

LAPORAN TUGAS BESAR

Implementasi Struktur Data Multi-Linked List untuk Mengelola Pengerjaan Project dalam kepanitiaan

Disusun untuk memenuhi Tugas Besar Mata Kuliah Struktur Data (STD)



Arunna Janu Rezasyah **103022400084**

Zulfan Falah Permana **103022400076**

Kelas: SE-48-01

Dosen Pengampu : Nurjayanti

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA - UNIVERSITAS TELKOM

DESEMBER 2025

A. Tujuan Tugas

Tujuan dari tugas besar mata kuliah Struktur Data ini adalah untuk mempelajari dan mengimplementasikan konsep struktur data Multi-Linked List (MLL) menggunakan bahasa pemrograman C++. Pada tugas ini, kami mengimplementasikan struktur data tersebut berdasarkan studi kasus Simulator Organisasi yang mengelola data Divisi, Panitia, dan Project.

Pengerjaan tugas besar ini bertujuan untuk memahami cara merancang struktur data yang saling terhubung, khususnya implementasi relasi One-to-Many (antara Divisi dengan Panitia) dan relasi Many-to-Many (antara Panitia dengan Project). Selain itu, tugas ini mencakup implementasi operasi manipulasi data seperti penambahan elemen (*insert*), pencarian (*searching*), penghapusan data beserta relasinya (*cascading delete*), serta penelusuran (*traversal*) untuk menampilkan keterkaitan antara divisi, panitia, dan project yang sedang dikerjakan.

B. Deskripsi Studi Kasus dan Struktur Data

Deskripsi Studi Kasus Studi kasus yang digunakan pada tugas besar ini adalah Simulator Organisasi. Sistem ini dirancang untuk mengelola sumber daya manusia dan kegiatan dalam sebuah organisasi. Data yang dikelola meliputi:

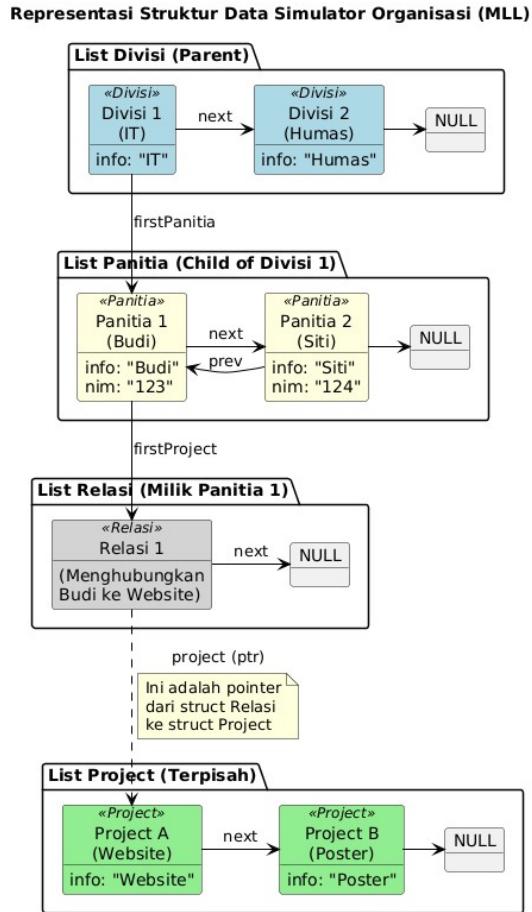
1. Divisi sebagai pengelompokan unit kerja.
2. Panitia (Anggota) yang bernaung di bawah divisi tertentu.
3. Project yang merupakan tugas atau acara yang harus dikerjakan oleh panitia.

