

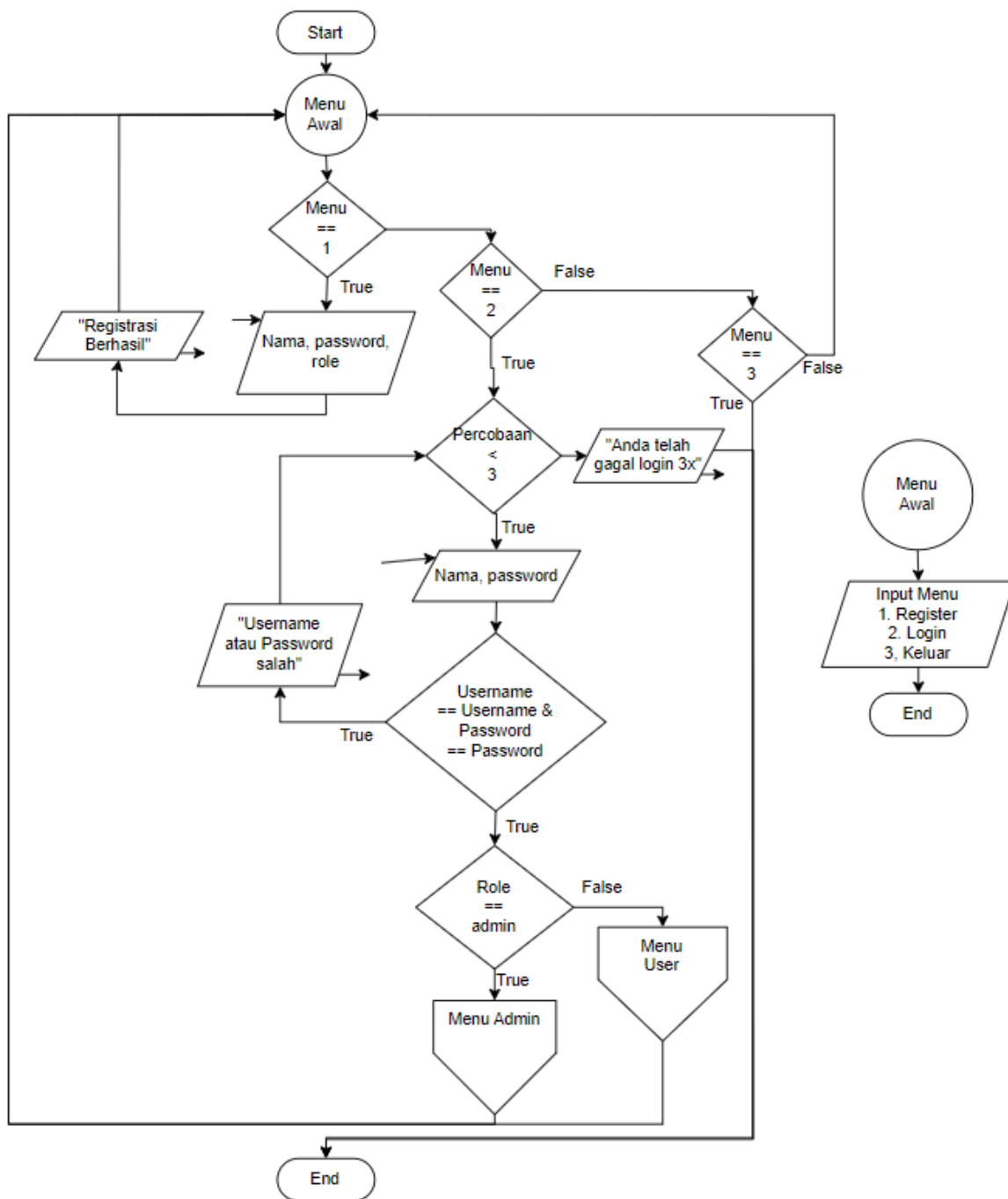
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



