

**LAPORAN PRAKTIKUM**  
**POSTTEST 3**  
**ALGORITMA PEMROGRAMAN**  
**LANJUT**



**Disusun oleh:**

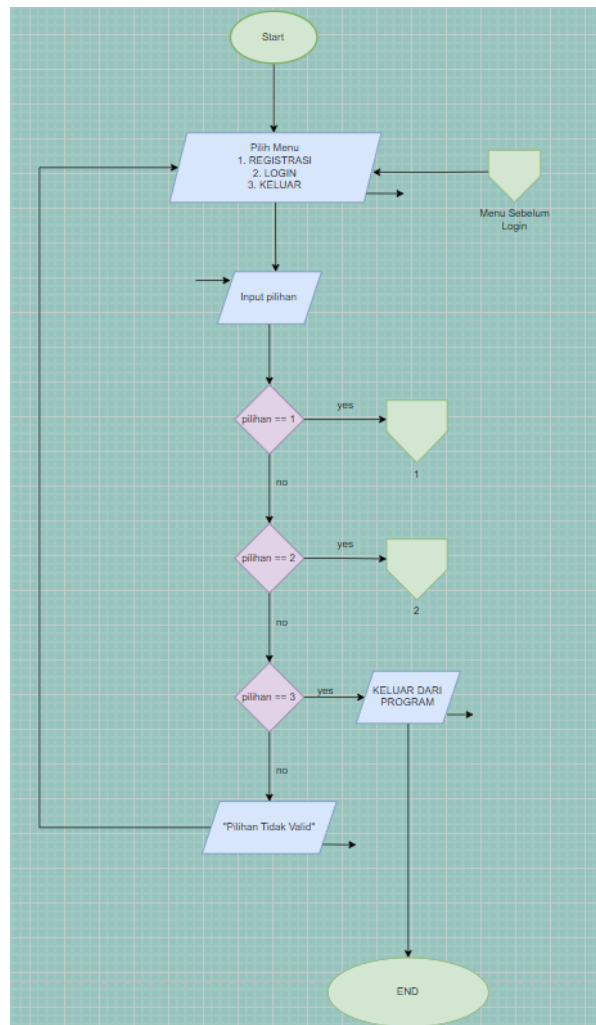
**Rusdiana**

**(2409106021)**

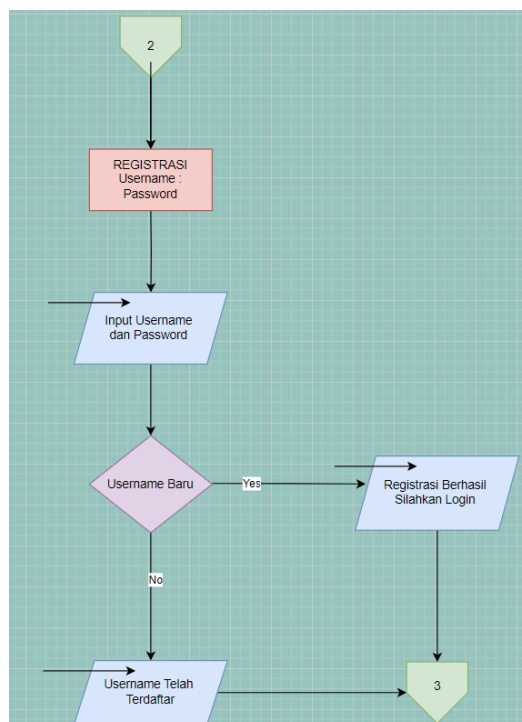
**Kelas A1'24**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

