

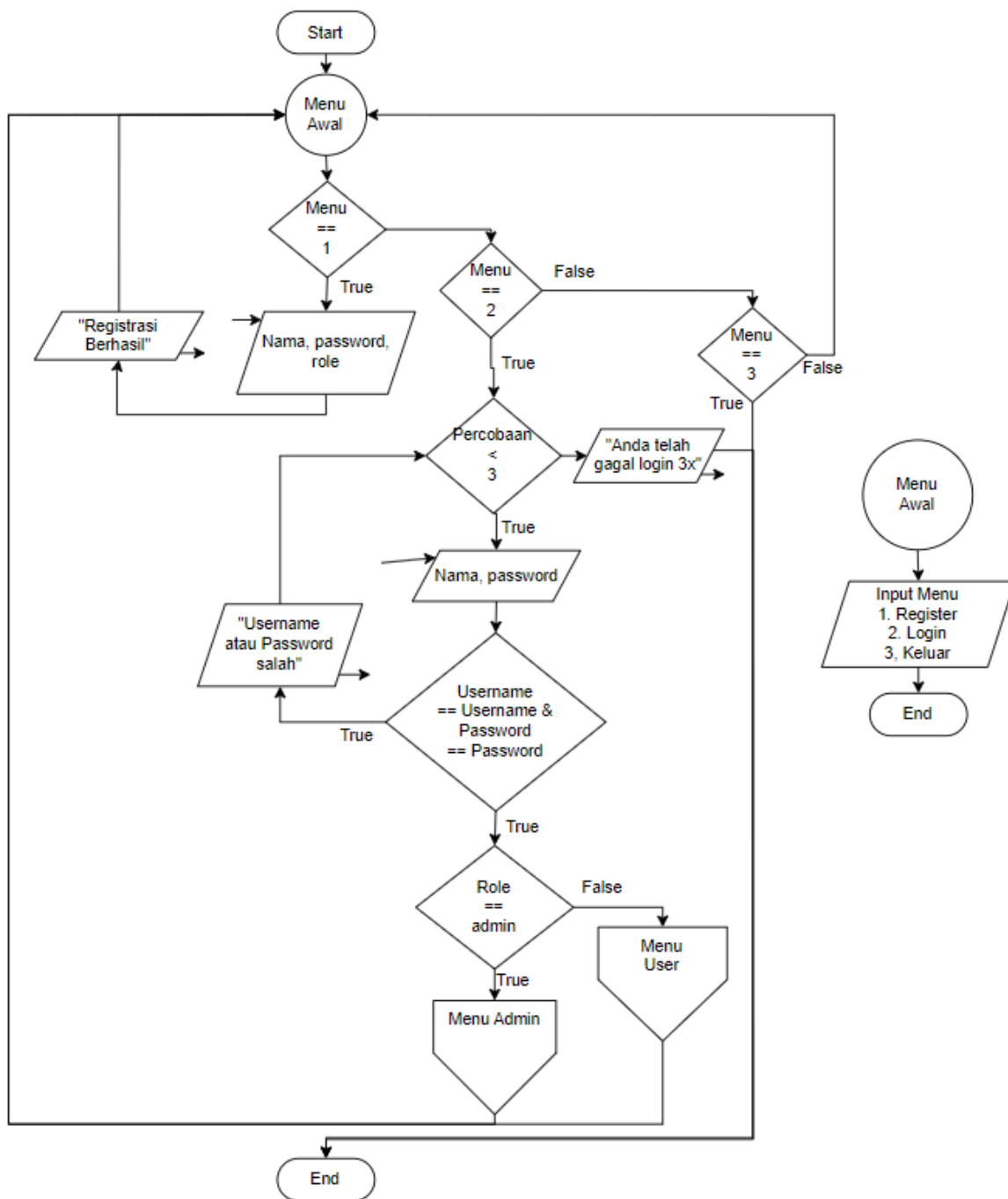
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



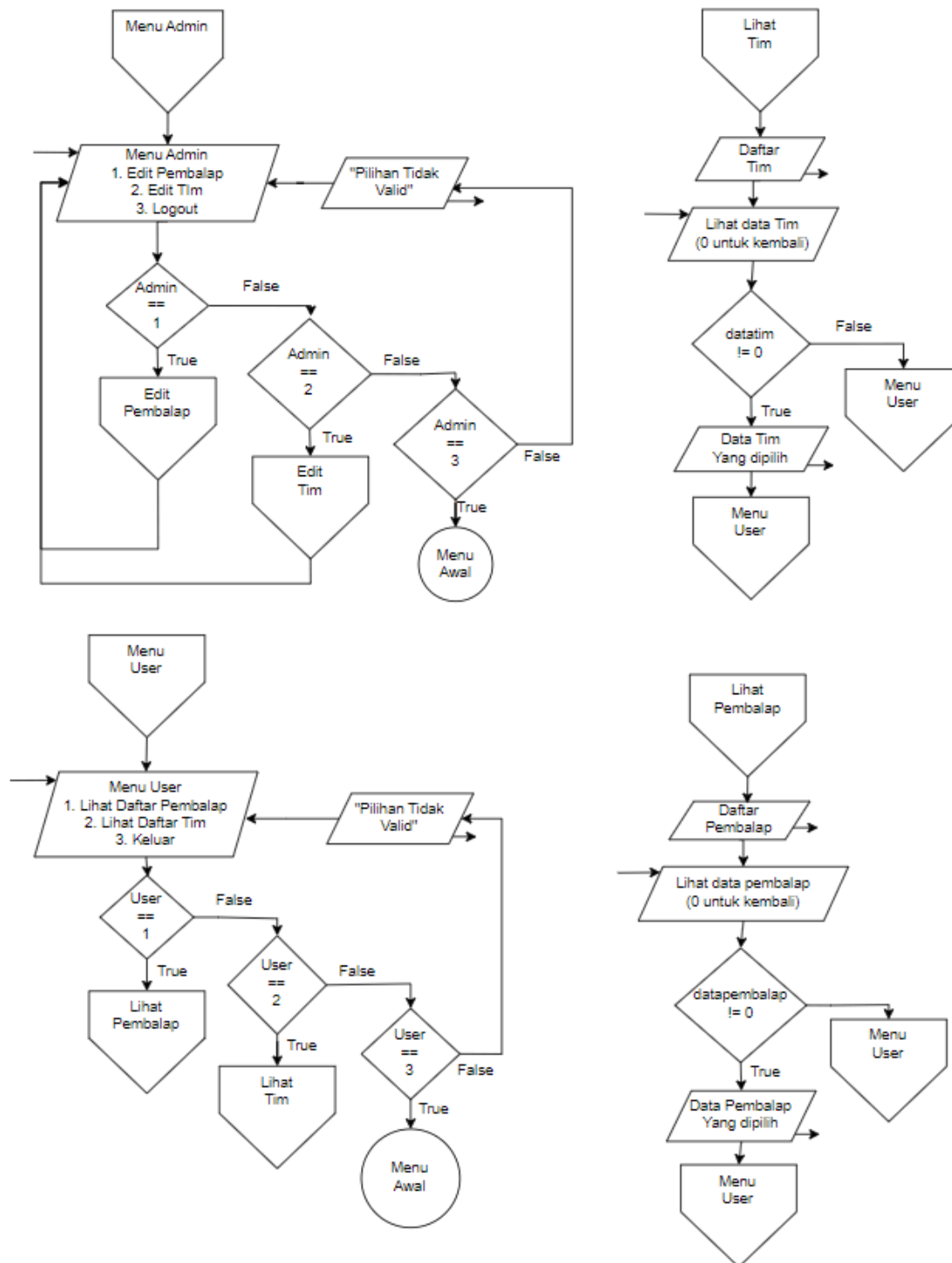
**Disusun oleh:**  
**Wahyu Aditya (2409106067)**  
**Kelas (B1 '24)**

