

RANCANG BANGUN SISTEM INFORMASI MANAJEMEN GYM MENGGUNAKAN STRUKTUR DATA NON-LINEAR (BST) DAN MULTI-LINKED LIST



Oleh:

1. LUTFI SHIDQI MARDIAN - 103112400077
2. ABISAR FATHIR - 103112400068
3. DIMAS RAMADHANI - 103112400065

S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

2025

Daftar Isi

BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan	1
BAB II ANALISIS STRUKTUR DATA	2
2.1 Binary Search Tree (BST) – Modul Data Member	2
2.2 Multi-Linked List (MLL) - Modul Paket Latihan.....	2
2.3 Relasi (Associative List) - Modul Transaksi.....	2
BAB III PEMBAGIAN TUGAS DAN KONTRIBUTSI	3
BAB IV IMPLEMENTASI ALGORITMA	4
4.1 Algoritma Insert Member (Rekursif BST).....	4
4.2 Algoritma Menambah Gerakan ke Paket (Traversal MLL)	5
BAB V HASIL PENGUJIAN (SCREENSHOTS)	6
5.1 Tampilan Menu Utama.....	6
5.2 Tampilan Menu Member.....	6
5.3 Tambah Member	7
5.4 Statistik Member.....	7
5.5 Traversal.....	7
5.6 Menu Latihan	8
5.7 Tambah Paket (Menu Kesulitan).....	8
5.8 Tambah Gerakan	9
5.9 Statistik Paket.....	9
5.10 Menu Relasi	10
5.11 Assign Paket ke Member.....	10
BAB VI SOURCE CODE LENGKAP	11
6.1 Header File (gym.h).....	11
6.2 Implementasi (gym.cpp, relasi.cpp, main.cpp)	14
BAB VII LINK DEMO PROGRAM.....	15
BAB VIII KESIMPULAN.....	15

BAB I PENDAHULUAN

1.1 Latar Belakang

Pusat kebugaran (Gym) modern membutuhkan sistem pengelolaan data yang cepat dan efisien. Pengelolaan data member yang bertambah seiring waktu, variasi paket latihan yang kompleks, serta pencatatan pengambilan paket oleh member seringkali menjadi kendala jika tidak dikelola dengan struktur data yang tepat. Oleh karena itu, diperlukan sebuah sistem berbasis CLI (*Command Line Interface*) yang mampu menangani operasi pencarian, penambahan, dan penghapusan data dengan performa optimal.

1.2 Tujuan

Berdasarkan latar belakang yang telah diuraikan, tujuan dalam laporan ini adalah:

1. Mengimplementasikan struktur data **Binary Search Tree (BST)** untuk manajemen data member agar proses pencarian (*searching*) lebih efisien.
2. Mengimplementasikan **Multi-Linked List (MLL)** untuk merepresentasikan data hierarkis antara Paket Latihan dan Gerakan.
3. Mengimplementasikan **Single Linked List** sebagai tabel relasi untuk menangani hubungan *Many-to-Many* antara Member dan Paket.

BAB II ANALISIS STRUKTUR DATA

Pada sistem ini, kami menerapkan tiga topik utama mata kuliah Struktur Data:

2.1 Binary Search Tree (BST) – Modul Data Member

Kami memilih BST untuk menyimpan data member karena karakteristiknya yang terurut.

1. **Implementasi:** Setiap *node* merepresentasikan satu member dengan *nama_member* sebagai *key*.
2. **Analisis Kompleksitas:**
 1. **Searching:** Pada kasus rata-rata (*average case*), pencarian member memiliki kompleksitas waktu **$O(\log n)$** , jauh lebih cepat dibandingkan Linked List biasa **$O(n)$** .
 2. **Sorting:** Dengan melakukan traversal *Inorder*, data otomatis tampil terurut secara alfabetis (A-Z).

2.2 Multi-Linked List (MLL) - Modul Paket Latihan

Struktur data ini digunakan untuk menangani hubungan "Satu Paket memiliki Banyak Gerakan" (*One-to-Many*).

