LAPORAN PRAKTIKUM
Modul 6
DOUBLE LINKED LIST (BAGIAN PERTAMA)

Disusun Oleh:
Nama : Ganes Gemi Putra
NIM : 2311104075
Kelas : SE-07-02

Dosen : WAHYU ANDI SAPUTRA

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

1. **Tujuan**
   1. **Memahami konsep modul linked list.**
   2. **Mengaplikasikan konsep double linked list dengan menggunakan pointer dan dengan bahasa C**

2. **Landasan Teori**

1. Struktur Data Double Linked List (DLL)

- **Definisi: Double Linked List (DLL) adalah struktur data berbentuk daftar berantai (linked list) di mana setiap elemen (node) memiliki dua pointer, yaitu**

pointer ke elemen sebelumnya (prev) dan pointer ke elemen berikutnya (next). Struktur ini memungkinkan traversing (penelusuran) baik dari awal ke akhir maupun dari akhir ke awal.

- **Karakteristik: DLL memungkinkan akses dua arah antara node, sehingga lebih fleksibel dalam operasi penambahan dan penghapusan data, terutama jika dibandingkan dengan Single Linked List.**
- **Kelebihan dan Kekurangan: Keuntungan utama DLL adalah kemudahan dalam penghapusan dan penyisipan node pada posisi tertentu, karena node dapat diakses dari dua arah. Namun, penggunaan memori menjadi lebih besar karena adanya dua pointer di setiap node.**

2. Operasi Dasar pada Double Linked List

- **Penambahan Elemen (Insertion): Pada DLL, elemen baru dapat ditambahkan di awal (head), di akhir (tail), atau di tengah-tengah list dengan mengatur ulang pointer next dan prev di node yang relevan.**
- **Penghapusan Elemen (Deletion): Elemen dapat dihapus dari awal, akhir, atau posisi tertentu dalam DLL. Operasi penghapusan melibatkan pengaturan ulang pointer di node sebelum dan setelah node yang akan dihapus.**
- **Pencarian Elemen (Searching): DLL dapat digunakan untuk mencari elemen tertentu dengan melakukan iterasi mulai dari head atau tail.**

3. Penerapan Double Linked List pada Sistem Penyimpanan Data

Dalam konteks program pengelolaan kendaraan, DLL digunakan untuk menyimpan informasi mengenai kendaraan, seperti Nomor Polisi, Warna Kendaraan, dan Tahun Pembuatan. Data ini disimpan dalam bentuk node pada list, sehingga memudahkan proses pencarian dan penghapusan berdasarkan kriteria tertentu (misalnya berdasarkan nomor polisi).

4. Konsep Modularitas dalam Pemrograman

- **Fungsi dan Prosedur: Program dirancang secara modular, dengan memecah setiap operasi (seperti insertLast, findElm, deleteFirst, deleteLast, dll.) ke dalam fungsi atau prosedur terpisah. Hal ini membuat program lebih terstruktur, mudah dibaca, dan mudah dikelola.**
- **Penggunaan Pointer: Dalam C++, pointer digunakan untuk menangani memori secara dinamis. Pointer memberikan fleksibilitas dalam mengelola node pada list, khususnya untuk menghubungkan elemen-elemen dalam DLL.**

5. Prinsip Low Coupling dan High Cohesion dalam Desain Program

- **Low Coupling: Program diorganisir dengan fungsi-fungsi terpisah untuk setiap operasi yang berdiri sendiri, sehingga meminimalkan ketergantungan antar fungsi. Ini memudahkan pengujian dan pemeliharaan program.**
- **High Cohesion: Setiap fungsi hanya bertanggung jawab untuk satu tugas spesifik (misalnya findElm hanya untuk pencarian, deleteFirst hanya untuk penghapusan elemen pertama), yang meningkatkan keterbacaan dan memudahkan debugging.**

6. Penggunaan Struktur Data dalam C++

- **Struct: Program menggunakan struct untuk mendefinisikan infotype (berisi data**

kendaraan) dan elmlist (untuk node list), yang merupakan cara efektif untuk mengelompokkan data terkait di C++.