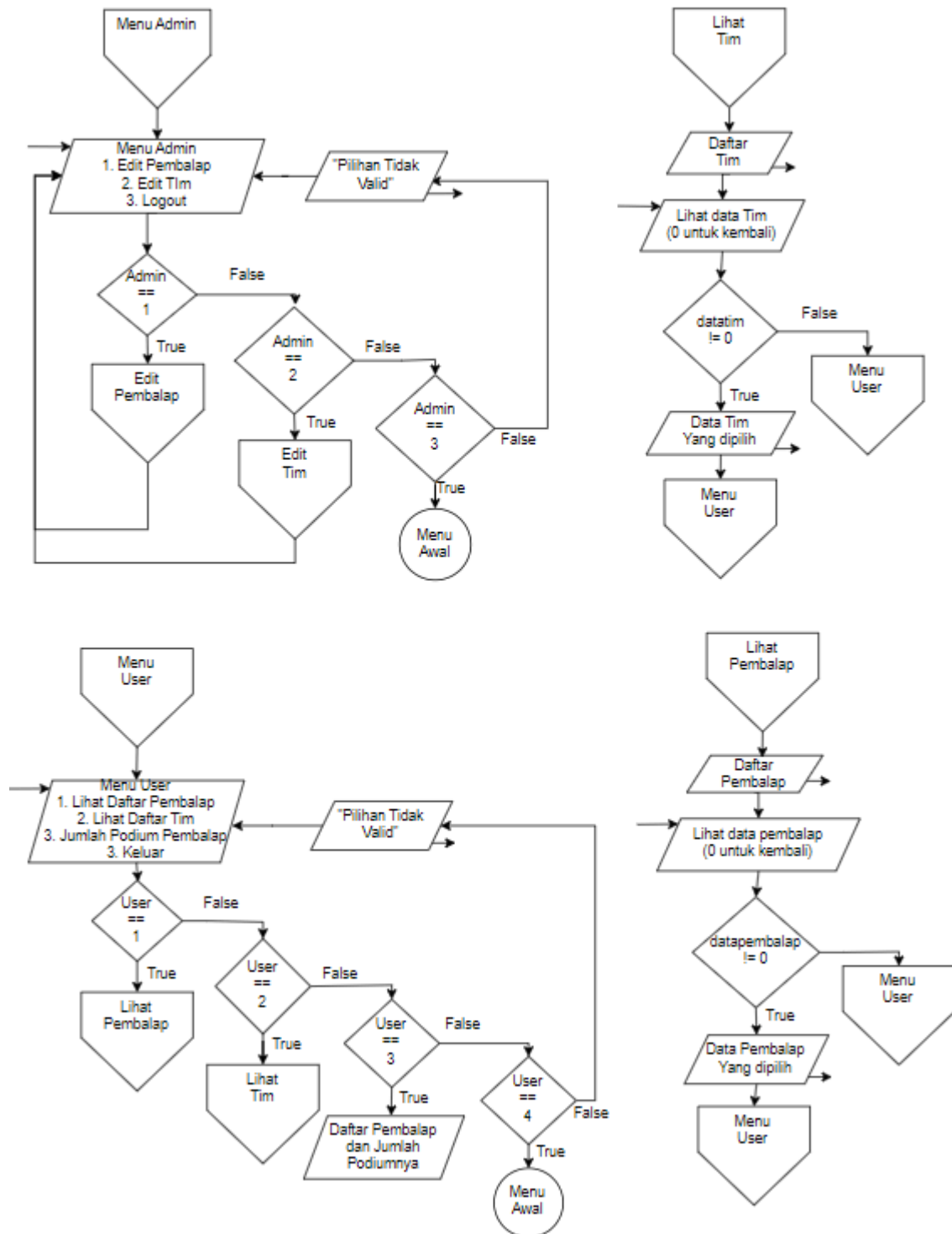
Disusun oleh:
Wahyu Aditya (2409106067)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

