LAPORAN PRAKTIKUM STRUKTUR DATA

MODUL VI
DOUBLY LINKED LIST (BAGIAN PERTAMA)

Disusun Oleh :

NAMA : ADIKA AUNURFIKRI NOVIYANTO

NIM : 103112400195

Dosen

FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

A. Dasar Teori

C++ adalah pengembangan dari dari bahasa c yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), beriorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

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

Guided 1

```
#ifndef DOUBLYLIST_H
#define DOUBLYLIST_H
#include <iostream>
#include <string>
using namespace std;

struct kendaraan {
   string nopol;
   string warna;
   int thnbuat;
};

typedef kendaraan infotype;

struct ElmList;
typedef ElmList* address;

struct ElmList {
   infotype info;
   address next;
   address prev;
};

struct List {
   address first;
   address last;
};


void createList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void printInfo(List L);
```

```
void insertLast(List &L, address P);
address findElm(List L, string nopol);
void deleteFirst(List &L, address &P);
void deleteLast(List &L, address &P);
void deleteAfter(List &L, address Prec, address &P);

#endif
```

DoublyList.cpp

```
#include "Doublylist.h"

void createList(List &L) {
    L.first = NULL;
    L.last = NULL;
}

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

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

void printInfo(List L) {
    address P = L.last;
    cout << "\nDATA LIST 1\n";
    while (P != NULL) {
        cout << "no polisi : " << P->info.nopol << endl;
        cout << "warna     : " << P->info.warna << endl;
        cout << "tahun     : " << P->info.thnbuat << endl;
```

```
        cout << endl;
        P = P->prev;
    }
}

void insertLast(List &L, address P) {
    if (L.first == NULL) {
        L.first = P;
        L.last = P;
    } else {
        L.last->next = P;
        P->prev = L.last;
        L.last = P;
    }
}

address findElm(List L, string nopol) {
    address P = L.first;
    while (P != NULL && P->info.nopol != nopol) {
        P = P->next;
    }
    return P;
}

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