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

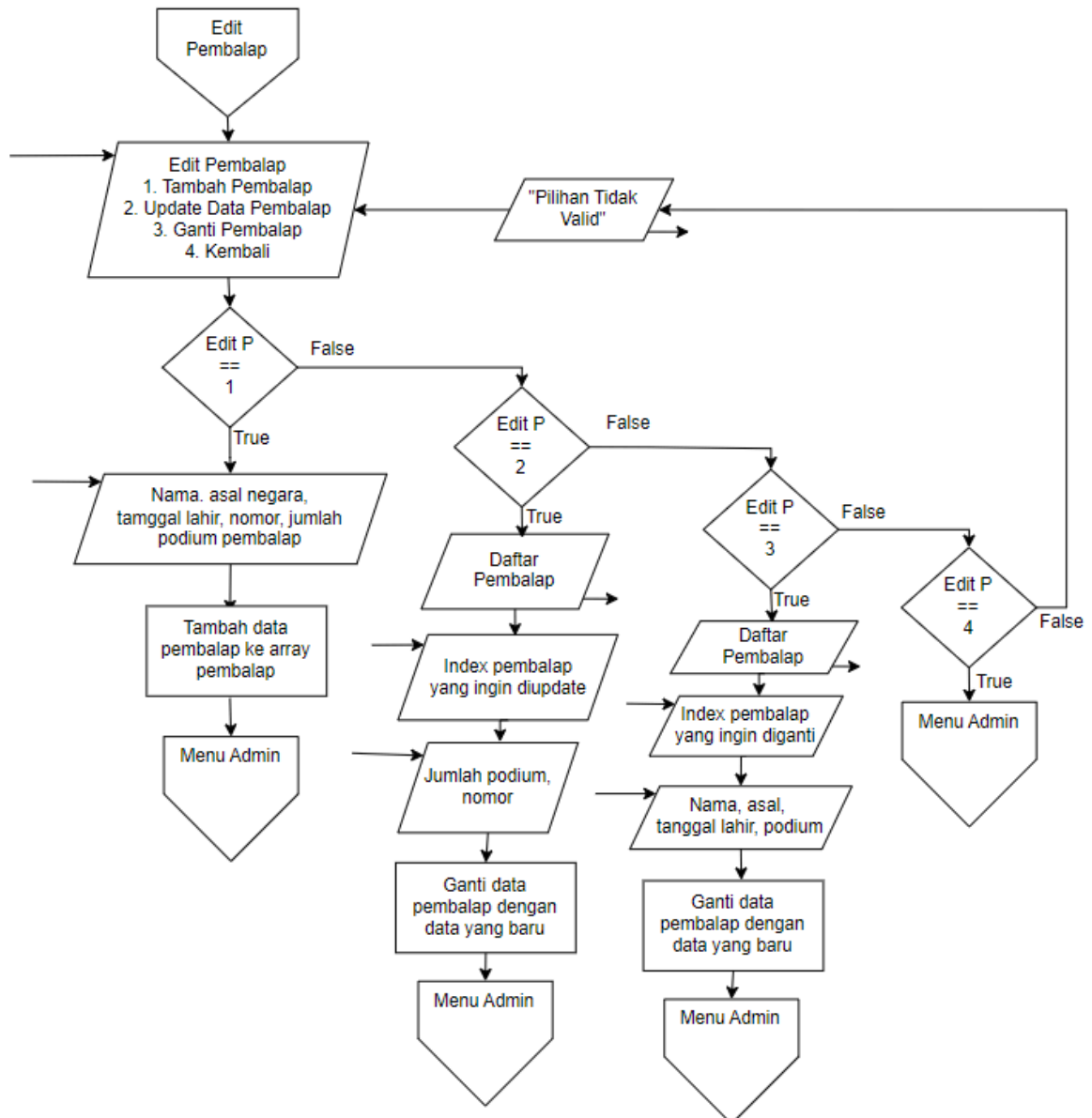
## 1. Flowchart



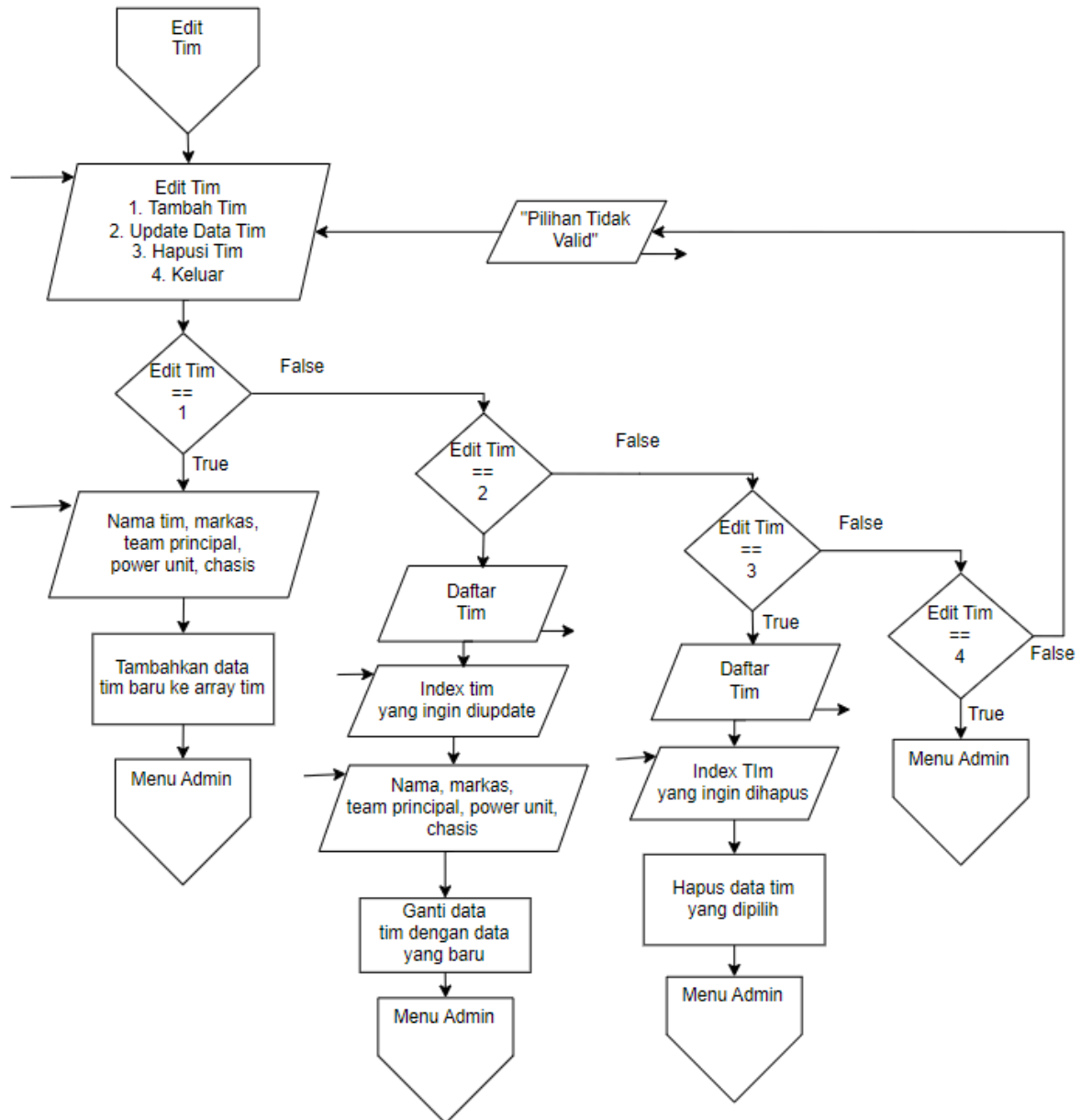
Gambar 1.1 Flowchart



Gambar 1.2 Fungsi Flowchart 2



Gambar 1.3 Fungsi Flowchart 3



Gambar 1.4 Fungsi Flowchart 4

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program yang dibuat pada post test kedua kali ini adalah program crud (create, read, update, delete) dari tema yang sudah dipilih. Tema yang dipilih praktikan adalah sistem pendataan pembalap dan tim formula 1. Sistem ini memungkinkan pengguna untuk melihat data pembalap dan tim yang sedang berlomba di kejuaraan formula 1, sementara admin bisa mengedit data-data tersebut.

## 2.2 Penjelasan Alur & Algoritma

Pada awal program terdapat tiga pilihan yaitu registrasi, login, dan keluar. Jika pengguna memilih keluar, maka program akan berhenti. Sementara jika pengguna memilih registrasi, maka sistem akan meminta pengguna untuk membuat username dan password. Sementara di menu login, sistem akan meminta pengguna untuk memasukkan username dan password. Setelah itu sistem akan memeriksa apakah username dan password yang dimasukkan benar, jika salah maka pengguna akan diminta untuk memasukkan username dan password ulang. Terdapat batasan kesalahan jika pengguna salah memasukkan username dan password yaitu sebanyak tiga kali. Jika salah sudah lebih dari tiga kali maka program akan berhenti, jika belum maka sistem akan meminta pengguna memasukkan username dan password lagi. Sementara jika username dan password yang dimasukkan benar, sistem akan memeriksa lagi role yang dimiliki pengguna, apakah rolenya admin atau user. Jika rolenya admin, jalankan menu admin. Jika rolenya user, jalankan menu user.

Pada menu admin sendiri terdapat tiga pilihan, yaitu edit pembalap, edit tim, dan keluar. Jika memilih keluar maka program akan keluar dari menu admin dan kembali ke menu awal. Sementara jika memilih pilihan edit pembalap, terdapat pilihan lagi yaitu tambah pembalap, update data pembalap, ganti pembalap, dan kembali. Sementara jika memilih pilihan edit tim, akan terdapat pilihan yaitu tambah tim, update data tim, hapus tim, dan kembali. Pada kedua pilihan, baik itu edit pembalap dan edit tim, jika memilih kembali maka sistem akan kembali ke menu admin

Pada pilihan edit pembalap, Jika memilih tambah pembalap, admin akan diminta untuk memasukkan data pembalap seperti nama, tim, nomor, asal negara, tanggal lahir, dan jumlah podium dari pembalap. Jika memilih update data pembalap, sistem akan menampilkan daftar pembalap, setelah itu admin akan diminta memasukkan indeks nomor pembalap yang ingin diupdate datanya, dan setelah itu admin diminta memasukkan