

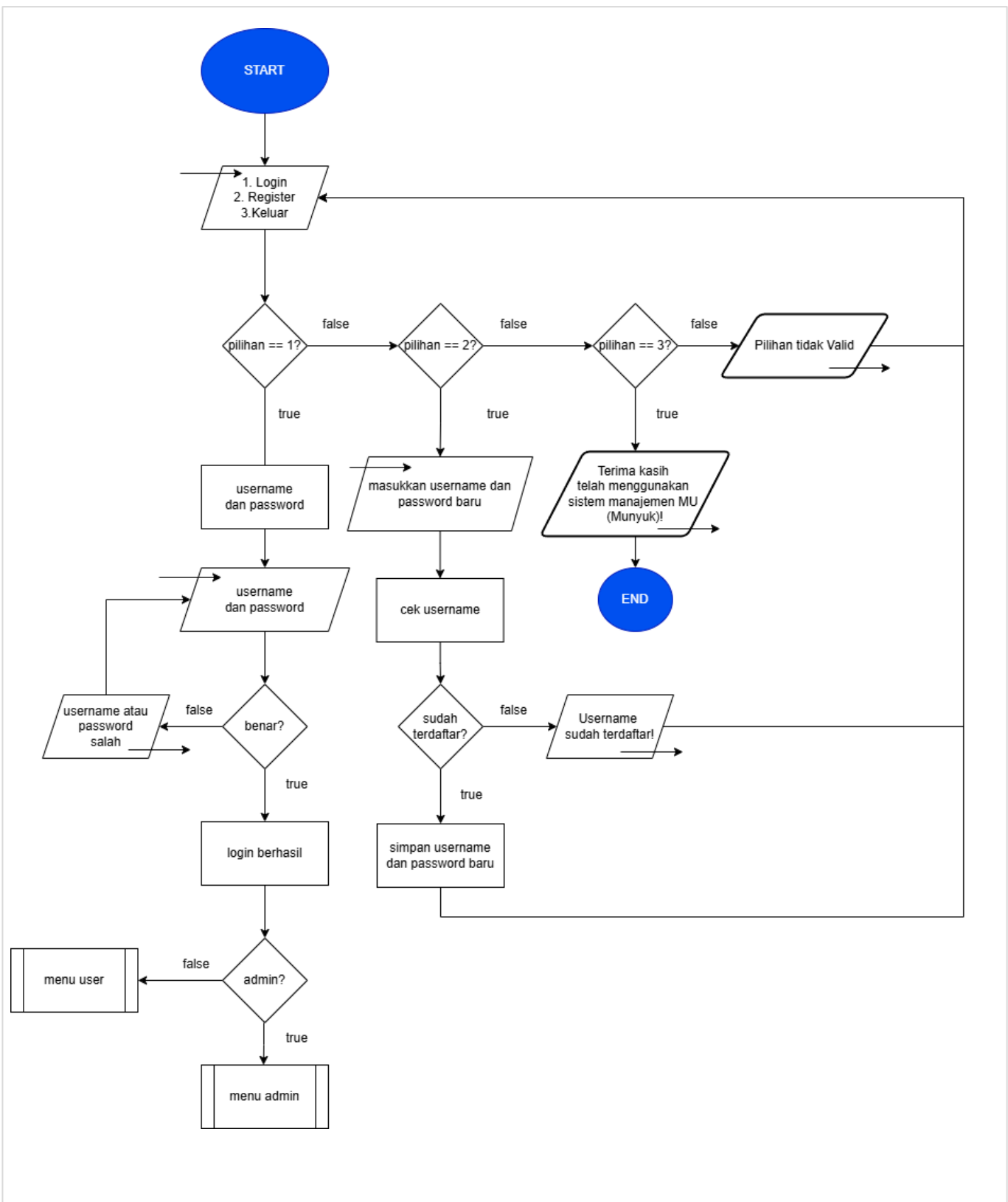
LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR