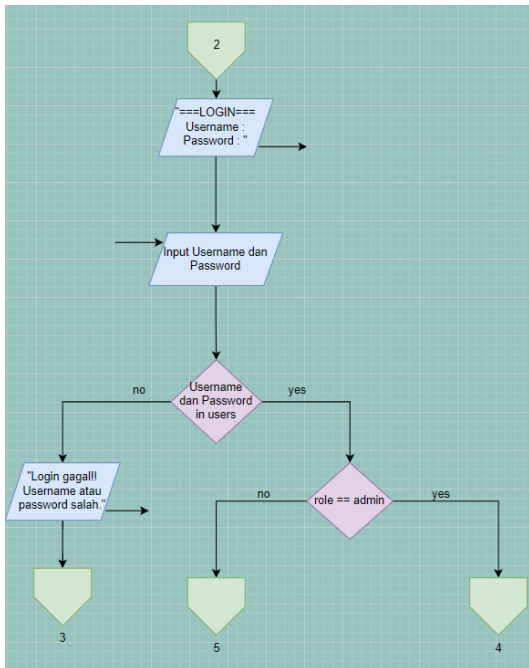
## 1. Flowchart



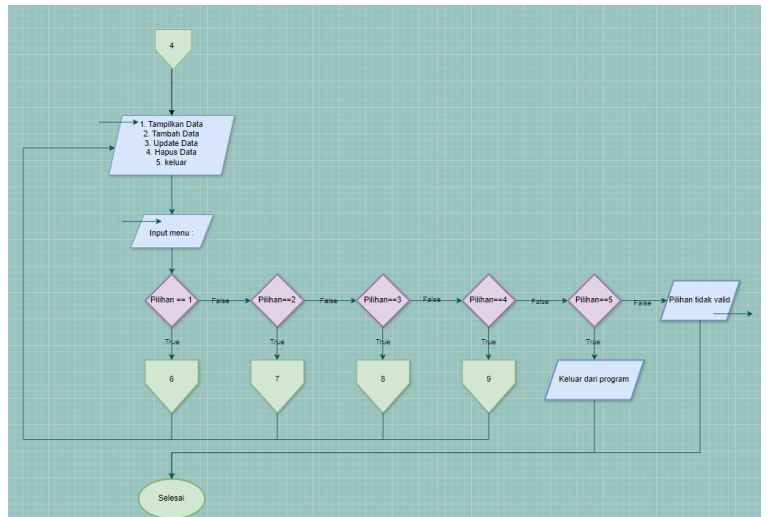
Gambar 1.1 Menu Sebelum Login



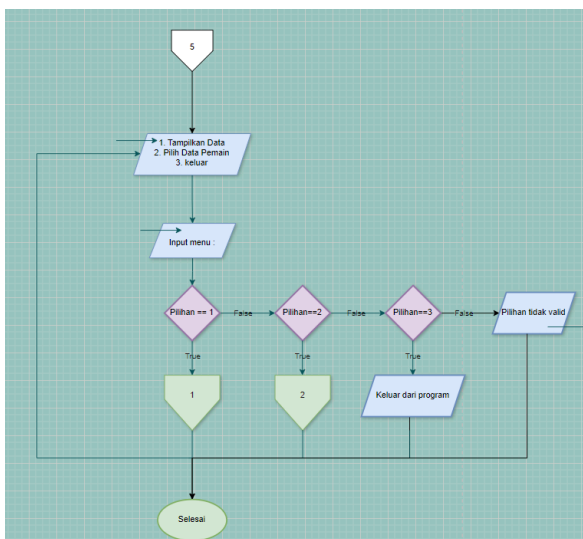
Gambar 1.2 Menu Registrasi



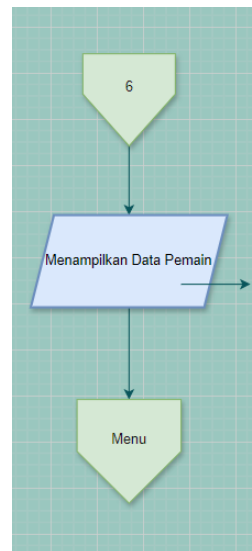
Gambar 1.3 Menu Login



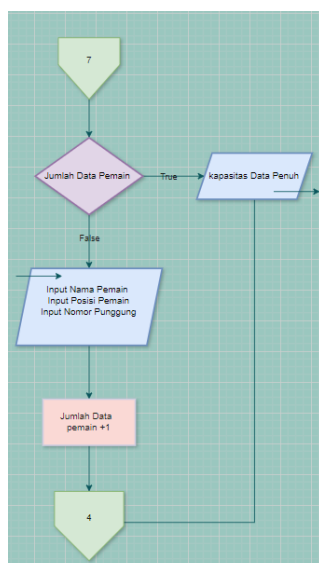
Gambar 1.4 Menu Admin



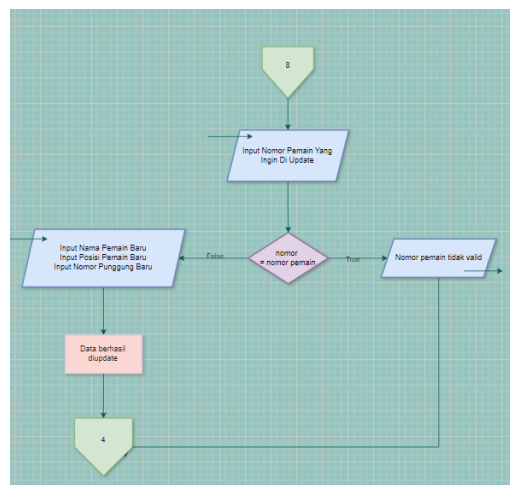
Gambar 1.5 Menu User



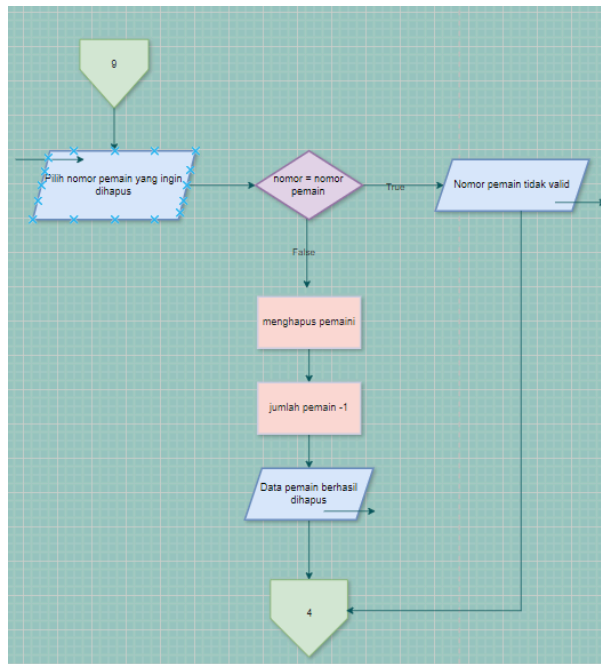
Gambar 1.6 Tampilan Data



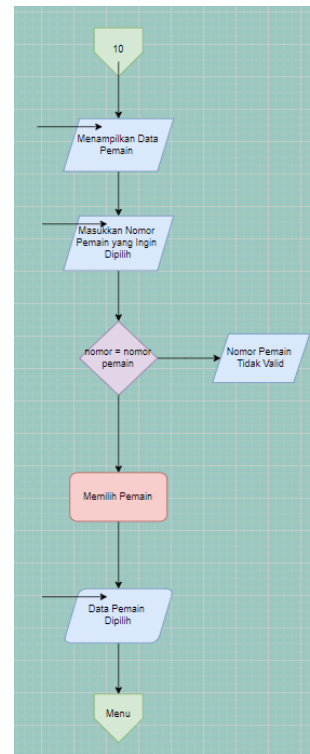
Gambar 1.7 Tambah Data



Gambar 1.8 Update Data



Gambar 1.9 Hapus Data



Gambar 1.10 Pilih Data Pemain

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini adalah sebuah manajemen “Data Pemain Bola” yang ditulis dalam bahasa C++. Pada program ini terdapat menu sebagai user baru dan menu sebagai admin, di mana menu user sendiri dapat mengakses tampilan, pilihan, serta keluar. Lalu untuk menu pada admin dapat mengakses bermacam - macam pilihan yang dapat dipilih memungkinkan pengguna admin untuk melakukan berbagai operasi CRUD (Create, Read, Update, Delete) pada data pemain bola, seperti menambahkan, menampilkan, mengupdate, dan menghapus data pemain. Selain itu, program ini juga dilengkapi dengan sistem login untuk mengamankan akses ke aplikasi.