data baru, dalam hal ini data pembalap yang bisa diupdate hanya jumlah podium dan nomor pembalap saja karena data yang lain tidak akan berubah. Jika memilih ganti pembalap, sistem akan menampilkan daftar pembalap dan admin diminta memasukkan indeks nomor pembalap yang ingin diganti, setelah itu admin akan memasukkan data seperti nama, nomor, asal negara tanggal lahir dan jumlah podium dari pembalap. Sistem ganti pembalap ini berfungsi karena jumlah pembalap dalam satu musim hanya bisa bertambah (jika ada tim baru) dan tidak bisa berkurang. Jadi daripada membuat sistem hapus pembalap, lebih baik membuat sistem ganti pembalap.

Pada pilihan edit tim, jika memilih tambah tim, admin akan diminta untuk memasukkan data tim baru seperti nama, markas, team principal, power unit, dan chasis. Sementara jika memilih update data tim, sistem akan menampilkan daftar tim yang ada dan admin diminta memasukka indeks nomor tim yang ingin diupdate, setelah itu admin bisa mengubah data-data tim yang ada. Sementara jika memilih pilihan hapus tim, sistem akan menampilkan list tim yang ada dan admin akan diminta untuk memasukkan indeks nomor tim yang ingin dihapus, setelah itu tim yang dipilih akan terhapus.

Sementara pada menu user sendiri terdapat tiga pilihan yaitu lihat daftar pembalap, lihat daftar tim, dan logout. Jika user memilih logout, program menu user akan berhenti dan program akan kembali ke menu awal. Jika user memilih lihat daftar pembalap, sistem akan menampilkan list pembalap yang ada, setelah itu user akan diberi pilihan apakah ingin melihat data salah satu pembalap, jika iya user diminta memasukkan indeks dari pembalap tersebut dan data dari pembalap tersebut akan ditampilkan, jika tidak maka program akan kembali ke menu user. Sementara jika user memilih lihat daftar tim, sistem akan menampilkan list tim yang ada, setelah itu user akan diberi pilihan apakah ingin melihat data salah satu tim, jika iya user diminta memasukkan indeks dari tim tersebut dan data dari tim tersebut akan ditampilkan, jika tidak maka program akan kembali ke menu user.

### 3. Source Code

#### A. Menu Awal

Menu ini akan tampil saat pengguna baru menjalankan program.

#### Source Code:

```
int login () {
    system("cls");
    string inputpin;
    int percobaan = 0;
    while (percobaan < maks) {
        cout << "Selamat Datang di ATM" << endl;
        cout << "Masukkan Pin Anda : ";
        cin >> inputpin;
        if (inputpin == pin) {
            return true;
        }
    }
}
```

```

    }
    else {
        cout << "Pin Salah" << endl;
        percobaan++;
    }
}
return false;
}

```

Gambar 3.1 Menu Awal

## B. Fitur Registrasi

Fitur ini berfungsi untuk registrasi pengguna.

