

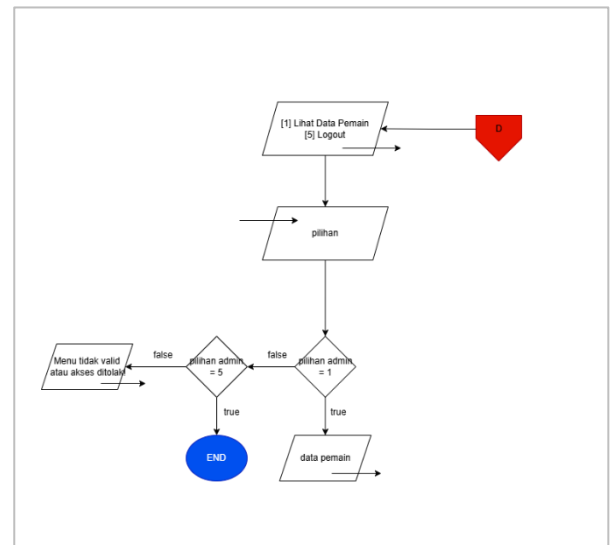
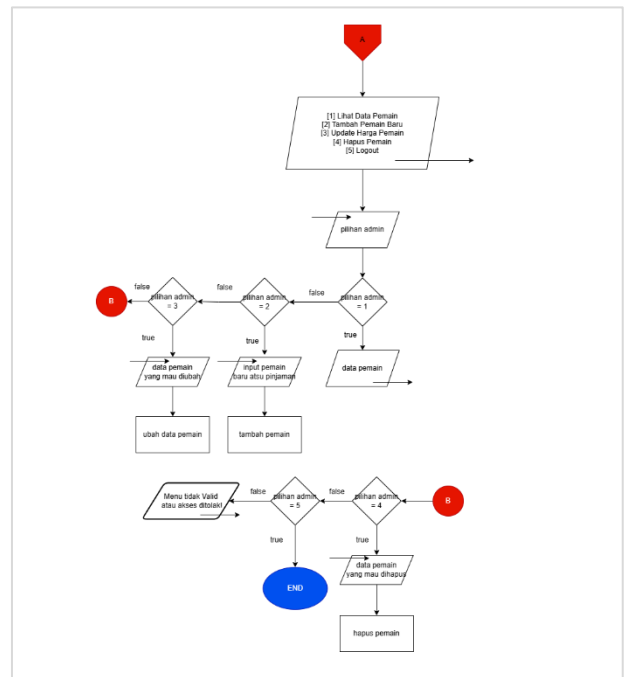
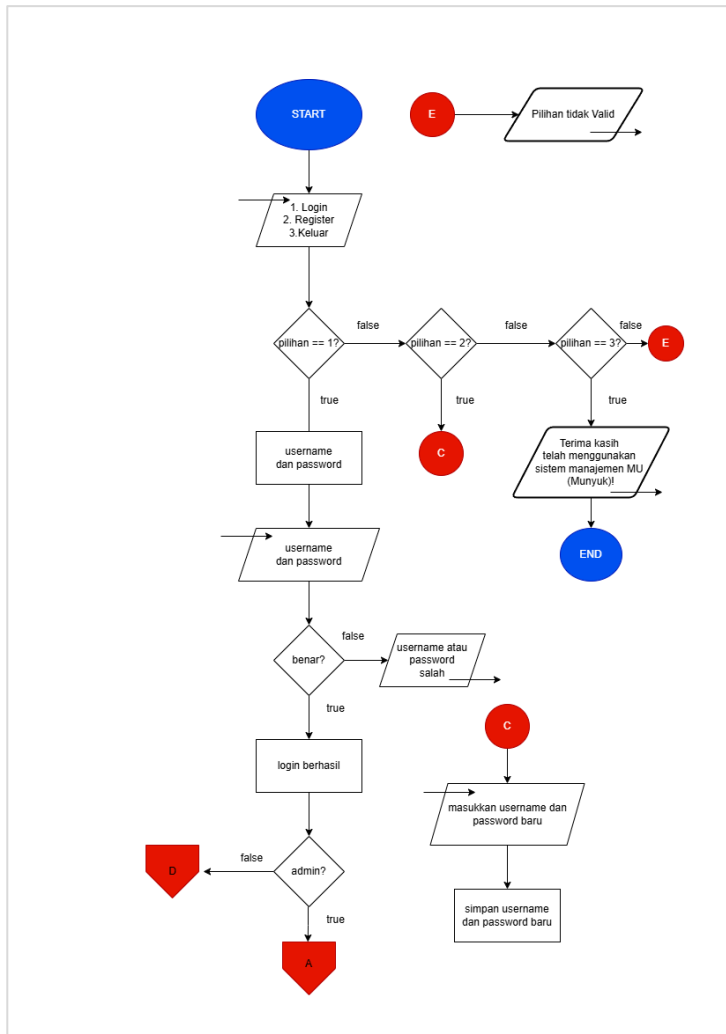
LAPORAN PRAKTIKUM
POSTTEST (5)
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Ferly Ahmad Nabil (2509106024)
Kelas (A2 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Deskripsi Singkat Program