- **Pointer dan Alokasi Dinamis: Fungsi alokasi digunakan untuk membuat node baru secara dinamis, dan fungsi dealokasi untuk menghapus node dari memori. Ini adalah penerapan dari pengelolaan memori dinamis di C++ yang penting dalam pemrograman dengan linked list.**

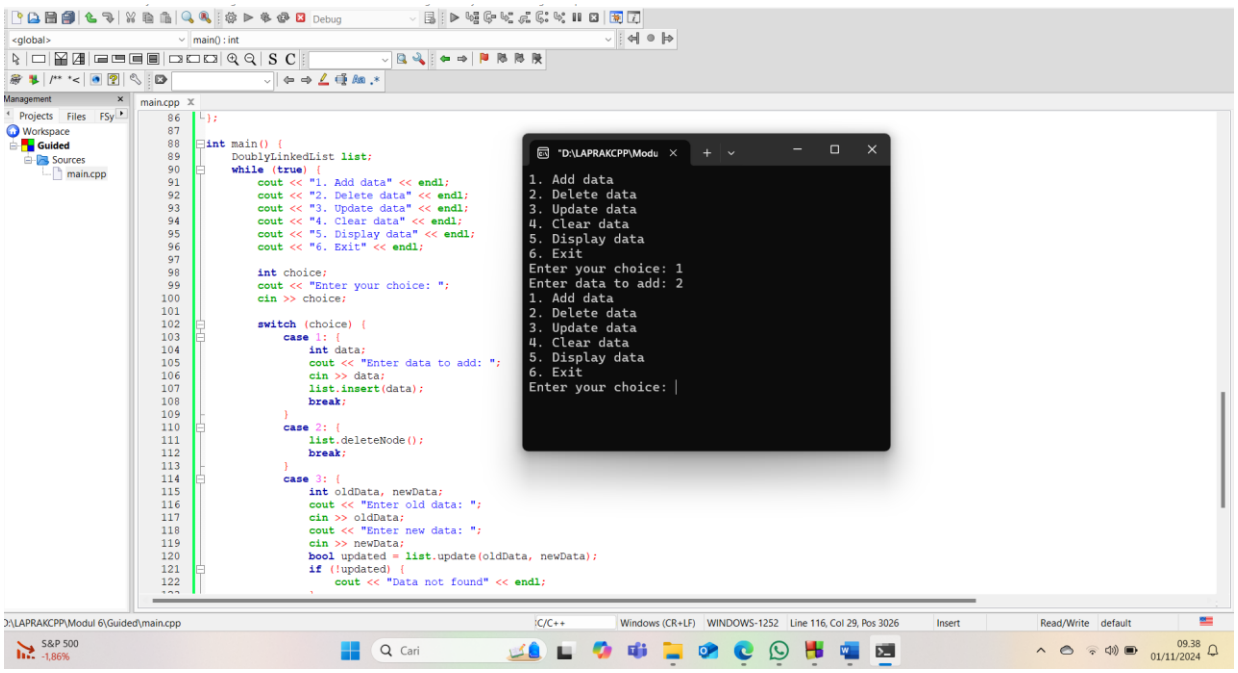
7. **Implementasi Data Abstrak (Abstract Data Type)**

- **Dalam pemrograman berbasis linked list, penggunaan struktur data seperti DLL dan berbagai operasi yang dilakukan mencerminkan konsep abstrak dari "List". Implementasi ini memungkinkan program untuk memanipulasi data kendaraan tanpa memperhatikan detail bagaimana data tersebut disimpan, menjadikannya lebih modular dan mudah diperluas.**