Source code:

```

if (pilihanmenu == "1") {
    if (jumlahuser >= maxuser) {
        cout << "Maksimum User Tercapai" << endl;
        continue;
    }
    cout << "\n=== Registrasi ===\n";
    cout << "Masukkan username: ";
    cin >> usernames[jumlahuser];
    cout << "Masukkan password: ";
    cin >> passwords[jumlahuser];

    cout << "Set role (user/admin): ";
    cin >> roles[jumlahuser];

    if (roles[jumlahuser] != "admin") {
        roles[jumlahuser] = "user";
    }

    jumlahuser++;
    cout << "Registrasi Berhasil" << endl;
    string lanjut;
    cout << "Enter Untuk Melanjutkan...";
    cin.ignore();
    cin.get();
    system("cls");
    continue;
}

```



Gambar 3.2 Fitur Registrasi

### C. Fitur Login

Fitur ini digunakan jika pengguna ingin login ke dalam program.

**Source code:**

```
else if (pilihanmenu == "2") {
    int percobaan = 0;
    bool loginberhasil = false;

    while (percobaan < 3) {
        string username, password;
        cout << "=== Login ===" << endl;
        cout << "Masukkan Username : ";
        cin >> username;
        cout << "Masukkan Password : ";
        cin >> password;
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");

        for (int i = 0; i < jumlahuser; i++) {
            if (usernames[i] == username && passwords[i] == password) {
                loginberhasil = true;

                if (roles[i] == "admin") {
                    //program admin }

                else {
                    //program user }
                }
            }
        }
    }
}
```

Gambar 3.3 Fitur Login

### D. Menu Admin

Menu akan ditampilkan jika yang login adalah admin.

**Source Code:**

```

if (roles[i] == "admin") {
while (true) {
string pilihanadmin;
cout << "Menu Admin" << endl;
cout << "1. Edit Pembalap" << endl;
cout << "2. Edit Tim" << endl;
cout << "3. Logout" << endl;
cout << "Pilih menu : ";
cin >> pilihanadmin;
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");

```

Gambar 3.4 Menu Admin

#### E. Pilihan Menu Edit Pembalap

Menu ini akan ditampilkan jika admin ingin mengedit pembalap.

##### Source Code:

```

if (pilihanadmin == "1") {
string pilihanpembalap;
cout << "\n1. Tambah Pembalap\n";
cout << "2. Update Data Pembalap\n";
cout << "3. Ganti Pembalap\n";
cout << "4. Kembali\n";
cout << "Masukkan Pilihan: ";
cin >> pilihanpembalap;
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");

```

Gambar 3.5 Menu Edit Pembalap

#### F. Pilihan Menu Tambah Pembalap

Fitur ini digunakan jika admin ingin menambahkan pembalap.

##### Source Code:

```

if (pilihanpembalap == "1") {
    if (jumlahpembalap >= maxpembalap) {
        cout << "\nKapasitas pembalap penuh! Tidak bisa menambah lagi.\n";
    } else {
        cout << "\nMasukkan data pembalap baru:\n";
        cin.ignore();
        cout << "Nama          : "; getline(cin, pembalap[jumlahpembalap][0]);
        cout << "Tim            : "; getline(cin, pembalap[jumlahpembalap][1]);
        cout << "Asal Negara   : "; getline(cin, pembalap[jumlahpembalap][2]);
        cout << "Nomor         : "; getline(cin, pembalap[jumlahpembalap][3]);
        cout << "Tanggal Lahir: "; getline(cin, pembalap[jumlahpembalap][4]);
        cout << "Jumlah Podium: "; getline(cin, pembalap[jumlahpembalap][5]);

        jumlahpembalap++;
        cout << "Pembalap berhasil ditambahkan!\n";
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    }
}

```

Gambar 3.6 Tambah Pembalap

## G. Pilihan Menu Update Data Pembalap

Fitur ini digunakan jika admin ingin mengupdate data pembalap.

### Source Code:

```

else if (pilihanpembalap == "2") {
    int updatepembalap;
    cout << "\nDaftar Pembalap F1 2025:\n";
    for (int i = 0; i < jumlahpembalap; i++) {
        cout << i + 1 << ". " << pembalap[i][0] << " (" << pembalap[i][1] << ")\n";
    }
    cout << "\nMasukkan nomor pembalap yang ingin diupdate (0 untuk kembali): ";
    cin >> updatepembalap;

    if (updatepembalap >= 1 && updatepembalap <= jumlahpembalap) {
        int index = updatepembalap - 1;
        cin.ignore();
        cout << "\nMasukkan data baru:\n";
        cout << "Nomor : "; getline(cin, pembalap[index][3]);
    }
}

```

```

cout << "Jumlah Podium : "; getline(cin, pembalap[index][5]);

cout << "Data pembalap berhasil diperbarui!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
} else if (updatepembalap != 0) {
cout << "Pilihan tidak valid!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}
}

```

Gambar 3.7 Update Data Pembalap

## H. Pilihan Menu Ganti Pembalap

Fitur ini digunakan jika admin ingin mengganti pembalap di suatu tim dengan pembalap yang lain.

### Source Code:

```

else if (pilihanpembalap == "3") {
int gantipembalap;
cout << "\nDaftar Pembalap F1 2025:\n";
for (int i = 0; i < jumlahpembalap; i++) {
cout << i + 1 << ". " << pembalap[i][0] << " (" << pembalap[i][1] << ")\n";
}
cout << "\nMasukkan nomor pembalap yang ingin diganti (0 untuk kembali): ";
cin >> gantipembalap;

if (gantipembalap >= 1 && gantipembalap <= jumlahpembalap) {
int index = gantipembalap - 1;
cin.ignore();
cout << "\nMasukkan data pembalap:\n";
cout << "Nama          : "; getline(cin, pembalap[index][0]);
cout << "Asal Negara    : "; getline(cin, pembalap[index][2]);
cout << "Nomor         : "; getline(cin, pembalap[index][3]);
cout << "Tanggal Lahir: "; getline(cin, pembalap[index][4]);
}
}

```

```

cout << "Jumlah Podium: "; getline(cin, pembalap[index][5]);

cout << "Pembalap berhasil diganti!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
} else if (gantipembalap != 0) {
cout << "Pilihan tidak valid!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}
}

```

Gambar 3.8 Ganti Pembalap

### I. Pilihan Menu Edit Tim

Menu ini digunakan jika admin ingin mengedit tim.

#### Source Code:

```

else if (pilihanadmin == "2") {
string pilihantim;
cout << "\n1. Tambah Tim\n";
cout << "2. Update Data Tim\n";
cout << "3. Hapus Tim\n";
cout << "4. Kembali\n";
cout << "Masukkan Pilihan: ";
cin >> pilihantim;
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}

```

Gambar 3.9 Menu Edit Tim

### J. Pilihan Menu Tambah Tim

Fitur ini digunakan jika admin ingin menambah tim.