Struktur Data Struktur data disusun menggunakan konsep Multi-Linked List (MLL) yang kompleks untuk menangani relasi antar entitas, dimana:

1. List Parent (Induk) merepresentasikan Divisi. Setiap node divisi memiliki *pointer* yang menunjuk ke daftar anggotanya.
2. List Child (Anak) merepresentasikan Panitia. Ini adalah list dinamis (Double Linked List) yang bergantung pada divisi tertentu (Relasi *One-to-Many*: Satu divisi memiliki banyak panitia).
3. List Relasi merepresentasikan hubungan kerja antara Panitia dan Project. Setiap node panitia memiliki list relasi yang menunjuk ke Project yang sedang dikerjakannya (Relasi *Many-to-Many*: Satu panitia bisa mengerjakan banyak project, dan satu project bisa dikerjakan banyak panitia).
4. List Project berdiri sebagai list terpisah yang menyimpan data detail mengenai proyek-proyek yang tersedia.

Struktur ini sangat sesuai untuk direpresentasikan menggunakan MLL karena mampu memodelkan hubungan nyata dalam organisasi secara fleksibel, di mana keanggotaan bersifat hierarkis (Divisi ke Panitia) namun penugasan kerja bersifat jaringan (Panitia ke Project).

Representasi Visual (Diagram)



C. Deskripsi Implementasi Kode Program

1. Struktur Data yang Digunakan

Struktur data yang digunakan pada program ini adalah Multi-Linked List (MLL) yang menghubungkan tiga entitas utama: Divisi, Panitia, dan Project. Model relasi yang diterapkan adalah kombinasi hierarki dan jaringan:

- List Divisi (Parent): Implementasi Single Linked List (SLL).
 - Setiap node memiliki pointer `firstPanitia` yang menunjuk ke anggota pertama (Child).
 - Pointer `next` menunjuk ke divisi berikutnya.

- List Panitia (Child): Implementasi Double Linked List (DLL).
 - Setiap node memiliki pointer firstProject yang menunjuk ke daftar tugas/relasi (Sub-child).
 - Pointer next dan prev menghubungkan antar panitia dalam satu divisi.
- List Project : Implementasi Single Linked List (SLL).
 - Berdiri sendiri sebagai referensi data proyek.
- List Relasi: Implementasi Double Linked List (DLL).
 - Merupakan node kecil yang dimiliki oleh Panitia.
 - Memiliki pointer project yang menyimpan alamat memori dari node di List Project (menghubungkan Panitia ke Project tanpa menduplikasi data).

2. Inisialisasi List

Fungsi createListDivisi dan createListProject digunakan untuk menginisialisasi pointer utama (head) agar bernilai nullptr. Hal ini memastikan list dalam keadaan kosong sebelum operasi penambahan data dilakukan.

3. Pembuatan Elemen (Allocation)

Terdapat fungsi alokasi memori untuk setiap entitas:

- newDivisi: Membuat node divisi baru dan mengeset firstPanitia ke nullptr.
- newPanitia: Membuat node panitia baru dan mengeset firstProject, next, dan prev ke nullptr.
- newProject: Membuat node project baru.
- newRelasi: Membuat node relasi baru yang menyimpan alamat project yang dituju.

4. Penambahan Data (Insert Logic)

- Insert Divisi & Project: Menggunakan logika insertLast. Node baru ditambahkan di akhir list menggunakan iterasi (while) hingga node terakhir ditemukan.
- Insert Panitia: Menambahkan panitia ke dalam divisi tertentu. Fungsi ini mencari ujung dari list panitia dalam divisi tersebut, lalu menyambungkan pointer next dan prev (karena menggunakan DLL).
- Assign Project (Membuat Relasi): Fungsi assignProject menghubungkan panitia dengan project. Fungsi ini membuat node relasi baru, lalu menambahkannya ke dalam list firstProject milik panitia tersebut.

5. Pencarian Data (Search)

Fungsi pencarian (findDivisi, findPanitia, findProject) menggunakan pendekatan iteratif (*linear search*).

- Case Insensitive: Pencarian dilengkapi dengan helper function toLower untuk memastikan pencarian string tidak terpengaruh oleh huruf besar/kecil.
- Find Relasi: Digunakan untuk mengecek apakah seorang panitia sudah mengerjakan project tertentu sebelum ditambahkan, guna mencegah duplikasi tugas.

