head dan Tail

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

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

3. Pengecekan Kondisi Linked List

```
int isEmpty()
{
    if (head == NULL)
        return 1; // true
    else
        return 0; // false
}
```

4. Pembuatan Simpul (Node)

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

5. Penambahan Simpul (Node)

```
// Tambah Depan
void insertDepan(string
data) {
    // Buat Node baru
    buatNode(data);

    if (isEmpty() == 1)
    {
        head = baru;
        tail = head;
        baru->next = head;
    }
    else
    {
       while (tail->next != head)
       {
           tail = tail->next;
       }
       baru->next = head;
       head = baru;
       tail->next = head;
    }
}
```

6. Penghapusan Simpul (Node)

```
void hapusBelakang()
{
   if (isEmpty() == 0)
   {
       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;
      }
}
```

7. Menampilkan Data Linked List

```
void tampil()
{
   if (isEmpty() == 0)
   {
       tail = head;
       do
       {
           cout << tail->data << ends;
           tail = tail->next;
       } while (tail != head);
         cout << endl;
    }
}
```

B. Guided

1. Guided 1 : Linked List Non Circular

Source Code

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

// Deklarasi Struct Node
struct Node {
    int data;
    Node* next;
};

Node* head;
Node* tail;

// Inisialisasi Node
void init() {
    head = NULL;
    tail = NULL;
}

// Pengecekan apakah list kosong
bool isEmpty() {
    return head == NULL;
}

// Tambah Node di depan
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;
    }
}

// Tambah Node di belakang
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;
    }
}

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

// Tambah Node di posisi tengah
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;
    }
}

// Hapus Node di depan
void hapusDepan() {
    if (!isEmpty()) {
        Node* hapus = head;
```

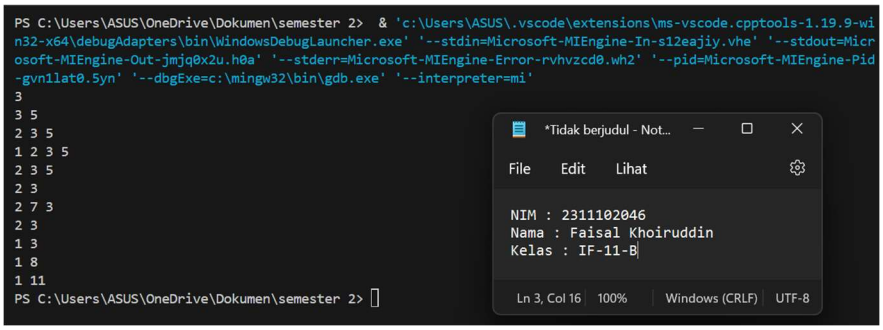
```
        if (head->next != NULL) {
            head = head->next;
            delete hapus;
        } else {
            head = tail = NULL;
            delete hapus;
        }
    } else {
        cout << "List kosong!" << endl;
    }
}

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

// Hapus Node di posisi tengah
void hapusTengah(int posisi) {
    if (posisi < 1 || posisi >
hitungList()) {
        cout << "Posisi diluar jangkauan"
<< endl;
    } else if (posisi == 1) {
        cout << "Posisi bukan posisi
tengah" << endl;
    } else {
        Node* hapus;
        Node* bantu = head;
        for (int nomor = 1; nomor < posisi
- 1; nomor++) {
```

```
            bantu = bantu->next;
        }
        hapus = bantu->next;
        bantu->next = hapus->next;
        delete hapus;
    }
}

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

// Ubah data Node di posisi tengah
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;
            for (int nomor = 1; nomor <
posisi; nomor++) {
                bantu = bantu->next;
            }
            bantu->data = data;
        }
    } else {
        cout << "List masih kosong!" <<
endl;
    }
}

// Ubah data Node di belakang
void ubahBelakang(int data) {
    if (!isEmpty()) {
        tail->data = data;
```

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

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

// Tampilkan semua data Node di list
void tampil() {
    if (!isEmpty()) {
        Node* bantu = head;
        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



Deskripsi:

Program tersebut merupakan single linked list. Program tersebut melakukan operasi seperti menambahkan Node ke depan, belakang, atau posisi tertentu dalam list, menghapus Node dari depan, belakang, atau posisi tertentu dalam list, mengubah data Node di depan, belakang, atau posisi tertentu dalam list, dan menghapus semua Node dalam list. Program tersebut menampilkan output bilangan dari insert depan, insert belakang, hapusDepan, hapusBelakang, insertTengah, hapusTengah, ubahDepan, dan ubahBelakang

- #include <iostream> ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- using namespace std; ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- struct Node ➔ mendefinisikan struct node
- int data; ➔ deklarasi variabel data dengan nama tipe data integer
- Node* next; ➔ pointer next untuk menunjuk ke node berikutnya
- Node* head; ➔ pointer head untuk menunjuk ke node pertama dalam list

- Node* tail; ➔ pointer tail untuk menunjuk ke node terakhir dalam list
- void init() ➔ prosedur init untuk menginisialisasi list
- head = NULL; ➔ mengatur head menjadi null
- tail = NULL; ➔ mengatur tail menjadi null
- bool isEmpty() ➔ fungsi untuk memeriksa apakah list kosong
- return head == NULL; ➔ mengembaikan true jika head yaitu null atau list kosong
- void insertDepan(int nilai) ➔ prosedur untuk memasukkan node di depan list
- Node* baru = new Node; ➔ membuat node
- baru->data = nilai; ➔ mengatur data ke node baru
- baru->next = NULL; ➔ mengatur next dari node baru menjadi null
- if (isEmpty()) ➔ percabangan if, jika list kosong
- head = tail = baru; ➔ node baru menjadi head dan tail
- else ➔ jika list tidak kosong
- baru->next = head; ➔ node baru menjadi head dan tail
- head = baru; ➔ mengatur head menjadi node baru
- void insertBelakang(int nilai) ➔ prosedur untuk memasukkan node di belakang list
- Node* baru = new Node; ➔ membuat node baru
- baru->data = nilai; ➔ mengatur data ke node baru
- baru->next = NULL; ➔ mengatur nest dari node baru menjadi null
- if (isEmpty()) ➔ jika list kosong
- head = tail = baru; ➔ node baru menjadi head dan tail
- else { ➔ jika list tidak kosong
- tail->next = baru; ➔ mengatur next dari tail saat ini menjadi

node baru

- tail = baru; ➔ mengatur tail menjadi node baru
- int hitungList() ➔ prosedur untuk menghitung jumlah node dalam list
- Node* hitung = head; ➔ membuat pointer hitung yang menunjuk head
- int jumlah = 0; ➔ variabel untuk menyimpan jumlah node
- while (hitung != NULL) { ➔ selama hitung bukan sama dengan null (selama ada node dalam list)
- jumlah++; ➔ jumlah++ (increment) selama ada node dalam list
- hitung = hitung->next; ➔ menggeser hitung ke node berikutnya
- return jumlah; ➔ mengembalikan jumlah node dalam list
- void insertTengah(int data, int posisi) ➔ prosedur untuk memasukkan node di Tengah list
- if (posisi < 1 || posisi > hitungList()) ➔ jika posisi < 1 atau posisi lebih dari hitungList (posisi tidak valid)
- cout << "Posisi diluar jangkauan" << endl; ➔ cout menampilkan statement Posisi diluar jangkauan
- else if (posisi == 1) { ➔ jila posisi sama dengan 1
- cout << "Posisi bukan posisi tengah" << endl; ➔ cout menampilkan statement Posisi bukan posisi tengah
- else ➔ jika posisi valid
- Node* baru = new Node(); ➔ membuat node baru
- baru->data = data; ➔ mengatur data ke node baru
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- int nomor = 1; ➔ nomor sama dengan 1
- while (nomor < posisi - 1) ➔ menggeser bant uke posisi sebelum posisi di tuju
- bantu = bantu->next; ➔ bantu sama dengan bantu -> next

- nomor++; ➔ nomor++ (increment)
- baru->next = bantu->next; ➔ mengatur next dari node baru menjadi node setelah bantu
- bantu->next = baru; ➔ mengatur next dari bantu menjadi node baru
- void hapusDepan() ➔ prosedur untuk menghapus node di depan
- if (!isEmpty()) { ➔ jika list tidak kosong
- Node* hapus = head; ➔ membuat pointer hapus yang menunjuk ke head
- if (head->next != NULL) { ➔ jika ada lebih dari satu node dalam list
- head = head->next; ➔ mengatur head menjadi node setelah head saat ini
- delete hapus; ➔ menghapus node yang ditunjuk oleh hapus
- } else { ➔ jika list kosong
- head = tail = NULL; ➔ mengatur head dan tail menjadi null
- delete hapus; ➔ menghapus node yang ditunjuk oleh hapus
- } else { ➔ jika list kosong
- cout << "List kosong!" << endl; ➔ cout menampilkan statement List kosong
- void hapusBelakang() { ➔ prosedur untuk menghapus node di belakang list
- if (!isEmpty()) { ➔ jika list tidak kosong
- if (head != tail) { ➔ jika ada lebih dari satu node dalam list
- Node* hapus = tail; ➔ membuat pointer hapus yang menunjuk ke tail
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- while (bantu->next != tail) { ➔ Menggeser bantu ke node sebelum tail

- bantu = bantu->next; ➔ bantu sama dengan bantu->next
- tail = bantu; ➔ mengatur tail menjadi node yang ditunjuk oleh bantu
- tail->next = NULL; ➔ mengatur next dari tail menjadi NULL
- delete hapus; ➔ menghapus node yang ditunjuk oleh hapus
- } else { ➔ Jika hanya ada satu node dalam list
- head = tail = NULL; ➔ Mengatur head dan tail menjadi NULL
- } else { ➔ Jika list kosong
- cout << "List kosong!" << endl; ➔ cout menampilkan statement List kosong!
- void hapusTengah(int posisi) { ➔ prosedur untuk menghapus node di tengah list
- if (posisi < 1 || posisi > hitungList()) { ➔ jika posisi kurang dari 1 atau posisi lebih dari hitungList (Jika posisi tidak valid)
- cout << "Posisi diluar jangkauan" << endl; ➔ cout menampilkan statement Posisi diluar jangkauan
- } else if (posisi == 1) { ➔jika posisi sama dengan 1 (depan list)
- cout << "Posisi bukan posisi tengah" << endl; ➔ cout menampilkan statement Posisi bukan posisi tengah
- } else { ➔ jika posisi valid
- Node* hapus; ➔ pointer hapus node
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- for (int nomor = 1; nomor < posisi - 1; nomor++) { ➔ perulangan for untuk menggeser bantu ke posisi sebelum posisi yang dituju
- bantu = bantu->next; ➔ bantu sama dengan -> next
- hapus = bantu->next; ➔ membuat pointer hapus yang menunjuk ke node setelah bantu
- bantu->next = hapus->next; ➔ mengatur next dari bantu menjadi node setelah node yang ditunjuk oleh hapus

- delete hapus; ➔ menghapus node yang ditunjuk oleh hapus
- void ubahDepan(int data) { ➔ prosedur untuk mengubah data node di depan list
- if (!isEmpty()) { ➔ jika list tidak kosong
- head->data = data; ➔ mengubah data head menjadi data baru
- } else { ➔ jika list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- void ubahTengah(int data, int posisi) { ➔ prosedur untuk mengubah data node di tengah list
- if (!isEmpty()) { ➔ jika list tidak kosong
- if (posisi < 1 || posisi > hitungList()) { ➔ jika posisi kurang dari 1 atau posisi lebih dari hitungList() (jika posisi tidak valid)
- cout << "Posisi di luar jangkauan" << endl; ➔ cout menampilkan statement Posisi di luar jangkauan
- } else if (posisi == 1) { ➔ jika posisi sama dengan 1 (depan list)
- cout << "Posisi bukan posisi tengah" << endl; ➔ cout menampilkan statement posisi bukan posisi tengah
- } else { ➔ Jika posisi valid
- Node* bantu = head; ➔
- for (int nomor = 1; nomor < posisi; nomor++) { ➔ perulangan for untuk menggeser bantu ke posisi yang dituju
- bantu = bantu->next; ➔ bantuk sama dengan bantu->next
- bantu->data = data; ➔ mengubah data node yang ditunjuk oleh bantu menjadi data baru
- } else { ➔ jika list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- void ubahBelakang(int data) { ➔ prosedur untuk mengubah data node di belakang list

- if (!isEmpty()) { ➔ jika list tidak kosong
- tail->data = data; ➔ mengubah data tail menjadi data baru
- } else { ➔ Jika list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- void clearList() { ➔ prosedur untuk menghapus semua node dalam list
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- while (bantu != NULL) { ➔ selama bantu bukan NULL (selama ada node dalam list)
- Node* hapus = bantu; ➔ membuat pointer hapus yang menunjuk ke node yang ditunjuk oleh bantu
- bantu = bantu->next; ➔ menggeser bantu ke node berikutnya
- delete hapus; ➔ menghapus node yang ditunjuk oleh hapus
- head = tail = NULL; ➔ Mengatur head dan tail menjadi NULL
- cout << "List berhasil terhapus!" << endl; ➔ cout menampilkan statement List berhasil terhapus!
- void tampil() { ➔ prosedur untuk menampilkan semua node dalam list
- if (!isEmpty()) { ➔ jika list tidak kosong
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- while (bantu != NULL) { ➔ Selama bantu tidak sama dengan NULL (selama ada node dalam list)
- cout << bantu->data << " "; ➔ menampilkan data dari node yang ditunjuk oleh bantu
- bantu = bantu->next; ➔ menggeser bantu ke node berikutnya
- cout << endl; ➔ membuat baris baru setelah menampilkan semua node
- } else { ➔ jika list kosong

- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- int main() { ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- init(); ➔ menginisialisasi list
- insertDepan(3); tampil(); ➔ memasukkan 3 di depan list dan menampilkan list
- insertBelakang(5); tampil(); ➔ memasukkan 5 di belakang list dan menampilkan list
- insertDepan(2); tampil(); ➔ memasukkan 2 di depan list dan menampilkan list
- insertDepan(1); tampil(); ➔ memasukkan 1 di depan list dan menampilkan list
- hapusDepan(); tampil(); ➔ menghapus node depan dan menampilkan list
- hapusBelakang(); tampil(); ➔ menghapus node belakang dan menampilkan list
- insertTengah(7, 2); tampil(); ➔ memasukkan 7 di posisi 2 dan menampilkan list
- hapusTengah(2); tampil(); ➔ menghapus node di posisi 2 dan menampilkan list
- ubahDepan(1); tampil(); ➔ Mengubah node depan menjadi 1 dan menampilkan list
- ubahBelakang(8); tampil(); ➔ Mengubah node belakang menjadi 8 dan menampilkan list
- ubahTengah(11, 2); tampil(); ➔ Mengubah node di posisi 2 menjadi 11 dan menampilkan list
- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

Guided 2 : Linked List Circular

