

LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN
LANJUT



Disusun oleh:

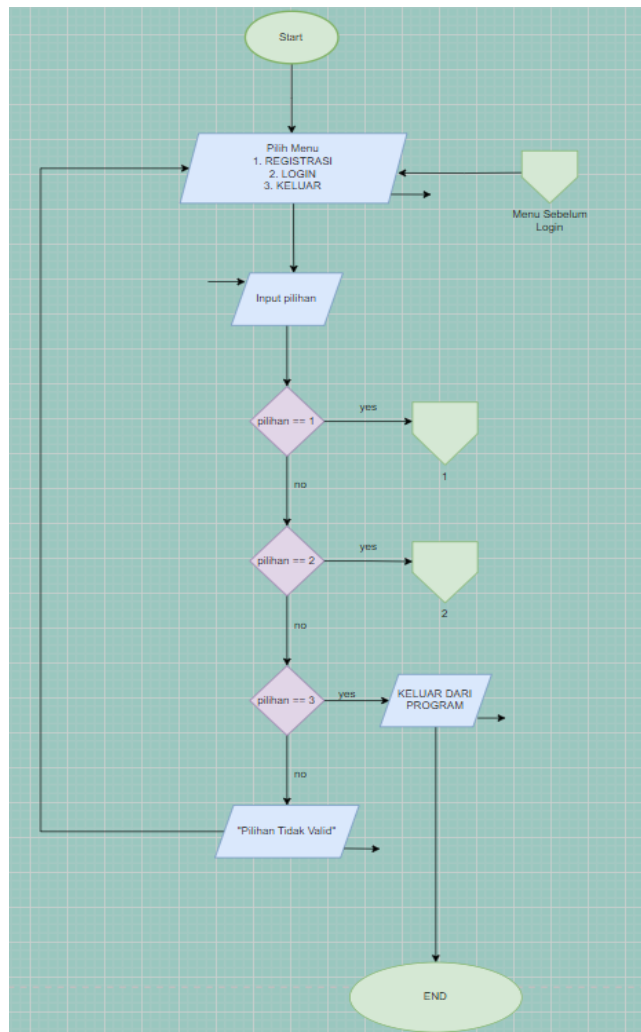
Rusdiana

(2409106021)

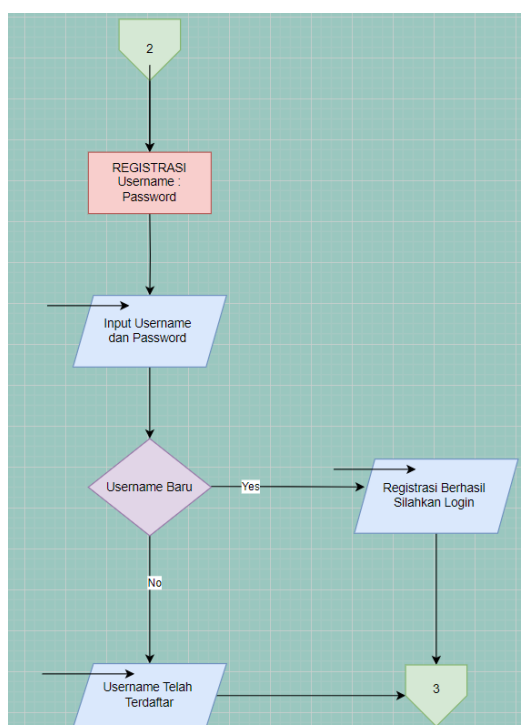
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