6. Penghapusan Data (Delete & Cascading)

Implementasi penghapusan memiliki logika penanganan memori yang ketat (*cascading delete*) untuk mencegah *memory leak*:

- Delete Panitia: Menghapus node panitia dari divisi. Sebelum node panitia dihapus, program akan menghapus seluruh node dalam list firstProject (relasi) milik panitia tersebut.
- Delete Divisi: Menghapus node divisi. Sebelum divisi dihapus, program akan mengiterasi dan menghapus seluruh panitia yang ada di dalamnya beserta relasi tugasnya.
- Delete Project: Ini adalah fungsi penghapusan paling kompleks.
 1. Menghapus node project dari List Project utama.
 2. Melakukan Traversing Menyeluruh: Program akan mengecek setiap Divisi \rightarrow setiap Panitia \rightarrow setiap Relasi.
 3. Jika ditemukan relasi yang menunjuk ke project yang akan dihapus, relasi tersebut diputus dan memori dealokasi secara otomatis.

7. Traversal dan Penampilan Data

Program menggunakan beberapa metode traversal untuk menampilkan informasi dari sudut pandang berbeda:

- View Hierarki (Divisi-Oriented):

Fungsi printDivisi menelusuri: Divisi \rightarrow Panitia \rightarrow Relasi Project.

Output: Menampilkan nama divisi, diikuti daftar anggota, dan di bawah setiap anggota tercantum project yang sedang dikerjakan.

- View Project (Project-Oriented):

Fungsi printProjectWithPanitia menelusuri: Project \rightarrow Mencari ke seluruh list Divisi & Panitia.

Output: Menampilkan nama project, jumlah staff yang terlibat, serta detail nama staff dan asal divisinya.

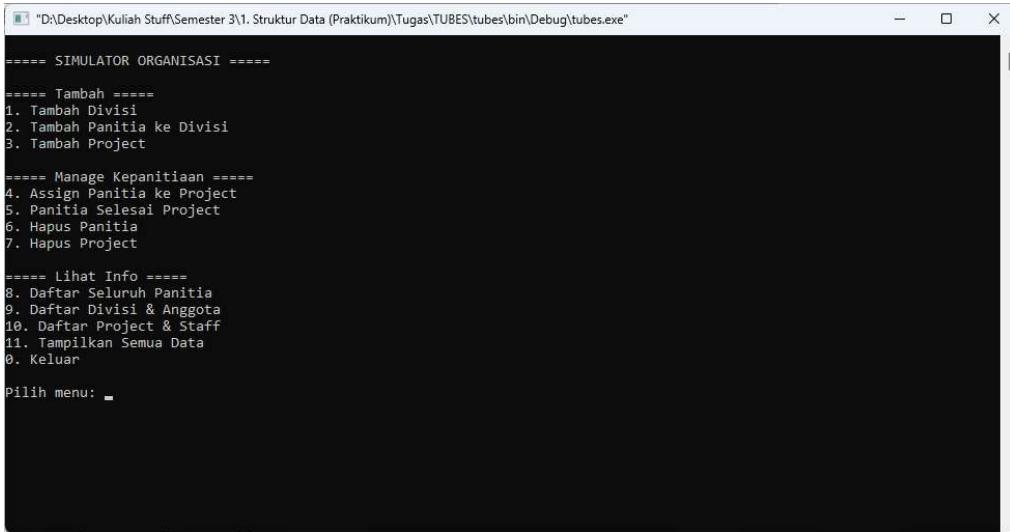
8. Program Utama (Main Program)

Pada fungsi main, alur program dikendalikan menggunakan menu interaktif (do-while loop) yang mencakup:

- Input data Divisi, Panitia, dan Project.
- Manajemen penugasan (Assign/Unassign Project).
- Operasi penghapusan data.
- Pemanggilan fungsi *view* untuk monitoring organisasi.