```
#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()) {
```

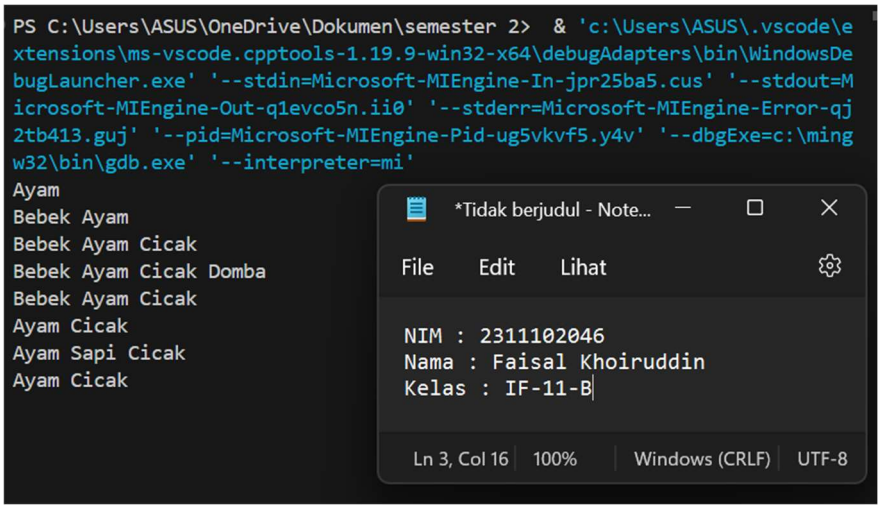
```
        hapus = head;
        tail = head;
        if (hapus->next == head) {
            head = NULL;
            tail = NULL;
            delete hapus;
        } else {
            while (hapus->next != head) {
                hapus = hapus->next;
            }
            while (tail->next != 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:

Program tersebut merupakan contoh dari single linked list. Program tersebut yaitu program single linked list circular. Program tersebut menampilkan output dari menu pilihan pengguna. Output yang ditampilkan pada program tersebut berupa operasi dari void insertDepan sampai void hapusTengah yang ditampilkan menggunakan void tampil.

- #include <iostream> ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- using namespace std; ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam

namespace std

- struct Node { ➔ mendefinisikan struct node
- string data; ➔ deklarasi variabel data dengan nama tipe data string
- Node *next; ➔ pointer next untuk menunjuk ke node berikutnya
- Node *head, *tail, *baru, *bantu, *hapus; ➔ deklarasi variabel pointer
- void init() { ➔ prosedur init untuk menginisialisasi list
- head = NULL; ➔ mengatur head menjadi null
- tail = head; ➔ mengatur tail menjadi head
- int isEmpty() { ➔ fungsi untuk mengecek apakah list masih kosong
- return head == NULL; ➔ mengembalikan true jika head adalah null
- void buatNode(string data) { ➔ prosedur untuk membuat node baru dengan parameter string data
- baru = new Node; ➔ membuat node baru
- baru->data = data; ➔ mengatur data pada node baru
- baru->next = NULL; ➔ mengatur next pada node baru menjadi null
- int hitungList() { ➔ fungsi untuk menghitung jumlah node dalam linked list
- bantu = head; ➔ mengatur bantu samadengan head
- int jumlah = 0; ➔ mengatur jumlah sama dengan 0
- while (bantu != NULL) { ➔ perulangan whileselama bantu tidak sama dengan null
- jumlah++; ➔ jumlah++(increment)
- bantu = bantu->next; ➔ mengatur bantu sama dengan next dari bantu

- return jumlah; ➔ mengembalikan jumlah
- void insertDepan(string data) { ➔ prosedur untuk menambahkan node di depan dengan parameter string data
- buatNode(data); ➔ membuat node baru
- if (isEmpty()) { ➔ percabangan if jika list kosong
- head = baru; ➔ mengatur head sama dengan node baru
- tail = head; ➔ mengatur tail sama dengan head
- baru->next = head; ➔ mnegatur next pada node baru sama dengan head
- } else { ➔ jika linked list tidak kosong
- while (tail->next != head) { ➔ selama next dari tail bukan head
- tail = tail->next; ➔ mengatur tail sama dengan next dari tail
- baru->next = head; ➔ mengatur next pada node baru sama dengan head
- head = baru; ➔ mengatur head sama dengan node baru
- tail->next = head; ➔ mengatur next pada tail sama dengan node baru
- void insertBelakang(string data) { ➔ prosedur untuk menambahkan node di belakang dengan parameter string data
- buatNode(data); ➔ membuat node baru
- if (isEmpty()) { ➔ percabangan if jika list kosong
- head = baru; ➔ mengatur head sama dengan node baru
- tail = head; ➔ mengatur tail sama dengan head
- baru->next = head; ➔ mengatur next pada node baru sama dengan head
- } else { ➔ jika linked list tidak kosong
- while (tail->next != head) { ➔ perulangan while selama next dari tail buka head
- tail = tail->next; ➔ mengatur tail sama dengan next dari tail

- tail->next = baru; ➔ mengatur next pada tail sama dengan node baru
- baru->next = head; ➔ mengatur next pada node baru sama dengan head
- void insertTengah(string data, int posisi) { ➔ prosedur untuk menambahkan node pada posisi tertentu dengan parameter string data, int posisi
- if (isEmpty()) { ➔ percabangan if jika linked list kosong
- head = baru; ➔ mengatur head sama dengan node baru
- tail = head; ➔ mengatur tail sama dengan head
- baru->next = head; ➔ mengatur next padda node baru sama dengan head
- } else { ➔ jika linked list tidak kosong
- baru->data = data; ➔ mengatur data pada node baru
- int nomor = 1; ➔ mengatur variabel nomor menjadi 1
- bantu = head; ➔ mengatur bantu sama dengan head
- while (nomor < posisi - 1) { ➔ perlangan while selama nomor kurang dari posisi - 1
- bantu = bantu->next; ➔ mengatur bantu sama dengan next dari bantu
- nomor++; ➔ nomor++(increment)
- baru->next = bantu->next; ➔ mengatur next pada node baru sama dengan next dari bantu
- bantu->next = baru; ➔
- void hapusDepan() { ➔ prosedur untuk menghapus node di depan
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- hapus = head; ➔ mengatur hapus sama dengan head
- tail = head; ➔ mengatur tail sama dengan head
- if (hapus->next == head) { ➔ percabangan if jika next dari

hapus sama dengan head

- head = NULL; ➔ mengatur head menjadi null
- tail = NULL; ➔ mengatur tail menjadi null
- delete hapus; ➔ menghapus node hapus
- } else { ➔ jika next dari hapus bukan head
- while (tail->next != hapus) { ➔ perulangan while selama next dari tail bukan hapus
- tail = tail->next; ➔ mengatur tail sama dengan next dari tail
- head = head->next; ➔ mengatur head sama dengan next dari head
- tail->next = head; ➔ mengatur next pada tail sama dengan head
- hapus->next = NULL; ➔ mengatur next pada hapus menjadi null
- delete hapus; ➔ menghapus node hapus
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampillkan pernyataan List masih kosong!
- void hapusBelakang() { ➔ prosedur untuk menghapus node di belakang
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- hapus = head; ➔ mengatur hapus sama dengan head
- tail = head; ➔ mengatur tail sama dengan head
- if (hapus->next == head) { ➔ percabangan if jika next dari hapus sama dengan head
- head = NULL; ➔ mengatur head menjadi null
- tail = NULL; ➔ mengatur tail menjadi nul
- delete hapus; ➔ menghapus node hapus
- } else { ➔ jika next dari hapus bukan head
- while (hapus->next != head) { ➔ perulangan while selama next

dari hapus bukan head

- hapus = hapus->next; ➔ mengatur hapus sama dengan next dari hapus
- while (tail->next != hapus) { ➔ perulangan while selama next dari tail bukan hapus
- tail = tail->next; ➔ mengatur tail sama dengan next dari tail
- tail->next = head; ➔ mengatur next pada tail sama dengan head
- hapus->next = NULL; ➔ mengatur next pada tail sama dengan head
- delete hapus; ➔ menghapus node hapus
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- void hapusTengah(int posisi) { ➔ prosedur untuk menghapus node di posisi tertentu
- if (!isEmpty()) { ➔ percabangan jika linked list tidak kosong
- int nomor = 1; ➔ mengatur nomor menjadi 1
- bantu = head; ➔ mengatur bantu sama dengan head
- while (nomor < posisi - 1) { ➔ perulangan whileselama nomor kurang dari posisi - 1
- bantu = bantu->next; ➔ mengatur bantu sama dengan next dari bantu
- nomor++; ➔ nomor++(increment)
- hapus = bantu->next; ➔ mengatur hapus sama dengan next dari bantu
- bantu->next = hapus->next; ➔ mengatur next pada bantu sama dengan next dari hapus
- delete hapus; ➔ menghapus node hapus
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan

statement List masih kosong!

- void clearList() { ➔ prosedur untuk menghapus semua node dalam linked list
- if (head != NULL) { ➔ percabangan if jika head tidak sama dengan null
- hapus = head->next; ➔ mengatur hapus sama dengan next dari head
- while (hapus != head) { ➔ perulangan while selama hapus tidak sama dengan head
- bantu = hapus->next; ➔ mengatur bantu sama dengan next dari hapus
- delete hapus; ➔ menghapus node hapus
- hapus = bantu; ➔ mengatur hapus sama dengan bantu
- delete head; ➔ menghapus node head
- head = NULL; ➔ mengatur head menjadi null
- cout << "List berhasil terhapus!" << endl; ➔ cout menampilkan statement List berhasil terhapus!
- void tampil() { ➔ prosedur untuk menampilkan semua node dalam linked list
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- tail = head; ➔ mengatur tail sama dengan head
- do { ➔ lakukan
- cout << tail->data << " "; ➔ cout menampilkan data pada node yang ditunjuk oleh tail
- tail = tail->next; ➔ mengatur tail sama dengan next dari tail
- } while (tail != head); ➔ perulangan while selama tail tidak sama dengan head
- cout << endl; ➔ cout untuk mencetak baris baru
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan

statement List masih kosong!