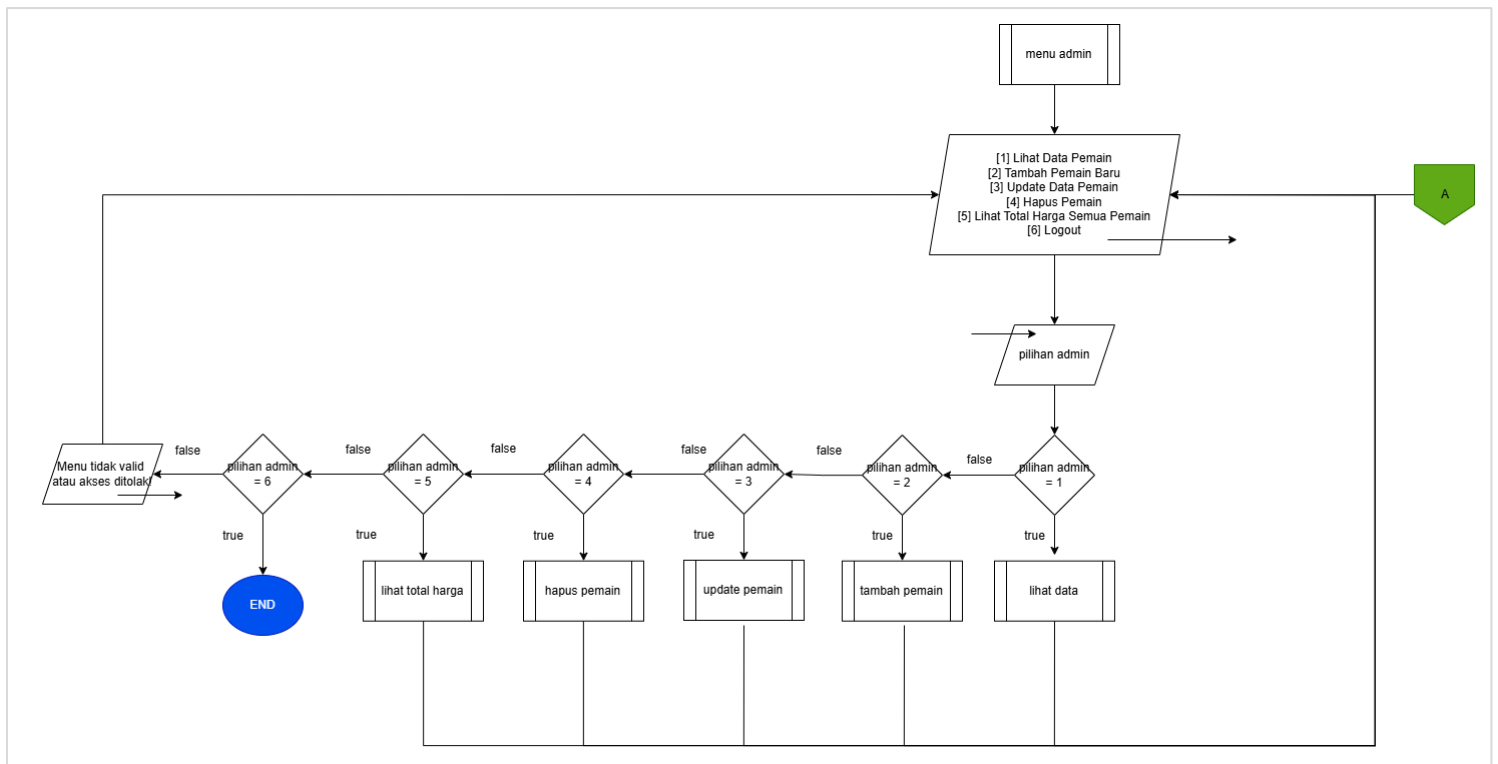
Disusun oleh:
Ferly Ahmad Nabil (2509106024)
Kelas (A2 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