D. Hasil Output analisis

- Main Menu



```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\bin\Debug\tubes.exe"

===== SIMULATOR ORGANISASI =====

===== Tambah =====
1. Tambah Divisi
2. Tambah Panitia ke Divisi
3. Tambah Project

===== Manage Kepanitiaan =====
4. Assign Panitia ke Project
5. Panitia Selesai Project
6. Hapus Panitia
7. Hapus Project

===== Lihat Info =====
8. Daftar Seluruh Panitia
9. Daftar Divisi & Anggota
10. Daftar Project & Staff
11. Tampilkan Semua Data
0. Keluar

Pilih menu: _
```

- **Menu 1: Tambah Divisi**

- **Deskripsi:** Menu ini berguna untuk mendaftarkan divisi baru ke dalam organisasi. User akan diminta memasukkan nama divisi. Divisi baru akan ditempatkan di urutan paling akhir dari list divisi.
- **Fungsi yang dijalankan:**
 - A. newDivisi(d): Mengalokasikan memori untuk node divisi baru.
 - B. insertLastDivisi(listDivisi, D): Menyambungkan node baru tersebut ke akhir list divisi (*Single Linked List*).



```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\bin\Debug\tubes.exe"

===== Lihat Info =====
8. Daftar Seluruh Panitia
9. Daftar Divisi & Anggota
10. Daftar Project & Staff
11. Tampilkan Semua Data
0. Keluar

Pilih menu: 1
Nama divisi: PDD

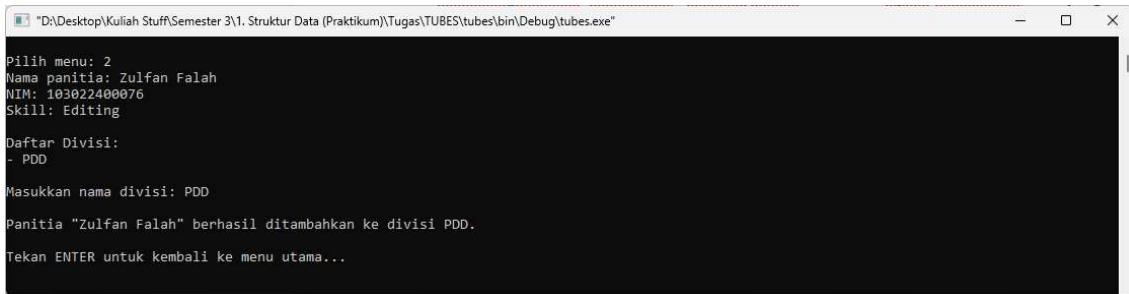
Divisi "PDD" berhasil ditambahkan.

Tekan ENTER untuk kembali ke menu utama..._
```

- **Menu 2: Tambah Panitia ke Divisi**

- **Deskripsi:** Menu ini berguna untuk menambahkan anggota (panitia) baru ke dalam divisi tertentu. User memasukkan data diri panitia, lalu memilih divisi mana yang akan dimasuki. Program akan mencari alamat divisi tersebut terlebih dahulu sebelum memasukkan panitia.
- **Fungsi yang dijalankan:**
 - A. findDivisi(listDivisi, namaDivisi): Mencari alamat node divisi tujuan berdasarkan nama yang diinput.

- B. newPanitia(p): Mengalokasikan memori untuk node panitia baru.
- C. insertPanitia(D, P): Menyambungkan node panitia ke dalam list child milik divisi tersebut (*Double Linked List*).



```
Pilih menu: 2
Nama panitia: Zulfan Falah
NIM: 103022400076
Skill: Editing

Daftar Divisi:
- PDD

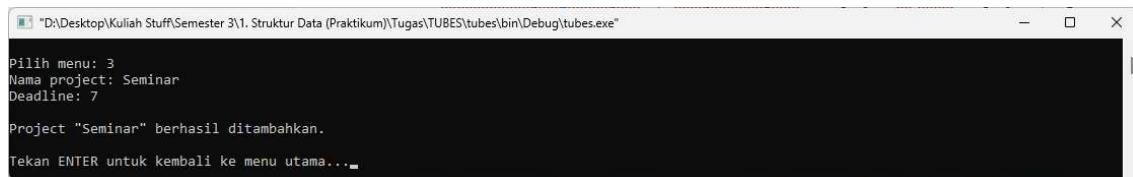
Masukkan nama divisi: PDD

Panitia "Zulfan Falah" berhasil ditambahkan ke divisi PDD.

Tekan ENTER untuk kembali ke menu utama...
```

