LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL XI
MULTI LINKED LIST

Disusun Oleh :
NAMA : Muhamad Naufal Ammar
NIM : 103112430036

Dosen
WAHYU ANDI SAPUTRA

PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

A. Dasar Teori

Circular Linked List merupakan pengembangan dari struktur linked list linier yang memiliki karakteristik khusus, yaitu node terakhir menunjuk kembali ke node pertama sehingga membentuk struktur melingkar. Struktur data ini dinilai efektif untuk sistem yang membutuhkan proses traversal berulang tanpa kondisi akhir yang eksplisit, seperti manajemen antrian dinamis, sistem penjadwalan, dan simulasi sistem berputar (Goodrich et al., 2016; Weiss, 2020). Penggunaan pointer yang saling terhubung memungkinkan efisiensi memori karena tidak memerlukan indeks seperti pada array, serta mendukung operasi insert dan delete secara dinamis dengan kompleksitas waktu yang relatif rendah. ADT Circular Linked List juga menekankan pemisahan antara definisi data dan implementasi operasinya, sehingga meningkatkan modularitas, keterbacaan kode, serta kemudahan pemeliharaan sistem perangkat lunak (Sedgewick & Wayne, 2018).

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

```
#ifndef MULTILIST_H
#define MULTILIST_H

#include <iostream>
using namespace std;

struct Anak {
    int info;
    Anak* next;
};

struct Induk {
    int info;
    Anak* firstAnak;
    Induk* next;
};

void createInduk(Induk*& p, int x);
void insertAnak(Induk* induk, int x);
void printAll(Induk* L);


#endif
```

*C:\Program Files\Notepad++\ch...

File Edit Search View Encoding Language Settings
Tools Macro Run Plugins Window ?

change.log

```
1  Nama : Muhamad Naufal Ammar
2  NIM : 103112430036
3  Kelas 12-IF-03
```

Ln: 3 Col: 15 Windows (CR LF) UTF-8 INS

```
#include "multilist.h"

void createInduk(Induk*& L, int x) {
    Induk* p = new Induk;
    p->info = x;
    p->firstAnak = NULL;
    p->next = L;
    L = p;
}

void insertAnak(Induk* induk, int x) {
    Anak* p = new Anak;
    p->info = x;
    p->next = NULL;
    if (induk->firstAnak == NULL) {
        induk->firstAnak = p;
    } else {
        Anak* q = induk->firstAnak;
        while (q->next != NULL) {
            q = q->next;
        }
        q->next = p;
    }
}

void printAll(Induk* L) {
    while (L != NULL) {
        cout << "Pegawai: " << L->info << endl;
        cout << "Induk: " << L->info << endl;
        Anak* A = L->firstAnak;
        while (A != NULL) {
            cout << A->info << endl;
            A = A->next;
        }
        cout << endl;
        L = L->next;
    }
}
```