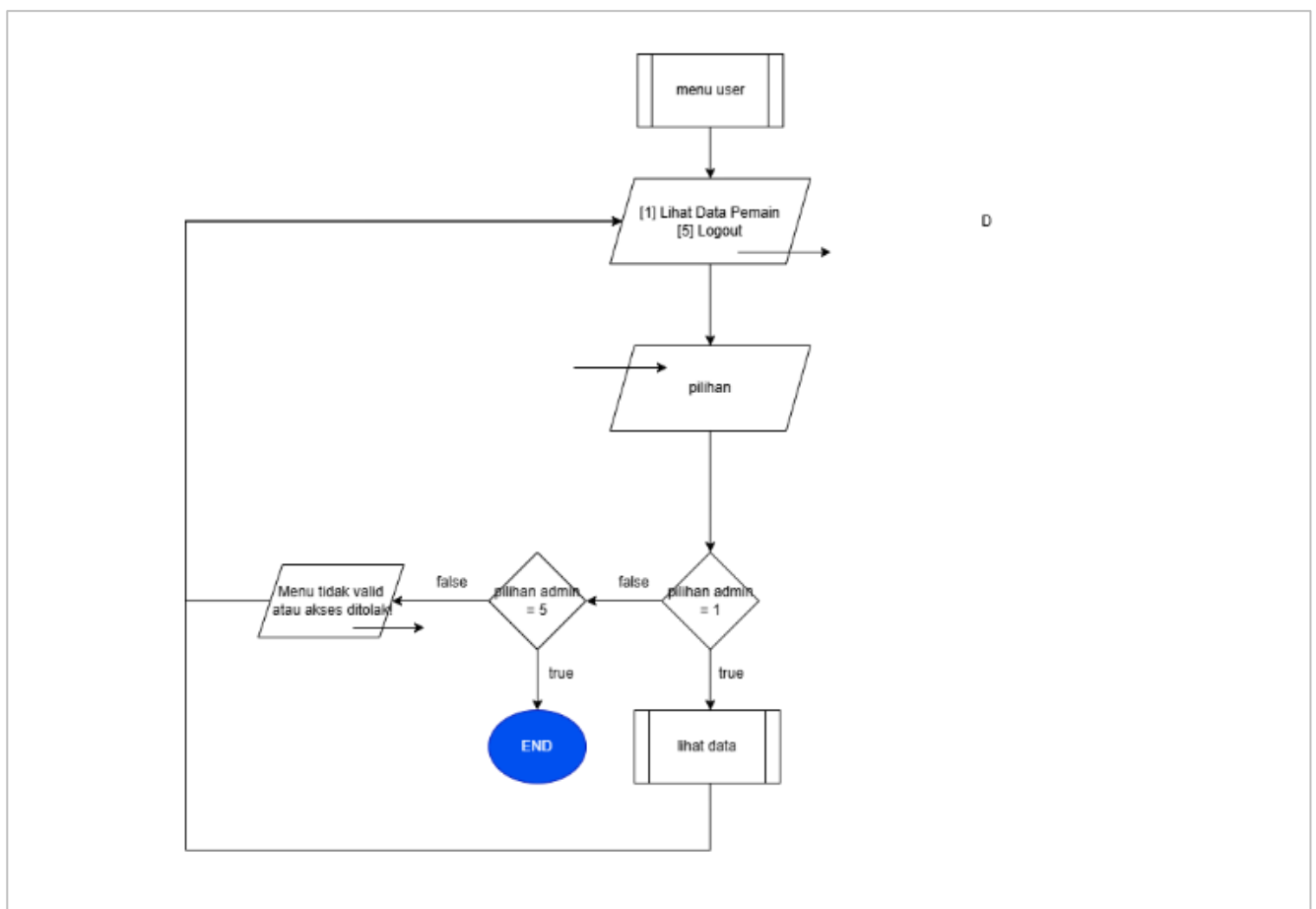
1. Flowchart



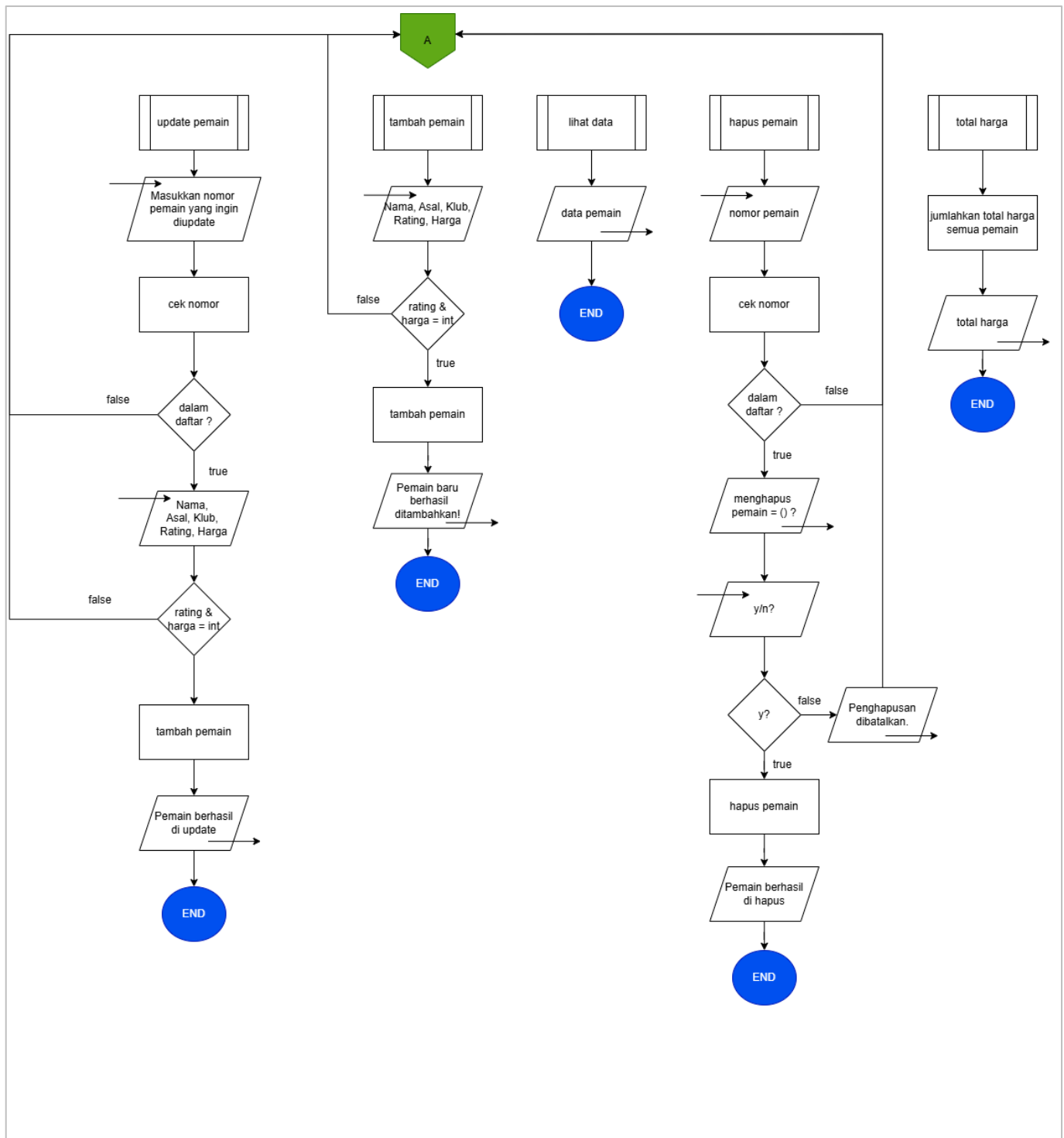
Gambar 1.1 Flowchart Bagian Awal



Gambar 1.2 Flowchart Menu Admin



Gambar 1.3 Flowchart Menu User biasa



Gambar 1.4 Flowchart Fungsi

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.

I. Modularisasi Program (Pemecahan File)

Program dibagi menjadi beberapa file agar struktur lebih rapi dan mudah dikembangkan:

- **main.py** → Program utama yang menjalankan menu dan mengatur alur sistem.
- **login.py** → Berisi fungsi autentikasi login, register, dan tampilan tabel utama menggunakan PrettyTable.
- **data.py** → Menyimpan data awal pemain dan akun.
- **hps.py** → Menyediakan fungsi tambahan seperti *clear_screen()* dan pengaturan warna

J. Tampilan Berwarna dan Interaktif

Program memanfaatkan dua library utama untuk tampilan terminal yang menarik:

Library	Kegunaan
colorma	Memberi warna pada teks
prettytable	Membuat tampilan tabel seperti daftar pemain dan hasil login agar lebih rapi.
datetime	Menampilkan waktu login secara otomatis

3. Source Code

A. Program Sistem Manajemen Tim Bola

```
1  pemain = {
2      1: {"nama": "Bruno Fernandes", "negara": "Portugal", "klub": "Sporting Lisbon", "rating": 90, "harga": 80000000},
3      2: {"nama": "Casemiro", "negara": "Brazil", "klub": "Real Madrid", "rating": 88, "harga": 70000000},
4      3: {"nama": "Benjamin Sesko", "negara": "Slovenia", "klub": "Salzburg", "rating": 92, "harga": 85000000},
5      4: {"nama": "Kobbie Mainoo", "negara": "Inggris", "klub": "Akademi MU", "rating": 81, "harga": 25000000},
6      5: {"nama": "Senne Lammes", "negara": "Belgia", "klub": "Royal Antwerp", "rating": 94, "harga": 18000000},
7  }
8  users = {
9      "admin": {"password": "024", "role": "admin"},
10     "user": {"password": "420", "role": "user"}
11 }
```

Gambar 3.1
Source Code di file *data.py*

```
1  from colorama import Fore, Style, init
2  from prettytable import PrettyTable
3  from hps import clear_screen
4  from data import users
5  from datetime import datetime
6
7  init(autoreset=True, convert=True, strip=False)
8
9  def login_user():
10     while True:
11         clear_screen()
12         tabel = PrettyTable()
13         tabel.field_names = [Fore.YELLOW + "Menu " + Style.RESET_ALL, Fore.YELLOW + "Ket" + Style.RESET_ALL]
14         tabel.add_row(["1", "Login"])
15         tabel.add_row(["2", "Registrasi"])
16         tabel.add_row(["3", "Keluar"])
17
18         print(Fore.RED + "=== SISTEM LOGIN MANCHESTER UNITED ===" + Style.RESET_ALL)