- int main() { ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- init();➔ menginisialisasi linked list
- insertDepan("Ayam"); ➔ menambahkan node dengan data ayam di depan
- tampil(); ➔ menampilkan semua node dalam linked list
- insertDepan("Bebek"); ➔ menambahkan node dengan data bebek di depan
- tampil(); ➔ menampilkan semua node dalam linked list
- insertBelakang("Cicak"); ➔ menambahkan node dengan data cicak di belakang
- tampil(); ➔ menampilkan semua node dalam linked list
- insertBelakang("Domba"); ➔ menambahkan node dengan data domba di belakang
- tampil(); ➔ menampilkan semua node dalam linked list
- hapusBelakang(); ➔ menghapus node di belakang
- tampil(); ➔ menampilkan semua node dalam linked list
- hapusDepan(); ➔ menghapus node di depan
- tampil(); ➔ menampilkan semua node dalam linked list
- insertTengah("Sapi", 2); ➔ menambahkan node dengan data sapi di posisi 2
- tampil(); ➔ menampilkan semua node dalam linked list
- hapusTengah(2); ➔ menghapus node di posisi 2
- tampil(); ➔ menampilkan semua node dalam linked list
- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

C. Unguided/Tugas

1. Unguided 1 : Soal mengenai Single Linked Lis

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

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 :
```

   - Tampil

```
-Tambah Depan

Masukkan Nama :
Masukkan NIM :

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**

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

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 terkahir menjadi berikut:

      Lucy 23300101

   g) Hapus data awal

h) Ubah data awal menjadi berikut:

   Bagas 2330002

i) Hapus data akhir

j) Tampilkan seluruh data

Source Code

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

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

Node *head;
Node *tail;

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

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

void insertDepan(string nama, string nim)
{
    Node *baru = new Node;
    baru->nama = nama;
    baru->nim = nim;
    baru->next = NULL;
    if (isEmpty()) {
        head = tail = baru;
    } else {
        baru->next = head;
        head = baru;
    }

}
```

```
void insertBelakang(string nama, string
nim) {
    Node *baru = new Node;
    baru->nama = nama;
    baru->nim = nim;
    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(string nama, string nim,
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->nama = nama;
        baru->nim = nim;
        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;
        string namaHapus = hapus->nama;
        if (head->next != NULL) {
            head = head->next;
        } else {
            head = tail = NULL;
        }
        delete hapus;
        cout << "\nData (" << namaHapus <<
") berhasil dihapus." << endl;
    } else {
        cout << "List kosong!" << endl;
    }
}

void hapusBelakangNIM(string nim) {
    if (!isEmpty()) {
        Node *hapus = tail;
        string namaHapus = hapus->nama;
        if (head != tail) {
            Node *bantu = head;
            while (bantu->next != tail) {
                bantu = bantu->next;
            }
            tail = bantu;
            tail->next = NULL;
        } else {
            head = tail = NULL;
        }
        delete hapus;
        cout << "\nData (" << namaHapus <<
") berhasil dihapus." << endl;
    } else {
        cout << "List kosong!" << endl;
    }
}

void hapusTengahNIM(string nim) {
    if (!isEmpty()) {
```

```
        Node *bantu = head;
        Node *sebelum = NULL;
        while (bantu != NULL && bantu->nim
!= nim) {
            sebelum = bantu;
            bantu = bantu->next;
        }
        if (bantu == NULL) {
            cout << "Data dengan NIM " <<
nim << " tidak ditemukan." << endl;
        } else {
            if (sebelum != NULL) {
                sebelum->next = bantu-
>next;
            } else {
                head = bantu->next;
            }
            string namaHapus = bantu-
>nama;
            delete bantu;
            cout << "\nData (" <<
namaHapus << ") berhasil dihapus." <<
endl;
        }
    } else {
        cout << "List kosong!" << endl;
    }
}
void ubahDepan(string nama, string nim) {
    if (!isEmpty()) {
        string namaLama = head->nama;
        head->nama = nama;
        head->nim = nim;
        cout << "\nData (" << namaLama <<
") telah diganti dengan data (" << nama <<
")." << endl;
    } else {
        cout << "List masih kosong!" <<
endl;
    }
}

void ubahTengah(string nama, string nim,
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++;
            }
            string namaLama = bantu->nama;
            bantu->nama = nama;
            bantu->nim = nim;
            cout << "\nData (" << namaLama
<< ") telah diganti dengan data (" << nama
<< ")." << endl;
        }
    } else {
        cout << "List masih kosong!" <<
endl;
    }
}

void ubahBelakang(string nama, string nim)
{
    if (!isEmpty()) {
        string namaLama = tail->nama;
        tail->nama = nama;
        tail->nim = nim;
        cout << "\nData (" << namaLama <<
") telah diganti dengan data (" << nama <<
")." << endl;
    } 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() {
    if (!isEmpty()) {
        Node* bantu = head;
        cout << "-------------------------
----------------------\n";
        cout << "| No |
Nama              |
NIM                 |\n";
        cout << "-------------------------
----------------------\n";
        int no = 1;
        while (bantu != NULL) {
            cout << "| " << left <<
setw(2) << no << " | " << left << setw(17)
<< bantu->nama << " | " << left <<
setw(19) << bantu->nim << " |\n";
            bantu = bantu->next;
            no++;
        }
        cout << "-------------------------
----------------------\n";
    } else {
        cout << "List masih kosong!" <<
endl;
    }
}

int main() {
    init();
    int pilihan;
    string nama, nim, namaHapus;
    int posisi;
    do {
        cout <<
"========================================
==" << endl;
        cout << "| PROGRAM SINGLE LINKED
LIST NON-CIRCULAR |" << endl;
```

```
        cout <<
"========================================
==" << endl;
        cout << "| No. | Pilihan
Menu                      |" << 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;
        switch (pilihan) {
            case 1:
                cout << "Masukkan Nama: ";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;
                insertDepan(nama, nim);
```

```
                cout << "-----------------
----------------\n";
                cout << "|
Nama              | NIM       |\n";
                cout << "-----------------
----------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " |\n";
                cout << "-----------------
----------------\n";
                cout << "\nData telah
ditambahkan.\n" << endl;
                cout <<
"================================\n" <<
endl;
                break;
            case 2:
                cout << "Masukkan Nama: ";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;
                insertBelakang(nama, nim);
                cout << "-----------------
----------------\n";
                cout << "|
Nama              | NIM       |\n";
                cout << "-----------------
----------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " |\n";
                cout << "-----------------
----------------\n";
                cout << "\nData telah
ditambahkan." << endl;
                cout <<
"================================\n" <<
endl;
                break;

            case 3:
                cout << "Masukkan Nama: ";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;
```

```
                cout << "Masukkan Posisi:
";
                cin >> posisi;
                insertTengah(nama, nim,
posisi);
                cout << "----------------
------------------------\n";
                cout << "|
Nama              | NIM       | Posisi
|\n";
                cout << "----------------
------------------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " | " << setw(6) << left <<
posisi << " |\n";
                cout << "----------------
------------------------\n";
                cout <<
"========================================
=\n" << endl;
                break;
             case 4:
                cout << "Masukkan Nama: ";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;
                ubahDepan(nama, nim);
                cout << "----------------
---------------\n";
                cout << "|
Nama              | NIM       |\n";
                cout << "----------------
---------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " |\n";
                cout << "----------------
---------------\n";
                cout <<
"================================\n" <<
endl;
                break;
            case 5:
                cout << "Masukkan Nama: ";
                cin >> nama;
```

```
                cout << "Masukkan NIM: ";
                cin >> nim;
                ubahBelakang(nama, nim);
                cout << "-----------------
----------------\n";
                cout << "|
Nama              | NIM       |\n";
                cout << "-----------------
----------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " |\n";
                cout << "-----------------
----------------\n";
                cout <<
"================================\n" <<
endl;
                break;
            case 6:
                 cout << "Masukkan Nama:
";
                cin >> nama;
                cout << "Masukkan NIM: ";
                cin >> nim;
                cout << "Masukkan Posisi:
";
                cin >> posisi;
                ubahTengah(nama, nim,
posisi);
                cout << "-----------------
------------------------\n";
                cout << "|
Nama              | NIM       | Posisi
|\n";
                cout << "-----------------
------------------------\n";
                cout << "| " << setw(17)
<< left << nama << " | " << setw(10) <<
left << nim << " | " << setw(6) << left <<
posisi << " |\n";
                cout << "-----------------
------------------------\n";
                cout <<
"========================================
=\n" << endl;
                break;
```

```
            case 7:
                if (!isEmpty()) {
                    nama = head->nama; //
Simpan nama sebelum node dihapus
                    nim = head->nim; //
Simpan nim sebelum node dihapus
                    hapusDepan();
                    cout << "----------
--------------------\n";
                    cout << "|
Nama               | NIM       |\n";
                    cout << "----------
--------------------\n";
                    cout << "| " <<
setw(17) << left << nama << " | " <<
setw(10) << left << nim << " |\n";
                    cout << "----------
--------------------\n";
                } else {
                    cout << "List masih
kosong!" << endl;
                }
                break;
            case 8:
                    cout << "Masukkan NIM:
";
                    cin >> nim;
                    if (!isEmpty()) {
                        Node *bantu =
head;
                        while (bantu !=
NULL && bantu->nim != nim) {
                            bantu = bantu-
>next;
                        }
                        if (bantu == NULL)
{
                            cout << "Data
dengan NIM " << nim << " tidak ditemukan."
<< endl;
                        } else {
                            nama = bantu-
>nama; // Simpan nama sebelum node dihapus
                            hapusBelakangN
IM(nim);
```

```
                                cout << "-----
-----------------------------\n";
                                cout << "|
Nama               | NIM       |\n";
                                cout << "-----
-----------------------------\n";
                                cout << "| "
<< setw(17) << left << nama << " | " <<
setw(10) << left << nim << " |\n";
                                cout << "-----
-----------------------------\n";
                            }
                        } else {
                            cout << "List
masih kosong!" << endl;
                        }
                        break;
                case 9:
                    cout << "Masukkan NIM: ";
                    cin >> nim;
                    if (!isEmpty()) {
                        Node *bantu = head;
                        while (bantu != NULL
&& bantu->nim != nim) {
                            bantu = bantu-
>next;
                        }
                        if (bantu == NULL) {
                            cout << "Data
dengan NIM " << nim << " tidak ditemukan."
<< endl;
                        } else {
                            nama = bantu-
>nama; // Simpan nama sebelum node dihapus
                            hapusTengahNIM(nim
);
                            cout << "---------
-------------------------\n";
                            cout << "|
Nama               | NIM       |\n";
                            cout << "---------
-------------------------\n";
                            cout << "| " <<
setw(17) << left << nama << " | " <<
setw(10) << left << nim << " |\n";
                            cout << "---------
```

```
------------------------\n";
                        cout << "Data
berhasil dihapus." << endl;
                    }
                } else {
                    cout << "List masih
kosong!" << endl;
                }
                break;
            case 10:
                clearList();
                break;
            case 11:
                tampil();
                break;
            case 0:
                cout << "Terima kasih
telah menggunakan program ini." << endl;
                break;
            default:
                cout << "Pilihan tidak
valid." << endl;
        }
    } while (pilihan != 0);
    return 0;
}
```

Screenshots Output

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>  & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.19.9-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-figmwhbv.rr1' '--stdout=Microsoft-MIEngine-Out-slfkpfo3.jpa' '--stderr=Microsoft-MIEngine-Error-r4o31n04.5ls' '--pid=Microsoft-MIEngine-Pid-yybjac1i.2qp' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
========================================
| No. | Pilihan Menu                     |
========================================
| 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
Masukkan Nama: Jawad
Masukkan NIM: 2300001
---------------------------------
| Nama              | NIM       |
---------------------------------
| Jawad             | 2300001     |
---------------------------------
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 100% Windows (CRLF) UTF-8

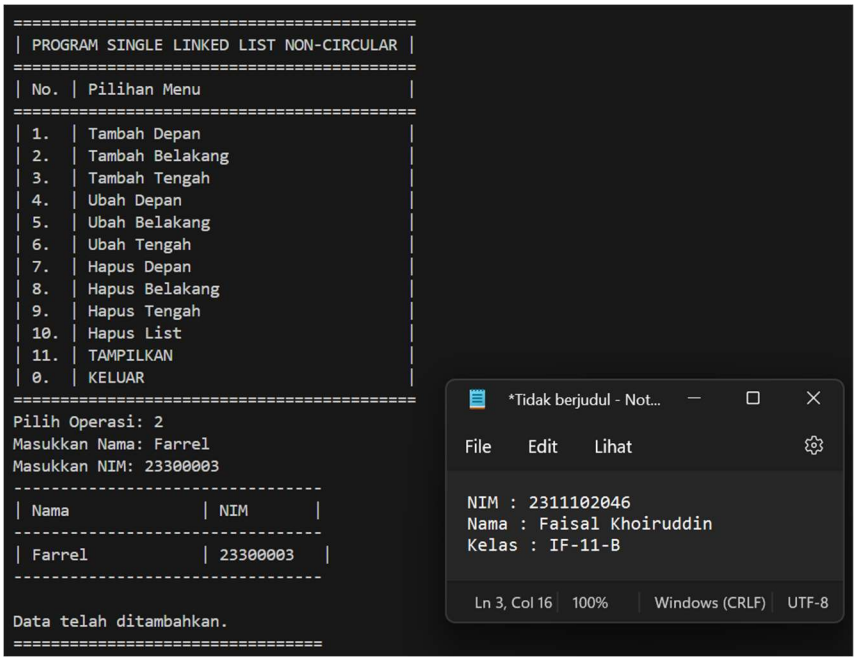
```
================================

========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
========================================
| No. | Pilihan Menu                     |
========================================
| 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
Masukkan Nama: Faisal
Masukkan NIM: 2311102046
---------------------------------
| Nama              | NIM       |
---------------------------------
| Faisal            | 2311102046 |
---------------------------------

Data telah ditambahkan.
================================
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