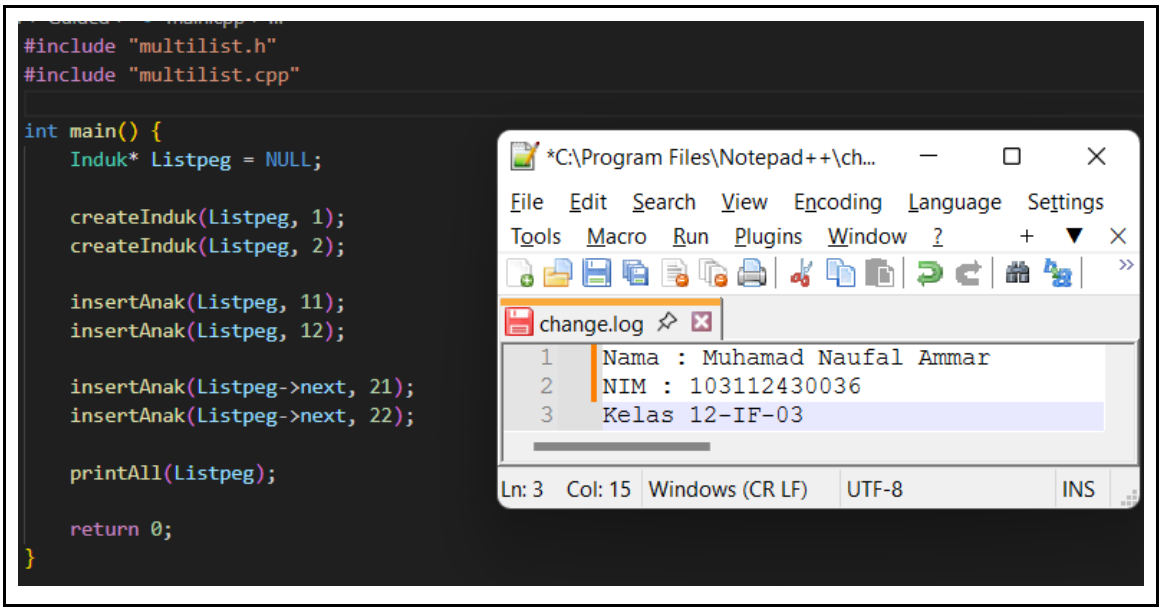
*C:\Program Files\Notepad++\ch...

File Edit Search View Encoding Language Settings
Tools Macro Run Plugins Window ?

change.log

```
1  Nama : Muhamad Naufal Ammar
2  NIM : 103112430036
3  Kelas 12-IF-03
```

Ln: 3 Col: 15 Windows (CR LF) UTF-8 INS



Screenshots Output

Deskripsi:

Kode ini adalah program C++ sederhana yang mendemonstrasikan penggunaan struktur data multilist untuk merepresentasikan hubungan induk-anak, seperti dalam konteks pegawai dan anak buah. Program dimulai dengan menginisialisasi pointer Listpeg ke NULL, kemudian membuat dua induk dengan ID 1 dan 2 menggunakan fungsi createInduk. Selanjutnya, anak-anak ditambahkan ke masing-masing induk: ID 11 dan 12 ke induk pertama, serta ID 21 dan 22 ke induk kedua melalui insertAnak. Akhirnya, fungsi printAll dipanggil untuk mencetak seluruh struktur multilist, dan program berakhir dengan return 0.

C. Unguided (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```
#ifndef CIRCULARLIST_H
#define CIRCULARLIST_H
#define Nil NULL
#include <string>
using namespace std;
struct mahasiswa {
    string nama;
    string nim;
    char jenis_kelamin;
    float ipk;
};

typedef mahasiswa infotype;
struct ElmList;
typedef ElmList* address;

struct ElmList {
    infotype info;
    address next;
};
struct List {
    address First;
};

void CreateList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void insertFirst(List &L, address P);
void insertAfter(List &L, address Prec, address P);
void insertLast(List &L, address P);
void deleteFirst(List &L, address &P);
void deleteAfter(List &L, address Prec, address &P);
void deleteLast(List &L, address &P);
address findElm(List L, infotype x);
void printInfo(List L);