19         print(tabel)
20         pilih = input(Fore.CYAN + "Pilih menu: " + Style.RESET_ALL)
21
22         if pilih == "1":
23             clear_screen()
24             print(Fore.YELLOW + "=== LOGIN ===" + Style.RESET_ALL)
25             username = input("Username: ").strip()
26             password = input("Password: ").strip()
27
28             if username in users and users[username]["password"] == password:
29                 waktu = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
30                 clear_screen()
31                 tabel_login = PrettyTable()
32                 tabel_login.field_names = ["Status", "Tanggal & Waktu Login"]
33                 tabel_login.add_row([" Login Berhasil", waktu])
34                 print(Fore.GREEN + str(tabel_login))
35                 return True
36             else:
37                 print(Fore.RED + "Username atau Password salah!" + Style.RESET_ALL)
```

```

35         role = users[username]["role"]
36         input("\nTekan Enter untuk melanjutkan ke sistem...")
37         return username, role
38     else:
39         input(Fore.RED + " Username atau password salah! Tekan Enter untuk ulang...")
40
41     elif pilih == "2":
42         clear_screen()
43         print(Fore.YELLOW + "=== REGISTER USER BARU ===" + Style.RESET_ALL)
44         new_user = input("Buat username: ").strip()
45         if new_user in users:
46             input(Fore.RED + " Username sudah digunakan! Tekan Enter...")
47             continue
48
49         new_pass = input("Buat password: ").strip()
50         role = input("Masukkan role (admin/user): ").lower().strip()
51         if role not in ["admin", "user"]:
52             input(Fore.RED + " Role tidak valid! Tekan Enter...")
53             continue
54
55         users[new_user] = {"password": new_pass, "role": role}
56         clear_screen()
57         tabel_reg = PrettyTable()
58         tabel_reg.field_names = ["Status", "Username", "Role"]
59         tabel_reg.add_row([" Registrasi Berhasil", new_user, role])
60         print(Fore.GREEN + str(tabel_reg))
61         input("\nTekan Enter untuk kembali ke menu login...")
62
63     elif pilih == "3":
64         clear_screen()
65         tabel_keluar = PrettyTable()
66         tabel_keluar.field_names = ["Status", "Pesan"]
67         tabel_keluar.add_row([" Keluar", "Terima kasih telah menggunakan sistem MUNYUK!"])
68         print(Fore.YELLOW + str(tabel_keluar))
69         exit()
70
71     else:
72         input(Fore.RED + " Pilihan tidak valid! Tekan Enter untuk ulang...")