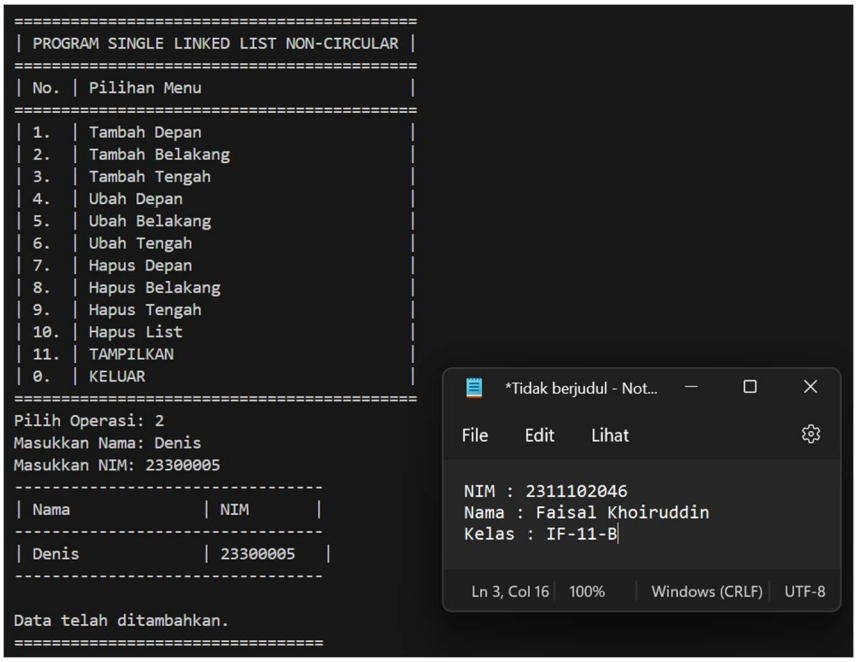
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Farrel
Masukkan NIM: 23300003

| Nama | NIM |

| Farrel | 23300003 |

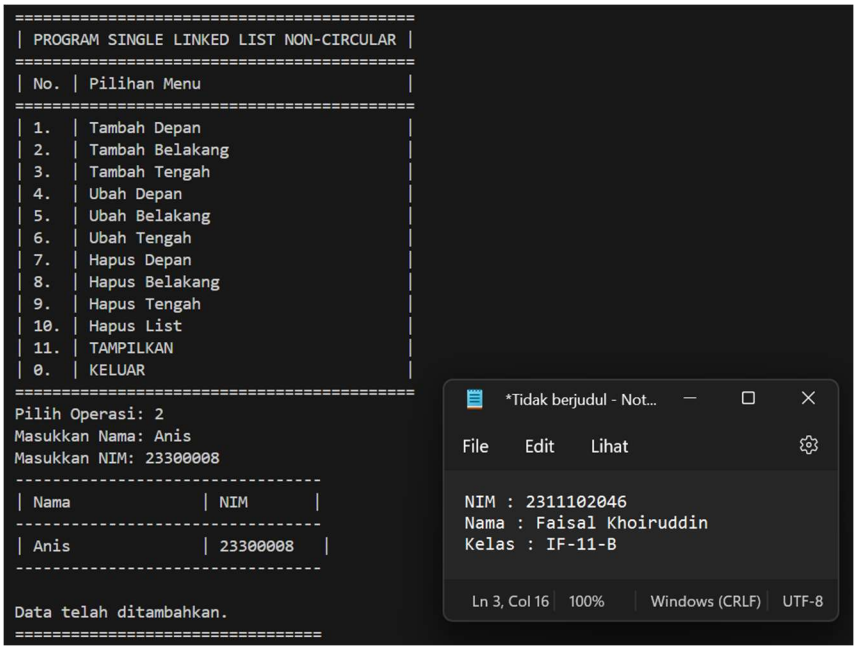
Data telah ditambahkan.
==========
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Denis
Masukkan NIM: 23300005

| Nama | NIM |

| Denis | 23300005 |

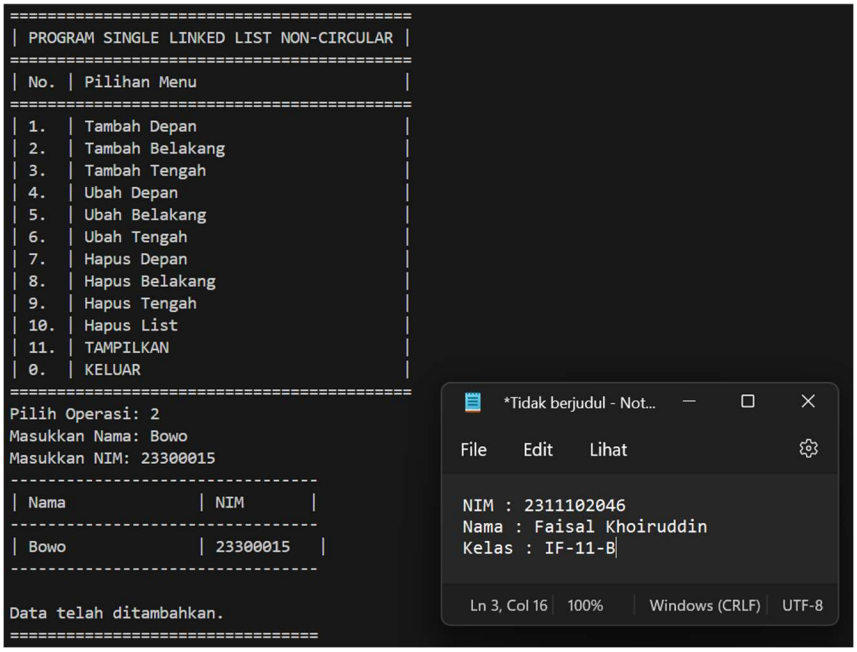
Data telah ditambahkan.
==========
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Anis
Masukkan NIM: 23300008

| Nama | NIM |

| Anis | 23300008 |

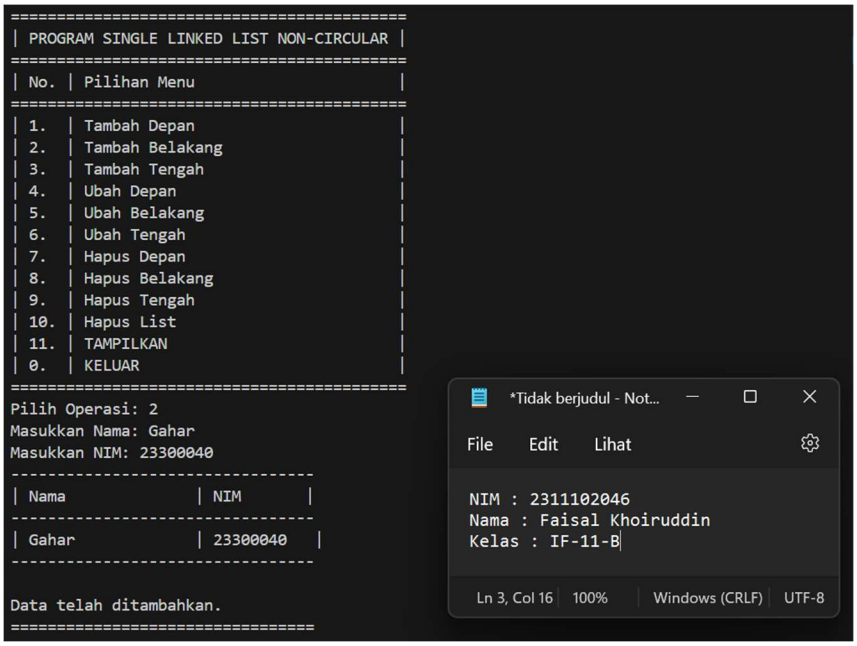
Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Bowo
Masukkan NIM: 23300015

| Nama | NIM |

| Bowo | 23300015 |

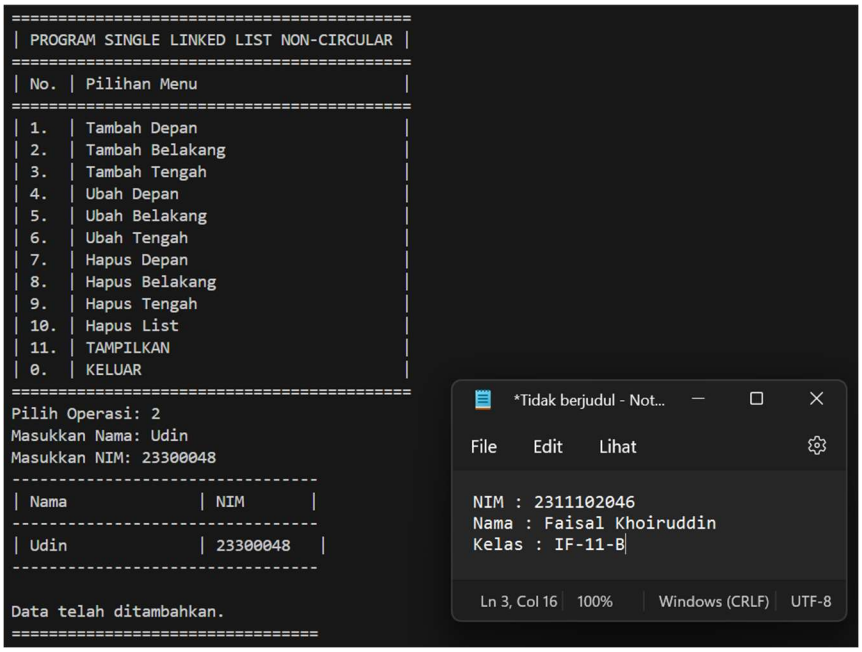
Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Gahar
Masukkan NIM: 23300040

| Nama | NIM |

| Gahar | 23300040 |

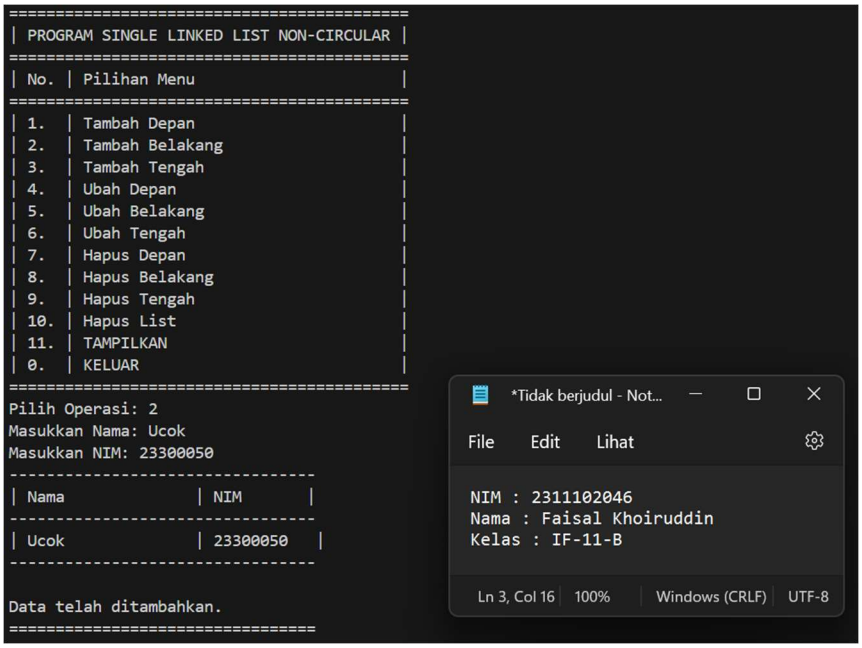
Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Udin
Masukkan NIM: 23300048

| Nama | NIM |

| Udin | 23300048 |

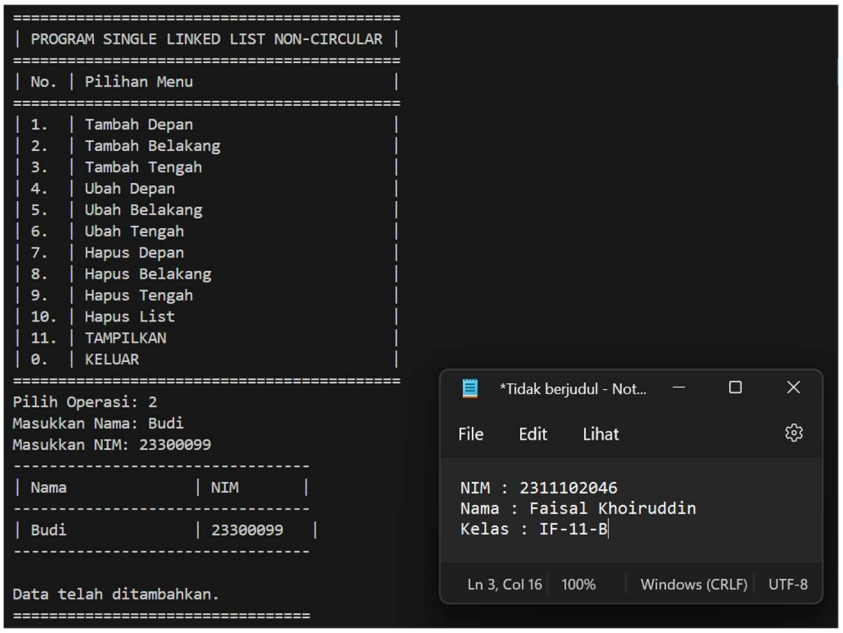
Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Ucok
Masukkan NIM: 23300050

| Nama | NIM |

| Ucok | 23300050 |

Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


==
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==
| No. | Pilihan Menu |
==
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
Masukkan Nama: Budi
Masukkan NIM: 23300099

| Nama | NIM |

| Budi | 23300099 |

Data telah ditambahkan.
================================
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

```
==========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==========================================
| No. | Pilihan Menu                       |
==========================================
| 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
------------------------------------------------
| No | Nama              | NIM               |
------------------------------------------------
| 1  | Jawad             | 2300001           |
| 2  | Faisal            | 2311102046        |
| 3  | Farrel            | 23300003          |
| 4  | Denis             | 23300005          |
| 5  | Anis              | 23300008          |
| 6  | Bowo              | 23300015          |
| 7  | Gahar             | 23300040          |
| 8  | Udin              | 23300048          |
| 9  | Ucok              | 23300050          |
| 10 | Budi              | 23300099          |
------------------------------------------------
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 100% Windows (CRLF) UTF-8