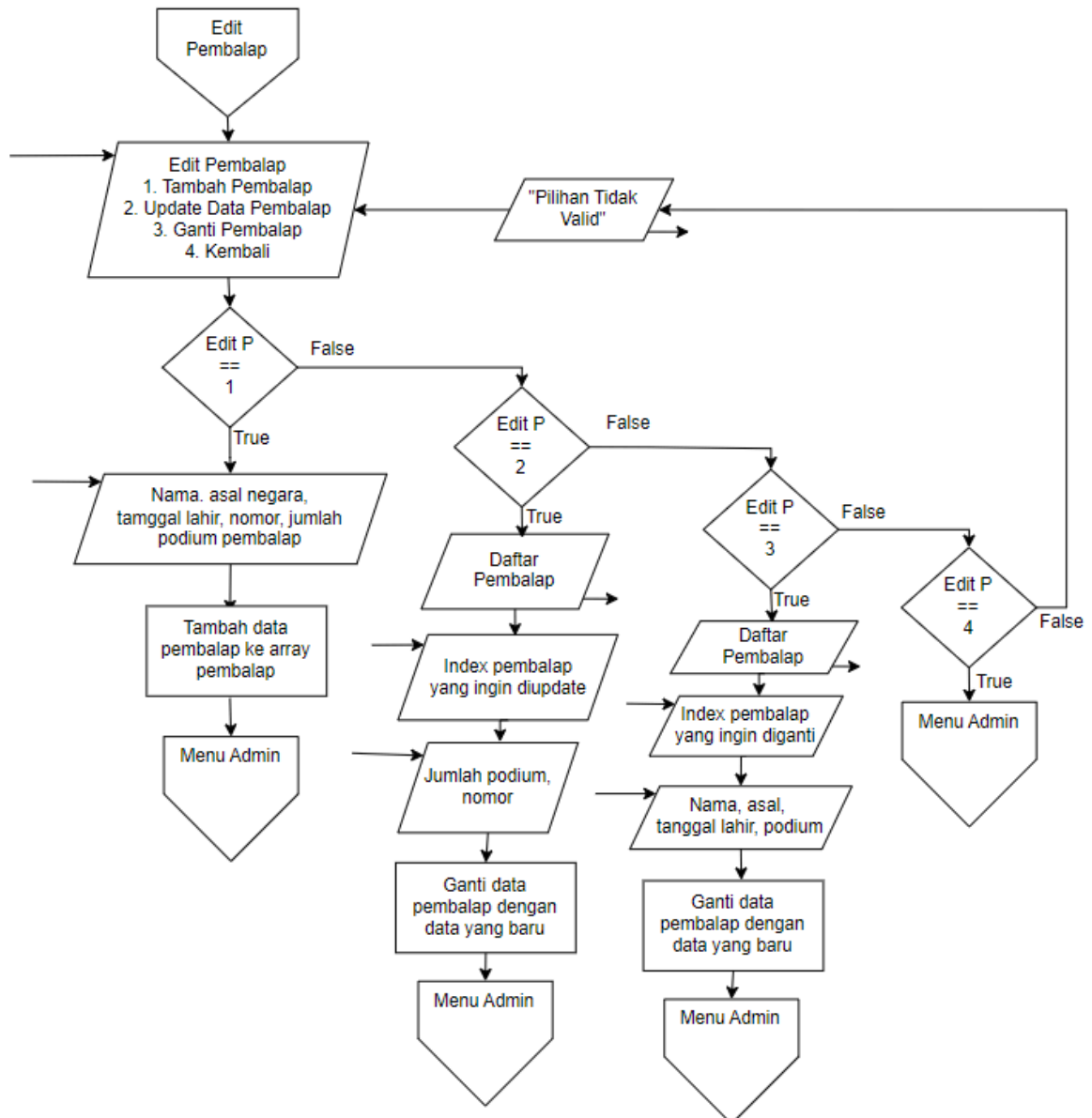
1. Flowchart



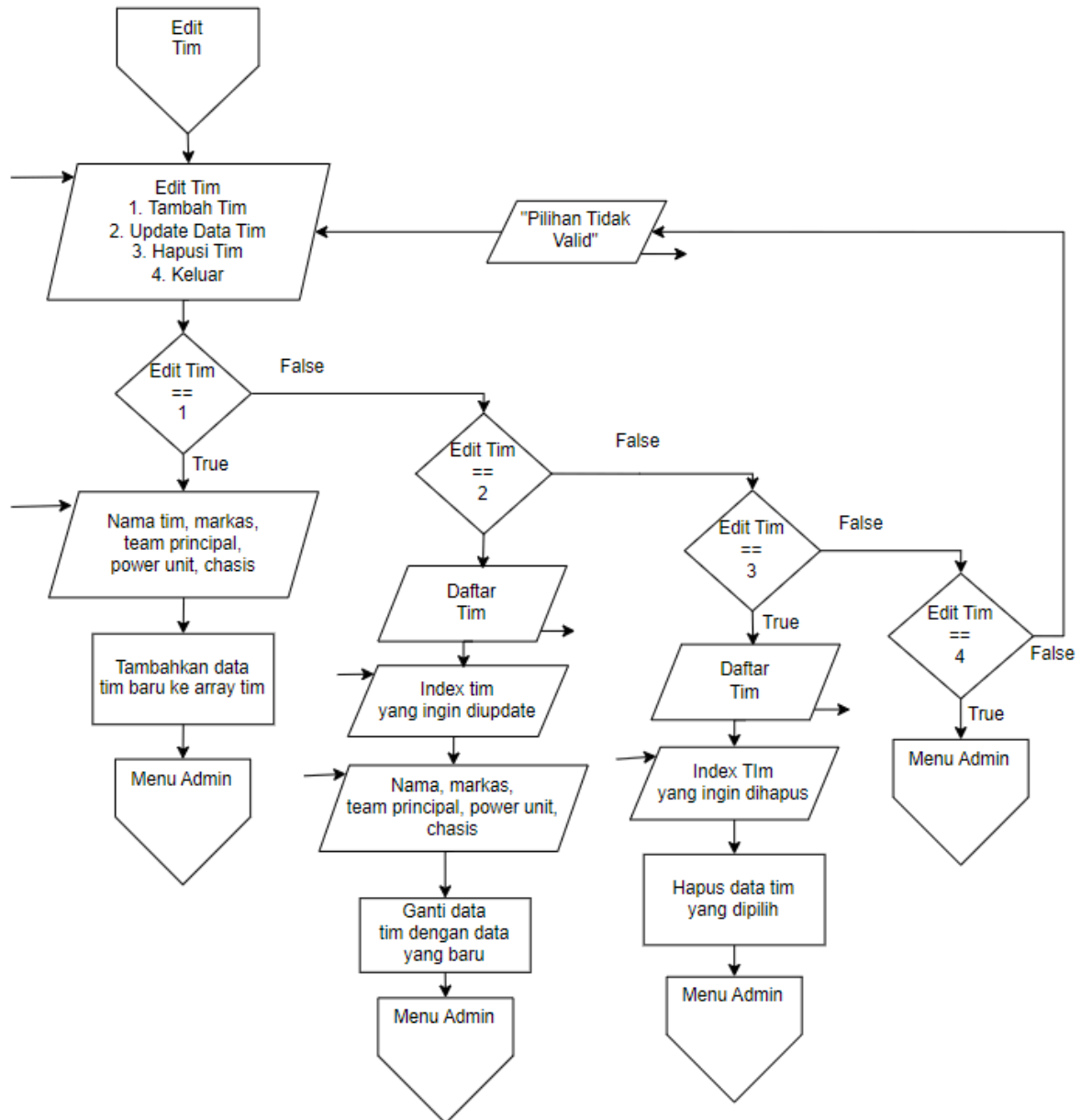
Gambar 1.1 Flowchart



Gambar 1.2 Fungsi Flowchart 2



Gambar 1.3 Fungsi Flowchart 3



Gambar 1.4 Fungsi Flowchart 4

2. Analisis Program

Program yang dibuat pada post test keenam kali ini adalah menerapkan minimal 3 sorting pada program crud (create, read, update, delete) yang sudah dibuat pada posttest sebelumnya. 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.