- **Tipe MLL:** Variasi *List of List* (Parent-Child).
- **Parent Node:** Menyimpan data Paket (Nama Paket, Tingkat Kesulitan).
- **Child Node:** Menyimpan data Gerakan (Nama Gerakan, Reps/Set) yang terhubung ke Parent-nya.
- **Keunggulan:** Memungkinkan fleksibilitas tinggi di mana setiap paket bisa memiliki jumlah gerakan yang berbeda-beda secara dinamis.
- **Generation:** LLM menghasilkan respons berdasarkan konteks yang diberikan

2.3 Relasi (Associative List) - Modul Transaksi

Hubungan antara Member dan Paket adalah *Many-to-Many* (Satu member bisa ambil banyak paket, satu paket bisa diambil banyak member).

- **Solusi:** Kami menggunakan *Single Linked List* tambahan sebagai tabel perantara yang menyimpan pointer/identitas Member dan Paket yang saling berhubungan.

BAB III PEMBAGIAN TUGAS DAN KONTRIBUSI

Pembagian tugas dilakukan secara merata 33%:

Dimas Ramadhani: Modul Member (Binary Search Tree) - 33%

- Merancang struct Tree dan fungsi dasar (createNode, insertNode).
- Membuat logika **Deletion pada BST** yang kompleks (menangani kasus penghapusan *leaf node*, node dengan 1 anak, dan node dengan 2 anak menggunakan konsep *inorder successor*).
- Membuat fitur Traversal (inorder, preorder, postorder) untuk pelaporan data.
- Membuat fitur statistik data member (Rata-rata berat badan, Max/Min).

Abisar Fathir: Modul Latihan (Multi-Linked List) - 33%

- Merancang struct MLL (Parent & Child) dan fungsi dasar (createPaket, createGerakan).
- Membuat logika manipulasi list yang kompleks: **Insert Last Child** (menambah gerakan ke paket tertentu) dan **Delete Parent** (menghapus paket beserta seluruh gerakannya/ *cascade delete*).
- Membuat fitur update data paket dan statistik paket latihan (Paket dengan gerakan terbanyak/tersedikit).

Lutfi Shidqi Mardian: Modul Relasi & Integrasi (Main Program) - 33%

- Merancang struct Relasi dan fungsi penghubung (assignPaketToMember).
- Melakukan validasi logika bisnis (contoh: memastikan member yang akan di-*assign* benar-benar ada di Tree, dan paketnya ada di MLL).
- Menyusun **Menu Utama (main.cpp)** agar semua modul terintegrasi dengan antarmuka yang ramah pengguna.
- Menyusun laporan akhir dan dokumentasi teknis.

BAB IV IMPLEMENTASI ALGORITMA

Berikut adalah logika utama dari program kami dalam notasi algoritmik:

4.1 Algoritma Insert Member (Rekursif BST)

```
PROCEDURE insertMember(input/output root : adrMember, input newNode :  
adrMember)  
BEGIN  
  IF root = NULL THEN  
    root <- newNode  
  ELSE IF newNode.info.nama < root.info.nama THEN  
    // Masuk ke sub-pohon kiri jika nama lebih kecil (alfabetis)  
    insertMember(root.left, newNode)  
  ELSE IF newNode.info.nama > root.info.nama THEN  
    // Masuk ke sub-pohon kanan jika nama lebih besar  
    insertMember(root.right, newNode)  
  ELSE  
    OUTPUT "Member sudah terdaftar (Duplikat)"  
  ENDIF  
END PROCEDURE
```

4.2 Algoritma Menambah Gerakan ke Paket (Traversal MLL)

```
PROCEDURE addGerakan(input paket : adrPaket, input newGerakan : adrGerakan)
VAR temp : adrGerakan
BEGIN
  IF paket.first = NULL THEN
    // Jika paket belum punya gerakan, ini jadi yang pertama
    paket.first <- newGerakan
  ELSE
    // Traversal sampai elemen terakhir (Insert Last)
    temp <- paket.first
    WHILE temp.next != NULL DO
      temp <- temp.next
    ENDWHILE
    temp.next <- newGerakan
  ENDIF
END PROCEDURE
```

BAB V HASIL PENGUJIAN (SCREENSHOTS)

5.1 Tampilan Menu Utama

```
===== GYM =====  
1. Info member  
2. Info latihan  
3. Relasi Member-Latihan  
4. Keluar  
-----  
Pilih: █
```

Deskripsi: Screenshot menampilkan menu utama dengan 4 pilihan: Info Member, Info Latihan, Relasi Member-Latihan, dan Keluar.

