

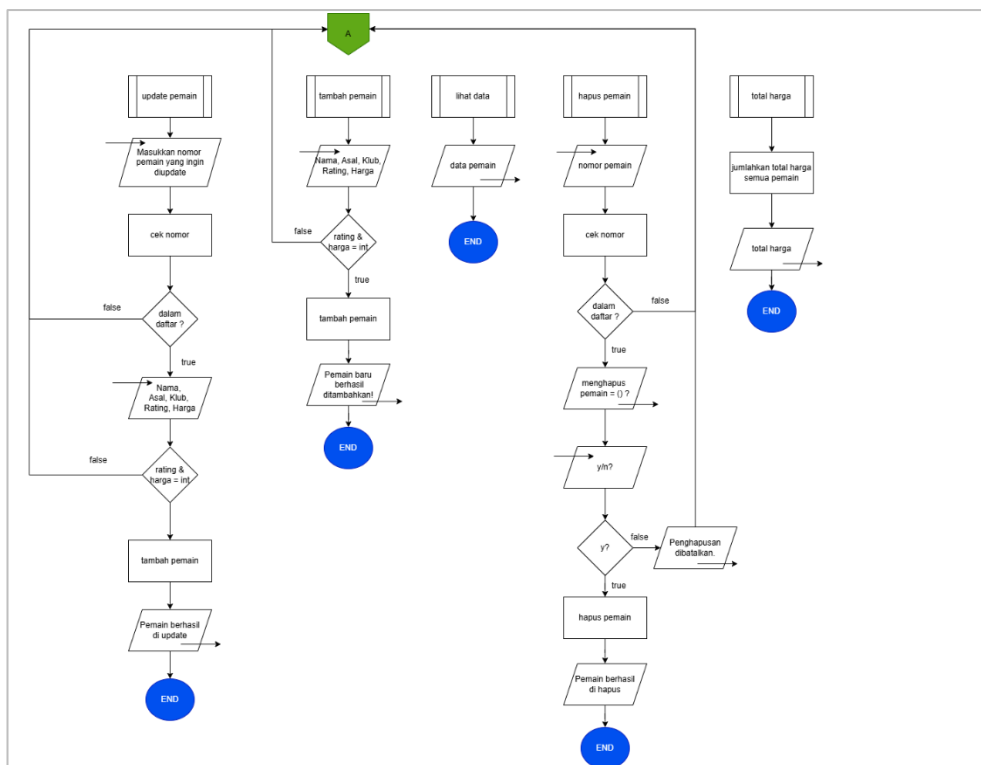
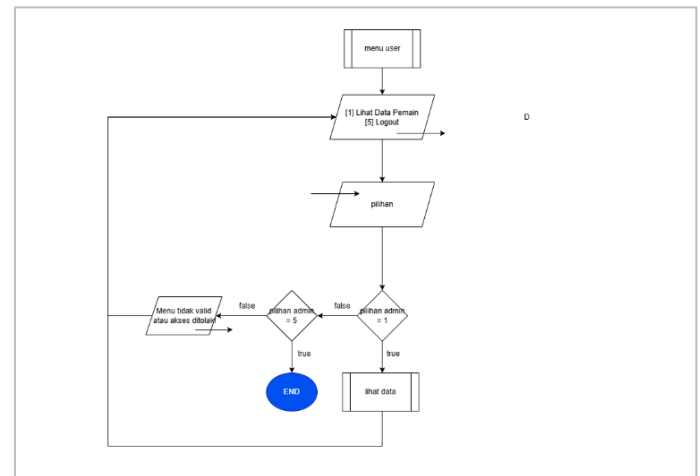
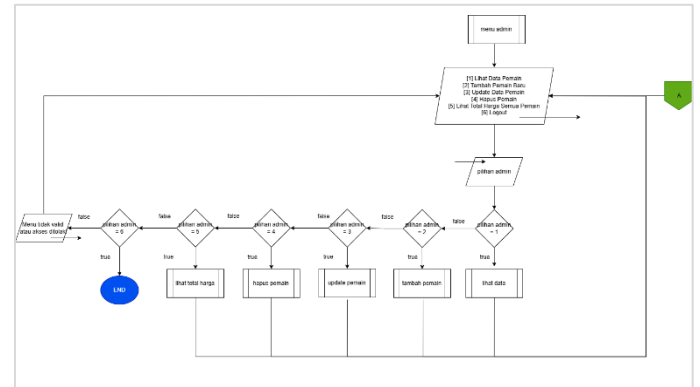
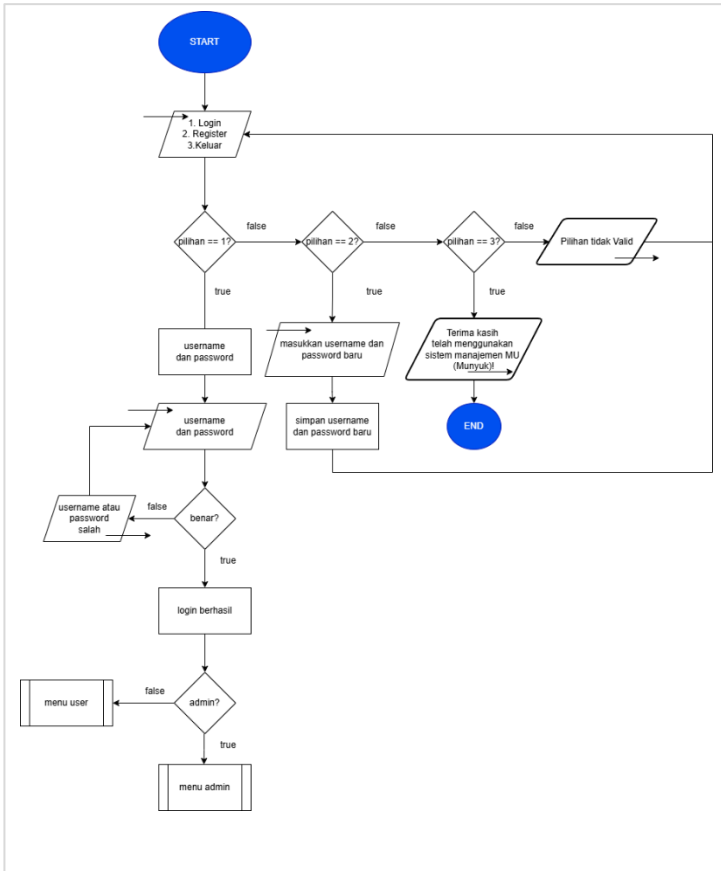
LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