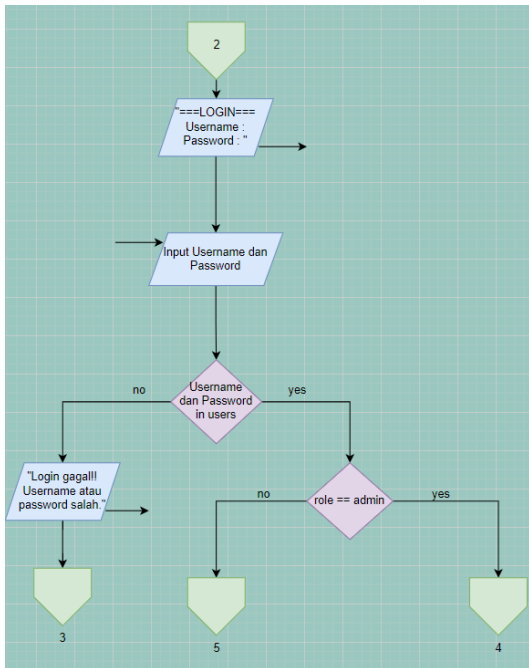
1. Flowchart



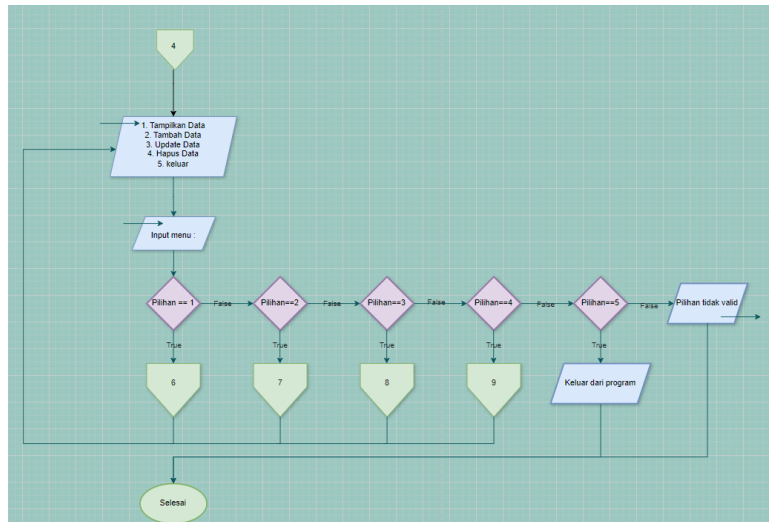
Gambar 1.1 Menu Sebelum Login



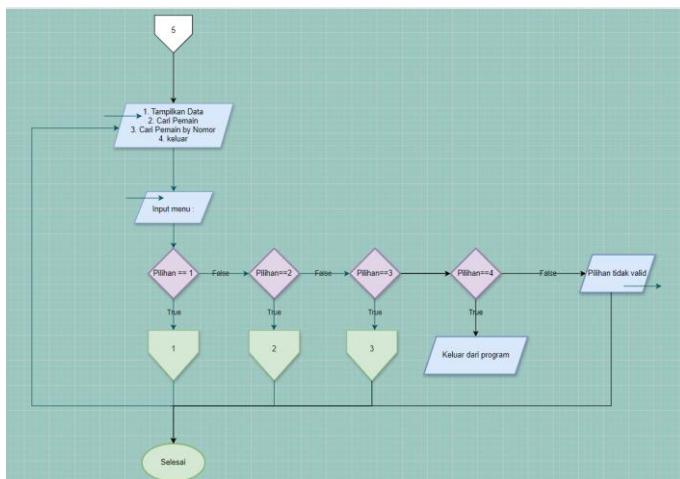
Gambar 1.2 Menu Registrasi



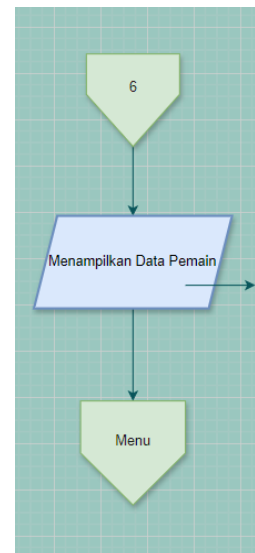
Gambar 1.3 Menu Login



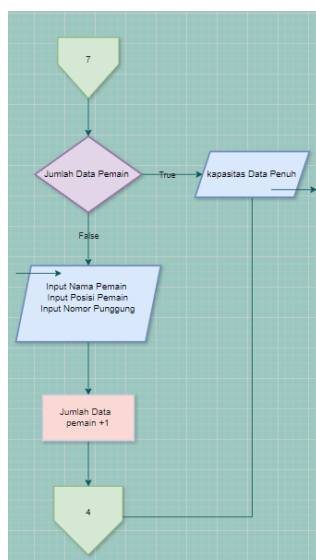
Gambar 1.4 Menu Admin



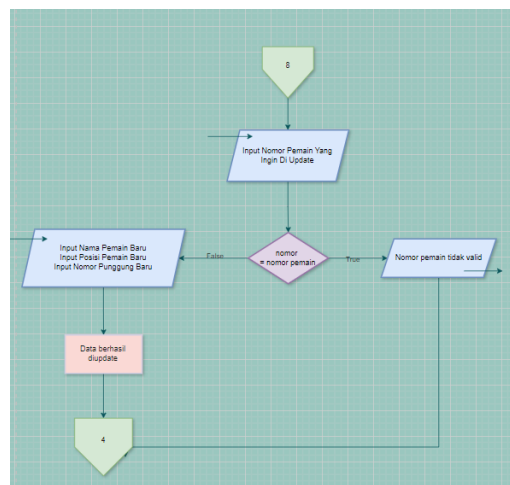
Gambar 1.5 Menu User



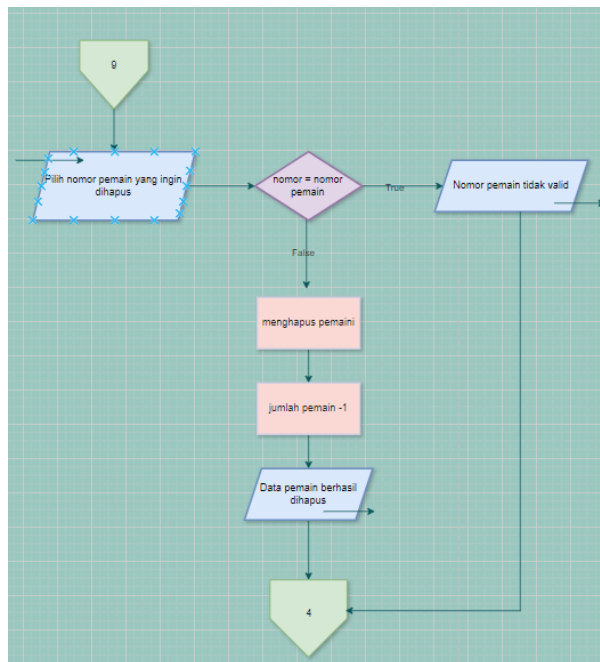
Gambar 1.6 Tampilan Data



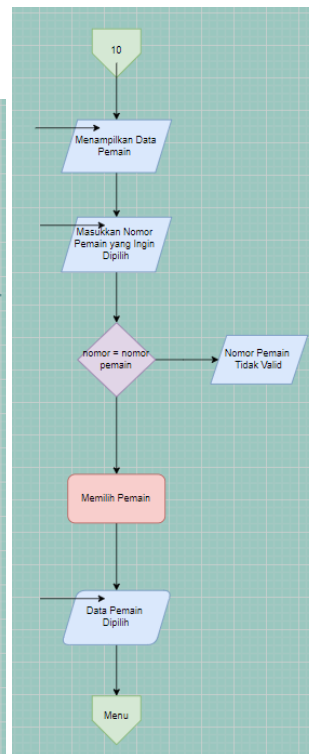
Gambar 1.7 Tambah Data



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 :

```
void login(int &userLogin, int &percobaanLogin) {
    const int MAX_PERCobaan = 3;

    while (percobaanLogin < MAX_PERCobaan) {
        cout << "\n===== LOGIN =====\n";
        string username, password;
        cout << "Username: ";
        getline(cin, username);
        cout << "Password: ";
        getline(cin, password);

        bool loginSukses = false;
        for (int i = 0; i < jumlahUser; i++) {
            if (username == users[i].username && password == users[i].password) {
                userLogin = i;
                loginSukses = true;
                percobaanLogin = 0;
                cout << "\nLogin berhasil!\n";
                break;
            }
        }

        if (!loginSukses) {
            percobaanLogin++;
            cout << "\nUsername atau password salah!\n";
            cout << "Percobaan tersisa: " << MAX_PERCobaan - percobaanLogin << endl;

            if (percobaanLogin >= MAX_PERCobaan) {
                cout << "\nAnda telah 3x salah login. Program berhenti.\n";
                exit(0);
            }
        } else {
            break;
        }
    }
}
```