```
==========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
==========================================
| No. | Pilihan Menu                       |
==========================================
| 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
Masukkan Nama: Wati
Masukkan NIM: 2330004
Masukkan Posisi: 4
------------------------------------------
| Nama              | NIM       | Posisi |
------------------------------------------
| Wati              | 2330004   | 4      |
------------------------------------------
==========================================
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 100% Windows (CRLF) UTF-8

```
========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
========================================
| No. | Pilihan Menu                     |
========================================
| 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
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 | 100% | Windows (CRLF) | UTF-8

```
-----------------------------------------------
| No | Nama              | NIM               |
-----------------------------------------------
| 1  | Jawad             | 2300001           |
| 2  | Faisal            | 2311102046        |
| 3  | Farrel            | 23300003          |
| 4  | Wati              | 2330004           |
| 5  | Denis             | 23300005          |
| 6  | Anis              | 23300008          |
| 7  | Bowo              | 23300015          |
| 8  | Gahar             | 23300040          |
| 9  | Udin              | 23300048          |
| 10 | Ucok              | 23300050          |
| 11 | Budi              | 23300099          |
-----------------------------------------------
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 | 100% | Windows (CRLF) | UTF-8

```
========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
========================================
| No. | Pilihan Menu                     |
========================================
| 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
Masukkan NIM: 23300005

Data (Denis) berhasil dihapus.
---------------------------------
| Nama              | NIM       |
---------------------------------
| Denis             | 23300005  |
---------------------------------
Data berhasil dihapus.
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 | 100% | Windows (CRLF) | UTF-8

```
=========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
=========================================
| No. | Pilihan Menu                      |
=========================================
| 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
Masukkan Nama: Owi
Masukkan NIM: 2330000
---------------------------------
| Nama              | NIM       |
---------------------------------
| Owi               | 2330000    |
---------------------------------

Data telah ditambahkan.
```

*Tidak berjudul - Not...

File Edit Lihat

```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

Ln 3, Col 16 100% Windows (CRLF) UTF-8

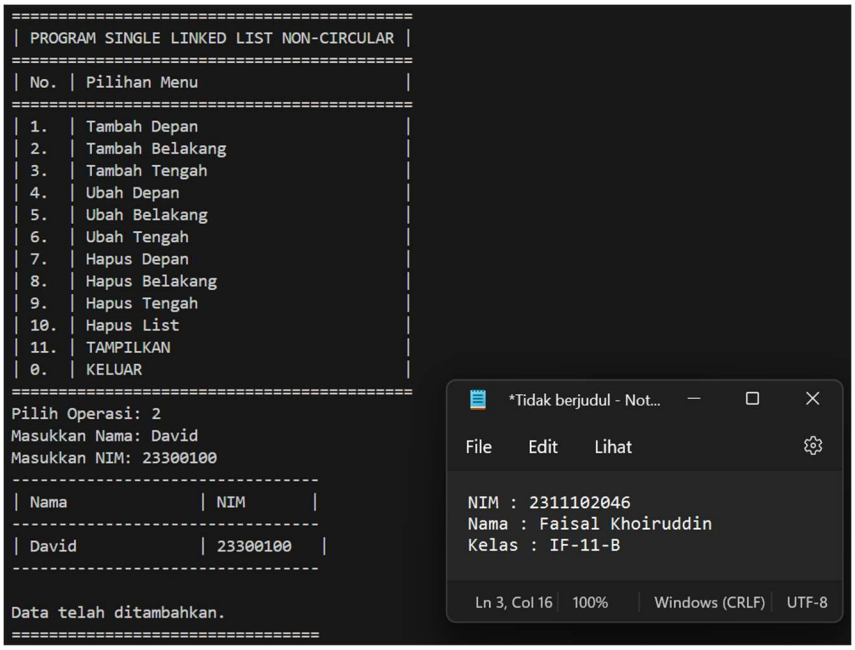
```
=========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
=========================================
| No. | Pilihan Menu                      |
=========================================
| 1.  | Tambah Depan                      |
| 2.  | Tambah Belakang                   |
| 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
-------------------------------------------------
| No | Nama              | NIM                |
-------------------------------------------------
| 1  | Owi               | 2330000            |
| 2  | Jawad             | 2300001            |
| 3  | Faisal            | 2311102046         |
| 4  | Farrel            | 23300003           |
| 5  | Wati              | 2330004            |
| 6  | Anis              | 23300008           |
| 7  | Bowo              | 23300015           |
| 8  | Gahar             | 23300040           |
| 9  | Udin              | 23300048           |
| 10 | Ucok              | 23300050           |
| 11 | Budi              | 23300099           |
```

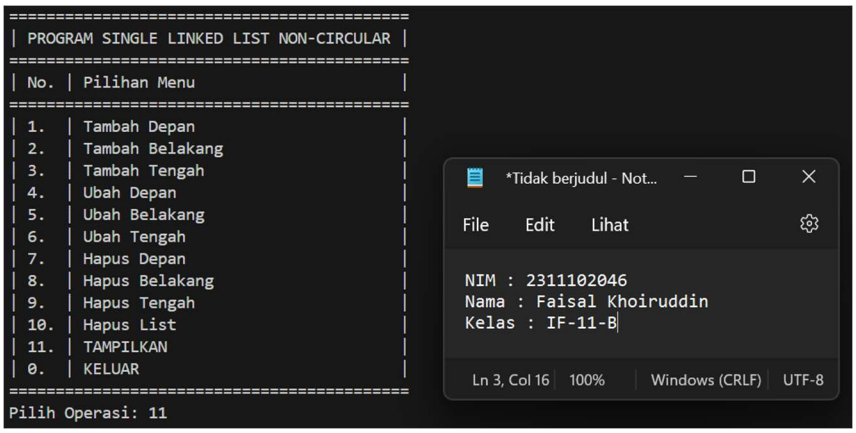
*Tidak berjudul - Not...

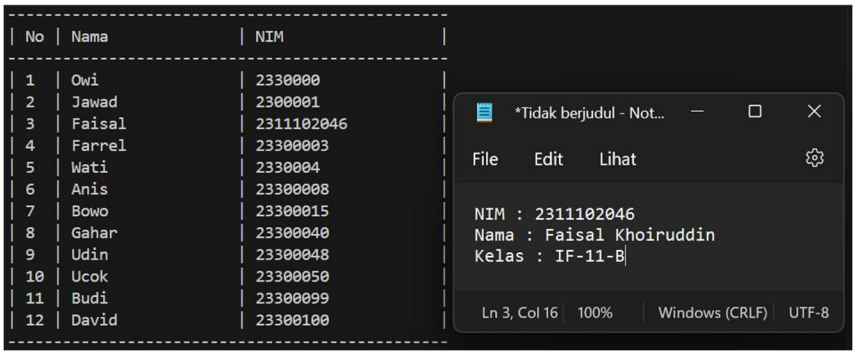
File Edit Lihat

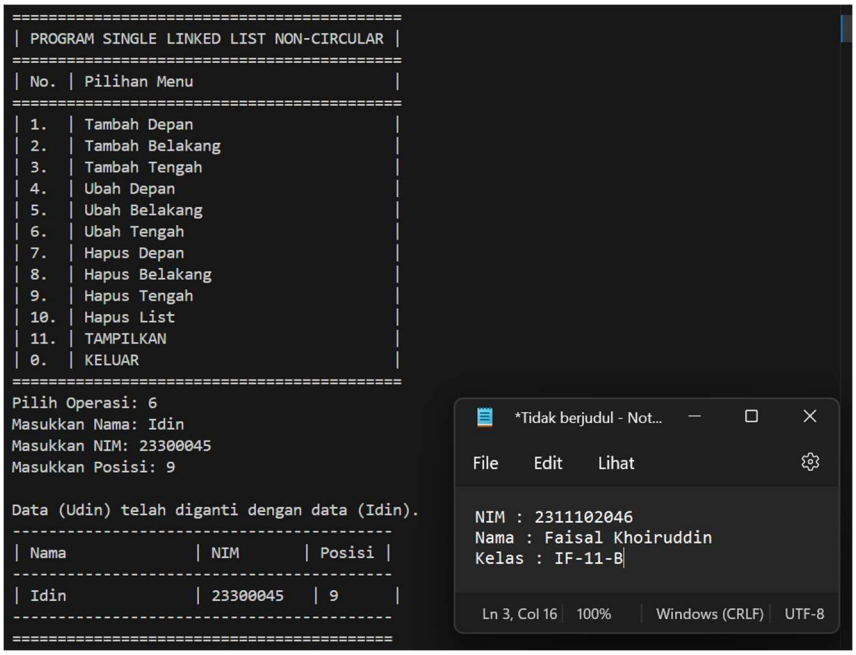
```
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
```

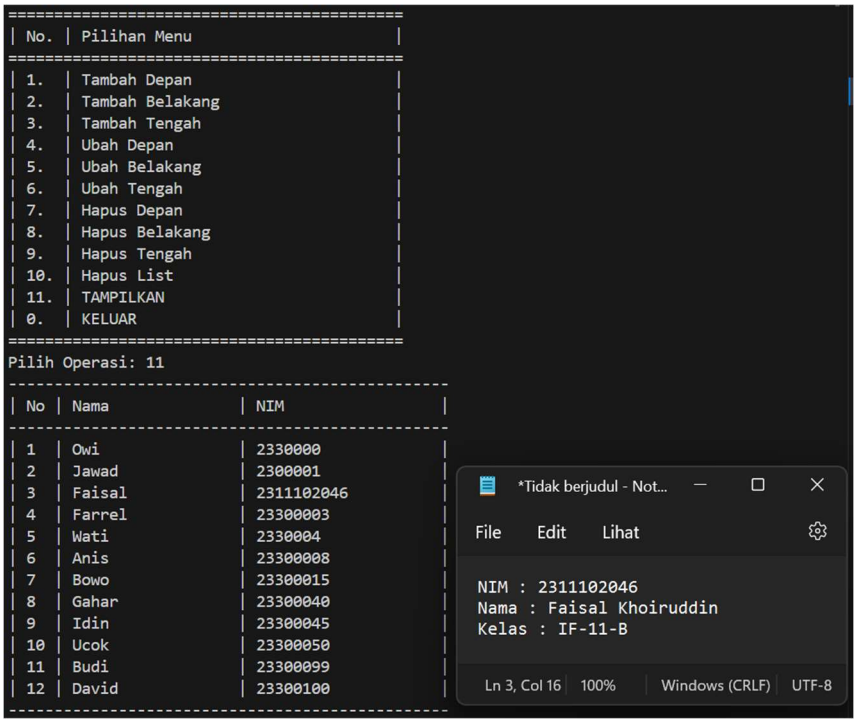
Ln 3, Col 16 100% Windows (CRLF) UTF-8


| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
No.	Pilihan Menu
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
Masukkan Nama: David
Masukkan NIM: 23300100
Nama	NIM
David	23300100
Data telah ditambahkan.
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

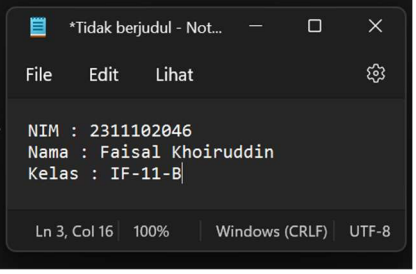

| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
No.	Pilihan Menu
1.	Tambah Depan
2.	Tambah Belakang
3.	Tambah Tengah
4.	Ubah Depan
5.	Ubah Belakang
6.	Ubah Tengah
7.	Hapus Depan
8.	Hapus Belakang
9.	Hapus Tengah
10.	Hapus List
11.	TAMPILKAN
0.	KELUAR
Pilih Operasi: 11
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


No	Nama	NIM
1	Owi	2330000
2	Jawad	2300001
3	Faisal	2311102046
4	Farrel	23300003
5	Wati	2330004
6	Anis	23300008
7	Bowo	23300015
8	Gahar	23300040
9	Udin	23300048
10	Ucok	23300050
11	Budi	23300099
12	David	23300100
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

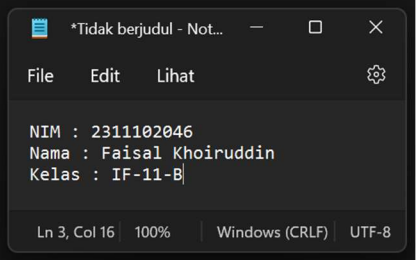

PROGRAM SINGLE LINKED LIST NON-CIRCULAR
No. | Pilihan Menu
1. | Tambah Depan
2. | Tambah Belakang
3. | Tambah Tengah
4. | Ubah Depan
5. | Ubah Belakang
6. | Ubah Tengah
7. | Hapus Depan
8. | Hapus Belakang
9. | Hapus Tengah
10. | Hapus List
11. | TAMPILKAN
0. | KELUAR
Pilih Operasi: 6
Masukkan Nama: Idin
Masukkan NIM: 23300045
Masukkan Posisi: 9
Data (Udin) telah diganti dengan data (Idin).
Nama | NIM | Posisi
Idin | 23300045 | 9
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


No. | Pilihan Menu
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
No | Nama | NIM
1 | Owi | 2330000
2 | Jawad | 2300001
3 | Faisal | 2311102046
4 | Farrel | 23300003
5 | Wati | 2330004
6 | Anis | 23300008
7 | Bowo | 23300015
8 | Gahar | 23300040
9 | Idin | 23300045
10 | Ucok | 23300050
11 | Budi | 23300099
12 | David | 23300100
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