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.

G. Fungsi (*def*), *Try*, *Except*, dan *Error Handling*

- Program menggunakan beberapa fungsi (*def*) untuk memisahkan tugas agar kode lebih rapi, seperti *clear_screen()* untuk membersihkan layar, *tampilkan_pemain()* untuk menampilkan data pemain, dan *total_harga_pemain()* untuk menghitung total harga pemain secara rekursif.
- Blok *try* dan *except* digunakan untuk menangani kesalahan input agar program tidak berhenti saat terjadi error. Misalnya saat pengguna memasukkan huruf pada input angka atau saat login salah.

- *Error handling* memberikan pesan kesalahan yang jelas dan membuat program tetap berjalan normal.
- Penggunaan *def* dan *try-except* membuat program lebih terstruktur, aman, dan mudah dikembangkan.

H. Fungsi Rekursif

- Program menggunakan fungsi rekursif bernama *total_harga_pemain()* untuk menghitung total harga seluruh pemain secara otomatis.
- Fungsi ini memanggil dirinya sendiri sampai semua data pemain dijumlahkan.
- Jika data pemain sudah habis (basis rekursi), fungsi akan berhenti dan mengembalikan nilai total.
- Penggunaan rekursi membuat proses perhitungan lebih efisien dan kode menjadi lebih sederhana tanpa perlu perulangan manual.