- **Menu 3: Tambah Project**

- **Deskripsi:** Menu ini berguna untuk membuat data project baru (misal: Seminar, Website). Project ini awalnya berdiri sendiri dan belum memiliki staff yang mengerjakannya.
- **Fungsi yang dijalankan:**
 - A. newProject(p): Mengalokasikan memori untuk node project baru.
 - B. insertLastProject(listProject, P): Menyambungkan node project ke akhir list project (*Single Linked List*).



```
Pilih menu: 3
Nama project: Seminar
Deadline: 7

Project "Seminar" berhasil ditambahkan.

Tekan ENTER untuk kembali ke menu utama....
```

- **Menu 4: Assign Panitia ke Project**

- **Deskripsi:** Menu ini adalah inti dari **Multi-Linked List**. Berguna untuk menugaskan seorang panitia untuk mengerjakan project tertentu. Program akan melakukan validasi bertahap: memastikan project ada, divisi ada, dan panitia ada.
- **Fungsi yang dijalankan:**
 - A. findProject(...): Mencari alamat project yang dituju.
 - B. findDivisi(...): Mencari alamat divisi asal panitia.
 - C. findPanitia(...): Mencari alamat panitia berdasarkan NIM.
 - D. findRelasi(...): Mengecek apakah panitia tersebut sudah pernah mengambil project ini sebelumnya (mencegah duplikasi).
 - E. assignProject(Pan, Proj): Membuat node relasi baru di bawah node panitia yang menunjuk ke node project.

```

D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe

Pilih menu: 4
Daftar Project:
- Seminar

Masukkan nama project: Seminar

Daftar Divisi:
- PDD

Masukkan nama divisi: PDD

Daftar Panitia:
1. Zulfan Falah (103022400076)

Masukkan NIM panitia: 103022400076

Panitia "Zulfan Falah" (PDD) berhasil menjadi staff project "Seminar".

Tekan ENTER untuk kembali ke menu utama...

```

- **Menu 5: Panitia Selesai Project**

- **Deskripsi:** Menu ini berguna jika seorang panitia telah menyelesaikan tugasnya atau ditarik dari sebuah project. Ini **hanya menghapus hubungan/relasi** kerja, tidak menghapus data panitia maupun data project-nya.
- **Fungsi yang dijalankan:**
 - A. findProject(...): Validasi keberadaan project.
 - B. findDivisi(...) & findPanitia(...): Mencari panitia yang dimaksud.
 - C. panitiaDoneProject(Pan, namaProject): Mencari node relasi spesifik pada panitia tersebut lalu menghapusnya (dealokasi node relasi).

```

D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe

Pilih menu: 5
Daftar Project:
- Seminar

Masukkan nama project: Seminar

Daftar Divisi & Panitia:
Divisi: PDD
- Zulfan Falah
* Seminar

Masukkan nama divisi: PDD

Daftar Panitia:
1. Zulfan Falah (103022400076)

Masukkan NIM panitia: 103022400076

Panitia "Zulfan Falah" selesai mengerjakan project "Seminar".

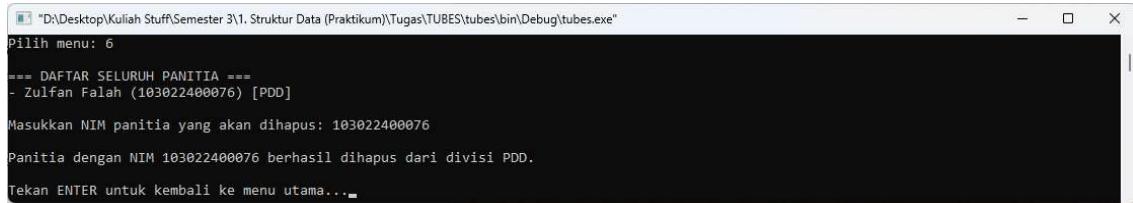
Tekan ENTER untuk kembali ke menu utama...