```

Gambar 3.2
Source Code di file *login.py*

```

1  from prettytable import PrettyTable
2  import random
3
4  def tampilkan_pemain(data_pemain):
5      tabel = PrettyTable()
6      tabel.field_names = ["No", "Nama", "Negara", "Klub Sebelumnya", "Rating", "Harga (€)"]
7      for i, data in data_pemain.items():
8          tabel.add_row([i, data['nama'], data['negara'], data['klub'], data['rating'], data['harga']])
9      print(tabel)
10
11  def tambah_pemain(pemain):
12      print("=== TAMBAH PEMAIN BARU ===")
13      nama = input("Nama pemain: ")
14      negara = input("Asal negara: ")
15      klub = input("Klub sebelumnya: ")
16
17      try:
18          rating_input = input("Rating (kosongkan untuk acak): ")
19          rating = int(rating_input) if rating_input else random.randint(70, 99)
20          harga_input = input("Harga (€) (kosongkan untuk acak): ")
21          harga = int(harga_input) if harga_input else random.randint(1_000_000, 100_000_000)
22


```

```

22
23     id_baru = max(pemain.keys()) + 1 if pemain else 1
24     pemain[id_baru] = {
25         "nama": nama, "negara": negara, "klub": klub,
26         "rating": rating, "harga": harga
27     }
28     input("Pemain baru berhasil ditambahkan!")
29 except ValueError:
30     input("Rating dan harga harus berupa angka!")
31
32 def hapus_pemain(pemain):
33     print("=== HAPUS PEMAIN ===")
34     try:
35         nomor = int(input("Masukkan nomor pemain yang akan dihapus: "))
36         if nomor in pemain:
37             print(f"Menghapus pemain: {pemain[nomor]['nama']}")
38             yakin = input("Yakin (y/n)? ").lower()
39             if yakin == "y":
40                 del pemain[nomor]
41                 input("Pemain berhasil dihapus!")
42             else:
43                 input("Dibatalkan.")
44         else:
45             input("Nomor pemain tidak valid!")
46     except ValueError:
47         input("Masukkan angka yang benar!")
48
49 def update_pemain(pemain):
50     print("=== UPDATE DATA PEMAIN ===")
51     try:
52         nomor = int(input("Masukkan nomor pemain yang ingin diupdate: "))
53         if nomor in pemain:
54             data = pemain[nomor]
55             print(f"\nData pemain saat ini: {data}")
56             print("Kosongkan input jika tidak ingin mengubah field tersebut.\n")
57
58             nama_baru = input(f>Nama baru [{data['nama']}]: ") or data['nama']
59             negara_baru = input(f">Negara baru [{data['negara']}]: ") or data['negara']
60             klub_baru = input(f">Klub baru [{data['klub']}]: ") or data['klub']
61             rating_baru = input(f">Rating baru [{data['rating']}]: ") or str(data['rating'])
62             harga_baru = input(f">Harga baru (€) [{data['harga']}]: ") or str(data['harga'])
63
64             if rating_baru.isdigit() and harga_baru.isdigit():
65                 pemain[nomor] = {
66                     "nama": nama_baru,
67                     "negara": negara_baru,
68                     "klub": klub_baru,
69                     "rating": int(rating_baru),
70                     "harga": int(harga_baru)
71                 }
72                 input("Data pemain berhasil diperbarui!")
73             else:
74                 input("Rating dan harga harus berupa angka!")
75         else:
76             input("Nomor pemain tidak ditemukan!")
77     except ValueError:
78         input("Masukkan angka yang benar!")