3. Source Code

A. Menu Awal

Menu ini akan tampil saat pengguna baru menjalankan program.

Source Code:

```
while (true) {
    string pilihanmenu;
    cout << " Selamat Datang " << endl;
    cout << " 1. Registrasi " << endl;
    cout << " 2. Login " << endl;
    cout << " 3. Keluar " << endl;
    cout << "Masukkan pilihan: ";
    cin >> pilihanmenu;
    enter();
}
```

Gambar 3.1 Menu Awal

B. Fitur Registrasi

Fitur ini berfungsi untuk registrasi pengguna.

Source code:

```
bool registrasiuser(User* users, int* jumlahuser, int maxuser) {
    if (*jumlahuser >= maxuser) {
        cout << "Maksimum User Tercapai\n";
        return false;
    }
    cout << "\n=== Registrasi ===\n";
    cout << "Masukkan username: ";
    cin >> users[*jumlahuser].username;
    cout << "Masukkan password: ";
    cin >> users[*jumlahuser].password;
    cout << "Set role (user/admin): ";
    cin >> users[*jumlahuser].role;
    if (users[*jumlahuser].role != "admin") {
        users[*jumlahuser].role = "user";
    }
    (*jumlahuser)++;
    cout << "Registrasi Berhasil\n";
}
```

```
    return true;
}
```

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 ===\n";
        cout << "Masukkan Username : ";
        cin >> username;
        cout << "Masukkan Password : ";
        cin >> password;
        enter();

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

                if (users[i].role == "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 (users[i].role == "admin") {
    while (true) {
        string pilihanadmin;
```

```

cout << "Menu Admin" << endl;
cout << "1. Edit Pembalap" << endl;
cout << "2. Edit Tim" << endl;
cout << "3. Kembali ke Menu Utama" << endl;
cout << "Pilih menu : ";
cin >> pilihanadmin;
enter();

```

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;
    enter();
}

```

Gambar 3.5 Menu Edit Pembalap

F. Pilihan Menu Tambah Pembalap

Fitur ini digunakan jika admin ingin menambahkan pembalap.

Source Code:

```

void tambahpembalap() {
    if (jumlahpembalap >= maxpembalap) {
        cout << "\nKapasitas pembalap penuh! Tidak bisa menambah lagi.\n";
    } else {
        Pembalap p;
        cin.ignore();
        cout << "\nMasukkan data pembalap baru:\n";
        cout << "Nama          : "; getline(cin, p.nama);
        cout << "Tim            : "; getline(cin, p.tim);
        cout << "Asal Negara   : "; getline(cin, p.asalNegara);
        cout << "Nomor         : "; getline(cin, p.nomor);
    }
}

```



```

    cout << "Tanggal Lahir: "; getline(cin, p.tanggalLahir);
    cout << "Jumlah Podium: "; cin >> p.jumlahPodium;
    pembalap[jumlahpembalap++] = p;
    cout << "Pembalap berhasil ditambahkan!" << endl;
}
}

```

Gambar 3.6 Tambah Pembalap

G. Pilihan Menu Update Data Pembalap

Fitur ini digunakan jika admin ingin mengupdate data pembalap.

Source Code:

```

void updatepembalap() {
    int updatepembalap;
    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].nomor);
        cout << "Jumlah Podium : "; cin >> pembalap[index].jumlahPodium;
        cout << "Data pembalap berhasil diperbarui!\n";
    } else if (updatepembalap != 0) {
        cout << "Pilihan tidak valid!\n";
    }
}
}

```

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:

```

string gantipembalap(Pembalap pembalap[], int& jumlahpembalap) {
    int gantipembalap;
    cout << "\nMasukkan nomor pembalap yang ingin diganti (0 untuk kembali): ";
    cin >> gantipembalap;
}

```

```

if (gantipembalap >= 1 && gantipembalap <= jumlahpembalap) {
    int index = gantipembalap - 1;
    string namaLama = pembalap[index].nama;
    cin.ignore();
    cout << "\nMasukkan data pembalap baru:\n";
    cout << "Nama          : "; getline(cin, pembalap[index].nama);
    cout << "Asal Negara   : "; getline(cin, pembalap[index].asalNegara);
    cout << "Nomor         : "; getline(cin, pembalap[index].nomor);
    cout << "Tanggal Lahir: "; getline(cin, pembalap[index].tanggalLahir);
    cout << "Jumlah Podium: "; cin >> pembalap[index].jumlahPodium;
    cout << "Pembalap " << namaLama << " berhasil diganti dengan " <<
pembalap[index].nama << "\n";
    return namaLama;
} else if (gantipembalap != 0) {
    cout << "Pilihan tidak valid!\n";
}
return "";
}

```

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;
    enter();
}

```

Gambar 3.9 Menu Edit Tim

J. Pilihan Menu Tambah Tim

Fitur ini digunakan jika admin ingin menambahkan tim.

Source Code:

```

void tambahtim() {

```

```

    if (jumahtim >= maxtim) {
        cout << "\nJumlah tim sudah penuh!" << endl;
    }
    cout << "Masukkan detail tim baru:" << endl;
    cout << "Nama Tim      : "; getline(cin, tim[jumahtim].nama);
    cout << "Markas        : "; getline(cin, tim[jumahtim].markas);
    cout << "Team Principal: "; getline(cin, tim[jumahtim].teamPrincipal);
    cout << "Power Unit    : "; getline(cin, tim[jumahtim].powerUnit);
    cout << "Chassis       : "; getline(cin, tim[jumahtim].chassis);
    jumahtim++;
    cout << "Tim berhasil ditambahkan!\n";
}

```

Gambar 3.10 Tambah Tim

K. Pilihan Menu Update Data Tim

Fitur ini digunakan jika admin ingin mengupdate data tim.

Source Code:

```

void updatetim() {
    int choice;
    cout << "\nMasukkan nomor tim yang ingin diperbarui (0 untuk kembali): ";
    cin >> choice;
    if (choice >= 1 && choice <= jumahtim) {
        int index = choice - 1;
        cout << "\nMasukkan data baru untuk tim " << tim[index].nama << ":\n";
        cout << "Nama Tim      : "; getline(cin, tim[index].nama);
        cout << "Markas        : "; getline(cin, tim[index].markas);
        cout << "Team Principal: "; getline(cin, tim[index].teamPrincipal);
        cout << "Power Unit    : "; getline(cin, tim[index].powerUnit);
        cout << "Chassis       : "; getline(cin, tim[index].chassis);
        cout << "Tim berhasil diperbarui!\n";
    } else if (choice != 0) {
        cout << "Pilihan tidak valid!\n";
    }
}

```

Gambar 3.11 Update Data Tim

L. Pilihan Menu Hapus Tim

Fitur ini digunakan jika admin ingin menghapus suatu tim.

Source Code:

```

void hapustim() {
    int choice;
    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++) {
            tim[i] = tim[i + 1];
        }
        jumlahtim--;
        cout << "Tim berhasil dihapus!\n";
    } else if (choice != 0) {
        cout << "Pilihan tidak valid!\n";
    }
}

```

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. Lihat jumlah podium pembalap\n";
        cout << "4. Kembali ke Menu Utama\n";
        cout << "Pilih menu: ";
        cin >> pilihanuser;
        enter();
    }
}

```

Gambar 3.13 Menu User

N. Pilihan Lihat Daftar Pembalap

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

Source Code:

```

void tampilkanpembalap() {
    for (int i = 0; i < jumlahpembalap - 1; i++) {

```

```

        for (int j = 0; j < jumlahpembalap - i - 1; j++) {
            if (pembalap[j].nama > pembalap[j + 1].nama) {
                Pembalap temp = pembalap[j];
                pembalap[j] = pembalap[j + 1];
                pembalap[j + 1] = temp;
            }
        }
    }

    cout << "\nDaftar Pembalap F1 2025:" << endl;
    for (int i = 0; i < jumlahpembalap; i++) {
        cout << i + 1 << ". " << pembalap[i].nama << endl;
    }
}

void lihatdatapembalap() {
    int pilihandatapembalap;
    cout << "\nMasukkan nomor pembalap untuk melihat detail (0 untuk kembali): ";
    cin >> pilihandatapembalap;
    if (pilihandatapembalap >= 1 && pilihandatapembalap <= jumlahpembalap) {
        int index = pilihandatapembalap - 1;
        cout << "\nDetail Pembalap:\n";
        cout << "Nama          : " << pembalap[index].nama << "\n";
        cout << "Tim            : " << pembalap[index].tim << "\n";
        cout << "Asal Negara   : " << pembalap[index].asalNegara << "\n";
        cout << "Nomor         : " << pembalap[index].nomor << "\n";
        cout << "Tanggal Lahir: " << pembalap[index].tanggalLahir << "\n";
        cout << "Jumlah Podium: " << pembalap[index].jumlahPodium << "\n";
    } else if (pilihandatapembalap != 0) {
        cout << "Pilihan tidak valid!\n";
    }
}
}

```

Gambar 3.14 Lihat Daftar Pembalap

O. Pilihan Lihat Daftar Tim

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

Source Code:

```

void tampilkantim() {
    for (int i = 0; i < jumlahtim - 1; i++) {
        int indeksMin = i;
    }
}

```

```

        for (int j = i + 1; j < jumlahtim; j++) {
            if (tim[j].nama < tim[indeksMin].nama) {
                indeksMin = j;
            }
        }
        if (indeksMin != i) {
            Tim temp = tim[i];
            tim[i] = tim[indeksMin];
            tim[indeksMin] = temp;
        }
    }

    cout << "Daftar Tim F1 2025:\n";
    for (int i = 0; i < jumlahtim; i++) {
        cout << i + 1 << ". " << tim[i].nama << endl;
    }
}

void lihatdatatim() {
    int pilihandatatim;
    cout << "\nMasukkan nomor tim untuk melihat detail (0 untuk kembali): ";
    cin >> pilihandatatim;
    if (pilihandatatim >= 1 && pilihandatatim <= jumlahtim) {
        int index = pilihandatatim - 1;
        cout << "\nDetail Tim:\n";
        cout << "Nama Tim      : " << tim[index].nama << "\n";
        cout << "Markas       : " << tim[index].markas << "\n";
        cout << "Team Principal : " << tim[index].teamPrincipal << "\n";
        cout << "Power Unit    : " << tim[index].powerUnit << "\n";
        cout << "Chassis       : " << tim[index].chassis << "\n";
    } else if (pilihandatatim != 0) {
        cout << "Pilihan tidak valid!\n";
    }
}
}