### Source Code:

```
if (pilihantim == "1") {
    if (jumlahtim >= maxtim) {
        cout << "\nJumlah tim sudah penuh!" << endl;
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    }

    cout << "Masukkan detail tim baru:" << endl;
    cout << "Nama Tim      : "; getline(cin, tim[jumlahtim][0]);
    cout << "Markas        : "; getline(cin, tim[jumlahtim][1]);
    cout << "Team Principal: "; getline(cin, tim[jumlahtim][2]);
    cout << "Power Unit   : "; getline(cin, tim[jumlahtim][3]);
    cout << "Chassis      : "; getline(cin, tim[jumlahtim][4]);
    jumlahtim++;
    cout << "Tim berhasil ditambahkan!\n";
    string lanjut;
    cout << "Enter Untuk Melanjutkan...";
    cin.ignore();
    cin.get();
    system("cls");
}
```

Gambar 3.10 Tambah Tim

### K. Pilihan Menu Update Data Tim

Fitur ini digunakan jika admin ingin mengupdate data tim.

### Source Code:

```
else if (pilihantim == "2") {
    int choice;
    cout << "\nDaftar Tim F1 2025:\n";
    for (int i = 0; i < jumlahtim; i++) {
        cout << i + 1 << ". " << tim[i][0] << "\n";
    }
    cout << "\nMasukkan nomor tim yang ingin diperbarui (0 untuk kembali): ";
    cin >> choice;

    if (choice >= 1 && choice <= jumlahtim) {
```

```

int index = choice - 1;
cout << "\nMasukkan data baru untuk tim " << tim[index][0] << ":\n";
cout << "Nama Tim      : "; cin.ignore(); getline(cin, tim[index][0]);
cout << "Markas        : "; getline(cin, tim[index][1]);
cout << "Team Principal: "; getline(cin, tim[index][2]);
cout << "Power Unit     : "; getline(cin, tim[index][3]);
cout << "Chassis        : "; getline(cin, tim[index][4]);
cout << "Tim berhasil diperbarui!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
} else if (choice != 0) {
cout << "Pilihan tidak valid!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}
}

```

Gambar 3.11 Update Data Tim

## L. Pilihan Menu Hapus Tim

Fitur ini digunakan jika admin ingin menghapus suatu tim.

### Source Code:

```

else if (pilihantim == "3") {
int choice;
cout << "\nDaftar Tim F1 2025:\n";
for (int i = 0; i < jumlahtim; i++) {
cout << i + 1 << ". " << tim[i][0] << "\n";
}
cout << "\nMasukkan nomor tim yang ingin dihapus (0 untuk kembali): ";
cin >> choice;

if (choice >= 1 && choice <= jumlahtim) {
int index = choice - 1;
for (int i = index; i < jumlahtim - 1; i++) {
for (int j = 0; j < 5; j++) {
tim[i][j] = tim[i + 1][j];
}
}
}
}

```

```

}
}
jumahtim--;
cout << "Tim berhasil dihapus!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
} else if (choice != 0) {
cout << "Pilihan tidak valid!\n";
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}
}
}

```

Gambar 3.12 Update Data Tim

## M. Menu User

Fitur ini digunakan jika pengguna login sebagai user

### Source Code:

```

else {
while (true) {
string pilihanuser;
cout << "\nMenu User:\n";
cout << "1. Lihat daftar pembalap F1 2025\n";
cout << "2. Lihat daftar tim F1 2025\n";
cout << "3. Keluar ke menu utama\n";
cout << "Pilih menu: ";
cin >> pilihanuser;
string lanjut;
cout << "Enter Untuk Melanjutkan...";
cin.ignore();
cin.get();
system("cls");
}
}
}

```

Gambar 3.13 Menu User



## N. Pilihan Lihat Daftar Pembalap

Fitur ini digunakan jika user ingin melihat daftar pembalap yang ada.

### Source Code:

```
if (pilihanuser == "1") {
    cout << "\nDaftar Pembalap F1 2025:\n";
    for (int i = 0; i < jumlahpembalap; i++) {
        cout << i + 1 << ". " << pembalap[i][0] << " (" << pembalap[i][1] << ")\n";
    }
    int pilihandatapembalap;
    cout << "\nMasukkan nomor pembalap untuk melihat detail (0 untuk kembali): ";
    cin >> pilihandatapembalap;
    string lanjut;
    cout << "Enter Untuk Melanjutkan...";
    cin.ignore();
    cin.get();
    system("cls");
    if (pilihandatapembalap >= 1 && pilihandatapembalap <= jumlahpembalap) {
        int index = pilihandatapembalap - 1;
        cout << "\nDetail Pembalap:\n";
        cout << "Nama          : " << pembalap[index][0] << "\n";
        cout << "Tim            : " << pembalap[index][1] << "\n";
        cout << "Asal Negara: " << pembalap[index][2] << "\n";
        cout << "Nomor         : " << pembalap[index][3] << "\n";
        cout << "Tanggal Lahir: " << pembalap[index][4] << "\n";
        cout << "Jumlah Podium: " << pembalap[index][5] << "\n";
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    } else if (pilihandatapembalap != 0) {
        cout << "Pilihan tidak valid!\n";
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    }
}
```

Gambar 3.14 Lihat Daftar Pembalap

## O. Pilihan Lihat Daftar Tim

Fitur ini digunakan jika user ingin melihat daftar tim yang ada.

### Source Code:

```
else if (pilihanuser == "2") {
    cout << "\nDaftar Tim F1 2025:\n";
    for (int i = 0; i < jumlahtim; i++) {
        cout << i + 1 << ". " << tim[i][0] << "\n";
    }
    int pilihandatatim;
    cout << "\nMasukkan nomor tim untuk melihat detail (0 untuk kembali): ";
    cin >> pilihandatatim;
    string lanjut;
    cout << "Enter Untuk Melanjutkan...";
    cin.ignore();
    cin.get();
    system("cls");
    if (pilihandatatim >= 1 && pilihandatatim <= jumlahtim) {
        int index = pilihandatatim - 1;
        cout << "\nDetail Tim:\n";
        cout << "Nama Tim      : " << tim[index][0] << "\n";
        cout << "Markas        : " << tim[index][1] << "\n";
        cout << "Team Principal : " << tim[index][2] << "\n";
        cout << "Power Unit     : " << tim[index][3] << "\n";
        cout << "Chassis        : " << tim[index][4] << "\n";
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    } else if (pilihandatatim != 0) {
        cout << "Pilihan tidak valid!\n";
        string lanjut;
        cout << "Enter Untuk Melanjutkan...";
        cin.ignore();
        cin.get();
        system("cls");
    }
}
```