```

Gambar 3.3
Source Code di file *pemain.py*



```

1  import os
2  from prettytable import PrettyTable
3  from colorama import Fore, Style, init
4
5  init(autoreset=True, convert=True, strip=False)
6
7  def total_harga_pemain(pemain):
8      return sum(data["harga"] for data in pemain.values())
9
10 def tampilkan_pemain(pemain):
11     clear_screen()
12     print(Fore.RED + "=== DAFTAR PEMAIN MANCHESTER UNITED ===" + Style.RESET_ALL)
13     tabel = PrettyTable()
14     tabel.field_names = ["No", "Nama", "Negara", "Klub", "Rating", "Harga (€)"]
15     for i, data in pemain.items():
16         tabel.add_row([
17             i,
18             data["nama"],
19             data["negara"],
20             data["klub"],
21             data["rating"],
22             f"€{data['harga']:,}"
23         ])
24     print(Fore.CYAN + str(tabel) + Style.RESET_ALL)
25
26 def tampilkan_user(users):
27     print(Fore.YELLOW + "\n=== DAFTAR USER TERDAFTAR ===" + Style.RESET_ALL)
28     tabel = PrettyTable()
29     tabel.field_names = ["Username", "Role"]
30     for username, data in users.items():
31         tabel.add_row([username, data["role"]])
32     print(Fore.CYAN + str(tabel) + Style.RESET_ALL)
33
34 def clear_screen():
35     os.system("cls" )

```

Gambar 3.4
Source Code di file *hpss.py*

```

1  from colorama import Fore, Style, init
2  from login import login_user, clear_screen
3  from hps import tampilkan_pemain, total_harga_pemain
4  from data import pemain
5
6  init(autoreset=True, convert=True, strip=False)
7
8  user_login, role = login_user()
9
10 while True:
11     clear_screen()
12     print(Fore.RED + "=== MANCHESTER UNITED MANAGEMENT SYSTEM ===")
13     print(Fore.GREEN + f"Login sebagai: {user_login} ({role})")
14     print("-----")
15     print("[1] Lihat Data Pemain")
16     if role == "admin":
17         print("[2] Tambah Pemain Baru")
18         print("[3] Update Data Pemain")
19         print("[4] Hapus Pemain")
20         print("[5] Lihat Total Harga Semua Pemain")
21     print("[6] Logout")
22     print("-----")
23
24     menu = input(Fore.BLUE + "Pilih menu: " + Style.RESET_ALL)
25
26     if menu == "1":
27         tampilkan_pemain(pemain)
28         input("\nTekan Enter untuk kembali...")
29
30     elif menu == "2" and role == "admin":
31         clear_screen()
32         print(Fore.YELLOW + "=== TAMBAH PEMAIN BARU ===")
33         nama = input("Nama pemain: ")
34         negara = input("Asal negara: ")
35         klub = input("Klub sebelumnya: ")
36         rating = int(input("Rating (0-100): "))
37         harga = int(input("Harga (€): "))
38         id_baru = max(pemain.keys()) + 1
39         pemain[id_baru] = {"nama": nama, "negara": negara, "klub": klub, "rating": rating, "harga": harga}
40         input(Fore.GREEN + " Pemain berhasil ditambahkan!")
41
42     elif menu == "3" and role == "admin":
43         tampilkan_pemain(pemain)
44         nomor = int(input("Masukkan nomor pemain: "))
45         if nomor in pemain:
46             data = pemain[nomor]
47             print("\nKosongkan jika tidak ingin mengubah field.")
48             nama_baru = input(f>Nama [{data['nama']}]: ") or data["nama"]
49             negara_baru = input(f">Negara [{data['negara']}]: ") or data["negara"]
50             klub_baru = input(f">Klub [{data['klub']}]: ") or data["klub"]
51             rating_baru = input(f">Rating [{data['rating']}]: ") or data["rating"]
52             harga_baru = input(f">Harga [{data['harga']}]: ") or data["harga"]
53             pemain[nomor] = {
54                 "nama": nama_baru, "negara": negara_baru, "klub": klub_baru,
55                 "rating": int(rating_baru), "harga": int(harga_baru)
56             }
57             input(Fore.GREEN + " Data pemain diperbarui!")
58
59     elif menu == "4" and role == "admin":
60         tampilkan_pemain(pemain)
61         nomor = int(input("Masukkan nomor pemain yang ingin dihapus: "))
62         if nomor in pemain:
63             del pemain[nomor]
64             input(Fore.GREEN + " Pemain berhasil dihapus!")
65

```

```

65
66     elif menu == "5" and role == "admin":
67         clear_screen()
68         total = total_harga_pemain(pemain)
69         print(Fore.MAGENTA + f" Total harga semua pemain: €{total:,}")
70         input("\nTekan Enter untuk kembali...")
71
72     elif menu == "6":
73         clear_screen()
74         print(Fore.YELLOW + "Logout berhasil. Terima kasih telah menggunakan sistem MUNYUK!")
75         break
76
77     else:
78         input(Fore.RED + " Menu tidak valid atau akses ditolak!")