### **2.2 Penjelasan Alur & Algoritma**

Program ini terdiri dari beberapa tahap utama, yang mencakup proses login, navigasi menu, serta operasi CRUD (Create, Read, Update, Delete) pada data produk. Berikut adalah alur kerja program :

#### **1. Login Pengguna**

- ☐ Pengguna harus memasukkan username dan 3 digit terakhir NIM untuk masuk ke sistem.
- ☐ Jika pengguna salah memasukkan data login sebanyak tiga kali, program akan berhenti.
- ☐ Jika login berhasil, pengguna diarahkan ke menu utama.

#### **2. Menampilkan Menu Utama**

- ☐ Program menampilkan opsi yang bisa dipilih oleh pengguna:
  1. Tampilkan Data Pemain
  2. Tambah Data Pemain
  3. Update Data Pemain
  4. Hapus Data Pemain
  5. Keluar
- ☐ Pengguna memasukkan nomor pilihan menu, yang akan menentukan langkah selanjutnya.

#### **3. CRUD Data Produk**

- **Tampilan Data Pemain:** Program mencetak daftar data pemain dalam format tabel agar lebih rapi dan mudah dibaca.
- **Tambah Data Pemain:** Pengguna menginputkan nama pemain, posisi pemain, dan nomor punggung pemain, yang kemudian disimpan dalam array dua dimensi.

- **Update Data Pemain:** Pengguna memilih Pemain berdasarkan nomor urutnya, lalu memasukkan data baru untuk mengganti informasi yang ada.
- **Hapus Data Pemain :** Pengguna memilih nomor pemain yang ingin dihapus, lalu program menggeser data dalam array agar tetap terstruktur.

#### 4. Keluar dari Program

- Jika pengguna memilih opsi Keluar, program akan menampilkan pesan perpisahan dan berhenti.

#### Penjelasan Fungsi dalam Program:

- **TampilkanData()** Menampilkan daftar pilihan menu utama.
- **TambahData()** Menambahkan data pemain baru ke dalam array jika kapasitas belum penuh.
- **TampilkanData()** Menampilkan semua data pemain yang telah disimpan dalam format tabel.
- **UbahData()** Mengubah informasi data pemain tertentu berdasarkan input pengguna.
- **HapusData()** Menghapus data pemain yang dipilih dan menyesuaikan susunan array.
- **Main()** Mengontrol seluruh proses eksekusi program dari login hingga interaksi dengan menu.

### 3. Source Code

#### A. Login

Pada Fitur ini menu yang digunakan untuk menvalidasi user yang ingin menggunakan aplikasi

#### Source Code :

```
const int max_kesempatan = 3; // Maksimal percobaan login
int kesempatan = 0; // Menghitung jumlah percobaan login

while (true) {
    if (loggedInUserIndex == -1) {
        // Menu sebelum login
        cout << "\n===== Selamat Datang =====> endl;
        cout << "1. Register" << endl;
        cout << "2. Login" << endl;
        cout << "3. Keluar" << endl;
        cout << "=====> endl;
        cout << "Pilih menu (1-3): ";
        cin >> pilihan;
        cin.ignore();

        if (pilihan == 1) {
            // Register User
            if (jumlah_user < 100) {
                cout << "\n===== Register User =====> endl;
                cout << "Masukkan Username: ";
                getline(cin, users[jumlah_user].username);
                cout << "Masukkan Password: ";
                getline(cin, users[jumlah_user].password);
                users[jumlah_user].isAdmin = false;
                jumlah_user++;
                cout << "Registrasi berhasil!\n";
            } else {
                cout << "Kapasitas user penuh!\n";
            }
        } else if (pilihan == 2) {
            // Login User
            if (kesempatan < max_kesempatan) {
                cout << "\n===== Login =====> endl;
                string input_username, input_password;
                cout << "Username: ";
                getline(cin, input_username);
                cout << "Password: ";
                getline(cin, input_password);

                bool login_berhasil = false;
                for (int i = 0; i < jumlah_user; i++) {
                    if (input_username == users[i].username && input_password == users[i].password) {
                        cout << "Login berhasil!\n";
                        loggedInUserIndex = i;
                        login_berhasil = true;
                        break;
                    }
                }
            }
        }
    }
}
```



```
    }  
    if (!login_berhasil) {  
        kesempatan++;  
        cout << "Username atau password salah!\n";  
        cout << "Sisa percobaan login: " << max_kesempatan - kesempatan << "\n" << endl;  
    }  
    } else {  
        cout << "Anda telah mencoba login sebanyak " << max_kesempatan << " kali. Program berhenti.\n";  
        return 0; // Keluar dari program  
    }  
} else if (pilihan == 3) {  
    // Keluar  
    cout << "Terima kasih telah menggunakan program ini!" << endl;  
    break;  
} else {  
    cout << "Pilihan tidak valid! Silakan coba lagi.\n";  
}
```