```

- **Menu 6: Hapus Panitia**

- **Deskripsi:** Menu ini berguna untuk mengeluarkan panitia dari organisasi (misal: mengundurkan diri). User hanya perlu memasukkan NIM. Program di main.cpp akan mencari NIM tersebut ada di divisi mana, lalu menghapusnya.
- **Fungsi yang dijalankan:**
 - A. findPanitia(...): Digunakan dalam *looping* di main program untuk menemukan lokasi panitia di setiap divisi.

- B. `deletePanitia(D, nim)`: Menghapus node panitia dari list divisi, sekaligus membersihkan semua tugas/relasi yang sedang dikerjakannya (otomatis *cascading delete* relasi).



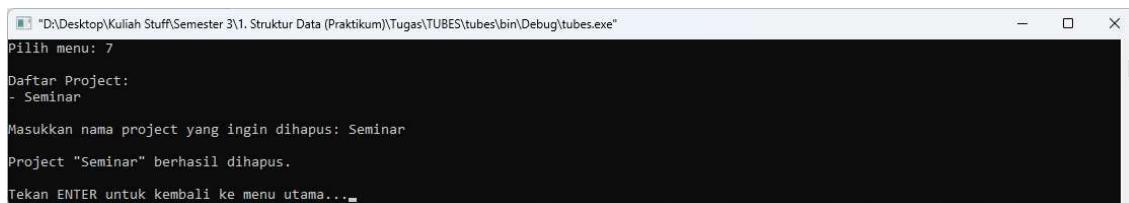
```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
Pilih menu: 6
*** DAFTAR SELURUH PANITIA ===
- Zulfan Falah (103022400076) [PDD]

Masukkan NIM panitia yang akan dihapus: 103022400076
Panitia dengan NIM 103022400076 berhasil dihapus dari divisi PDD.

Tekan ENTER untuk kembali ke menu utama..."
```

- **Menu 7: Hapus Project**

- **Deskripsi:** Menu ini berguna untuk menghapus project (misal: project batal). Operasi ini sangat krusial karena **berdampak pada banyak panitia**. Jika project dihapus, maka tugas tersebut juga harus hilang dari daftar tugas semua panitia yang mengerjakannya.
- **Fungsi yang dijalankan:**
 - A. `deleteProject(listProject, namaProject, listDivisi)`: Fungsi kompleks yang menghapus node project utama, lalu menelusuri seluruh divisi dan panitia untuk menghapus semua node relasi yang terhubung ke project tersebut (*Cascading Delete*).



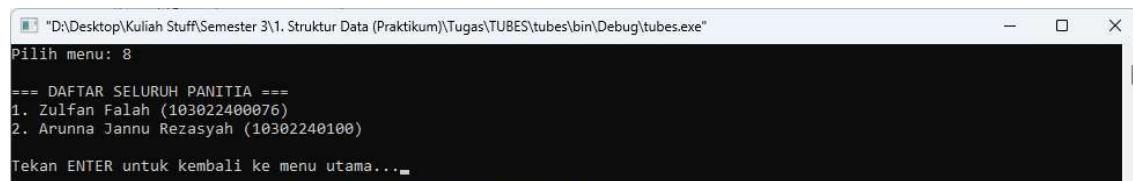
```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
Pilih menu: 7
Daftar Project:
- Seminar

Masukkan nama project yang ingin dihapus: Seminar
Project "Seminar" berhasil dihapus.

Tekan ENTER untuk kembali ke menu utama..."
```