3. Source Code

A. Program Sistem Managemen Tim Bola

Source Code:

```
import os

akun = {
    "admin": {"password": "024", "role": "admin"},
    "user": {"password": "420", "role": "user"}
}

pemain = {
    1: {"nama": "Bruno Fernandes", "negara": "Portugal", "klub": "Sporting
Lisbon", "rating": 90, "harga": 80000000},
    2: {"nama": "Casemiro", "negara": "Brazil", "klub": "Real Madrid", "rating":
88, "harga": 70000000},
    3: {"nama": "Benjamin Sesko", "negara": "Slovenia", "klub": "Salzburg",
"rating": 92, "harga": 85000000},
    4: {"nama": "Kobbie Mainoo", "negara": "Inggris", "klub": "Akademi MU",
"rating": 81, "harga": 2500000},
    5: {"nama": "Senne Lammes", "negara": "Belgia", "klub": "Royal Antwerp",
"rating": 94, "harga": 18000000},
}

login = False
role = ""
user_login = ""

def clear_screen():
    os.system('cls')

def tampilkan_pemain(data_pemain):
    print("=== DAFTAR PEMAIN MANCHESTER UNITED ===")
    print("No | Nama | Negara | Klub Sebelumnya | Rating
| Harga (€)")
    print("-----")
    for i, data in data_pemain.items():
        print(f"{i:<2} | {data['nama']:<18} | {data['negara']:<12} |
{data['klub']:<18} | {data['rating']:<6} | {data['harga']}")
        print("-----")
    print("-----")
```

```

def tambah_pemain():
    global pemain
    clear_screen()
    print("=== TAMBAH PEMAIN BARU ===")
    try:
        nama = input("Nama pemain: ")
        negara = input("Asal negara: ")
        klub = input("Klub sebelumnya: ")
        rating = int(input("Rating performa (0-100): "))
        harga = int(input("Harga pasar (€): "))

        id_baru = max(pemain.keys()) + 1 if pemain else 1
        pemain[id_baru] = {
            "nama": nama,
            "negara": negara,
            "klub": klub,
            "rating": rating,
            "harga": harga
        }
        input("Pemain baru berhasil ditambahkan!")
    except ValueError:
        input("Input rating atau harga harus berupa angka!")

def hapus_pemain():
    global pemain
    clear_screen()
    print("=== HAPUS PEMAIN ===")
    try:
        nomor = int(input("Masukkan nomor pemain yang akan dihapus: "))
        if nomor in pemain:
            print("Menghapus pemain:", pemain[nomor]["nama"])
            yakin = input("Yakin (y/n)? ").lower()
            if yakin == "y":
                del pemain[nomor]
                input("Pemain berhasil dihapus!")
            else:
                input("Penghapusan dibatalkan.")
        else:
            input("Nomor pemain tidak valid!")
    except ValueError:
        input("Masukkan angka yang benar!")

def total_harga_pemain(keys=None):
    global pemain
    if keys is None:
        keys = list(pemain.keys())

    if not keys:
        return 0

```

```

def tambah_pemain():
    global pemain
    clear_screen()
    print("=== TAMBAH PEMAIN BARU ===")
    try:
        nama = input("Nama pemain: ")
        negara = input("Asal negara: ")
        klub = input("Klub sebelumnya: ")
        rating = int(input("Rating performa (0-100): "))
        harga = int(input("Harga pasar (€): "))

        id_baru = max(pemain.keys()) + 1 if pemain else 1
        pemain[id_baru] = {
            "nama": nama,
            "negara": negara,
            "klub": klub,
            "rating": rating,
            "harga": harga
        }
        input("Pemain baru berhasil ditambahkan!")
    except ValueError:
        input("Input rating atau harga harus berupa angka!")

def hapus_pemain():
    global pemain
    clear_screen()
    print("=== HAPUS PEMAIN ===")
    try:
        nomor = int(input("Masukkan nomor pemain yang akan dihapus: "))
        if nomor in pemain:
            print("Menghapus pemain:", pemain[nomor]["nama"])
            yakin = input("Yakin (y/n)? ").lower()
            if yakin == "y":
                del pemain[nomor]
                input("Pemain berhasil dihapus!")
            else:
                input("Penghapusan dibatalkan.")
        else:
            input("Nomor pemain tidak valid!")
    except ValueError:
        input("Masukkan angka yang benar!")

def total_harga_pemain(keys=None):
    global pemain
    if keys is None:
        keys = list(pemain.keys())

    if not keys:

```

```

        return 0

    k = keys[0]
    return pemain[k]["harga"] + total_harga_pemain(keys[1:])

def login_user(akun_data):
    global login, role, user_login
    try:
        user = input("Username: ")
        pw = input("Password: ")

        if user in akun_data and akun_data[user]["password"] == pw:
            login = True
            role = akun_data[user]["role"]
            user_login = user
        else:
            raise ValueError("Username atau password salah!")
    except ValueError as e:
        input(f"{e} Tekan Enter untuk lanjut...")

while not login:
    clear_screen()
    print("=== MANAJEMEN TIM MANCHESTER UNITED (MUNYUK) ===")
    print("[1] Login")
    print("[2] Register")
    print("[3] Keluar")
    pilih = input("Pilih menu: ")

    if pilih == "1":
        clear_screen()
        print("=== LOGIN ===")
        login_user(akun)

    elif pilih == "2":
        clear_screen()
        print("=== REGISTER AKUN BARU ===")
        user_baru = input("Masukkan username baru: ")
        if user_baru in akun:
            input("Username sudah terdaftar! Tekan Enter untuk kembali...")
        else:
            pw_baru = input("Masukkan password: ")
            akun[user_baru] = {"password": pw_baru, "role": "user"}
            input("Registrasi berhasil! Silakan login kembali...")

    elif pilih == "3":
        clear_screen()
        print("Terima kasih telah menggunakan sistem manajemen MU (Munyak)!")
        exit()