## B. Menu

Terdapat menu pada user yang dapat memilih fitur pada aplikasi

### Source Code

```
// Menu User Biasa
    cout << "\n===== Menu User Biasa =====" << endl;
    cout << "1. Tampilkan Data" << endl;
    cout << "2. Pilih Pemain" << endl;
    cout << "3. Keluar" << endl;
    cout << "===== " << endl;
    cout << "Pilih menu (1-3): ";
    cin >> pilihan;
    cin.ignore();
```

Pada menu Admin terdapat beberapa fitur pada aplikasi yang dapat di akses

### Source Code

```
// Main program
    while (true) {
        cout << "\n==== Manajemen Data Pemain Bola =====" << endl;
        cout << "1. Tampilkan Data" << endl;
        cout << "2. Tambah Data" << endl;
        cout << "3. Update Data" << endl;
        cout << "4. Hapus Data" << endl;
        cout << "5. Keluar" << endl;
        int pilihan;
        cout << "Pilih menu (1-5): ";
        cin >> pilihan;
```

## C. CRUD Manajemen Data Pemain Bola

Fitur CRUD dalam pemrograman ini mengacu pada empat operasi utama yaitu Create (membuat/menambahkan data baru), Read (Mengambil dan menampilkan data yang tersimpan), Update (Mengubah atau memperbarui data yang sudah ada.), Delet (Menghapus data dari Array.) yang digunakan untuk mengelola data produk dalam Array.

### Source Code ( Admin ):

```
if (pilihan == 1) {
    // Tampilkan Data
    if (users[loggedInUserIndex].jumlah_data == 0) {
        cout << "Tidak ada data pemain.\n";
    } else {
        cout << "\n===== Data Pemain Bola =====" << endl;
        cout << setw(5) << "No" << setw(20) << "Nama Pemain" << setw(20) << "Posisi" << setw(20) << "Nomor" << endl;
        cout << "Punggug" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < users[loggedInUserIndex].jumlah_data; i++) {
            cout << setw(5) << i + 1 << setw(20) << users[loggedInUserIndex].dataPemain[i].nama
                << setw(20) << users[loggedInUserIndex].dataPemain[i].posisi
                << setw(15) << users[loggedInUserIndex].dataPemain[i].nomorPunggug << endl;
        }
    }
}
```

```

    }
    cout << endl;
}
} else if (pilihan == 2) {
    // Tambah Data
    if (users[loggedInUserIndex].jumlah_data < 100) {
        cout << "\n===== Tambah Data Pemain =====" << endl;
        cout << "Masukkan Nama Pemain: ";
        getline(cin, users[loggedInUserIndex].dataPemain[users[loggedInUserIndex].jumlah_data].nama);
        cout << "Masukkan Posisi Pemain: ";
        getline(cin, users[loggedInUserIndex].dataPemain[users[loggedInUserIndex].jumlah_data].posisi);
        cout << "Masukkan Nomor Punggung: ";
        cin >> users[loggedInUserIndex].dataPemain[users[loggedInUserIndex].jumlah_data].nomorPunggung;
        cin.ignore();
        users[loggedInUserIndex].jumlah_data++;
        cout << "Data berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas data penuh!\n";
    }
} else if (pilihan == 3) {
    // Update Data
    if (users[loggedInUserIndex].jumlah_data > 0) {
        cout << "\n===== Update Data Pemain =====" << endl;
        cout << setw(5) << "No" << setw(20) << "Nama Pemain" << setw(20) << "Posisi" << setw(15) << "Nomor
Punggung" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < users[loggedInUserIndex].jumlah_data; i++) {
            cout << setw(5) << i + 1 << setw(20) << users[loggedInUserIndex].dataPemain[i].nama
                << setw(20) << users[loggedInUserIndex].dataPemain[i].posisi
                << setw(15) << users[loggedInUserIndex].dataPemain[i].nomorPunggung << endl;
        }
        cout << endl;

        int index;
        cout << "Masukkan nomor pemain yang ingin diupdate: ";
        cin >> index;
        cin.ignore();
        if (index >= 1 && index <= users[loggedInUserIndex].jumlah_data) {
            cout << "Masukkan Nama Pemain Baru: ";
            getline(cin, users[loggedInUserIndex].dataPemain[index - 1].nama);
            cout << "Masukkan Posisi Pemain Baru: ";
            getline(cin, users[loggedInUserIndex].dataPemain[index - 1].posisi);
            cout << "Masukkan Nomor Punggung Baru: ";
            cin >> users[loggedInUserIndex].dataPemain[index - 1].nomorPunggung;
            cin.ignore();
            cout << "Data berhasil diupdate!\n";
        } else {
            cout << "Nomor pemain tidak valid!\n";
        }
    } else {
        cout << "Tidak ada data untuk diupdate!\n";
    }
} else if (pilihan == 4) {
    // Hapus Data