5.2 Tampilan Menu Member

```
===== Member =====  
1. Tambah member  
2. Update progress member  
3. Lihat detail member  
4. Lihat semua member  
5. Hapus member  
6. Statistik Member  
7. Kembali  
-----  
Pilihan: █
```

Deskripsi: Menu manajemen member dengan 7 opsi termasuk statistik.

5.3 Tambah Member

```
===== Tambah Member =====  
Nama : Abi  
Target Berat Badan : 80  
Tinggi Badan : 185|
```

Deskripsi: Form input member dengan validasi.

5.4 Statistik Member

```
===== STATISTIK MEMBER =====  
  
=== STATISTIK MEMBER ===  
Total Member : 2 orang  
Rata-rata Target Berat : 80.00 kg  
  
Target Berat Tertinggi : Dimas (80.00 kg)  
Target Berat Terendah : Dimas (80.00 kg)  
  
Tekan Enter untuk kembali...
```

Deskripsi: Menampilkan total member, rata-rata, max, min, dan min target berat.

5.5 Traversal

```
===== LIHAT SEMUA MEMBER =====  
1. Inorder (Alfabetis)  
2. Preorder  
3. Postorder  
-----  
Pilihan: 1  
  
Daftar Member:  
- Abi (Target BB: 80.00 kg, TB: 185.00 cm)  
- Dimas (Target BB: 80.00 kg, TB: 170.00 cm)  
|
```

Deskripsi: Hasil traversal BST

5.6 Menu Latihan

```
===== MENU LATIHAN =====  
1. Tambah paket latihan  
2. Tambah gerakan ke paket  
3. Hapus paket latihan  
4. Hapus gerakan dari paket  
5. Update paket latihan  
6. Update gerakan  
7. Lihat semua paket  
8. Lihat detail paket  
9. Statistik Paket  
10. Kembali  
-----  
Pilihan: |
```

Deskripsi: Menu manajemen paket Latihan dengan 10 opsi.

5.7 Tambah Paket (Menu Kesulitan)

```
===== TAMBAH PAKET LATIHAN =====  
Nama Paket          : Push  
  
Pilih Kesulitan:  
1. Beginner  
2. Intermediate  
3. Advanced  
Pilihan (1-3)       : 1  
-----  
Paket Push (Beginner) berhasil ditambahkan.  
|
```

Deskripsi: Input paket dengan menu kesulitan (1-3)

5.8 Tambah Gerakan

```
===== TAMBAH GERAKAN KE PAKET =====  
Nama Paket      : Push  
Nama Gerakan    : BenchPress  
Berdasarkan waktu? (y/n): n  
Reps            : 10  
Sets            : 3
```

Deskripsi: Form tambah gerakan dengan pilihan waktu/reps.

5.9 Statistik Paket

```
===== STATISTIK PAKET =====  
  
=== STATISTIK PAKET LATIHAN ===  
Total Paket : 1 paket  
  
Paket dengan Gerakan Terbanyak : Push (2 gerakan)  
Paket dengan Gerakan Tersedikit: Push (2 gerakan)  
  
=== PAKET PER KESULITAN ===  
  
Paket dengan kesulitan "Beginner":  
1. Push (2 gerakan)  
  
Total: 1 paket  
  
Paket dengan kesulitan "Intermediate":  
(Tidak ada paket dengan kesulitan ini)  
  
Paket dengan kesulitan "Advanced":  
(Tidak ada paket dengan kesulitan ini)  
  
Tekan Enter untuk kembali...
```

Deskripsi: Statistik paket termasuk breakdown per kesulitan.

5.10 Menu Relasi

```
===== MENU RELASI =====  
1. Assign paket ke member  
2. Lihat paket member  
3. Hapus assignment  
4. Lihat semua assignment  
5. Kembali  
-----  
Pilihan:
```

Deskripsi: Menu manage assignment member-paket.