#endif
```

*C:\Program Files\Notepad++\ch...

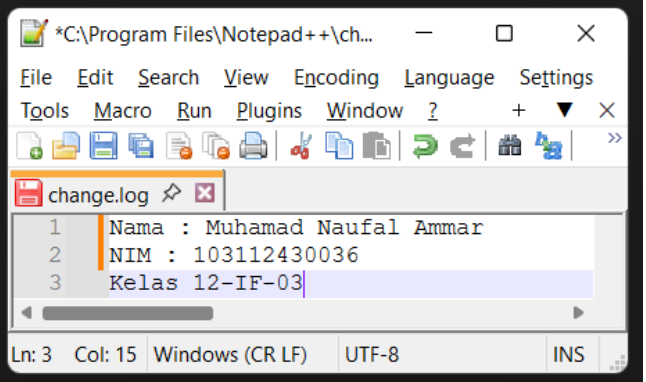
File Edit Search View Encoding Language Settings
Tools Macro Run Plugins Window ?

change.log

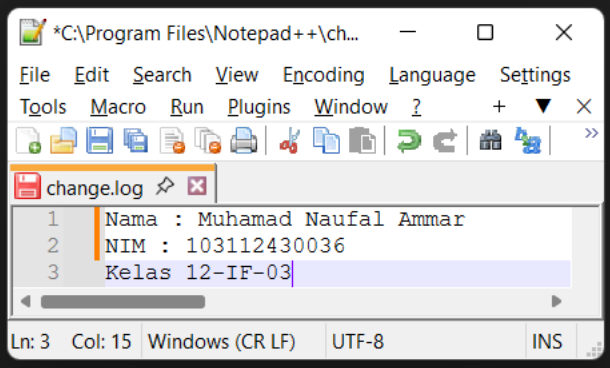
```
1  Nama : Muhamad Naufal Ammar
2  NIM : 103112430036
3  Kelas 12-IF-03
```

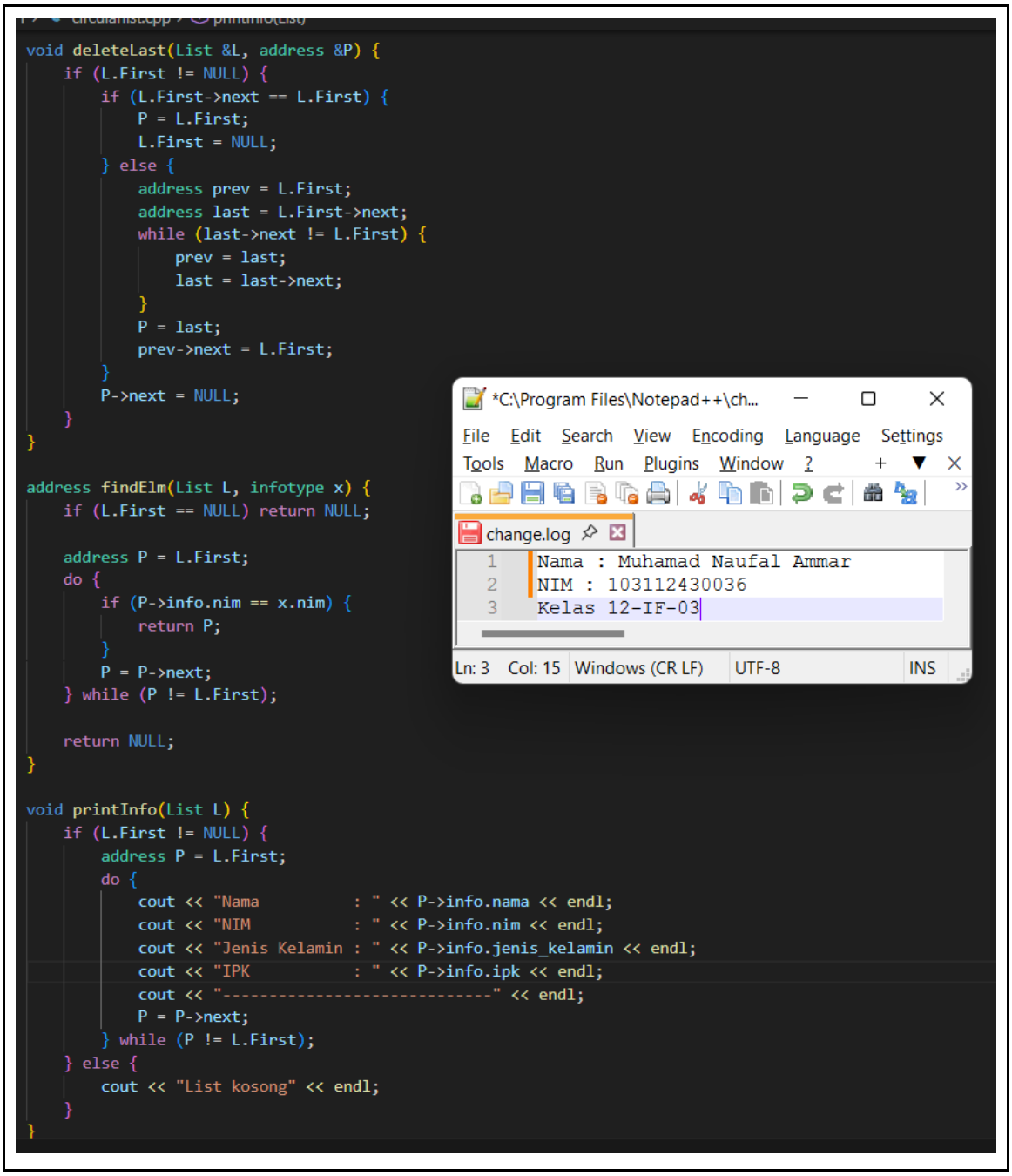
Ln: 3 Col: 15 Windows (CR LF) UTF-8 INS