B. Registrasi

```
void registerUser() {
    if (jumlahUser < 100) {
        cout << "\n===== REGISTER =====\n";
        User baru;
        cout << "Username: ";
        getline(cin, baru.username);
        cout << "Password: ";
        getline(cin, baru.password);
        baru.isAdmin = 0;
        users[jumlahUser] = baru;
        jumlahUser++;
        cout << "\nRegistrasi berhasil!\n";
    } else {
        cout << "\nKapasitas user penuh!\n";
    }
}

void tampilkanMenuAdmin(int &userLogin) {
    cout << "\n===== MENU ADMIN =====\n";
    cout << "1. Tampilkan Data Pemain\n";
    cout << "2. Tambah Pemain\n";
    cout << "3. Tambah Pemain (Reference)\n"; // Opsi baru
    cout << "4. Edit Pemain\n";
    cout << "5. Edit Pemain (Pointer)\n"; // Opsi baru
    cout << "6. Hapus Pemain\n";
    cout << "7. Cari Pemain by Nomor (Pointer)\n"; // Opsi baru
    cout << "8. Logout\n";
    cout << "=====\n";
    cout << "Pilihan: ";

    int pilihan;
    cin >> pilihan;
    cin.ignore();

    switch (pilihan) {
        case 1:
            tampilkanDataPemain();
            break;
        case 2:
            tambahPemain();
            break;
        case 3: {
            Pemain p;
            tambahPemainDenganReference(p); // Menggunakan fungsi dengan reference
            break;
        }
        case 4:
            editPemain();
            break;
        case 5: {
            tampilkanDataPemain();
```

```

        int nomor;
        cout << "\nPilih nomor pemain yang akan diedit: ";
        cin >> nomor;
        cin.ignore();

        if (nomor >= 1 && nomor <= jumlahPemain) {
            editPemainDenganPointer(&dataPemain[nomor-1]); // Menggunakan fungsi dengan
pointer
            cout << "\nData pemain setelah edit:\n";
            tampilkanPemainPointer(&dataPemain[nomor-1]); // Menggunakan fungsi dengan
pointer
        } else {
            cout << "\nNomor tidak valid!\n";
        }
        break;
    }
    case 6:
        hapusPemain();
        break;
    case 7: {
        int nomor;
        cout << "\nMasukkan nomor punggung pemain: ";
        cin >> nomor;
        cin.ignore();

        Pemain* ptrPemain = cariPemainByNomor(nomor); // Menggunakan fungsi yang
mengembalikan pointer
        if (ptrPemain != nullptr) {
            cout << "\nPemain ditemukan:\n";
            tampilkanPemainPointer(ptrPemain); // Menggunakan fungsi dengan pointer
        } else {
            cout << "\nPemain dengan nomor " << nomor << " tidak ditemukan!\n";
        }
        break;
    }
    case 8:
        userLogin = -1;
        cout << "\nLogout berhasil!\n";
        break;
    default:
        cout << "\nPilihan tidak valid!\n";
    }
}

```

B. Menu

Terdapat menu pada user yang dapat memilih fitur pada aplikasi

Source Code

```
cout << "\n===== MENU USER =====\n";
cout << "1. Lihat Data Pemain\n";
cout << "2. Cari Pemain\n";
cout << "3. Cari Pemain by Nomor (Pointer)\n"; // Opsi baru
cout << "4. Logout\n";
cout << "=====\n";
cout << "Pilihan: ";
```

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

Source Code

```
cout << "\n===== MENU ADMIN =====\n";
cout << "1. Tampilkan Data Pemain\n";
cout << "2. Tambah Pemain\n";
cout << "3. Tambah Pemain (Reference)\n"; // Opsi baru
cout << "4. Edit Pemain\n";
cout << "5. Edit Pemain (Pointer)\n"; // Opsi baru
cout << "6. Hapus Pemain\n";
cout << "7. Cari Pemain by Nomor (Pointer)\n"; // Opsi baru
cout << "8. Logout\n";
cout << "=====\n";
cout << "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):

```
int pilihan;
cin >> pilihan;
cin.ignore();

switch (pilihan) {
    case 1:
        tampilkanDataPemain();
        break;
    case 2:
        tambahPemain();
        break;
    case 3: {
        Pemain p;
        tambahPemainDenganReference(p); // Menggunakan fungsi dengan reference
```

```

        break;
    }
    case 4:
        editPemain();
        break;
    case 5: {
        tampilkanDataPemain();
        int nomor;
        cout << "\nPilih nomor pemain yang akan diedit: ";
        cin >> nomor;
        cin.ignore();

        if (nomor >= 1 && nomor <= jumlahPemain) {
            editPemainDenganPointer(&dataPemain[nomor-1]); // Menggunakan fungsi dengan
pointer
            cout << "\nData pemain setelah edit:\n";
            tampilkanPemainPointer(&dataPemain[nomor-1]); // Menggunakan fungsi dengan
pointer
        } else {
            cout << "\nNomor tidak valid!\n";
        }
        break;
    }
    case 6:
        hapusPemain();
        break;
    case 7: {
        int nomor;
        cout << "\nMasukkan nomor punggung pemain: ";
        cin >> nomor;
        cin.ignore();