```

Gambar 3.15 Lihat Daftar Tim

P. Pilihan Lihat Jumlah Podium Pembalap

Fitur ini digunakan jika user ingin melihat jumlah podium yang dimiliki setiap pembalap

```

void jumlahpodium() {
    for (int i = 1; i < jumlahpembalap; i++) {
        Pembalap key = pembalap[i];
    }
}

```

```

    int j = i - 1;

    while (j >= 0 && pembalap[j].jumlahPodium < key.jumlahPodium) {
        pembalap[j + 1] = pembalap[j];
        j--;
    }
    pembalap[j + 1] = key;
}
cout << "\nJumlah Podium Pembalap :" << endl;
for (int i = 0; i < jumlahpembalap; i++) {
    cout << i+1 << ". " << pembalap[i].nama << " (" <<
pembalap[i].jumlahPodium << " podium)" << endl;
}
}

```

Gambar 3.16 Lihat Jumlah Podium Pembalap

4. Uji Coba dan Hasil Output

A. Registrasi

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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

```

Gambar 4.1 Registrasi

B. Gagal Login

```

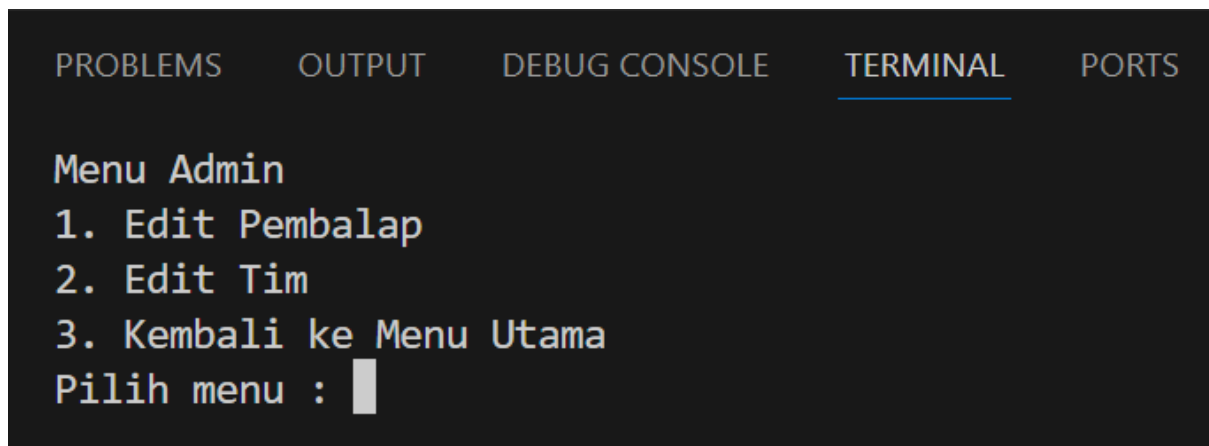
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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

```

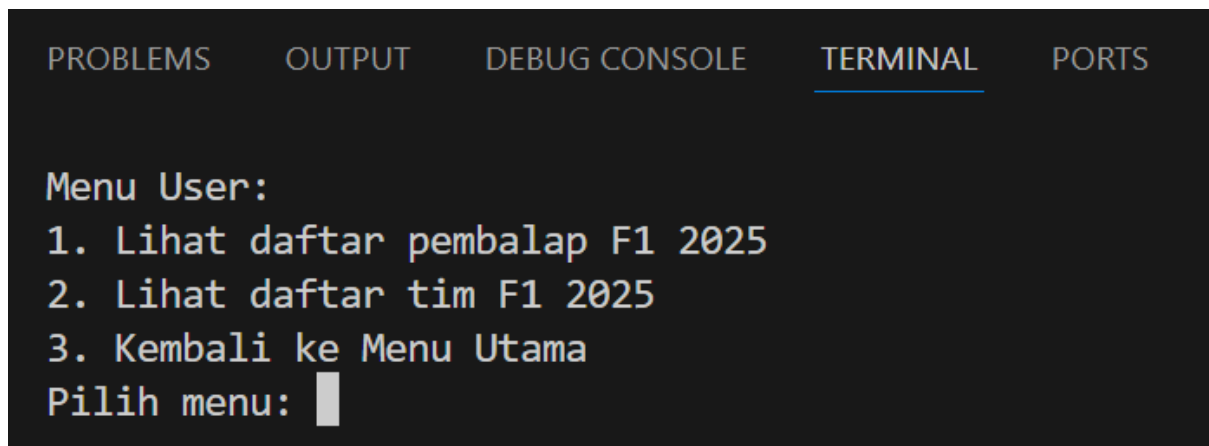
Gambar 4.2 Login Gagal

C. Login Admin



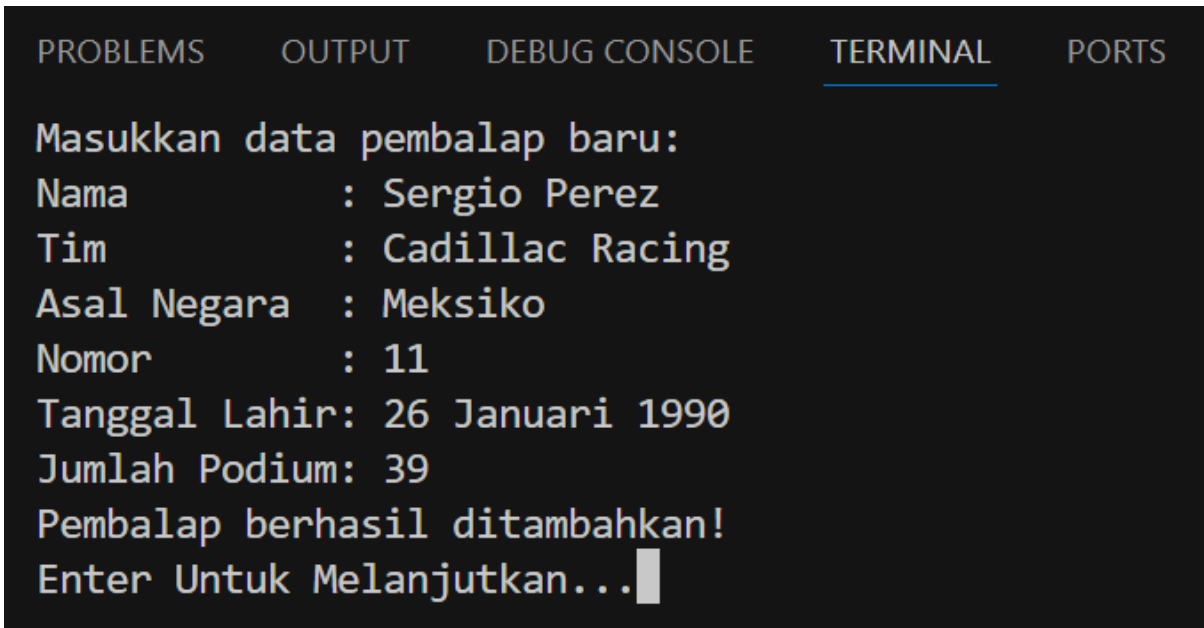
Gambar 4.3 Login Admin

D. Login User



Gambar 4.4 Login User

E. Tambah Pembalap

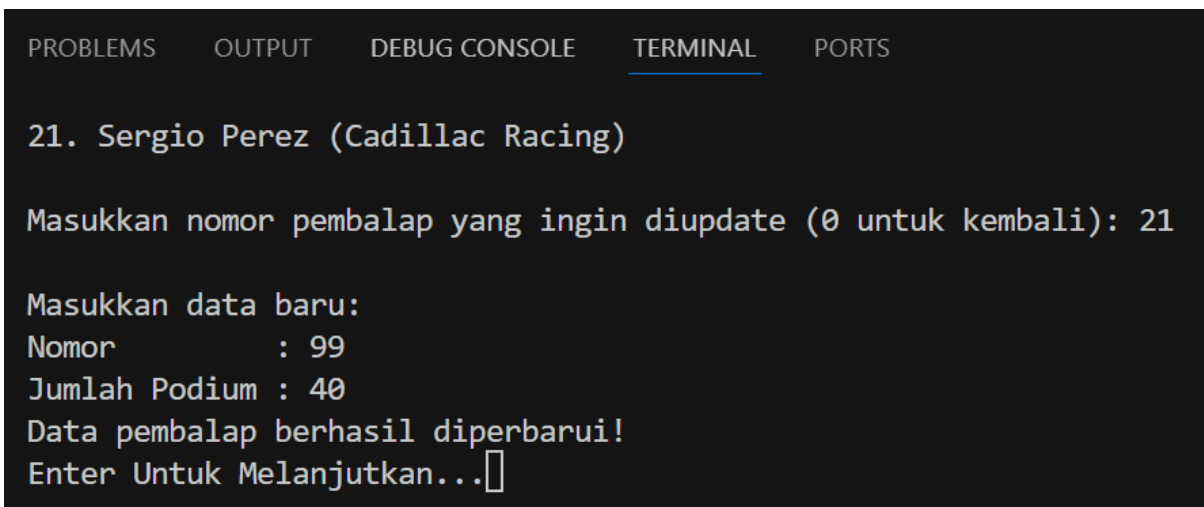


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukkan data pembalap baru:
Nama      : Sergio Perez
Tim       : Cadillac Racing
Asal Negara : Meksiko
Nomor     : 11
Tanggal Lahir: 26 Januari 1990
Jumlah Podium: 39
Pembalap berhasil ditambahkan!
Enter Untuk Melanjutkan...
```