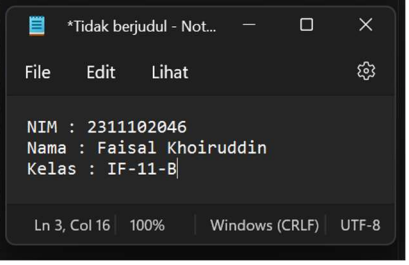
```
=========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
=========================================
| No. | Pilihan Menu                      |
=========================================
| 1.  | Tambah Depan                      |
| 2.  | Tambah Belakang                   |
| 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
Masukkan Nama: Lucy
Masukkan NIM: 23300101

Data (David) telah diganti dengan data (Lucy).
---------------------------------
| Nama              | NIM       |
---------------------------------
| Lucy              | 23300101  |
---------------------------------
=================================
```

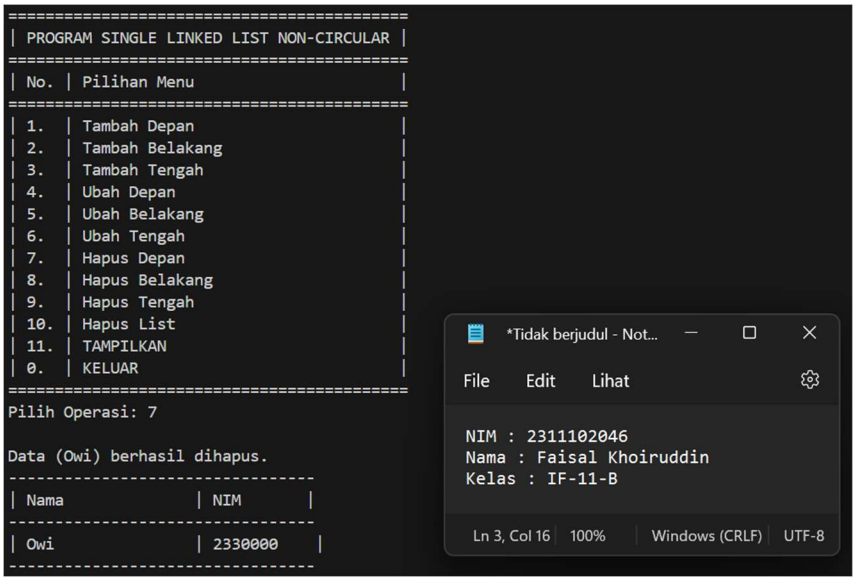


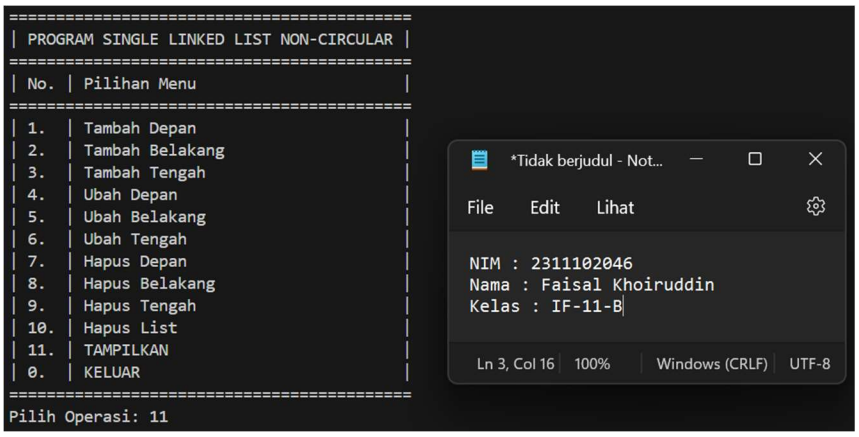

```
=========================================
| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |
=========================================
| No. | Pilihan Menu                      |
=========================================
| 1.  | Tambah Depan                      |
| 2.  | Tambah Belakang                   |
| 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
```

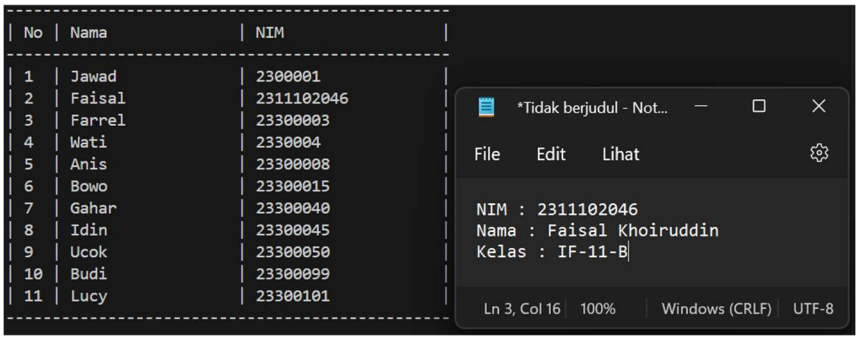


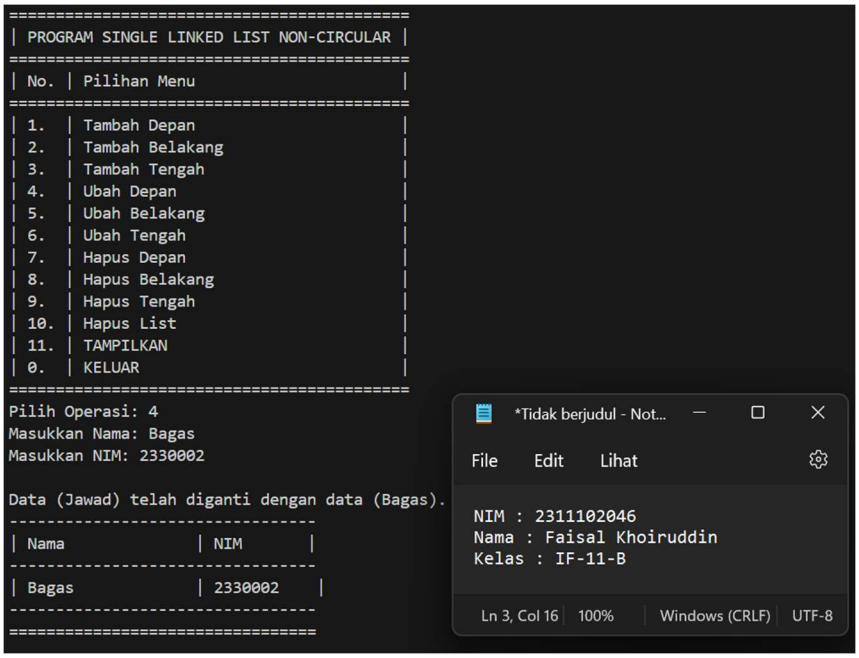

```
-----------------------------------------------
| No | Nama              | NIM               |
-----------------------------------------------
| 1  | Owi               | 2330000           |
| 2  | Jawad             | 2300001           |
| 3  | Faisal            | 2311102046        |
| 4  | Farrel            | 23300003          |
| 5  | Wati              | 2330004           |
| 6  | Anis              | 23300008          |
| 7  | Bowo              | 23300015          |
| 8  | Gahar             | 23300040          |
| 9  | Idin              | 23300045          |
| 10 | Ucok              | 23300050          |
| 11 | Budi              | 23300099          |
| 12 | Lucy              | 23300101          |
-----------------------------------------------
```

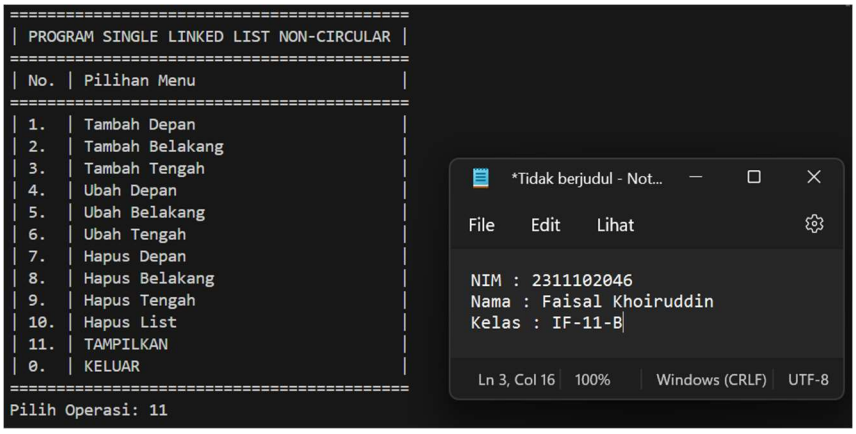


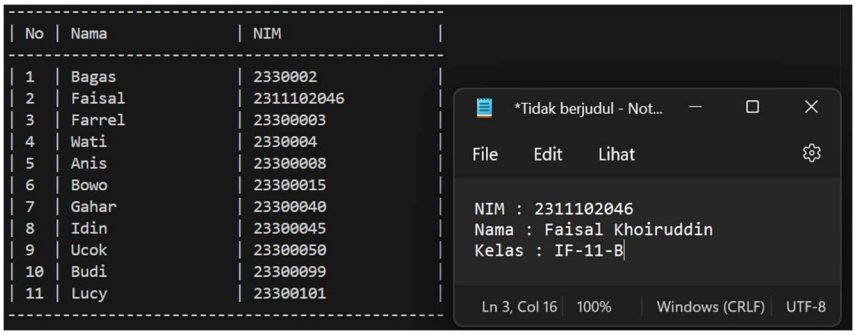

PROGRAM SINGLE LINKED LIST NON-CIRCULAR
No. | Pilihan Menu
1. | Tambah Depan
2. | Tambah Belakang
3. | Tambah Tengah
4. | Ubah Depan
5. | Ubah Belakang
6. | Ubah Tengah
7. | Hapus Depan
8. | Hapus Belakang
9. | Hapus Tengah
10. | Hapus List
11. | TAMPILKAN
0. | KELUAR
Pilih Operasi: 7
Data (Owi) berhasil dihapus.
Nama | NIM
Owi | 2330000
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

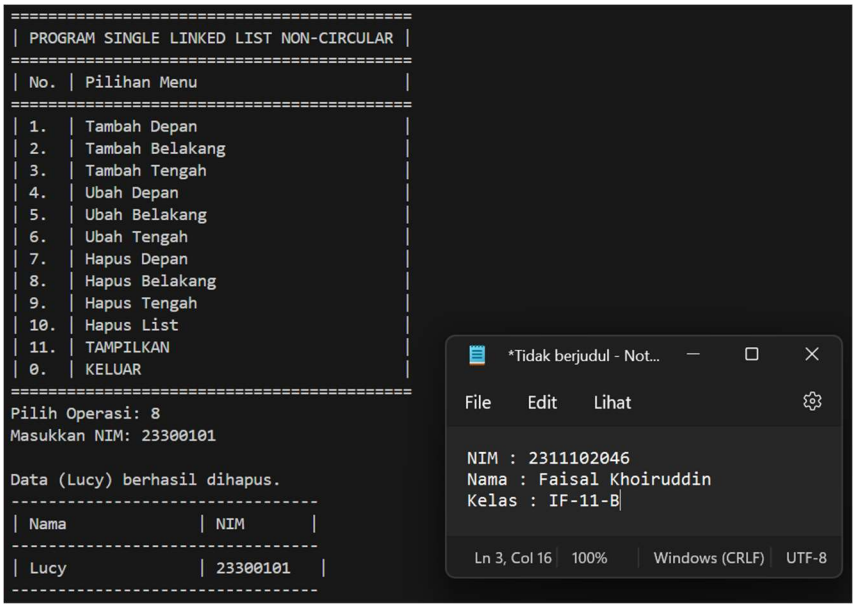

PROGRAM SINGLE LINKED LIST NON-CIRCULAR
No. | Pilihan Menu
1. | Tambah Depan
2. | Tambah Belakang
3. | Tambah Tengah
4. | Ubah Depan
5. | Ubah Belakang
6. | Ubah Tengah
7. | Hapus Depan
8. | Hapus Belakang
9. | Hapus Tengah
10. | Hapus List
11. | TAMPILKAN
0. | KELUAR
Pilih Operasi: 11
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8