Gambar 3.15 Lihat Daftar Tim

## **4. Uji Coba dan Hasil Output**

### **4.1 Uji Coba**

1. Skenario 1

Pengguna melakukan registrasi

2. Skenario 2

Pengguna gagal login sebanyak 3 kali

3. Skenario 3

Pengguna berhasil login sebagai admin

4. Skenario 4

Pengguna berhasil login sebagai user

5. Skenario 5

Admin menambah pembalap

6. Skenario 6

Admin mengupdate data pembalap

7. Skenario 7

Admin mengganti pembalap

8. Skenario 8

Admin menambah tim

9. Skenario 9

Admin mengupdate data tim

10. Skenario 10

Admin menghapus tim

#### 11. Skenario 11

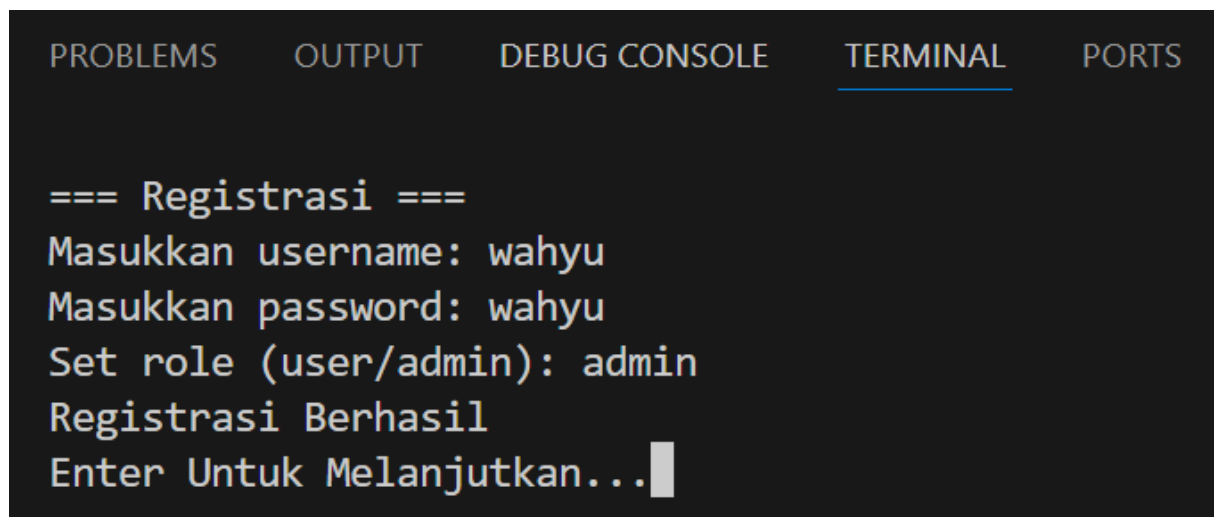
User melihat daftar pembalap

#### 12. Skenario 12

User melihat daftar tim

## 4.2 Hasil Output

### 1. Skenario 1

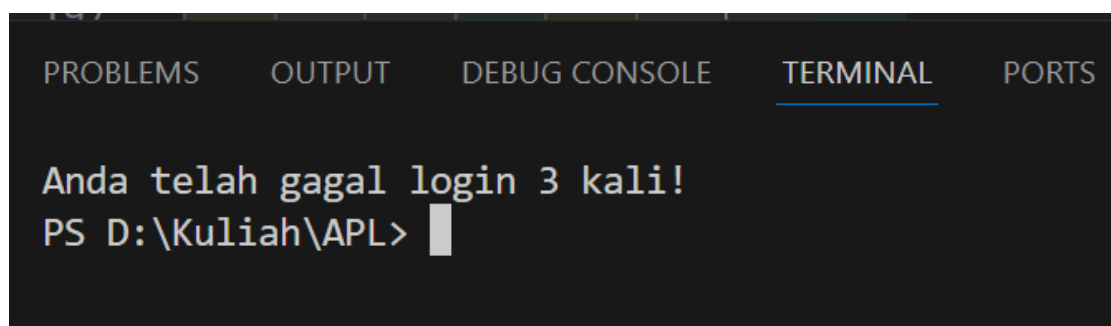


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=== Registrasi ===
Masukkan username: wahyu
Masukkan password: wahyu
Set role (user/admin): admin
Registrasi Berhasil
Enter Untuk Melanjutkan...
```

Gambar 4.2.1 Output Registrasi

### 2. Skenario 2



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Anda telah gagal login 3 kali!
PS D:\Kuliah\APL>
```

Gambar 4.2.2 Output Login Gagal

### 3. Skenario 3

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Menu Admin
1. Edit Pembalap
2. Edit Tim
3. Logout
Pilih menu : █
```

Gambar 4.2.3 Output Menu Admin

4. Skenario 4

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Menu User:
1. Lihat daftar pembalap F1 2025
2. Lihat daftar tim F1 2025
3. Keluar ke menu utama
Pilih menu: █
```

Gambar 4.2.4 Output Menu User

5. Skenario 5

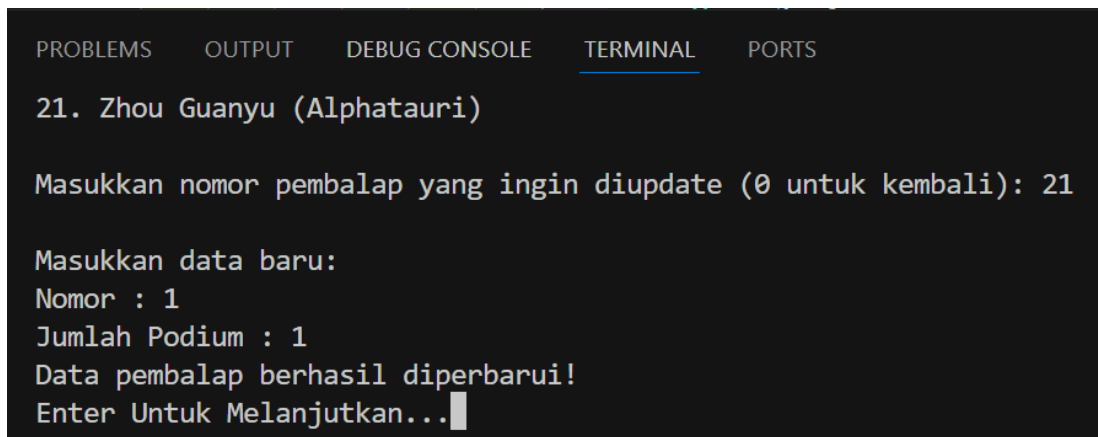
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukkan data pembalap baru:

Nama      : Zhou Guanyu
Tim       : Alphatauri
Asal Negara : China
Nomor     : 99
Tanggal Lahir: 20 Januari 2001
Jumlah Podium: 0
Pembalap berhasil ditambahkan!
Enter Untuk Melanjutkan... █
```

Gambar 4.2.5 Output Tambah Pembalap

6. Skenario 6



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

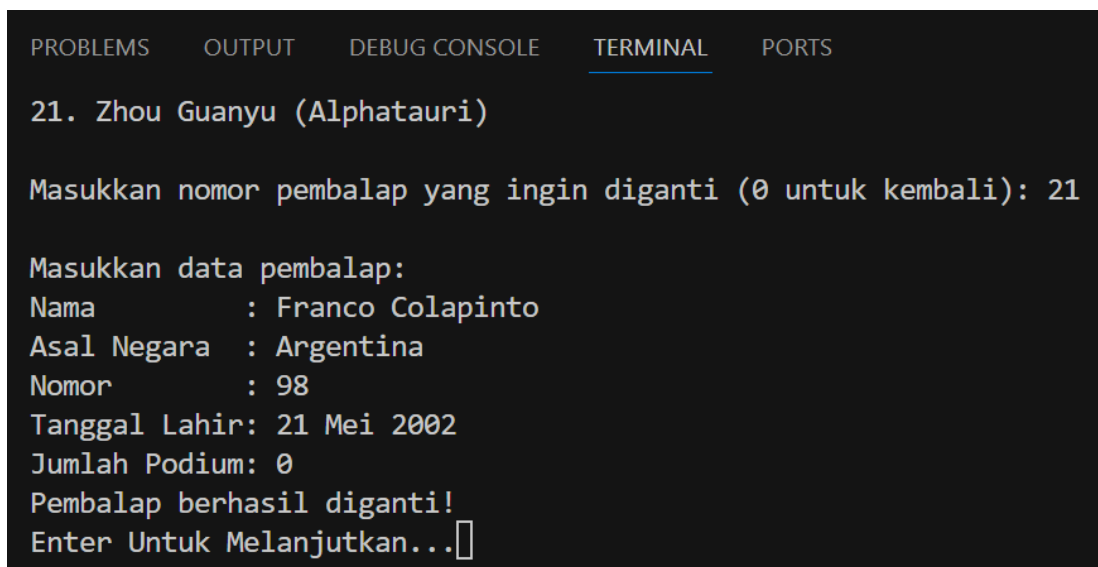
21. Zhou Guanyu (Alphatauri)

Masukkan nomor pembalap yang ingin diupdate (0 untuk kembali): 21

Masukkan data baru:
Nomor : 1
Jumlah Podium : 1
Data pembalap berhasil diperbarui!
Enter Untuk Melanjutkan...█
```