A. Multiuser (Admin dan User Biasa)

- Admin memiliki hak akses penuh untuk melihat, menambah, mengubah, dan menghapus data pemain.
- User biasa hanya dapat melihat daftar pemain.
- Sistem login membedakan kedua peran ini berdasarkan data akun yang tersimpan.

B. Registrasi Akun Baru

- Pengguna dapat membuat akun baru melalui menu register.
- Akun baru yang terdaftar otomatis berstatus sebagai *user biasa*.

C. Autentikasi Login

- Pengguna harus login terlebih dahulu sebelum masuk ke sistem.
- Jika username atau password salah, sistem akan menolak dan meminta pengguna mencoba lagi.

D. CRUD (Create, Read, Update, Delete)

- **Create** → Menambahkan pemain baru dengan data lengkap seperti nama, negara asal, klub sebelumnya, rating, dan harga.
- **Read** → Menampilkan seluruh daftar pemain yang tersimpan.
- **Update** → Mengubah harga pemain (misalnya saat performa berubah).
- **Delete** → Menghapus pemain yang dijual atau dipinjam ke klub lain.

E. Validasi Input

- Pengecekan dilakukan dengan perulangan dan kondisi if.

F. Clear Terminal (Tampilan Bersih)

- Menggunakan *import os* dan perintah `os.system('cls')` agar tampilan menu selalu bersih dan rapi setiap kali berpindah halaman.

3. Source Code

A. Program Sistem Managemen Tim Bola

Source Code:

```
import os

akun = [["admin", "024", "admin"], ["user", "420", "user"]]
pemain = [
    ["Bruno Fernandes", "Portugal", "Sporting Lisbon", 90, 80000000],
    ["Casemiro", "Brazil", "Real Madrid", 88, 70000000],
    ["Benjamin Sesko", "Slovenia", "Salzburg", 92, 85000000],
    ["Kobbie Mainoo", "Inggris", "Akademi MU", 81, 2500000],
    ["Senne Lammes", "Belgia", "Royal Antwerp", 94, 18000000],
]

login = False
role = ""

while not login:
    os.system('cls')
    print("=== MANAJEMEN TIM MANCHESTER UNITED / MUNYUK ===")
    print("[1] Login")
    print("[2] Register")
    print("[3] Keluar")
    pilih = input("Pilih menu: ")

    if pilih == "1":
        user = input("Username: ")
        pw = input("Password: ")
```

```

        i = 0
        while i < len(akun):
            if akun[i][0] == user and akun[i][1] == pw:
                login = True
                role = akun[i][2]
                break
            i = i + 1

        if login == False:
            input("Username / Password salah! Tekan Enter untuk
lanjut...")

        elif pilih == "2":
            os.system('cls')
            print("=== REGISTER AKUN BARU ===")
            user_baru = input("Masukkan username baru: ")
            pw_baru = input("Masukkan password: ")
            akun.append([user_baru, pw_baru, "user"])
            input("Registrasi berhasil! Silakan login kembali...")

        elif pilih == "3":
            os.system('cls')
            print("Terima kasih telah menggunakan sistem manajemen MU
(Munyak)!")
            exit()

    else:
        input("Pilihan tidak valid! Tekan Enter...")

while True:
    os.system('cls')
    print("=== MANCHESTER UNITED MANAGEMENT SYSTEM ===")
    print("Login sebagai:", role)
    print("-----")
    print("[1] Lihat Data Pemain")
    if role == "admin":
        print("[2] Tambah Pemain Baru")
        print("[3] Update Harga Pemain")
        print("[4] Hapus Pemain")
    print("[5] Logout")
    print("-----")
    menu = input("Pilih menu: ")