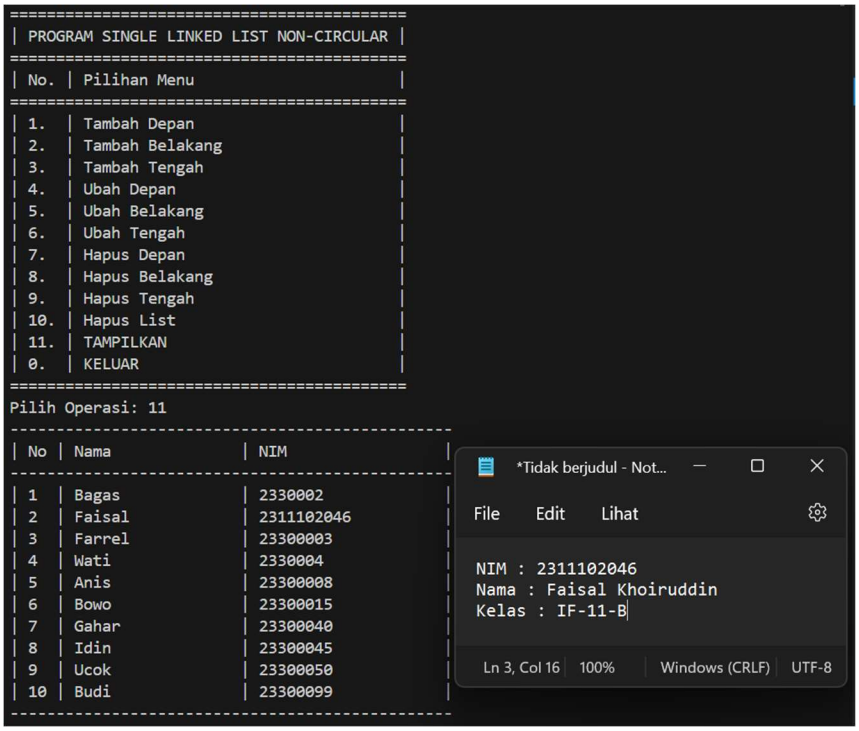

No | Nama | NIM
1 | Jawad | 2300001
2 | Faisal | 2311102046
3 | Farrel | 23300003
4 | Wati | 2330004
5 | Anis | 23300008
6 | Bowo | 23300015
7 | Gahar | 23300040
8 | Idin | 23300045
9 | Ucok | 23300050
10 | Budi | 23300099
11 | Lucy | 23300101
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


PROGRAM SINGLE LINKED LIST NON-CIRCULAR
No. | Pilihan Menu
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
Masukkan Nama: Bagas
Masukkan NIM: 2330002
Data (Jawad) telah diganti dengan data (Bagas).
Nama | NIM
Bagas | 2330002
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


PROGRAM SINGLE LINKED LIST NON-CIRCULAR
No. | Pilihan Menu
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
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8


No | Nama | NIM
1 | Bagas | 2330002
2 | Faisal | 2311102046
3 | Farrel | 23300003
4 | Wati | 2330004
5 | Anis | 23300008
6 | Bowo | 23300015
7 | Gahar | 23300040
8 | Idin | 23300045
9 | Ucok | 23300050
10 | Budi | 23300099
11 | Lucy | 23300101
*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoiruddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8





Deskripsi:

Program tersebut program menu Linked List Non Circular untuk menyimpan Nama dan NIM mahasiswa, dengan menggunakan input dari user. Pengguna diminta memilih operasi dengan memasukkan angka dari menu Tambah Depan sampai menu keluar. Output program tersebut yaitu menampilkan nama

dan nim mahasiswa.

- #include <iostream> ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- #include <iomanip> ➔ standar untuk menentukan beberapa manipulator yang masing-masing mengambil satu argumen
- using namespace std; ➔ digunakan untuk mendeklarasikan/ memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- struct Node { ➔ mendefinisikan struct node
- string nama; ➔ deklarasi variabel nama dengan nama tipe data string
- string nim; ➔ deklarasi variabel nim dengan nama tipe data string
- Node *next; ➔ pointer next untuk menunjuk ke node berikutnya
- Node *head; ➔ pointer head untuk menunjuk ke node pertama dalam list
- Node *tail; ➔ pointer tail untuk menunjuk ke node terakhir dalam list
- void init() { ➔ prosedur init untuk menginisialisasi list
- head = NULL; ➔ mengatur head menjadi null
- tail = NULL; ➔ mengatur tail menjadi null
- bool isEmpty() { ➔ fungsi untuk memeriksa apakah list kosong
- return head == NULL; ➔ mengembaikan true jika head yaitu null atau list kosong
- void insertDepan(string nama, string nim) { ➔ prosedur untuk memasukkan node di depan list
- Node *baru = new Node; ➔ membuat node baru

- baru->nama = nama; ➔ mengatur nama ke node baru
- baru->nim = nim; ➔ mengatur nim ke node baru
- baru->next = NULL; ➔ mengatur next dari node baru menjadi null
- if (isEmpty()) { ➔ percabangan if, jika list kosong
- head = tail = baru; ➔ node baru menjadi head dan tail
- } else { ➔ jika list tidak kosong
- baru->next = head; ➔ node baru menjadi head dan tail
- head = baru; ➔ mengatur head menjadi node baru
- void insertBelakang(string nama, string nim) { ➔ prosedur untuk memasukkan node di belakang list
- Node *baru = new Node; ➔ membuat node baru
- baru->nama = nama; ➔ mengatur nama ke node baru
- baru->nim = nim; ➔ mengatur nama ke node baru
- baru->next = NULL; ➔ mengatur nest dari node baru menjadi null
- if (isEmpty()) { ➔ jika list kosong
- head = tail = baru; ➔ node baru menjadi head dan tail
- } else { ➔ jika list tidak kosong
- tail->next = baru; mengatur next dari tail saat ini menjadi node baru
- tail = baru; ➔ mengatur tail menjadi node baru
- int hitungList() { ➔ prosedur untuk menghitung jumlah node dalam list
- Node *hitung = head; ➔ membuat pointer hitung yang menunjuk head
- int jumlah = 0; ➔ variabel untuk menyimpan jumlah node
- while (hitung != NULL) { ➔ selama hitung bukan sama dengan null (selama ada node dalam list)

- jumlah++; ➔ jumlah++ (increment) selama ada node dalam list
- hitung = hitung->next; ➔ menggeser hitung ke node berikutnya
- return jumlah; ➔ mengembalikan jumlah node dalam list
- insertTengah(string nama, string nim, int posisi) { ➔ prosedur untuk memasukkan node di Tengah list
- if (posisi < 1 || posisi > hitungList()) { ➔ jika posisi < 1 atau posisi lebih dari hitungList (posisi tidak valid)
- cout << "Posisi diluar jangkauan" << endl; ➔ cout menampilkan statement Posisi diluar jangkauan
- } else if (posisi == 1) { ➔ jila posisi sama dengan 1
- cout << "Posisi bukan posisi tengah" << endl; ➔ cout menampilkan statement Posisi bukan posisi tengah
- } else { ➔ jika posisi valid
- Node *baru = new Node(); ➔ membuat node baru
- baru->nama = nama; ➔ mengatur nama ke node baru
- baru->nim = nim; ➔ mengatur nim ke node baru
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke head
- int nomor = 1; ➔ nomor sama dengan 1
- while (nomor < posisi - 1) { ➔ menggeser bantu ke posisi sebelum posisi di tuju
- bantu = bantu->next; ➔ bantu sama dengan bantu -> next
- nomor++; ➔ nomor++ (increment)
- baru->next = bantu->next; ➔ mengatur next dari node baru menjadi node setelah bantu
- bantu->next = baru; ➔ mengatur next dari bantu menjadi node baru
- void hapusDepan() { ➔ prosedur untuk menghapus node di depan

- if (!isEmpty()) { ➔ percabangan if jika list tidak kosong
- Node *hapus = head;➔ membuat pointer hapus yang menunjuk ke head
- string namaHapus = hapus->nama; ➔ Menyimpan nama dari node yang akan dihapus
- if (head->next != NULL) { ➔ percabangan if Jika ada lebih dari satu node
- head = head->next; ➔ mengatur head menjadi node kedua
- } else { ➔ jika hanya ada satu node
- head = tail = NULL; ➔ mengatur head dan tail menjadi NULL
- delete hapus; ➔ menghapus node pertama
- cout << "\nData (" << namaHapus << ") berhasil dihapus." << endl; ➔ menampilkan pesan bahwa data telah dihapus
- } else { ➔ jika linked list kosong
- cout << "List kosong!" << endl; ➔ menampilkan pesan bahwa list kosong
- void hapusBelakangNIM(string nim) { ➔ prosedur untuk menghapus node di belakang list dengan parameter nim
- if (!isEmpty()) { ➔ jika linked list tidak kosong
- Node *hapus = tail; ➔ membuat pointer baru yang menunjuk ke node terakhir
- string namaHapus = hapus->nama; ➔ menyimpan nama dari node yang akan dihapus
- if (head != tail) { ➔ jika ada lebih dari satu node
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- while (bantu->next != tail) { ➔ mencari node sebelum tail
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- tail = bantu; ➔ mengatur tail menjadi node sebelum tail lama

- tail->next = NULL; ➔ Mengatur pointer next dari tail baru menjadi NULL
- } else { ➔ jika hanya ada satu node
- head = tail = NULL; ➔ mengatur head dan tail menjadi NUL
- delete hapus; ➔ menghapus node terakhir
- cout << "\nData (" << namaHapus << ") berhasil dihapus." << endl; ➔ cout menampilkan pesan bahwa data telah dihapus
- } else { ➔ jika linked list kosong
- cout << "List kosong!" << endl; ➔ cout menampilkan menampilkan pesan bahwa list kosong
- void hapusTengahNIM(string nim) { ➔ prosedur untuk menghapus node di tengah list dengan parmeter nim
- if (!isEmpty()) { ➔ jika linked list tidak kosong
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- Node *sebelum = NULL; ➔ membuat pointer sebelum yang menunjuk ke NULL
- while (bantu != NULL && bantu->nim != nim) { ➔ mencari node dengan nim yang diberikan
- sebelum = bantu; ➔ memindahkan pointer sebelum ke node saat ini
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- if (bantu == NULL) { ➔ jika node dengan nim yang diberikan tidak ditemukan
- cout << "Data dengan NIM " << nim << " tidak ditemukan." << endl; ➔ menampilkan pesan bahwa data tidak ditemukan
- } else { ➔ jika node dengan nim yang diberikan ditemukan
- if (sebelum != NULL) { ➔ percabangan if jika node yang ditemukan bukan node pertama
- sebelum->next = bantu->next; ➔ mengatur pointer next dari

node sebelum menjadi node setelah node yang ditemukan

- } else { ➔ Jika node yang ditemukan adalah node pertama
- head = bantu->next; ➔ mengatur head menjadi node kedua
- string namaHapus = bantu->nama; ➔ menyimpan nama dari node yang akan dihapus
- delete bantu; ➔ menghapus node yang ditemukan
- cout << "\nData (" << namaHapus << ") berhasil dihapus." << endl; ➔ menampilkan pesan bahwa data telah dihapus
- } else { ➔ jika linked list kosong
- cout << "List kosong!" << endl; ➔ cout menampilkan pesan bahwa list kosong
- void ubahDepan(string nama, string nim) { ➔ prosedur ubahDepan untuk mengubah data pada node pertama dengan parameter nama dan nim
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- string namaLama = head->nama; ➔ menyimpan nama lama sebelum diubah
- head->nama = nama; ➔ mengubah nama pada node pertama
- head->nim = nim; ➔ mengu bah nim pada node pertama
- cout << "\nData (" << namaLama << ") telah diganti dengan data (" << nama << ")." << endl; ➔ cout menampilkan pernyataan bahwa data telah berhasil diubah
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement bahwa list kosong
- void ubahTengah(string nama, string nim, int posisi) { ➔ prosedur ubahTengah untuk mengubah data pada node di posisi tengah dengan parameter string nama, string nim, int posisi
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- if (posisi < 1 || posisi > hitungList()) { ➔ percabangan if jika ika posisi kurang dari 1 atau posisi lebih dari hitungList())

- cout << "Posisi di luar jangkauan" << endl; ➔ cout meanmpilkan statement posisi di luar jangkauan
- } else if (posisi == 1) { ➔ jika posisi sama dengan 1
- cout << "Posisi bukan posisi tengah" << endl; ➔ cout menampilkan statement posisi bukan posisi tengah
- } else { ➔ jika posisi valid
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- int nomor = 1; ➔ menginisialisasi nomor sebagai 1
- while (nomor < posisi) { ➔ perulangan while dengan kondisi selama nomor kurang dari posisi
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- nomor++; ➔ nomor++(increment)
- string namaLama = bantu->nama; ➔ menyimpan nama lama sebelum diubah
- bantu->nama = nama; ➔ mengubah nama pada node di posisi tengah
- bantu->nim = nim; ➔ mengubah nim pada node di posisi tengah
- cout << "\nData (" << namaLama << ") telah diganti dengan data (" << nama << ")." << endl; ➔ cout menampilkan statement data telah berhasil diubah
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan statement List masih kosong!
- void ubahBelakang(string nama, string nim) { ➔ prosedur ubahBelakang untuk mengubah data pada node terakhir dengan perameter string nama, string nim
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- string namaLama = tail->nama; ➔ menyimpan nama lama sebelum diubah

- tail->nama = nama; ➔ mengubah nama pada node terakhir
- tail->nim = nim; ➔ mengubah nim pada node terakhir
- cout << "\nData (" << namaLama << ") telah diganti dengan data (" << nama << ")." << endl; ➔ cout menampilkan statement data telah berhasil diubah
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan pernyataan list kosong
- void clearList() { ➔ prosedur untuk menghapus semua node dalam linked list
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- Node *hapus; ➔ membuat pointer hapus
- while (bantu != NULL) { ➔ perulangan while selama bantu tidak sama dengan NULL
- hapus = bantu; ➔ meny impan alamat node yang akan dihapus
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- delete hapus; ➔ menghapus node
- head = tail = NULL; ➔ mengatur head dan tail menjadi NULL
- cout << "List berhasil terhapus!" << endl; ➔ cout menampilkan statement list berhasil terhapus
- void tampil() { ➔ prosedur untuk menampilkan semua node dalam linked list
- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- Node* bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- cout << "----------\n"; ➔ cout mencetak garis pembatas
- cout << "| No | Nama | NIM |\n"; ➔ cout mencetak header tabel