```

```

        if (users[loggedInUserIndex].jumlah_data > 0) {
            cout << "\n===== Hapus Data Pemain =====" << endl;
            cout << setw(5) << "No" << setw(20) << "Nama Pemain" << setw(20) << "Posisi" << setw(15) << "Nomor Punggung" << endl;
            cout << "-----" << endl;
            for (int i = 0; i < users[loggedInUserIndex].jumlah_data; i++) {
                cout << setw(5) << i + 1 << setw(20) << users[loggedInUserIndex].dataPemain[i].nama
                    << setw(20) << users[loggedInUserIndex].dataPemain[i].posisi
                    << setw(15) << users[loggedInUserIndex].dataPemain[i].nomorPunggung << endl;
            }
            cout << endl;

            int index;
            cout << "Masukkan nomor pemain yang ingin dihapus: ";
            cin >> index;
            cin.ignore();
            if (index >= 1 && index <= users[loggedInUserIndex].jumlah_data) {
                for (int i = index - 1; i < users[loggedInUserIndex].jumlah_data - 1; i++) {
                    users[loggedInUserIndex].dataPemain[i] = users[loggedInUserIndex].dataPemain[i + 1];
                }
                users[loggedInUserIndex].jumlah_data--;
                cout << "Data berhasil dihapus!\n";
            } else {
                cout << "Nomor pemain tidak valid!\n";
            }
        } else {
            cout << "Tidak ada data untuk dihapus!\n";
        }
    } else if (pilihan == 5) {
        // Keluar
        loggedInUserIndex = -1;
        cout << "Logout berhasil!\n";
    } else {
        cout << "Pilihan tidak valid! Silakan coba lagi.\n";
    }
}

```

### Source Code ( User Baru )

```

// Menu User Biasa
cout << "\n===== Menu User Biasa =====" << endl;
cout << "1. Tampilkan Data" << endl;
cout << "2. Pilih Pemain" << endl;
cout << "3. Keluar" << endl;
cout << "===== " << endl;
cout << "Pilih menu (1-3): ";
cin >> pilihan;
cin.ignore();

if (pilihan == 1) {
    // Tampilkan Data
    if (users[loggedInUserIndex].jumlah_data == 0) {
        cout << "Tidak ada data pemain.\n";
    } else {
        cout << "\n===== Data Pemain Bola =====" << endl;
    }
}