Gambar 4.2.6 Output Update Data Pembalap

7. Skenario 7



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

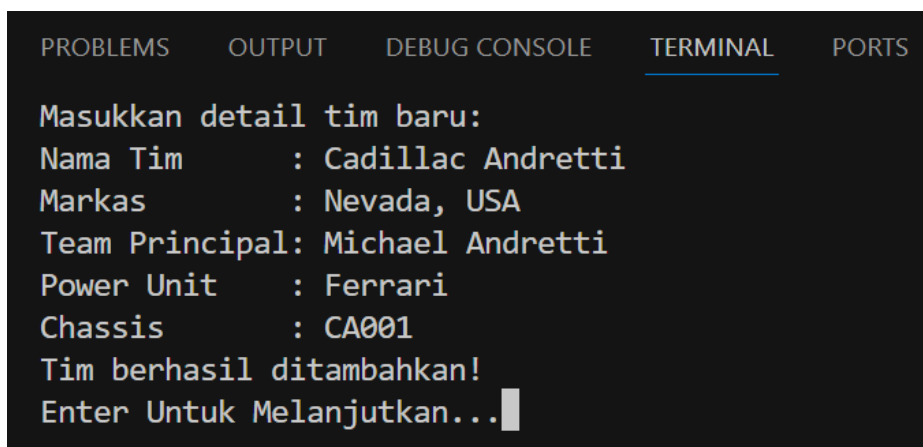
21. Zhou Guanyu (Alphatauri)

Masukkan nomor pembalap yang ingin diganti (0 untuk kembali): 21

Masukkan data pembalap:
Nama          : Franco Colapinto
Asal Negara   : Argentina
Nomor         : 98
Tanggal Lahir: 21 Mei 2002
Jumlah Podium: 0
Pembalap berhasil diganti!
Enter Untuk Melanjutkan...█
```

Gambar 4.2.7 Output Ganti Pembalap

8. Skenario 8

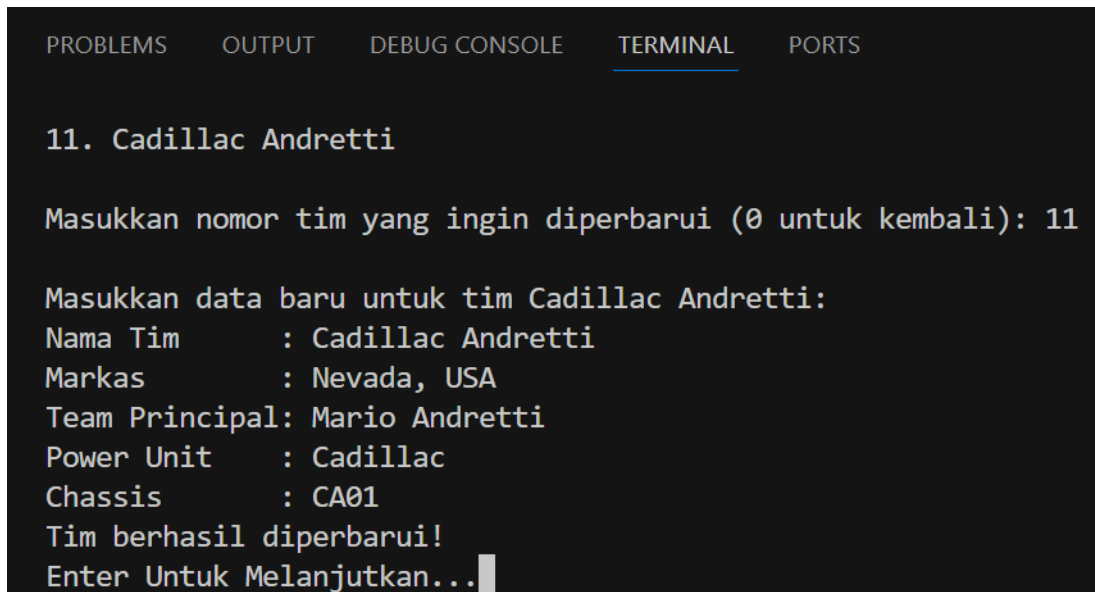


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukkan detail tim baru:
Nama Tim      : Cadillac Andretti
Markas       : Nevada, USA
Team Principal: Michael Andretti
Power Unit   : Ferrari
Chassis      : CA001
Tim berhasil ditambahkan!
Enter Untuk Melanjutkan...█
```

Gambar 4.2.8 Output Tambah Tim

## 9. Skenario 9



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

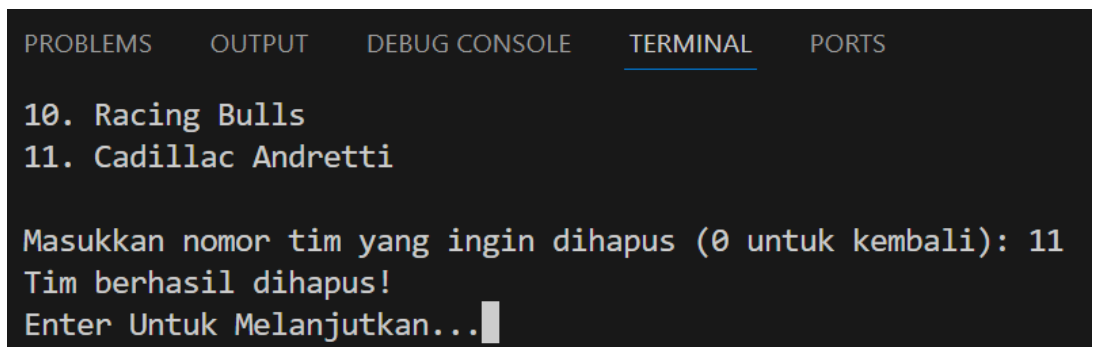
11. Cadillac Andretti

Masukkan nomor tim yang ingin diperbarui (0 untuk kembali): 11

Masukkan data baru untuk tim Cadillac Andretti:
Nama Tim      : Cadillac Andretti
Markas        : Nevada, USA
Team Principal: Mario Andretti
Power Unit     : Cadillac
Chassis        : CA01
Tim berhasil diperbarui!
Enter Untuk Melanjutkan...
```

Gambar 4.2.9 Output Update Data Tim

## 10. Skenario 10



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

10. Racing Bulls
11. Cadillac Andretti

Masukkan nomor tim yang ingin dihapus (0 untuk kembali): 11
Tim berhasil dihapus!
Enter Untuk Melanjutkan...
```