```

Gambar 3.5
Source Code di file utama : *main.py*

4. Hasil Output

```

=== SISTEM LOGIN MANCHESTER UNITED ===
+-----+-----+
| Menu | Ket |
+-----+-----+
| 1 | Login |
| 2 | Registrasi |
| 3 | Keluar |
+-----+-----+
Pilih menu: █

```

Gambar 4.1
Tampilan Awal setelah program dimulai

```

+-----+-----+
|      Status      | Tanggal & Waktu Login |
+-----+-----+
| Login Berhasil | 04/11/2025 19:05:38 |
+-----+-----+

Tekan Enter untuk melanjutkan ke sistem... █

```

Gambar 4.2
Tampilan Login Berhasil

```
=== 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: 
```

Gambar 4.3
Hasil Login berhasil sebagai admin

```
=== DAFTAR PEMAIN MANCHESTER UNITED ===
+-----+-----+-----+-----+-----+-----+
| No | Nama | Negara | Klub | Rating | Harga (€) |
+-----+-----+-----+-----+-----+-----+
| 1 | Bruno Fernandes | Portugal | Sporting Lisbon | 90 | €80,000,000 |
| 2 | Casemiro | Brazil | Real Madrid | 88 | €70,000,000 |
| 3 | Benjamin Sesko | Slovenia | Salzburg | 92 | €85,000,000 |
| 4 | Kobbie Mainoo | Inggris | Akademi MU | 81 | €2,500,000 |
| 5 | Senne Lammes | Belgia | Royal Antwerp | 94 | €18,000,000 |
+-----+-----+-----+-----+-----+-----+

Tekan Enter untuk kembali...
```

Gambar 4.4
Tampilan Lihat Data Pemain (Read)

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

Gambar 4.5
Tampilan Menambah Pemain Baru (Create)

=== DAFTAR PEMAIN MANCHESTER UNITED ===

No	Nama	Negara	Klub	Rating	Harga (€)
1	Bruno Fernandes	Portugal	Sporting Lisbon	90	€80,000,000
2	Casemiro	Brazil	Real Madrid	88	€70,000,000
3	Benjamin Sesko	Slovenia	Salzburg	92	€85,000,000
4	Kobbie Mainoo	Inggris	Akademi MU	81	€2,500,000
5	Senne Lammes	Belgia	Royal Antwerp	94	€18,000,000
6	Bryan Mbeumo	Kamerun	Brentford	89	€71,000,000

Tekan Enter untuk kembali...

Gambar 4.6
Tampilan data setelah di tambahkan dan sebelum di *update*

4	Kobbie Mainoo	Inggris	Akademi MU	81	€2,500,000
5	Senne Lammes	Belgia	Royal Antwerp	94	€18,000,000
6	Bryan Mbeumo	Kamerun	Brentford	89	€71,000,000

Masukkan nomor pemain: 6

Kosongkan jika tidak ingin mengubah field.

Nama [Bryan Mbeumo]: Mathis De Ligt

Negara [Kamerun]: Belanda

Klub [Brentford]: Juventus

Rating [89]: 92

Harga [71000000]: 62000000

Data pemain diperbarui!

Gambar 4.7
Tampilan ingin (*Update*) Data Pemain

=== DAFTAR PEMAIN MANCHESTER UNITED ===

No	Nama	Negara	Klub	Rating	Harga (€)
1	Bruno Fernandes	Portugal	Sporting Lisbon	90	€80,000,000
2	Casemiro	Brazil	Real Madrid	88	€70,000,000
3	Benjamin Sesko	Slovenia	Salzburg	92	€85,000,000
4	Kobbie Mainoo	Inggris	Akademi MU	81	€2,500,000
5	Senne Lammes	Belgia	Royal Antwerp	94	€18,000,000
6	Mathis De Ligt	Belanda	Juventus	92	€62,000,000

Tekan Enter untuk kembali...

Gambar 4.8
Tampilan seluruh data Setelah di *update* dan sebelum di jual (*Delete*)

```

=== DAFTAR PEMAIN MANCHESTER UNITED ===
+-----+-----+-----+-----+-----+-----+
| No | Nama | Negara | Klub | Rating | Harga (€) |
+-----+-----+-----+-----+-----+-----+
| 1 | Bruno Fernandes | Portugal | Sporting Lisbon | 90 | €80,000,000 |
| 2 | Casemiro | Brazil | Real Madrid | 88 | €70,000,000 |
| 3 | Benjamin Sesko | Slovenia | Salzburg | 92 | €85,000,000 |
| 5 | Senne Lammes | Belgia | Royal Antwerp | 94 | €18,000,000 |
| 6 | Mathis De Ligt | Belanda | Juventus | 92 | €62,000,000 |
+-----+-----+-----+-----+-----+-----+
Masukkan nomor pemain yang ingin dihapus: 4