- cout << "---\n"; ➔ cout mencetak garis pembatas
- int no = 1; ➔ menginisialisasi no sebagai 1
- while (bantu != NULL) { ➔ perulangan while selama bantu tidak sama dengan null
- cout << "| " << left << setw(2) << no << " | " << left << setw(17) << bantu->nama << " | " << left << setw(19) << bantu->nim << " |\n"; ➔ cout mencetak nomor, nama, dan nim
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- no++; ➔ no++(increment)
- cout << "---\n"; ➔ cout mencetak garis pembatas
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout menampilkan pernyataan List masih kosong!
- int main() { ➔ merupakan fungsi utama yang akan dieksekusi saat program dijalankan.
- init(); ➔ inisialisasi linked list
- int pilihan; ➔ deklarasi variabel pilihan untuk menyimpan pilihan menu
- string nama, nim, namaHapus; ➔ deklarasi variabel nama, nim, namaHapus dengan tipe data string
- int posisi; ➔ deklarasi variabel posisi dengan tipe data integer
- do { ➔ memulai perulangan do-while
- cout << "==
  " << endl; ➔ cout menampilkan batas garis
- cout << "| PROGRAM SINGLE LINKED LIST NON-CIRCULAR |" << endl;c➔ cout menampilkan header dengan judul program single linked list non-circular
- cout <<

"==
" << endl; ➔ cout menampilkan batas garis

- cout << "| No. | Pilihan Menu |" << endl; ➔ cout menampilkkan nomor dan pilihan menu
- cout << "==
" << endl; ➔ cout menampilkan batas garis
- cout << "| 1. | Tambah Depan |" << endl; ➔ cout menampilkan pilihan menu nomor 1 Tambah Depan
- cout << "| 2. | Tambah Belakang |" << endl; ➔ cout menampilkan pilihan menu nomor 2 Tambah Belakang
- cout << "| 3. | Tambah Tengah |" << endl; ➔ cout menampilkan pilihan menu nomor 3 Tambah Tengah
- cout << "| 4. | Ubah Depan |" << endl; ➔ cout menampilkan pilihan menu nomor 4 Ubah Depan
- cout << "| 5. | Ubah Belakang |" << endl;➔ cout menampilkan pilihan menu nomor 5 Ubah Belakang
- cout << "| 6. | Ubah Tengah |" << endl; ➔ cout menampilkan pilihan menu nomor 6 Ubah Tengah
- cout << "| 7. | Hapus Depan |" << endl; ➔ cout menampilkan pilihan menu nomor 7 Hapus Depan
- cout << "| 8. | Hapus Belakang |" << endl;➔ cout menampilkan pilihan menu nomor 8 Hapus Belakang
- cout << "| 9. | Hapus Tengah |" << endl; ➔ cout menampilkan pilihan menu nomor 9 Hapus Tengah
- cout << "| 10. | Hapus List |" << endl; ➔ cout menampilkan pilihan menu nomor 10 Hapus List
- cout << "| 11. | TAMPILKAN |" << endl; ➔ cout menampilkan menu nomor 11 Tampilkan
- cout << "| 0. | KELUAR |" << endl; ➔ cout menampilkan pilihan menu nomor 0 Keluar
- cout << "==

" << endl; ➔ cout menampilkan garis pembatas

- cout << "Pilih Operasi: "; ➔ cout menempilkan statement Pilih Operasi untuk meminta input pilihan operasi dari pengguna
- cin >> pilihan; ➔ menerima input dari pengguna dan menyimpan pada variabel pilihan
- switch (pilihan) { ➔ operasi switch statement berdasarkan pilihan
- case 1: ➔ jika pilihan nomor 1 yaitu Tambah depan
- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM
- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- insertDepan(nama, nim); ➔ menambahkan node baru di depan linked list
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout menampilkan nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "=================================\n" << endl; ➔ cout menampilkan garis pembatas
- break; ➔ mengahiri case 1
- case 2: ➔ jika pilihan adalah 2 yaitu Tambah Belakang

- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM
- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- insertBelakang(nama, nim); ➔ menambahkan node baru di belakang linked list
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout menampilkan nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "\nData telah ditambahkan." << endl; ➔ cout menampilkan statement Data telah ditambahkan
- cout << "=================================\n" << endl; ➔ cout menampilkan garis pembatas
- break; ➔ mengakhiri case 2
- case 3: ➔ jika pilihan adalah nomor 3 yaitu Tambah Tengah
- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM

- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- cout << "Masukkan Posisi: "; ➔ cout meminta user untuk memasukan posisi
- cin >> posisi; ➔ menerima input dari pengguna dan menyimpan pada variabel posisi
- insertTengah(nama, nim, posisi); ➔ menambahkan node baru di posisi tertentu dalam linked list
- cout << "----------\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM | Posisi |\n"; ➔ cout menampilkan nama dan nim
- cout << "----------\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " | " << setw(6) << left << posisi << " |\n"; ➔ cout mencetak nama dan nim
- cout << "----------\n"; ➔ cout menampilkan garis pembatas
- cout << "==========\n" << endl; ➔ cout menampilkan garis pembatas
- break; ➔ mengakhiri case 3
- case 4: ➔ jika pilihan nomor 4 yaitu Ubah depan
- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM
- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- ubahDepan(nama, nim); ➔ mengubah data pada node pertama

dengan nama dan nim baru

- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout menampilkan nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "=================================\n" << endl; ➔ cout menampilkan garis pembatas
- break; ➔ mengakhiri case 4
- case 5: ➔ jika pilihan nomor 5 yaitu Ubah Belakang
- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM
- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- ubahBelakang(nama, nim); ➔ mengubah data pada node terakhir dengan nama dan nim baru
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout menampilkan nama dan nim
- cout << "---------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10)

<< left << nim << " |\n"; ➔ cout mencetak nama dan nim

- cout << "--------------------------------\n"; ➔ cout menampilkan garis pembatas
- cout << "==================================\n" << endl; ➔ cout menampilkan garis pembatas
- break; ➔ mengakhiri case 5
- case 6: ➔ jika pilihan nomor 6 yaitu Ubah Tengah
- cout << "Masukkan Nama: "; ➔ cout meminta user untuk memasukan nama
- cin >> nama; ➔ menerima input dari pengguna dan menyimpan pada variabel nama
- cout << "Masukkan NIM: "; ➔ cout meminta user untuk memasukan NIM
- cin >> nim; ➔ menerima input dari pengguna dan menyimpan pada variabel nim
- cout << "Masukkan Posisi: "; ➔ cout meminta user untuk memasukan posisi
- cin >> posisi; ➔ menerima input dari pengguna dan menyimpan pada variabel posisi
- ubahTengah(nama, nim, posisi); ➔ mengubah data pada node di posisi tertentu dengan nama dan nim baru
- cout << "---\n"; ➔ cout menampilkan garis pembatas
- cout << "| Nama | NIM | Posisi |\n"; ➔ cout menampilkan nama dan nim
- cout << "---\n"; ➔ cout menampilkan garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " | " << setw(6) << left << posisi << " |\n";
- cout << "---\n"; ➔ cout mencetak nama dan nim
- cout <<

"==========\n" << endl; ➔ cout mencetak nama dan nim

- break; ➔ mengakhiri case 6
- case 7: ➔ jika pilihan nomor 7 yaitu hapus Depan
- if (!isEmpty()) { ➔ jika linked list tidak kosong
- nama = head->nama; ➔ simpan nama sebelum node dihapus
- nim = head->nim; ➔ simpan nim sebelum node dihapus
- hapusDepan(); ➔ hapus node pertama
- cout << "----------\n"; ➔ cout mencetak garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout mencetak header tabel
- cout << "----------\n"; ➔ cout mencetak garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim yang telah dihapus
- cout << "----------\n"; ➔ cout mencetak garis pembatas
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout mencetak pesan bahwa list kosong
- break; ➔ mengakhiri case 7
- case 8: ➔ jika pilihan nomor 8 yaitu hapus Belakang
- cout << "Masukkan NIM: "; ➔ cout meminta input NIM dari pengguna
- cin >> nim; ➔ menerima input NIM dan menyimpan pada variabel nim
- if (!isEmpty()) { ➔ jika linked list tidak kosong
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama

- while (bantu != NULL && bantu->nim != nim) { ➔ mencari node dengan NIM yang sama dengan input
- bantu = bantu->next; ➔ pindahkan pointer bantu ke node selanjutnya
- if (bantu == NULL) { ➔ percabangan if jika node dengan NIM yang sama tidak ditemukan
- cout << "Data dengan NIM " << nim << " tidak ditemukan." << endl; ➔ cout mencetak pesan bahwa data tidak ditemukan
- } else { ➔ jika node dengan NIM yang sama ditemukan
- nama = bantu->nama; ➔ Simpan nama sebelum node dihapus
- hapusBelakangNIM(nim); ➔ menghapus node dengan NIM yang sama dengan input
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout mencetak header tabel
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim yang telah dihapus
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout mencetak pesan bahwa list kosong
- break; ➔ mengakhiri case 8
- case 9: ➔ jika pilihan nomor 9 yaitu hapus Tengah
- cout << "Masukkan NIM: "; ➔ cout meminta input NIM dari pengguna
- cin >> nim; ➔ menerima input NIM dan menyimpan pada variabel nim

- if (!isEmpty()) { ➔ percabangan if jika linked list tidak kosong
- Node *bantu = head; ➔ membuat pointer bantu yang menunjuk ke node pertama
- while (bantu != NULL && bantu->nim != nim) { ➔ mencari node dengan NIM yang sama dengan input
- bantu = bantu->next; ➔ memindahkan pointer bantu ke node selanjutnya
- if (bantu == NULL) { ➔ percabangan if jika node dengan NIM yang sama tidak ditemukan
- cout << "Data dengan NIM " << nim << " tidak ditemukan." << endl; ➔ cout mencetak pesan bahwa data tidak ditemukan
- } else { ➔ jika node dengan NIM yang sama ditemukan
- nama = bantu->nama; ➔ menyimpan nama sebelum node dihapus
- hapusTengahNIM(nim); ➔ menghapus node dengan NIM yang sama dengan input
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- cout << "| Nama | NIM |\n"; ➔ cout mencetak header tabel
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- cout << "| " << setw(17) << left << nama << " | " << setw(10) << left << nim << " |\n"; ➔ cout mencetak nama dan nim yang telah dihapus
- cout << "---------------------------------\n"; ➔ cout mencetak garis pembatas
- cout << "Data berhasil dihapus." << endl; ➔ cout mencetak pesan bahwa data telah berhasil dihapus
- } else { ➔ jika linked list kosong
- cout << "List masih kosong!" << endl; ➔ cout mencetak pesan bahwa list kosong

- break; ➔ mengakhiri case 9
- case 10: ➔ jika pilihan nomor 10 yaitu Hapus List
- clearList(); ➔ menghapus semua node dalam linked list
- break; ➔ mengakhiri case 10
- case 11: ➔ jika pilihan nomor 11 yaitu Tampilkan
- tampil(); ➔ menampilkan semua node dalam linked list
- break; ➔ mengakhiri case 11
- case 0: ➔ jika pilihan nomor 0 yaitu Keluar
- cout << "Terima kasih telah menggunakan program ini." << endl; ➔ cout menampilkan statement Terima kasih telah menggunakan program ini
- break; ➔ mengakhiri case 0
- default: ➔ jika pengguna memilih opsi yang tidak valid yaitu diluar pilihan menu tambah depan sampai keluar
- cout << "Pilihan tidak valid." << endl; ➔ cout menapilkan statement Pilihan tidak valid
- } while (pilihan != 0); ➔ perulangan while mengulangi loop selama kondisi pilihan tidak sama dengan 0
- return 0; ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

D. Kesimpulan

1) Linked List Non Circular

Linked list non circular yaitu linked list dengan node pertama (head) dan node terakhir (tail) yang tidak saling terhubung. Operasi pada linked list non circular diantaranya deklarasi simpul (node), membuat dan menginisialisasi pointer head dan tail, pengecekan kondisi linked list, penambahan simpul (node), penghapusan simpul (node), dan tampil data linked list.

2) Linked List Circular

Linked list circular merupakan linked list yang tidak memiliki akhir karena node terakhir (tail) tidak bernilai 'NULL', tetapi terhubung dengan node pertama (head).. Pada cirlular linked list, node pertama dan node terakhir saling terhubung satu sama lain sehingga membentuk lingkaran. Linked list circular dapat digunakan untuk menyimpan data yang perlu diakses secara berulang, seperti daftar putar lagu, daftar pesan dalam antrian, atau penggunaan memori berulang dalam suatu aplikasi. operasi pada linked list circular diantaranya deklarasi simpul (node), membuat dan menginisialisasi pointer head dan tail, pengecekan kondisi linked list, pembuatan simpul (node), penambahan simpul (node), penghapusan simpul (node), dan menampilkan data linked list.

E. Referensi

[1] Asisten Praktikum, Struktur Data “ARRAY", WhatsApp, 2024.

[2] BobTylerMSFT. (n.d.).<iomanip>. diakses dari

https://learn.microsoft.com/id-id/cpp/standard-library/iomanip?view=msvc-170

[3] geeksforgeeks.org. Introduction to Circular Linked List. diakses dari

https://www.geeksforgeeks.org/circular-linked-list/