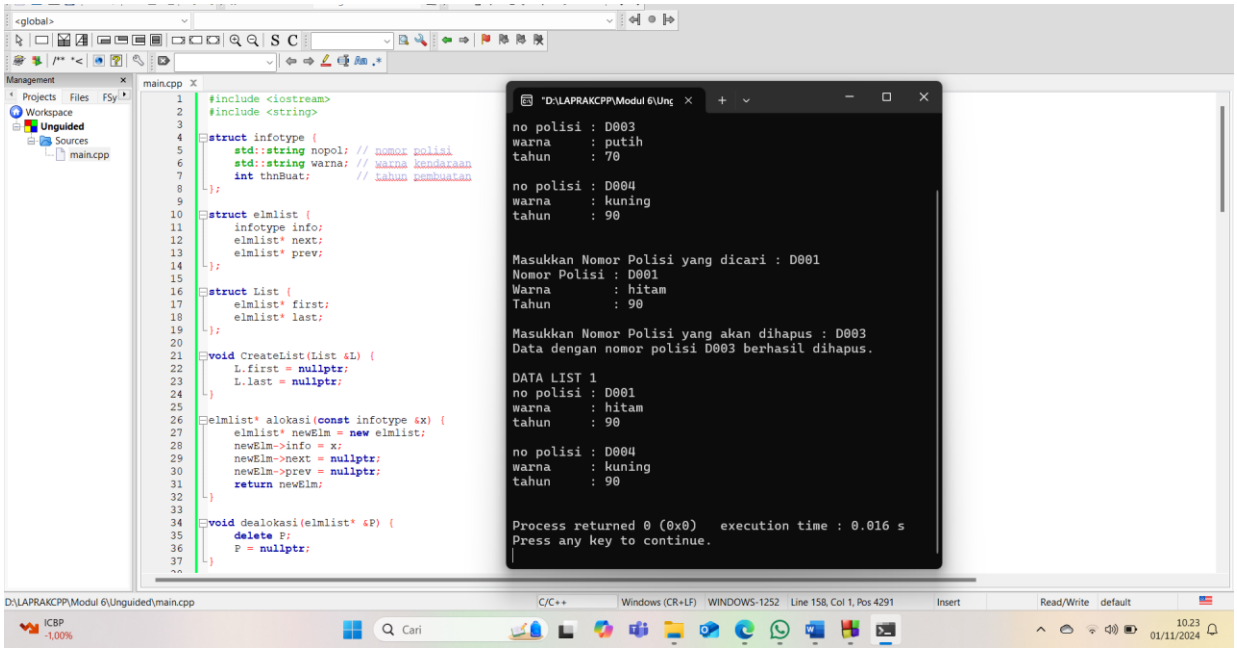
8. **Contoh Aplikasi Linked List pada Dunia Nyata**

- **Aplikasi dari Double Linked List ini banyak digunakan pada pengelolaan data yang membutuhkan akses dinamis, seperti Sistem Manajemen Kendaraan, Daftar Kontak, atau Manajemen Stok Barang. DLL sangat cocok untuk situasi di mana data perlu diakses, dihapus, atau dimodifikasi dari posisi mana pun dalam list.**

3. **Guided**



4. Unguided



5. Kesimpulan

- **Efisiensi dan Fleksibilitas**: Double Linked List memungkinkan penambahan, pencarian, dan penghapusan data dengan lebih fleksibel dan efisien, terutama karena elemen dalam list dapat diakses dua arah, dari awal maupun akhir list. Hal ini sangat bermanfaat dalam pengelolaan data yang dinamis seperti manajemen kendaraan.

- **Struktur Modular**: Program dirancang secara modular dengan memisahkan setiap operasi (seperti penambahan, penghapusan, dan pencarian) ke dalam fungsi terpisah. Ini mendukung prinsip **low coupling** dan **high cohesion**, yang menjadikan program lebih mudah dipahami, dipelihara, dan diperluas.

- **Pengelolaan Memori Dinamis**: Penggunaan pointer dan alokasi memori dinamis memungkinkan penghematan memori dan peningkatan kinerja karena memori hanya digunakan sesuai kebutuhan. Ini penting dalam aplikasi yang berhubungan dengan data yang sering berubah.

- **Aplikasi dalam Kehidupan Nyata**: Struktur Double Linked List dapat diimplementasikan pada berbagai sistem manajemen data di dunia nyata, seperti manajemen kendaraan, daftar kontak, dan inventaris barang, di mana data sering ditambahkan, dihapus, atau dimodifikasi.