```cpp
#include "circularlist.h"
#include <iostream>

void CreateList(List &L) {
    L.First = NULL;
}

address alokasi(infotype x) {
    address P = new ElmList;
    P->info = x;
    P->next = NULL;
    return P;
}

void dealokasi(address &P) {
    delete P;
    P = NULL;
}

void insertFirst(List &L, address P) {
    if (L.First == NULL) {
        L.First = P;
        P->next = P;
    } else {
        address last = L.First;
        while (last->next != L.First) {
            last = last->next;
        }
        P->next = L.First;
        last->next = P;
        L.First = P;
    }
}

void insertAfter(List &L, address Prec, address P) {
    if (Prec != NULL) {
        P->next = Prec->next;
        Prec->next = P;
    }
}
```



```
void insertLast(List &L, address P) {
    if (L.First == NULL) {
        L.First = P;
        P->next = P;
    } else {
        address last = L.First;
        while (last->next != L.First) {
            last = last->next;
        }
        last->next = P;
        P->next = L.First;
    }
}

void deleteFirst(List &L, address &P) {
    if (L.First != NULL) {
        if (L.First->next == L.First) {
            P = L.First;
            L.First = NULL;
        } else {
            address last = L.First;
            while (last->next != L.First) {
                last = last->next;
            }
            P = L.First;
            L.First = L.First->next;
            last->next = L.First;
        }
        P->next = NULL;
    }
}

void deleteAfter(List &L, address Prec, address &P) {
    if (Prec != NULL && Prec->next != L.First) {
        P = Prec->next;
        Prec->next = P->next;
        P->next = NULL;
    }
}
```





```
void deleteLast(List &L, address &P) {
    if (L.First != NULL) {
        if (L.First->next == L.First) {
            P = L.First;
            L.First = NULL;
        } else {
            address prev = L.First;
            address last = L.First->next;
            while (last->next != L.First) {
                prev = last;
                last = last->next;
            }
            P = last;
            prev->next = L.First;
        }
        P->next = NULL;
    }
}

address findElm(List L, infotype x) {
    if (L.First == NULL) return NULL;

    address P = L.First;
    do {
        if (P->info.nim == x.nim) {
            return P;
        }
        P = P->next;
    } while (P != L.First);

    return NULL;
}

void printInfo(List L) {
    if (L.First != NULL) {
        address P = L.First;
        do {
            cout << "Nama          : " << P->info.nama << endl;
            cout << "NIM           : " << P->info.nim << endl;
            cout << "Jenis Kelamin : " << P->info.jenis_kelamin << endl;
            cout << "IPK           : " << P->info.ipk << endl;
            cout << "----------------------------" << endl;
            P = P->next;
        } while (P != L.First);
    } else {
        cout << "List kosong" << endl;
    }
}
```

*C:\Program Files\Notepad++\ch...

File Edit Search View Encoding Language Settings
Tools Macro Run Plugins Window ?

change.log