- **Menu 8: Daftar Seluruh Panitia**

- **Deskripsi:** Menampilkan daftar absen seluruh anggota organisasi yang dikelompokkan per divisi, namun hanya menampilkan info dasar (Nama & NIM) tanpa detail project.
- **Fungsi yang dijalankan:**
 - A. `printPanitiaSimple(D->firstPanitia)`: Dipanggil berulang kali di dalam loop main.cpp untuk setiap divisi yang ada.



```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
Pilih menu: 8
*** DAFTAR SELURUH PANITIA ===
1. Zulfan Falah (103022400076)
2. Arunna Jannu Rezasyah (10302240100)

Tekan ENTER untuk kembali ke menu utama..."
```

- **Menu 9: Daftar Divisi & Anggota**

- **Deskripsi:** Menampilkan struktur hierarki organisasi secara lengkap. Mulai dari Nama Divisi, daftar Anggota, hingga Project apa saja yang sedang dikerjakan oleh masing-masing anggota.

- **Fungsi yang dijalankan:**

- A. printDivisi(listDivisi): Melakukan traversal menyeluruh (Divisi -> Panitia -> Relasi Project).

```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
--- DAFTAR DIVISI & ANGGOTA ---
Divisi: PDD
- Zulfan Falah
- Arunna Jannu Rezasyah

Tekan ENTER untuk kembali ke menu utama...
```

- **Menu 10: Daftar Project & Staff**

- **Deskripsi:** Menampilkan data dari sudut pandang Project. Berguna untuk melihat beban kerja setiap project (siapa saja staff yang terlibat dan berapa jumlahnya).
- **Fungsi yang dijalankan:**

- A. printProjectWithPanitia(listProject, listDivisi): Melakukan traversal dari Project, lalu mencari ke seluruh list panitia untuk menemukan siapa yang memiliki relasi ke project tersebut.

```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
--- DAFTAR PROJECT & STAFF ---
Seminar (2 Staff)
* Zulfan Falah - PDD
* Arunna Jannu Rezasyah - PDD

Tekan ENTER untuk kembali ke menu utama...
```

- **Menu 11: Tampilkan Semua Data**

- **Deskripsi:** Menu rangkuman yang menampilkan output dari Menu 9 dan Menu 10 secara bersamaan untuk memberikan gambaran total kondisi organisasi.

- **Fungsi yang dijalankan:**

- A. printDivisi(...)
- B. printProjectWithPanitia(...)

```
"D:\Desktop\Kuliah Stuff\Semester 3\1. Struktur Data (Praktikum)\Tugas\TUBES\tubes\bin\Debug\tubes.exe"
0. Keluar
Pilih menu: 11

--- DATA DIVISI ---
Divisi: PDD
- Zulfan Falah
  * Seminar
- Arunna Jannu Rezasyah
  * Seminar

--- DATA PROJECT ---
Seminar (2 Staff)
* Zulfan Falah - PDD
* Arunna Jannu Rezasyah - PDD

Tekan ENTER untuk kembali ke menu utama...
```

- **Menu 0: Keluar**
- **Deskripsi:** Mengakhiri loop do-while pada program utama (main.cpp) dan menutup aplikasi dengan return 0.

E. Kesimpulan

Tugas besar mata kuliah Struktur Data ini berhasil mengimplementasikan konsep **Multi-Linked List (MLL)** dalam studi kasus Simulator Organisasi yang melibatkan entitas Divisi, Panitia, dan Project. Struktur data yang dirancang mampu merepresentasikan relasi **One-to-Many** dan **Many-to-Many** secara efektif tanpa duplikasi data.

Seluruh operasi utama seperti penambahan, pencarian, penelusuran, serta penghapusan data dengan mekanisme **cascading delete** telah berjalan dengan baik dan menghasilkan data yang konsisten. Program juga mampu menampilkan informasi baik dari sudut pandang hierarki organisasi maupun dari sisi project, sehingga memudahkan pengelolaan dan monitoring data.

Dengan demikian, dapat disimpulkan bahwa penggunaan Multi-Linked List sangat sesuai untuk menangani sistem dengan hubungan data yang kompleks. Tugas besar ini telah memenuhi tujuan pembelajaran mata kuliah Struktur Data serta meningkatkan pemahaman dalam perancangan struktur data dan manajemen memori menggunakan bahasa pemrograman C++.