

Tugas Proceeding  
Modul 13 Struktur Data  
Multi Linked List



**Disusun Oleh:**

**Christian Felix Saliman Sugiono (2311104031)**

**S1SE0701**

**Dosen:**

**Yudha Islami Sulistya**

**Program Studi S1 Software Engineering**

**Fakultas Informatika**

**Telkom University**

**Purwokerto**

**2024**

**Tugas Pendahuluan Modul 8**

## STRUKTUR DATA - Ganjil 2024/2025

### "Multi Linked List"

#### A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 4 adalah Senin, 9 Oktober 2023 pukul 06.00 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan:  
**TP\_MOD\_[XX]\_NIM\_NAMA.pdf**

#### CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

**SELAMAT MENGERJAKAN^^**

## Latihan Modul 13

```
Latihan1.cpp

#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

struct Proyek {
    string namaProyek;
    int durasi;
};

struct Pegawai {
    string nama;
    string id;
    vector<Proyek> daftarProyek;
};

void tampilkanDataPegawai(const vector<Pegawai>& pegawaiList) {
    for (const auto& pegawai : pegawaiList) {
        cout << "Pegawai: " << pegawai.nama << " (ID: " << pegawai.id << ")\n";
        for (const auto& proyek : pegawai.daftarProyek) {
            cout << "  - Proyek: " << proyek.namaProyek << ", Durasi: " << proyek.durasi << " bulan\n";
        }
        cout << endl;
    }
}

int main() {
    vector<Pegawai> pegawaiList = {
        {"Andi", "P001", {}},
        {"Budi", "P002", {}},
        {"Citra", "P003", {}}
    };

    pegawaiList[0].daftarProyek.push_back({"Aplikasi Mobile", 12});
    pegawaiList[1].daftarProyek.push_back({"Sistem Akuntansi", 8});
    pegawaiList[2].daftarProyek.push_back({"E-commerce", 10});

    pegawaiList[0].daftarProyek.push_back({"Analisis Data", 6});

    auto& proyekAndi = pegawaiList[0].daftarProyek;
    proyekAndi.erase(
        remove_if(proyekAndi.begin(), proyekAndi.end(),
            [](const Proyek& p) { return p.namaProyek == "Aplikasi Mobile"; }),
        proyekAndi.end());

    tampilkanDataPegawai(pegawaiList);

    return 0;
}
```



Christian Felix  
x

Berikut ini adalah kodingan untuk latihan no 1 yaitu manajemen data pegawai dan proyek yang ditulis menggunakan bahasa C++. Kodingan ini menggunakan berbagai inputan extra seperti `<iostream>` untuk input/output, `<string>` untuk menangani string, `<vector>` untuk menyimpan daftar proyek atau pegawai secara dinamis, dan `<algorithm>` untuk manipulasi elemen dalam container seperti `remove_if`. Struktur `Proyek` digunakan untuk menyimpan nama proyek (string) dan durasi proyek (integer), sedangkan struktur `Pegawai` menyimpan data nama pegawai, ID, dan daftar proyek berupa `vector<Proyek>`. Fungsi `tampilkanDataPegawai` mencetak data pegawai dan proyek yang dikelolanya dengan loop. Program utama dimulai

dengan membuat daftar pegawai awal dalam vector dengan nama, ID, dan daftar proyek kosong. Proyek ditambahkan ke masing-masing pegawai menggunakan metode `push_back`. Misalnya, proyek seperti "Aplikasi Mobile" dengan durasi 12 bulan ditambahkan ke pegawai "Andi". Proyek tertentu juga dapat dihapus, seperti menghapus "Aplikasi Mobile" dari daftar proyek Andi menggunakan kombinasi `remove_if` dan `erase`. Setelah semua perubahan, fungsi `tampilkanDataPegawai` dipanggil untuk mencetak data pegawai dan proyek mereka, lalu program berakhir dengan `return 0` untuk menunjukkan eksekusi sukses. Program ini dirancang untuk mempermudah pengelolaan data pegawai dan proyek secara dinamis.

Untuk output dari kodingan kita adalah sebagai berikut ini:

```
Pegawai: Andi (ID: P001)
- Proyek: Analisis Data, Durasi: 6 bulan

Pegawai: Budi (ID: P002)
- Proyek: Sistem Akuntansi, Durasi: 8 bulan

Pegawai: Citra (ID: P003)
- Proyek: E-commerce, Durasi: 10 bulan
```

## latihan 2

```
Latihan2.cpp

#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

struct Buku {
    string judul;
    string tanggalPengembalian;
};

struct Anggota {
    string nama;
    string id;
    vector<Buku> daftarBuku;
};

void tampilkanDataAnggota(const vector<Anggota>& anggotaList) {
    for (const auto& anggota : anggotaList) {
        cout << "Anggota: " << anggota.nama << " (ID: " << anggota.id << ")\n";
        for (const auto& buku : anggota.daftarBuku) {
            cout << " - Buku: " << buku.judul << ", Pengembalian: " << buku.tanggalPengembalian << "\n";
        }
        cout << endl;
    }
}

int main() {
    vector<Anggota> anggotaList = {
        {"Rani", "A001", {}},
        {"Dito", "A002", {}},
        {"Vina", "A003", {}}
    };

    anggotaList[0].daftarBuku.push_back({"Pemrograman C++", "01/12/2024"});
    anggotaList[1].daftarBuku.push_back({"Algoritma Pemrograman", "15/12/2024"});

    anggotaList[0].daftarBuku.push_back({"Struktur Data", "10/12/2024"});

    // Hapus anggota Dito
    anggotaList.erase(
        remove_if(anggotaList.begin(), anggotaList.end(),
            [](const Anggota& a) { return a.id == "A002"; }),
        anggotaList.end());

    tampilkanDataAnggota(anggotaList);

    return 0;
}
```



Christian Felix  
x

Program ini merupakan sistem manajemen buku perpustakaan yang ditulis menggunakan bahasa pemrograman C++. Kodingan ini memanfaatkan inputan seperti `<iostream>` untuk input/output, `<string>` untuk menangani string, `<vector>` untuk menyimpan daftar buku atau anggota secara dinamis, dan `<algorithm>` untuk manipulasi data seperti penghapusan elemen. Struktur Buku digunakan untuk menyimpan judul buku dan tanggal pengembaliannya, sementara struktur Anggota menyimpan data nama, ID, dan daftar buku yang dipinjam berupa `vector<Buku>`. Fungsi `tampilkanDataAnggota` digunakan untuk mencetak data anggota dan daftar buku yang mereka pinjam. Di dalam main, program dimulai dengan membuat daftar

anggota perpustakaan awal yang berisi nama, ID, dan daftar buku kosong. Buku seperti "Pemrograman C++" dengan tanggal pengembalian "01/12/2024" ditambahkan ke anggota "Rani" menggunakan metode `push_back`. Program juga memungkinkan penghapusan anggota tertentu, seperti menghapus anggota dengan ID "A002" menggunakan kombinasi `remove_if` dan `erase`. Setelah semua perubahan dilakukan, data anggota dan buku yang dipinjam ditampilkan menggunakan fungsi `tampilkanDataAnggota`, dan program berakhir dengan `return 0` untuk menunjukkan eksekusi berhasil. Program ini dirancang untuk mempermudah pengelolaan data anggota dan buku perpustakaan.

Untuk output dari kodingan kita adalah sebagai berikut:

```
Anggota: Rani (ID: A001)
- Buku: Pemrograman C++, Pengembalian: 01/12/2024
- Buku: Struktur Data, Pengembalian: 10/12/2024
```