```

```

        if menu == "1":
            os.system('cls')
            print("=== DAFTAR PEMAIN MANCHESTER UNITED ===")
            print("No | Nama | Negara | Klub")
            print("Sebelumnya | Rating | Harga (€)")
            print("-----")
            i = 0
            while i < len(pemain):
                print(i+1, "|", pemain[i][0], "|", pemain[i][1], "|",
                    pemain[i][2], "|", pemain[i][3], "|", pemain[i][4])
                i = i + 1
            input("\nTekan Enter untuk kembali...")

        elif menu == "2" and role == "admin":
            os.system('cls')
            print("=== TAMBAH PEMAIN BARU ===")
            nama = input("Nama pemain: ")
            negara = input("Asal negara: ")
            klub = input("Klub sebelumnya: ")
            rating = input("Rating performa (0-100): ")
            harga = input("Harga pasar (€): ")

            cek_angka = True
            for huruf in rating:
                if huruf not in "0123456789":
                    cek_angka = False
            for huruf in harga:
                if huruf not in "0123456789":
                    cek_angka = False

            if cek_angka == True:
                pemain.append([nama, negara, klub, int(rating), int(harga)])
                input("Pemain baru berhasil ditambahkan!")
            else:
                input("Input rating atau harga harus angka!")

        elif menu == "3" and role == "admin":
            os.system('cls')
            print("=== UPDATE HARGA PEMAIN ===")
            index = input("Masukkan nomor pemain: ")

            angka_benar = True
            for huruf in index:
                if huruf not in "0123456789":
                    angka_benar = False

```

```

if angka_benar == True:
    index = int(index) - 1
    if index >= 0 and index < len(pemain):
        print("Nama pemain:", pemain[index][0])
        harga_baru = input("Masukkan harga baru (€): ")

        angka_baru = True
        for huruf in harga_baru:
            if huruf not in "0123456789":
                angka_baru = False

        if angka_baru == True:
            pemain[index][4] = int(harga_baru)
            input("Harga pemain berhasil diperbarui!")
        else:
            input("Harga harus berupa angka!")
    else:
        input("Nomor pemain tidak ditemukan!")
else:
    input("Masukkan angka yang benar!")

elif menu == "4" and role == "admin":
    os.system('cls')
    print("=== HAPUS PEMAIN ===")
    index = input("Masukkan nomor pemain yang akan dihapus: ")

    angka_valid = True
    for huruf in index:
        if huruf not in "0123456789":
            angka_valid = False

    if angka_valid == True:
        index = int(index) - 1
        if index >= 0 and index < len(pemain):
            print("Menghapus pemain:", pemain[index][0])
            yakin = input("Yakin (y/n)? ")
            if yakin == "y" or yakin == "Y":
                del pemain[index]
                input("Pemain berhasil dihapus!")
            else:
                input("Penghapusan dibatalkan.")
        else:
            input("Nomor pemain tidak valid!")
    else:
        input("Masukkan angka yang benar!")

elif menu == "5":
    break

```

```
else:
    input("Menu tidak valid atau akses ditolak!")
```

4. Hasil Output

```
=== MANAJEMEN TIM MANCHESTER UNITED / MUNYUK ===
[1] Login
[2] Register
[3] Keluar
Pilih menu: █
```

Gambar 4.1
Tampilan Awal setelah program dimulai

```
=== MANCHESTER UNITED MANAGEMENT SYSTEM ===
Login sebagai: admin
-----
[1] Lihat Data Pemain
[2] Tambah Pemain Baru
[3] Update Harga Pemain
[4] Hapus Pemain
[5] Logout
-----
Pilih menu: █
```

Gambar 4.2
Hasil *Login* berhasil sebagai admin

```
=== DAFTAR PEMAIN MANCHESTER UNITED ===
No | Nama | Negara | Klub Sebelumnya | Rating | Harga (€)
-----
1 | Bruno Fernandes | Portugal | Sporting Lisbon | 90 | 80000000
2 | Casemiro | Brazil | Real Madrid | 88 | 70000000
3 | Benjamin Sesko | Slovenia | Salzburg | 92 | 85000000
4 | Kobbie Mainoo | Inggris | Akademi MU | 81 | 25000000
5 | Senne Lamme | Belgia | Royal Antwerp | 94 | 18000000
Tekan Enter untuk kembali... █
```