```
    }
}

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

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

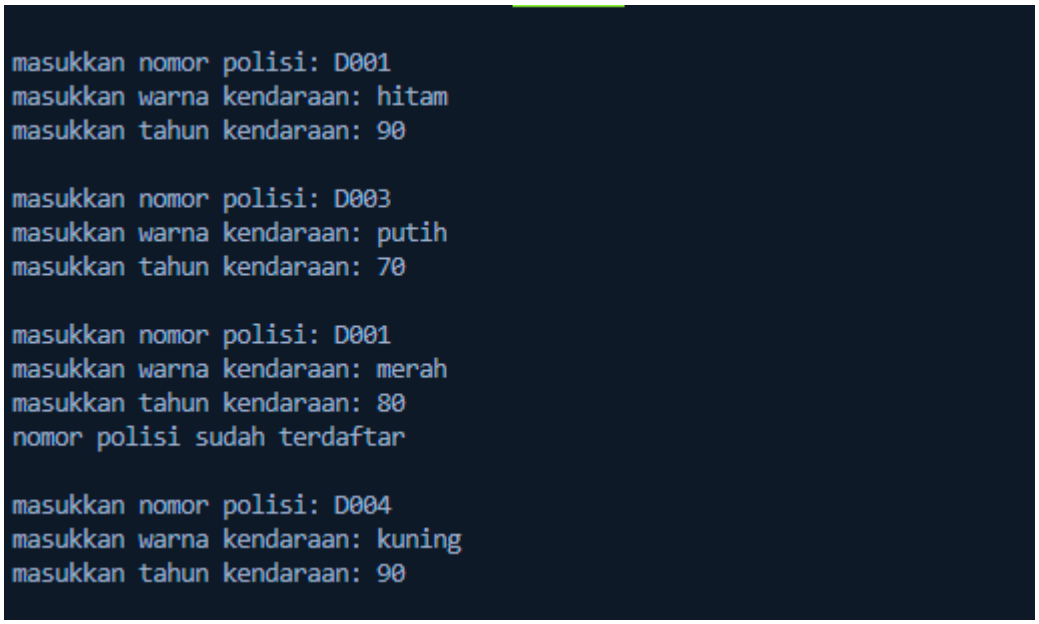
Main.cpp

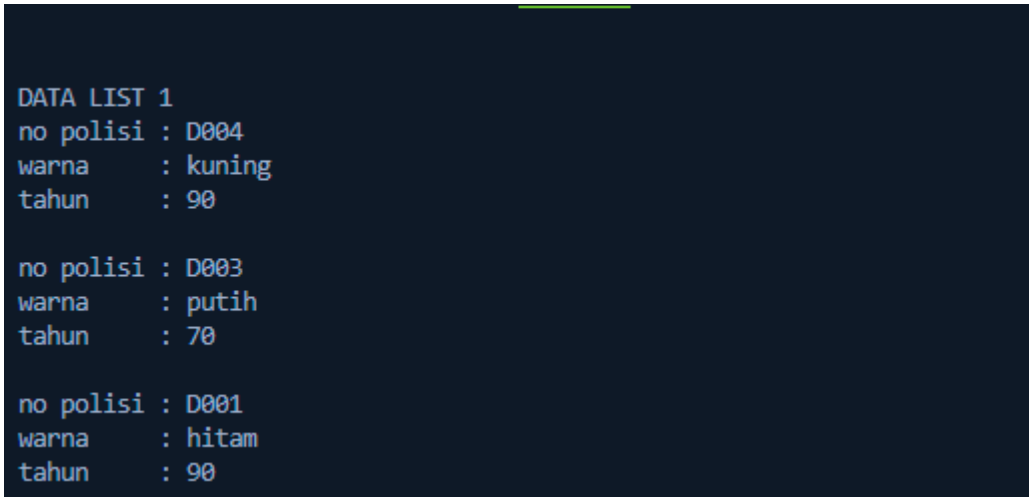
```
#include "Doublylist.h"

int main() {
    List L;
    createList(L);

    infotype x;
    address P;
    string cari;
    int n;

    cout << "Masukkan jumlah kendaraan: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << "\nmasukkan nomor polisi: ";
        cin >> x.nopol;
        cout << "masukkan warna kendaraan: ";
        cin >> x.warna;
        cout << "masukkan tahun kendaraan: ";
        cin >> x.thnbuat;

        if (findElm(L, x.nopol) != NULL) {
            cout << "nomor polisi sudah terdaftar\n";
            continue;
        }

        P = alokasi(x);
        insertLast(L, P);
    }

    printInfo(L);

    // CARI DATA
```

```
    cout << "Masukkan Nomor Polisi yang dicari : ";
    cin >> cari;
    P = findElm(L, cari);
    if (P != NULL) {
        cout << "\nNomor Polisi yang dicari : " << P->info.nopol << endl;
        cout << "Warna  : " << P->info.warna << endl;
        cout << "Tahun  : " << P->info.thnbuat << endl;
    } else {
        cout << "Data tidak ditemukan.\n";
    }

    // HAPUS DATA
    cout << "\nMasukkan Nomor Polisi yang akan dihapus : ";
    cin >> cari;
    P = findElm(L, cari);
    if (P == NULL) {
        cout << "Data tidak ditemukan.\n";
    } else {
        if (P == L.first)
            deleteFirst(L, P);
        else if (P == L.last)
            deleteLast(L, P);
        else
            deleteAfter(L, P->prev, P);

        cout << "Data dengan nomor polisi " << cari << " berhasil dihapus.\n";
        dealokasi(P);
    }

    printInfo(L);
    return 0;
}
```

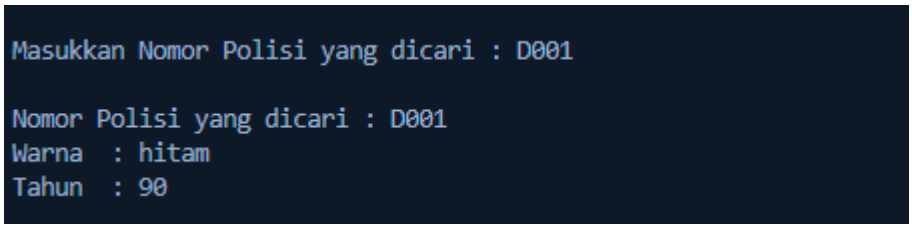
Screenshots Output:

1.

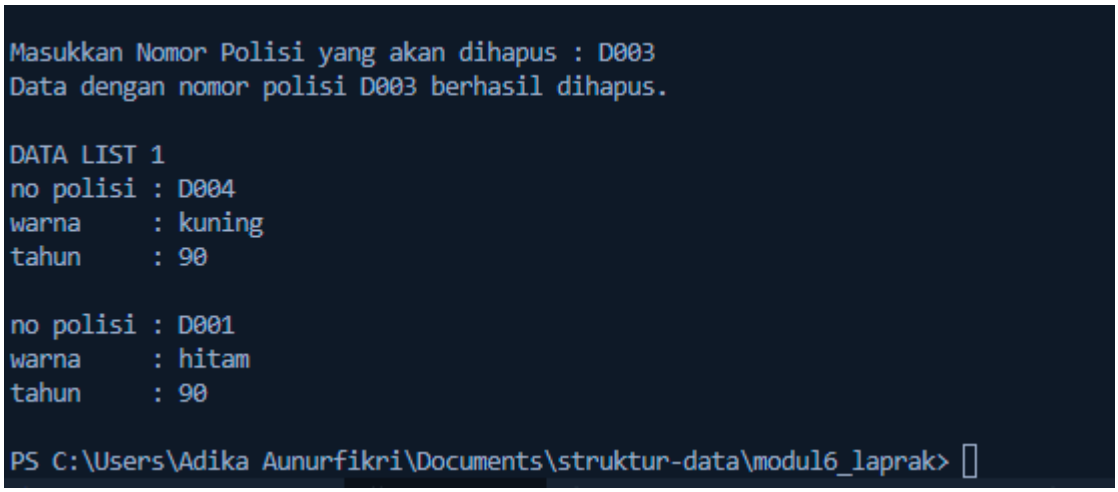




2.



3.



Deskripsi:

Program ini merupakan implementasi ADT Doubly Linked List dalam bahasa C++ untuk mengelola data kendaraan yang terdiri dari nomor polisi, warna kendaraan, dan tahun pembuatan. Struktur data yang digunakan memungkinkan penyimpanan dan pengelolaan data secara dinamis, di mana setiap elemen memiliki pointer ke elemen berikutnya (next) dan pointer ke elemen sebelumnya (prev).

C. Kesimpulan

Program ADT Doubly Linked List ini menunjukkan bagaimana struktur data linked list ganda dapat digunakan untuk menyimpan dan mengelola data kendaraan secara efisien. Dengan menggunakan dua pointer pada setiap elemen, proses penelusuran maju dan mundur menjadi lebih fleksibel, serta memungkinkan operasi penambahan, pencarian, dan penghapusan data dilakukan dengan mudah tanpa perlu menggeser elemen lain.

D. Referensi

- W3Resource. (2020). *C++ String Exercises: Convert digit/number to words*.
- GeeksforGeeks. (2020). *Loops in C++ (for, while, do-while).*