```

```

        cout << setw(5) << "No" << setw(20) << "Nama Pemain" << setw(20) << "Posisi" << setw(20) << "Nomor
Punggung" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < users[loggedInUserIndex].jumlah_data; i++) {
            cout << setw(5) << i + 1 << setw(20) << users[loggedInUserIndex].dataPemain[i].nama
                << setw(20) << users[loggedInUserIndex].dataPemain[i].posisi
                << setw(15) << users[loggedInUserIndex].dataPemain[i].nomorPunggung << endl;
        }
        cout << endl;
    }
} else if (pilihan == 2) {
    // Pilih Pemain
    if (users[loggedInUserIndex].jumlah_data > 0) {
        cout << "\n===== Pilih Pemain =====" << endl;
        cout << setw(5) << "No" << setw(20) << "Nama Pemain" << setw(20) << "Posisi" << setw(15) << "Nomor
Punggung" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < users[loggedInUserIndex].jumlah_data; i++) {
            cout << setw(5) << i + 1 << setw(20) << users[loggedInUserIndex].dataPemain[i].nama
                << setw(20) << users[loggedInUserIndex].dataPemain[i].posisi
                << setw(15) << users[loggedInUserIndex].dataPemain[i].nomorPunggung << endl;
        }
        cout << endl;

        int index;
        cout << "Masukkan nomor pemain yang ingin dipilih: ";
        cin >> index;
        cin.ignore();
        if (index >= 1 && index <= users[loggedInUserIndex].jumlah_data) {
            cout << "Anda memilih pemain: " << users[loggedInUserIndex].dataPemain[index - 1].nama << endl;
        } else {
            cout << "Nomor pemain tidak valid!\n";
        }
    } else {
        cout << "Tidak ada data pemain untuk dipilih!\n";
    }
} else if (pilihan == 3) {
    // Keluar
    loggedInUserIndex = -1;
    cout << "Logout berhasil!\n";
} else {
    cout << "Pilihan tidak valid! Silakan coba lagi.\n";
}
}
}
}

return 0;
}

```

## 4. Hasil Output

### 1. Menu Admin

```
=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login                                     |
| 2. Register                                |
| 3. Keluar                                  |
=====
Pilihan: 1

===== LOGIN =====
Username: nana
Password: 12

Username atau password salah!
Percobaan tersisa: 2

===== LOGIN =====
Username: nana
Password: 12

Username atau password salah!
Percobaan tersisa: 1

===== LOGIN =====
Username: nana
Password: 12

Username atau password salah!
Percobaan tersisa: 0

Anda telah 3x salah login. Program berhenti.
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL\PostTest\PostTest-3>
```

Gambar 4.1.1 Kondisi Gagal Login

```
=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login                                     |
| 2. Register                                |
| 3. Keluar                                  |
=====
Pilihan: 1

===== LOGIN =====
Username: Rusdiana
Password: 2409106021

Login berhasil!

===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
```

Gambar 4.1.2 Tampilan Login Admin

```
===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 2

===== TAMBAH PEMAIN =====
Nama: Neymar Jr
Posisi: Keeper
Nomor Punggung: 11

Data pemain berhasil ditambahkan!
```

Gambar 4.1.3 Menu Tambah Pemain

```
===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 1

===== DATA PEMAIN =====
No    Nama                Posisi              Nomor
-----
1     Neymar Jr           Keeper              11
```

Gambar 4.1.4 Menu Tampilan Data

```
Pilihan: 3

===== EDIT PEMAIN =====
No  Nama              Posisi      Nomor
-----
1   Neymar Jr         Keeper      11

Pilih nomor pemain yang akan diedit: 1
Nama [Neymar Jr]: Messi
Posisi [Keeper]: Gelandang
Nomor Punggung [11]: 12

Data pemain berhasil diupdate!

===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 1

===== DATA PEMAIN =====
No  Nama              Posisi      Nomor
-----
1   Messi              Gelandang   12
```

Gambar 4.1.5 Menu Edit Pemain

```
===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 4

===== HAPUS PEMAIN =====
No  Nama              Posisi      Nomor
-----
1   Neymar             Bekk        11
2   Pedri              Keeper      10

Pilih nomor pemain yang akan dihapus: 1

Data pemain berhasil dihapus!

===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 1

===== DATA PEMAIN =====
No  Nama              Posisi      Nomor
-----
1   Pedri              Keeper      10
```