```



```

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

while True:
    clear_screen()
    print("=== MANCHESTER UNITED MANAGEMENT SYSTEM ===")
    print(f"Login sebagai: {user_login} ({role})")
    print("-----")
    print("[1] Lihat Data Pemain")
    if role == "admin":
        print("[2] Tambah Pemain Baru")
        print("[3] Update Data Pemain")
        print("[4] Hapus Pemain")
        print("[5] Lihat Total Harga Semua Pemain ")
        print("[6] Logout")
    print("-----")
    menu = input("Pilih menu: ")

    if menu == "1":
        clear_screen()
        tampilkan_pemain(pemain)
        input("\nTekan Enter untuk kembali...")

    elif menu == "2" and role == "admin":
        tambah_pemain()

    elif menu == "3" and role == "admin":
        clear_screen()
        print("=== UPDATE DATA PEMAIN ===")
        try:
            nomor = int(input("Masukkan nomor pemain yang ingin diupdate: "))
            if nomor in pemain:
                data = pemain[nomor]
                print(f"\nData pemain saat ini:")
                print(f>Nama      : {data['nama']}")
                print(f>Negara    : {data['negara']}")
                print(f>Klub      : {data['klub']}")
                print(f>Rating   : {data['rating']}")
                print(f>Harga (€) : {data['harga']}")
                print("-----")
                print("Kosongkan input jika tidak ingin mengubah field tersebut.\n")

                nama_baru = input(f>Nama baru [{data['nama']}]: ") or data['nama']
                negara_baru = input(f>Negara baru [{data['negara']}]: ") or
data['negara']
                klub_baru = input(f>Klub baru [{data['klub']}]: ") or data['klub']
                rating_baru = input(f>Rating baru [{data['rating']}]: ") or
str(data['rating'])

```

```

        harga_baru = input(f"Harga baru (€) [{data['harga']}]: ") or
str(data['harga'])

        if rating_baru.isdigit() and harga_baru.isdigit():
            pemain[nomor] = {
                "nama": nama_baru,
                "negara": negara_baru,
                "klub": klub_baru,
                "rating": int(rating_baru),
                "harga": int(harga_baru)
            }
            input("Data pemain berhasil diperbarui!")
        else:
            input("Rating dan harga harus berupa angka!")
    else:
        input("Nomor pemain tidak ditemukan!")
except ValueError:
    input("Masukkan angka yang benar!")

elif menu == "4" and role == "admin":
    hapus_pemain()

elif menu == "5" and role == "admin":
    clear_screen()
    total = total_harga_pemain()
    print(f"Total harga semua pemain MU adalah: €{total:,}")
    input("\nTekan Enter untuk kembali...")

elif menu == "6":
    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

Gambar 4.2

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

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 Lammes | 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*)

```

=== 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 | 2500000
5 | Senne Lammes | Belgia | Royal Antwerp | 94 | 18000000
6 | Bryan Mbeumo | Kamerun | Brentford | 89 | 71000000

Tekan Enter untuk kembali...

```

Gambar 4.5
Tampilan data setelah di tambahkan dan sebelum di *update*

```

=== UPDATE DATA PEMAIN ===
Masukkan nomor pemain yang ingin diupdate: 1

Data pemain saat ini:
Nama      : Bruno Fernandes
Negara    : Portugal
Klub      : Sporting Lisbon
Rating    : 90
Harga (€) : 80000000
-----
Kosongkan input jika tidak ingin mengubah field tersebut.