        Pemain* ptrPemain = cariPemainByNomor(nomor); // Menggunakan fungsi yang
mengembalikan pointer
        if (ptrPemain != nullptr) {
            cout << "\nPemain ditemukan:\n";
            tampilkanPemainPointer(ptrPemain); // Menggunakan fungsi dengan pointer
        } else {
            cout << "\nPemain dengan nomor " << nomor << " tidak ditemukan!\n";
        }
        break;
    }
    case 8:
        userLogin = -1;
        cout << "\nLogout berhasil!\n";
        break;
    default:
        cout << "\nPilihan tidak valid!\n";
}
}

```

Source Code (User Baru)

```
void tampilkanMenuUser(int &userLogin) {
    cout << "\n===== MENU USER =====\n";
    cout << "1. Lihat Data Pemain\n";
    cout << "2. Cari Pemain\n";
    cout << "3. Cari Pemain by Nomor (Pointer)\n"; // Opsi baru
    cout << "4. Logout\n";
    cout << "=====\n";
    cout << "Pilihan: ";

    int pilihan;
    cin >> pilihan;
    cin.ignore();

    switch (pilihan) {
        case 1:
            tampilkanDataPemain();
            break;
        case 2:
            cariPemain();
            break;
        case 3: {
            int nomor;
            cout << "\nMasukkan nomor punggung pemain: ";
            cin >> nomor;
            cin.ignore();

            Pemain* ptrPemain = cariPemainByNomor(nomor); // Menggunakan fungsi yang
mengembalikan pointer
            if (ptrPemain != nullptr) {
                cout << "\nPemain ditemukan:\n";
                tampilkanPemainPointer(ptrPemain); // Menggunakan fungsi dengan pointer
            } else {
                cout << "\nPemain dengan nomor " << nomor << " tidak ditemukan!\n";
            }
            break;
        }
        case 4:
            userLogin = -1;
            cout << "\nLogout berhasil!\n";
            break;
        default:
            cout << "\nPilihan tidak valid!\n";
    }
}
```

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: 0211

Login berhasil!

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Cari Pemain by Nomor (Pointer)
4. Logout
=====
Pilihan:

```

Gambar 4.2.2 Login Sebagai User Baru

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Cari Pemain by Nomor (Pointer)
4. Logout
=====
Pilihan: 1

===== DATA PEMAIN =====
No  Nama          Posisi          Nomor
-----
1   Neymar        Gelandang      11

```

Gambar 4.2.3 Tampilan Lihat Data Pemain

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Cari Pemain by Nomor (Pointer)
4. Logout
=====
Pilihan: 2

Masukkan nama pemain: Neymar

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

```

Gambar 4.2.3 Menu Cari Pemain

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Cari Pemain by Nomor (Pointer)
4. Logout
=====
Pilihan: 3

Masukkan nomor punggung pemain: 21

Pemain dengan nomor 21 tidak ditemukan!

```

Logout

```

===== MENU USER =====
1. Lihat Data Pemain
2. Cari Pemain
3. Cari Pemain by Nomor (Pointer)
4. Logout
=====
Pilihan: 4

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-5>

```

Gambar 4.2.4 Cari Pemain by Nomor

Gambar 4.2.5

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 commit -m "finish post test 5"  
[main d7e9972] finish post test 5  
4 files changed, 480 insertions(+)  
create mode 100644 Kelas/Pertemuan-5/2409106021_Rusdiana.cpp  
create mode 100644 Kelas/Pertemuan-5/2409106021_Rusdiana.exe  
create mode 100644 PostTest/PostTest-5/2409106021_Rusdiana_PostTest5.cpp  
create mode 100644 PostTest/PostTest-5/2409106021_Rusdiana_PostTest5.exe  
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: 13, done.  
Counting objects: 100% (13/13), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (10/10), 944.89 KiB | 2.82 MiB/s, done.  
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), completed with 1 local object.  
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  
13ca614..d7e9972 main -> main  
PS C:\Users\ASUS\OneDrive\PDF\PRAKTIKUM-APL>
```

Gambar 5. 3 Git_Push