Gambar 4.1.6 Menu Hapus Pemain

```
===== MENU ADMIN =====
1. Tampilkan Data Pemain
2. Tambah Pemain
3. Edit Pemain
4. Hapus Pemain
5. Logout
=====
Pilihan: 5

Logout berhasil!

=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login |
| 2. Register |
| 3. Keluar |
=====
Pilihan: 3

Terima kasih telah menggunakan program ini!
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL\PostTest\PostTest-3>
```

Gambar 4.1.7 Menu Logout

## 2. Menu User Baru

```
=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login |
| 2. Register |
| 3. Keluar |
=====
Pilihan: 2

===== REGISTER =====
Username: nana
Password: 12

Registrasi berhasil!
```

Gambar 4.2.1 Registrasi Username Baru

```
=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login |
| 2. Register |
| 3. Keluar |
=====
Pilihan: 1

===== LOGIN =====
Username: nana
Password: 12

Login berhasil!

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: █
```

Gambar 4.2.2 Login Sebagai User Baru

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: 1

===== DATA PEMAIN =====
No    Nama             Posisi             Nomor
-----
1     Pedri              Keeper            12
2     Neymar             Gelandang         11

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: 

```

Gambar 4.2.3 Tampilan Lihat Data Pemain

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: 2

Masukkan nama pemain: Neymar

===== HASIL PENCARIAN =====
No    Nama             Posisi             Nomor
-----
2     Neymar             Gelandang         11

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: 

```

Gambar 4.2.3 Menu Cari Pemain

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Logout
=====
Pilihan: 3

Logout berhasil!

=====
|          SISTEM MANAJEMEN PEMAIN BOLA          |
=====
| 1. Login                                     |
| 2. Register                                |
| 3. Keluar                                   |
=====
Pilihan: 3

Terima kasih telah menggunakan program ini!
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL\PostTest\PostTest-3> 

```

Gambar 4.2.4 Menu Logout



## 4. Langkah-Langkah GIT

### 1. Git add

Git add adalah perintah Git yang digunakan untuk menambahkan perubahan pada file ke staging area sebelum dilakukan commit. Perintah ini memungkinkan Git mengetahui file mana saja yang akan dimasukkan dalam commit berikutnya

```
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> git add .  
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> █
```

Gambar 5. 1 Git\_add

### 2. Git Commit

Git commit adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository. Setiap commit akan memiliki hash unik, pesan commit, dan menyimpan snapshot dari perubahan yang dilakukan.

```
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> git add .  
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> git commit -m "finish post test 3"  
[main 6e64299] finish post test 3  
4 files changed, 178 insertions(+), 100 deletions(-)  
create mode 100644 Kelas/Pertemuan-2/pertemuan-2.cpp  
rename Kelas/{pertemuan_2 => Pertemuan-2}/pertemuan-2.exe (100%)  
create mode 100644 Kelas/Pertemuan-3/2409106021-Rusdiana-PostTest-3  
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> █
```

Gambar 5. 2 Git\_Commit

### 3. Git Push

Git push adalah perintah Git yang digunakan untuk mengirim (mengunggah) perubahan dari repository lokal ke repository remote (seperti GitHub, GitLab, atau Bitbucket). Perintah ini memastikan bahwa perubahan yang sudah dikomit di lokal tersedia di repository jarak jauh sehingga bisa diakses oleh orang lain atau untuk cadangan.

```
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> git push origin main --force  
Enumerating objects: 24, done.  
Counting objects: 100% (24/24), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (15/15), done.  
Writing objects: 100% (17/17), 1.85 MiB | 349.00 KiB/s, done.  
Total 17 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.  
remote: This repository moved. Please use the new location:  
remote: https://github.com/RusdianaNana/praktikum-apl.git  
To https://github.com/RusdianaNana/Praktikum-Apl.git  
49865d8..6e64299 main -> main  
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL> █
```

Gambar 5. 3 Git\_Push