Gambar 4.3
Tampilan Lihat Data Pemain (*Read*)

```
=== TAMBAH PEMAIN BARU ===
Nama pemain: Bryan Mbeumo
Asal negara: Kamerun
Klub sebelumnya: Brentford
Rating performa (0-100): 89
Harga pasar (€): 71000000
Pemain baru berhasil ditambahkan! █
```

Gambar 4.4
Tampilan Menambah Pemain Baru (*Create*)


```

=== UPDATE HARGA PEMAIN ===
Masukkan nomor pemain: 3
Nama pemain: Benjamin Sesko
Masukkan harga baru (€): 10000000
Harga pemain berhasil diperbarui!

```

Gambar 4.5
Tampilan (*Update*) Harga Pemain

```

=== HAPUS PEMAIN ===
Masukkan nomor pemain yang akan dihapus: 6
Menghapus pemain: Bryan Mbeumo
Yakin (y/n)? y
Pemain berhasil dihapus!

```

Gambar 4.6
Tampilan Pemain yang dijual/dipinjam (*Delete*)

```

=== DAFTAR PEMAIN MANCHESTER UNITED ===
No | Nama | Negara | Klub Sebelumnya | Rating | Harga (€)
-----
1 | Bruno Fernandes | Portugal | Sporting Lisbon | 90 | 80000000
2 | Casemiro | Brazil | Real Madrid | 88 | 70000000
3 | Benjamin Sesko | Slovenia | Salzburg | 92 | 10000000
4 | Kobbie Mainoo | Inggris | Akademi MU | 81 | 2500000
5 | Senne Lammes | Belgia | Royal Antwerp | 94 | 18000000

Tekan Enter untuk kembali...

```

Gambar 4.7
Tampilan Data Pemain yang sudah di (*CRUD*)

```

=== MANCHESTER UNITED MANAGEMENT SYSTEM ===
Login sebagai: user
-----
[1] Lihat Data Pemain
[5] Logout
-----
Pilih menu:

```

Gambar 4.8
Tampilan *Login* sebagai user (hanya bisa lihat data pemain dan *logout*)

```
=== REGISTER AKUN BARU ===  
Masukkan username baru: ewok  
Masukkan password: 108  
Registrasi berhasil! Silakan login kembali...
```

Gambar 4.9
Tampilan Registrasi atau buat akun baru

```
Terima kasih telah menggunakan sistem manajemen MU (Munyak)!  
PS D:\(KULIAH)\SEMESTER 1\praktikum Algoritma Pemograman Dasar\prakti
```

Gambar 4.10
Tampilan setelah Keluar dari program

5. Langkah-langkah GIT

5.1 GIT Init

```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git init
Reinitialized existing Git repository in D:/(KULIAH)/SEMESTER 1/praktikum Algori
tma Pemograman Dasar/.git/
```

Git init agar bisa *track progress* yang ada pada folder tersebut

5.2 GIT Add

```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git add .
```

Git Add menambah semua file yang ada didalam folder sementara sebelum file dikomit dalam repository

5.3 GIT Commit

```
Asus@Nabonkey MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemograman Dasa
r (main)
$ git commit -m "update"
[main a403ae8] update
5 files changed, 273 insertions(+), 1 deletion(-)
create mode 100644 praktikum-apd/kelas/pertemuan 5/main.py
create mode 100644 praktikum-apd/kelas/pertemuan 5/tuple.py
create mode 100644 praktikum-apd/post-test/post-test-apd-5/2509106024-FERLYAHMA
DNABIL-APD-5.pdf
create mode 100644 praktikum-apd/post-test/post-test-apd-5/2509106024-FERLYAHMA
DNABIL-APD-5.py
```

Git commit untuk menyimpan perubahan dalam *repository* dengan commit dan pesan "*update*".

5.4 GIT Remote

```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git remote add origin https://github.com/ferlynabil/Praktikum-APD.git
```

Git Remote digunakan untuk menambahkan *repository remote* dengan nama *origin*.

5.5 GIT Push

```
Asus@Nabonkey MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemograman Dasa
r (main)
$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 566.81 KiB | 13.82 MiB/s, done.
Total 13 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
Remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/ferlynabil/Praktikum-APD.git
40632e9..a403ae8 main -> main
```

GIT Push untuk mengunggah (*push*) perubahan ke *repository remote* pada *branch main*.