Nama baru [Bruno Fernandes]: Mathis De Ligt
Negara baru [Portugal]: Belanda
Klub baru [Sporting Lisbon]: Juventus
Rating baru [90]: 92
Harga baru (€) [80000000]: 62000000

```

Gambar 4.6
Tampilan ingin (*Update*) Data Pemain

```

=== DAFTAR PEMAIN MANCHESTER UNITED ===
No | Nama | Negara | Klub Sebelumnya | Rating | Harga (€)
-----
1 | Mathis De Ligt | Belanda | Juventus | 92 | 62000000
2 | Casemiro | Brazil | Real Madrid | 88 | 70000000
3 | Benjamin Sesko | Slovenia | Salzburg | 92 | 85000000
4 | Kobbie Mainoo | Inggris | Akademi MU | 81 | 2500000
5 | Senne Lammes | Belgia | Royal Antwerp | 94 | 18000000
6 | Bryan Mbeumo | Kamerun | Brentford | 89 | 71000000

Tekan Enter untuk kembali...

```

Gambar 4.7
Tampilan seluruh data Setelah di *update* dan sebelum di jual (*Delete*)

```

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

```

Gambar 4.8
Tampilan Pemain yang ingin dijual/dipinjam (*Delete*)

```

=== DAFTAR PEMAIN MANCHESTER UNITED ===
No | Nama                | Negara    | Klub Sebelumnya | Rating | Harga (€)
-----
1  | Mathis De Ligt      | Belanda   | Juventus         | 92     | 62000000
2  | Casemiro             | Brazil    | Real Madrid      | 88     | 70000000
3  | Benjamin Sesko       | Slovenia  | Salzburg         | 92     | 85000000
4  | Kobbie Mainoo        | Inggris   | Akademi MU       | 81     | 25000000
5  | Senne Lammes         | Belgia    | Royal Antwerp    | 94     | 18000000

Tekan Enter untuk kembali...

```

Gambar 4.9
Tampilan Data Pemain yang sudah di jual (*Delete*)

```

Total harga semua pemain MU adalah: €267,500,000
Tekan Enter untuk kembali...

```

Gambar 4.10
Tampilan Total Semua Harga Pemain (Rekursif)

```

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

```

Gambar 4.11
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 4.12
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 4.13
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 f8130e3] update
5 files changed, 369 insertions(+)
create mode 100644 praktikum-apd/kelas/pertemuan 7/coba 1.py
create mode 100644 praktikum-apd/kelas/pertemuan 7/main.py
create mode 100644 praktikum-apd/kelas/pertemuan 7/studi kasus.py
create mode 100644 praktikum-apd/post-test/post-test-apd-7/2509106024-FERLYAHMA
DNABIL-APD-7.pdf
create mode 100644 praktikum-apd/post-test/post-test-apd-7/2509106024-FERLYAHMA
DNABIL-APD-7.py
```

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

5.4 GIT Remote

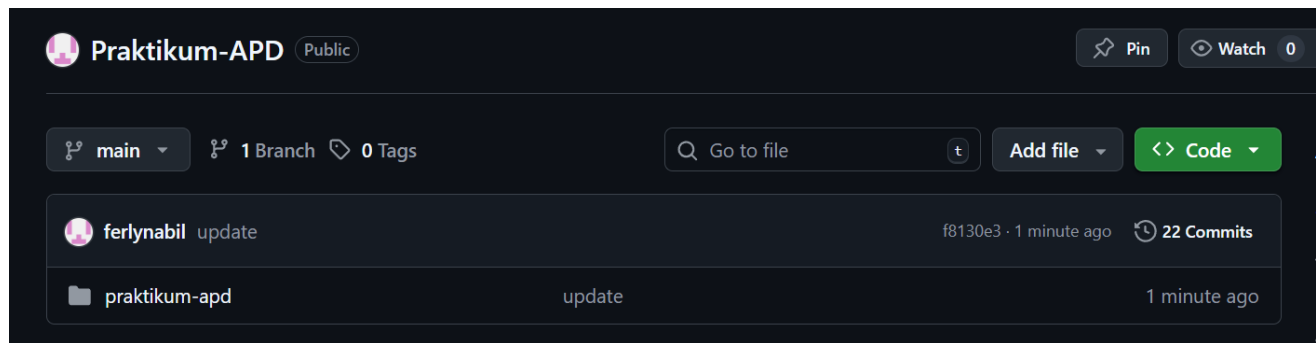
```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git remote add origin https://github.com/ferlynabil/Praktikum-APD.git
```

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: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 734.42 KiB | 28.25 MiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/ferlynabil/Praktikum-APD.git
0537909..f8130e3 main -> main
```

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



Semua Folder sudah ter *upload* pada Github