Gambar 4.2.10 Output Hapus Tim

## 11. Skenario 11

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Daftar Pembalap F1 2025:
1. Max Verstappen (Red Bull Racing)
2. Liam Lawson (Red Bull Racing)
3. Lando Norris (McLaren)
4. Oscar Piastri (McLaren)
5. Charles Leclerc (Scuderia Ferrari)
6. Lewis Hamilton (Scuderia Ferrari)
7. George Russell (Mercedes-AMG Petronas)
8. Kimi Antonelli (Mercedes-AMG Petronas)
9. Fernando Alonso (Aston Martin)
10. Lance Stroll (Aston Martin)
11. Pierre Gasly (Alpine)
12. Jack Doohan (Alpine)
13. Yuki Tsunoda (Racing Bulls)
14. Isaack Hadjar (Racing Bulls)
15. Carlos Sainz (Williams)
16. Alexander Albon (Williams)
17. Esteban Ocon (Haas)
18. Olliver Bearman (Haas)
19. Nico Hulkenberg (Sauber)
20. Gabriel Bortoleto (Sauber)
21. Franco Colapinto (Alphatauri)

Masukkan nomor pembalap untuk melihat detail (0 untuk kembali):
```

Gambar 4.2.11 Output User Lihat Daftar Pembalap

## 12. Skenario 12

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Daftar Tim F1 2025:
1. Red Bull Racing
2. Mercedes-AMG Petronas
3. Ferrari
4. McLaren
5. Aston Martin
6. Alpine
7. Williams
8. Haas
9. Sauber
10. Racing Bulls

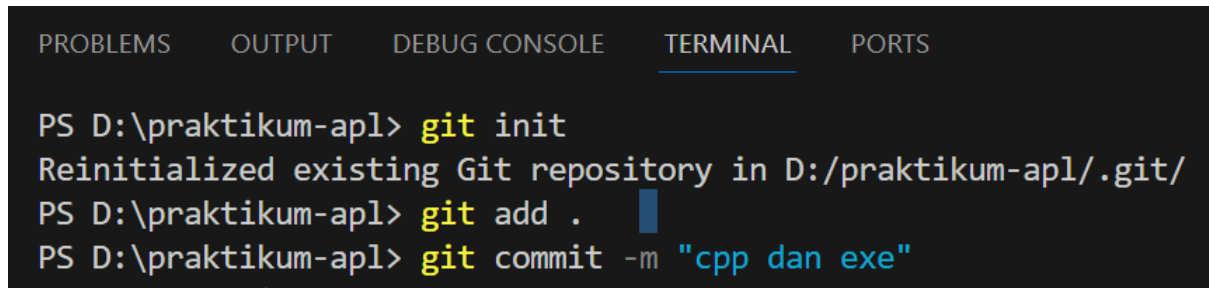
Masukkan nomor tim untuk melihat detail (0 untuk kembali):
```

Gambar 4.2.12 Output User Lihat Daftar Tim



## 5. Git

Tahap pertama yang dilakukan adalah menginisiasi git itu sendiri dengan cara mengetik “git init” pada terminal.

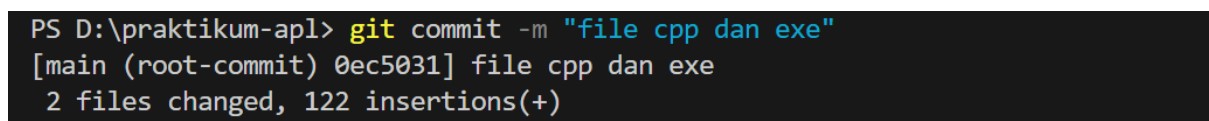


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\praktikum-apl> git init
Reinitialized existing Git repository in D:/praktikum-apl/.git/
PS D:\praktikum-apl> git add .
PS D:\praktikum-apl> git commit -m "cpp dan exe"
```

Gambar 5.1 Inisiasi Git dan git add

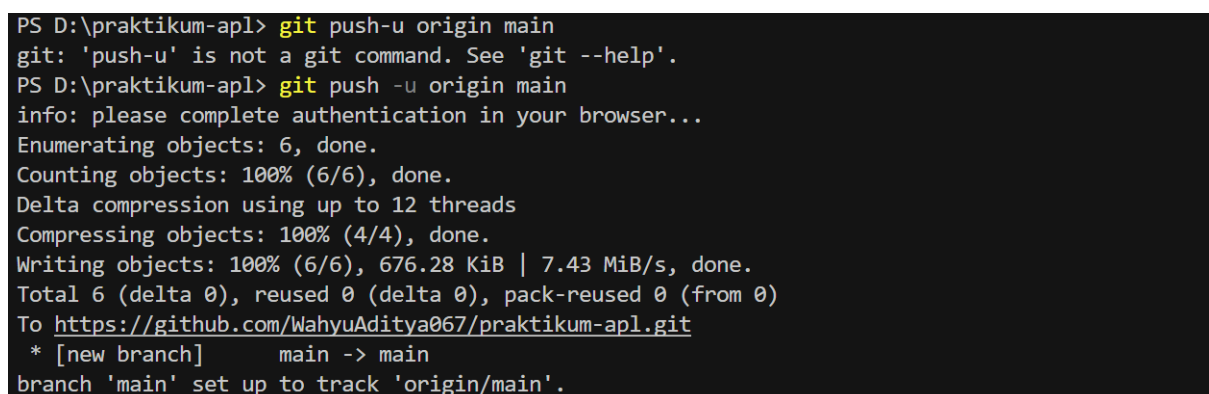
Setelah menginisiasi git, tahap selanjutnya adalah memilih file yang akan ditambahkan. Jika hanya file tertentu saja yang ingin diupload, ketik “git add (Nama file yang ingin dikirim)”, atau jika ingin langsung upload semua file maka langsung ketik “git add .”. Tambahkan juga pesan commit dengan cara mengetik “git commit -m ‘(Pesan commit)’ ”. Pesan commit sendiri dibuat untuk memudahkan mengetahui perubahan yang dilakukan.



```
PS D:\praktikum-apl> git commit -m "file cpp dan exe"
[main (root-commit) 0ec5031] file cpp dan exe
2 files changed, 122 insertions(+)
```

Gambar 5.2 Git commit

Karena git sudah dihubungkan dengan github pada posttest sebelumnya, maka langsung saja push file yang ada dari git ke repository github dengan cara ketik “git push -u origin main”.



```
PS D:\praktikum-apl> git push-u origin main
git: 'push-u' is not a git command. See 'git --help'.
PS D:\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 676.28 KiB | 7.43 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/WahyuAditya067/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.3 Git Push