```

Gambar 4.9
Tampilan Pemain yang ingin dijual/dipinjam (*Delete*)

```

=== DAFTAR PEMAIN MANCHESTER UNITED ===
+-----+-----+-----+-----+-----+-----+
| No | Nama | Negara | Klub | Rating | Harga (€) |
+-----+-----+-----+-----+-----+-----+
| 1 | Bruno Fernandes | Portugal | Sporting Lisbon | 90 | €80,000,000 |
| 2 | Casemiro | Brazil | Real Madrid | 88 | €70,000,000 |
| 3 | Benjamin Sesko | Slovenia | Salzburg | 92 | €85,000,000 |
| 5 | Senne Lammes | Belgia | Royal Antwerp | 94 | €18,000,000 |
| 6 | Mathis De Ligt | Belanda | Juventus | 92 | €62,000,000 |
+-----+-----+-----+-----+-----+-----+
Tekan Enter untuk kembali...

```

Gambar 4.10
Tampilan Data Pemain yang sudah di jual (*Delete*)

```

Total harga semua pemain: €315,000,000
.
Tekan Enter untuk kembali...

```

Gambar 4.11
Tampilan Total Semua Harga Pemain (Rekursif)

```
=== MANCHESTER UNITED MANAGEMENT SYSTEM ===
Login sebagai: user (user)
-----
[1] Lihat Data Pemain
[6] Logout
-----
Pilih menu: █
```

Gambar 4.12
Tampilan *Login* sebagai user (hanya bisa lihat data pemain dan *logout*)

```
+-----+-----+-----+
|      Status      | Username | Role |
+-----+-----+-----+
| Registrasi Berhasil |   ewok   | user |
+-----+-----+-----+
Tekan Enter untuk kembali ke menu login... █
```

Gambar 4.13
Tampilan Registrasi atau buat akun baru

```
+-----+-----+-----+
| Status |              Pesan              |
+-----+-----+-----+
| Keluar | Terima kasih telah menggunakan sistem MUNYUK! |
+-----+-----+-----+
PS D:\(KULIAH)\SEMESTER 1\praktikum Algoritma Pemograman Dasar
```

Gambar 4.14
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 170b3a8] update
13 files changed, 331 insertions(+)
 create mode 100644 praktikum-apd/kelas/pertemuan 8/library.py
 create mode 100644 praktikum-apd/kelas/pertemuan 8/main.py
 create mode 100644 praktikum-apd/kelas/pertemuan 8/th.py
 create mode 100644 praktikum-apd/log_aktivitas.txt
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8.pdf
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/__pycache__/data.cpython-311.pyc
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/__pycache__/hps.cpython-311.pyc
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/__pycache__/login.cpython-311.pyc
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/data.py
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/hps.py
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/login.py
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/main.py
 create mode 100644 praktikum-apd/post-test/post-test-apd-8/2509106024-FERLYAHMA
DNABIL-APD-8/pemain.py
```

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

5.4 GIT Remote

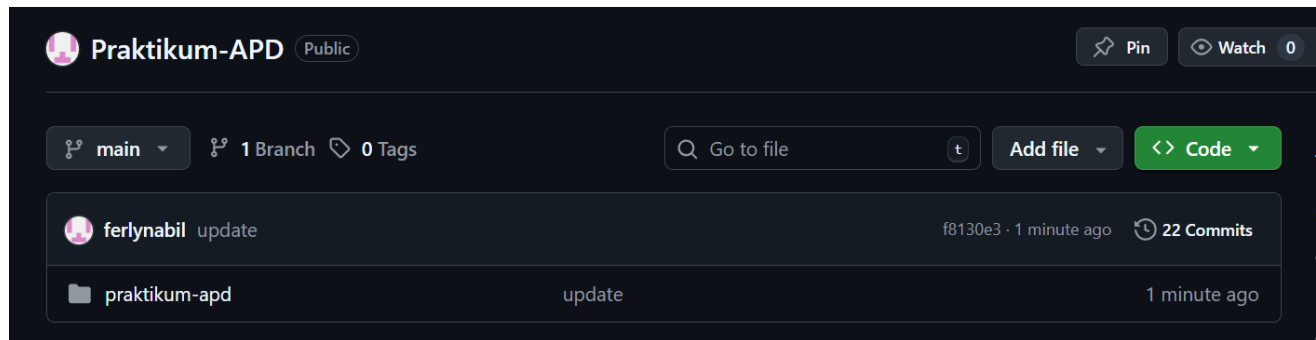
```
Asus@DESKTOP-8GBFI02 MINGW64 /d/(KULIAH)/SEMESTER 1/praktikum Algoritma Pemogram
an Dasar (main)
$ git remote add origin https://github.com/ferlynabil/Praktikum-APD.git
```

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: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 16 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (22/22), 2.20 MiB | 496.00 KiB/s, done.
Total 22 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/ferlynabil/Praktikum-APD.git
1c6f73c..170b3a8 main -> main
```

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



Semua Folder sudah ter *upload* pada Github