5.11 Assign Paket ke Member

```
===== ASSIGN PAKET KE MEMBER =====  
Nama Member      : Dimas  
Nama Paket       : Push  
-----  
Paket "Push (Beginner)" berhasil di-assign ke Dimas.  
Tekan Enter untuk kembali...|
```

Deskripsi: Form assignment dengan smart search dan validasi duplikat.

BAB VI SOURCE CODE LENGKAP

Berikut adalah implementasi kode program C++ yang telah kami buat:

6.1 Header File (gym.h)

```
#ifndef GYM_H
#define GYM_H
#include <iostream>
#include <iomanip>
#include <limits>
using namespace std;

// Tipe Pointer
typedef struct member* adrMember;
typedef struct jenisGerakan* adrGerakan;
typedef struct paketLatihan* adrPaket;
typedef struct relasi* adrRelasi;

// Struktur Data BST (Member)
struct infoMember {
    string nama_member;
    float target_berat;
    float tinggi_badan;
    float progress_berat[12];
};
struct member {
    infoMember info_member;
    adrMember left;
    adrMember right;
};

// Struktur Data MLL (Paket & Gerakan)
struct jenisGerakan {
```

```

    string nama_gerakan;
    bool isBerkasWaktu;
    int frekuensi;
    int sets;
    adrGerakan next;
};

struct paketLatihan {
    string nama_paket;
    string kesulitan;
    adrGerakan first;
    adrPaket next;
};

// Struktur Data Relasi
struct relasi {
    string nama_member;
    string nama_paket;
    string kesulitan_paket;
    adrRelasi next;
};

// Deklarasi Fungsi (Prototipe)
// --- Fungsi BST ---
void insertNode(adrMember &root, adrMember newNode);
adrMember createNode(infoMember info);
void deleteNode(adrMember root, string nama); // dan helper lainnya
void menuMember(adrMember &root);
// ... (dan fungsi traversal lainnya)

// --- Fungsi MLL ---
void insertPaket(adrPaket &head, adrPaket newPaket);
void insertGerakan(adrPaket paket, adrGerakan newGerakan);
void deletePaket(adrPaket &head, string nama, string kesulitan);

```

```
void menuLatihan(adrPaket &headPaket);

// --- Fungsi Relasi ---
void assignPaketToMember(adrRelasi &headRelasi, adrMember rootMember, adrPaket
headPaket);
void menuRelasi(adrRelasi &headRelasi, adrMember rootMember, adrPaket headPaket);

#endif
```

6.2 Implementasi (gym.cpp, relasi.cpp, main.cpp)

```
#include "gym.h"

int main() {
    adrMember root = nullptr;
    adrPaket headPaket = nullptr;
    adrRelasi headRelasi = nullptr;
    int pilihanMenu;

    do {
        system("cls");
        cout << "=== GYM MANAGEMENT SYSTEM ===\n";
        cout << "1. Info Member (BST)\n";
        cout << "2. Info Latihan (MLL)\n";
        cout << "3. Relasi Member-Latihan\n";
        cout << "4. Keluar\n";
        cout << "Pilih: "; cin >> pilihanMenu;

        switch (pilihanMenu) {
            case 1: menuMember(root); break;
            case 2: menuLatihan(headPaket); break;
            case 3: menuRelasi(headRelasi, root, headPaket); break;
        }
    } while (pilihanMenu != 4);
    return 0;
}
```


BAB VII LINK DEMO PROGRAM

Berikut adalah tautan video demonstrasi penggunaan program yang telah kami unggah:

- Platform: Youtube
- Link: <https://youtu.be/J464Y-MsMTw>

BAB VIII KESIMPULAN

Berdasarkan hasil perancangan dan pengujian, dapat disimpulkan bahwa:

1. Penggunaan **Binary Search Tree (BST)** sangat efektif untuk data member, memungkinkan pencarian data dengan cepat dan pengurutan data secara otomatis.
2. Struktur **Multi-Linked List** berhasil memodelkan paket latihan yang kompleks secara dinamis tanpa batasan array statis.
3. Sistem integrasi CLI berjalan dengan baik tanpa error (*bug-free*) pada operasi dasar CRUD (*Create, Read, Update, Delete*).