```
1  Nama : Muhamad Naufal Ammar
2  NIM : 103112430036
3  Kelas 12-IF-03
```

Ln: 3 Col: 15 Windows (CR LF) UTF-8 INS

```cpp
#include "circularlist.cpp"
address createData(string nama, string nim, char jenis_kelamin, float ipk) {
infotype x;
address P;
x.nama = nama;
x.nim = nim;
x.jenis_kelamin = jenis_kelamin;
x.ipk = ipk;
P = alokasi(x);
return P;
}

int main() {
        List L, A, B, L2;
        address P1 = Nil;
        address P2 = Nil;
        infotype x;
        CreateList(L);
        cout<<"coba insert first, last, dan after"<<endl;
        P1 = createData("Danu", "04", 'l', 4.0);
        insertFirst(L,P1);
        P1 = createData("Fahmi", "06", 'l',3.45);
        insertLast(L,P1);
        P1 = createData("Bobi", "02", 'l',3.71);
        insertFirst(L,P1);
        P1 = createData("Ali", "01", 'l', 3.3);
        insertFirst(L,P1);
        P1 = createData("Gita", "07", 'p', 3.75);
        insertLast(L,P1);
        x.nim = "07";
        P1 = findElm(L,x);
        P2 = createData("Cindi", "03", 'p', 3.5);
        insertAfter(L, P1, P2);
        x.nim = "02";
        P1 = findElm(L,x);
        P2 = createData("Hilmi", "08", 'p', 3.3);
        insertAfter(L, P1, P2);
        x.nim = "04";
        P1 = findElm(L,x);
        P2 = createData("Eli", "05", 'p', 3.4);
        insertAfter(L, P1, P2);
        printInfo(L);
        return 0;
}
```

*C:\Program Files\Notepad++\ch...

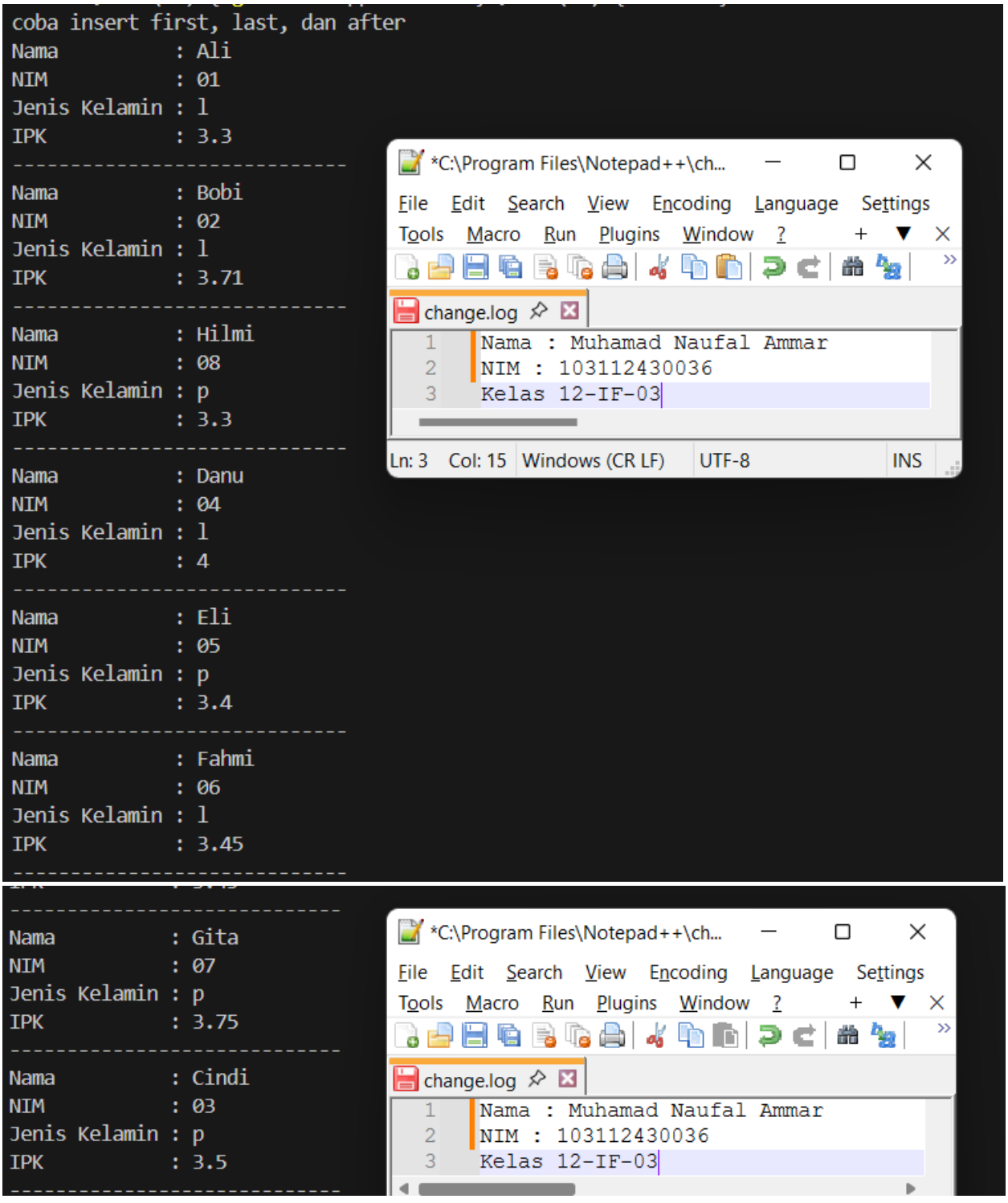
File Edit Search View Encoding Language Settings
Tools Macro Run Plugins Window ?

change.log

```
1  Nama : Muhamad Naufal Ammar
2  NIM : 103112430036
3  Kelas 12-IF-03
```

Ln: 3 Col: 15 Windows (CR LF) UTF-8 INS

Screenshots Output



Deskripsi:

Implementasi Abstract Data Type (ADT) Circular Linked List yang digunakan untuk mengelola data mahasiswa, di mana setiap elemen list berisi informasi nama, NIM, jenis kelamin, dan IPK, serta memiliki pointer next yang membentuk struktur melingkar dengan elemen terakhir menunjuk kembali ke elemen pertama. Implementasi ini menyediakan operasi dasar ADT seperti pembuatan list kosong, alokasi dan dealokasi memori, penambahan elemen di awal, tengah, dan akhir list, penghapusan elemen di awal, setelah elemen tertentu, dan di akhir list, pencarian data mahasiswa berdasarkan NIM, serta penampilan seluruh isi list ke layar. Struktur circular memungkinkan traversal data dilakukan secara berulang tanpa batas hingga kembali ke elemen awal, sehingga pengelolaan data menjadi efisien dan konsisten, terutama untuk kasus yang memerlukan perputaran data secara kontinu.

D. Kesimpulan

Berdasarkan implementasi dan landasan teori yang digunakan, ADT Circular Linked List pada kode tersebut telah memenuhi prinsip dasar struktur data dinamis yang efisien dan modular. Seluruh operasi utama seperti alokasi memori, penyisipan, penghapusan, pencarian, dan penelusuran data telah diimplementasikan dengan mempertahankan sifat circular, sehingga konsistensi struktur data tetap terjaga. Penerapan ADT ini sangat sesuai untuk pengelolaan data mahasiswa yang membutuhkan fleksibilitas tinggi dalam pengolahan data, sekaligus menjadi contoh penerapan konsep struktur data abstrak yang baik dan relevan dengan kebutuhan sistem modern.

E. Referensi

Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2016). *Data structures and algorithms in Java* (6th ed.). Wiley.

Sedgewick, R., & Wayne, K. (2018). *Algorithms* (4th ed.). Addison-Wesley.

Weiss, M. A. (2020). *Data structures and algorithm analysis in C++* (5th ed.). Pearson Education.