Gambar 4.5 Tambah Pembalap

F. Update Data Pembalap



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

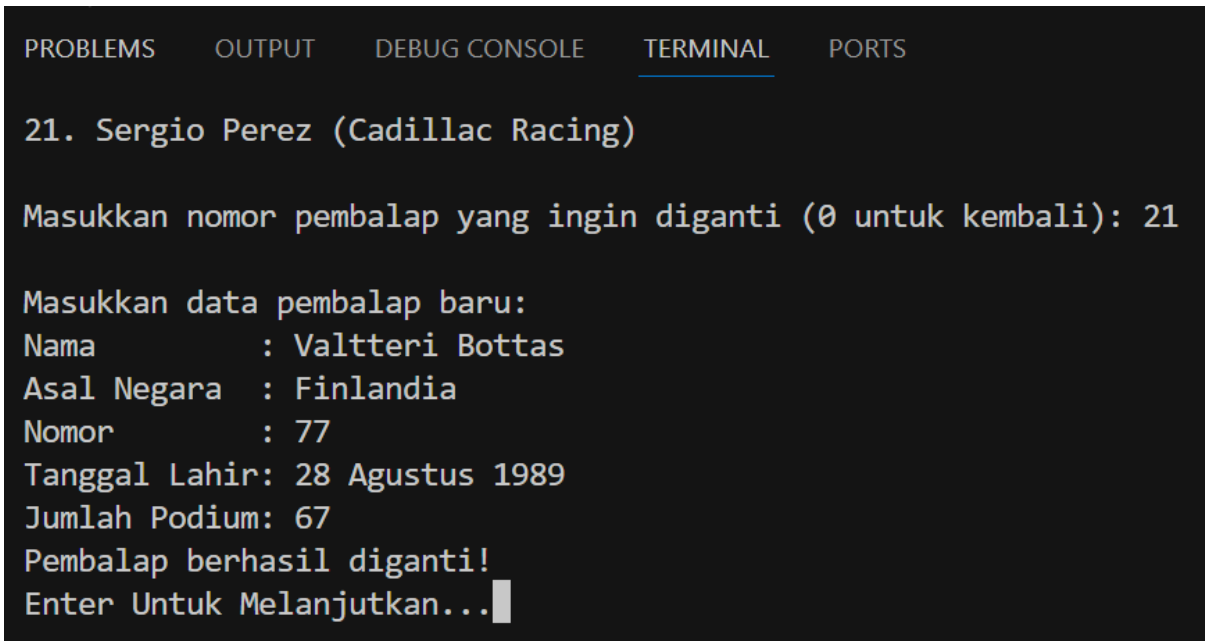
21. Sergio Perez (Cadillac Racing)

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

Masukkan data baru:
Nomor      : 99
Jumlah Podium : 40
Data pembalap berhasil diperbarui!
Enter Untuk Melanjutkan...
```

Gambar 4.6 Update Data Pembalap

G. Ganti Pembalap



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

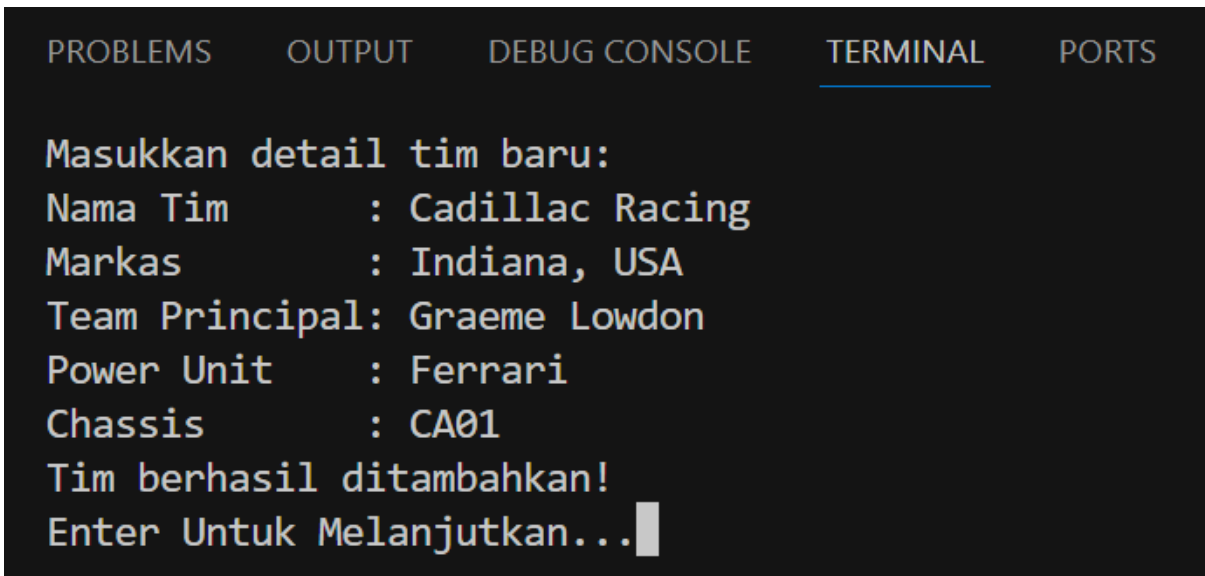
21. Sergio Perez (Cadillac Racing)

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

Masukkan data pembalap baru:
Nama           : Valtteri Bottas
Asal Negara    : Finlandia
Nomor          : 77
Tanggal Lahir  : 28 Agustus 1989
Jumlah Podium  : 67
Pembalap berhasil diganti!
Enter Untuk Melanjutkan...
```

Gambar 4.7 Ganti Pembalap

H. Tambah Tim

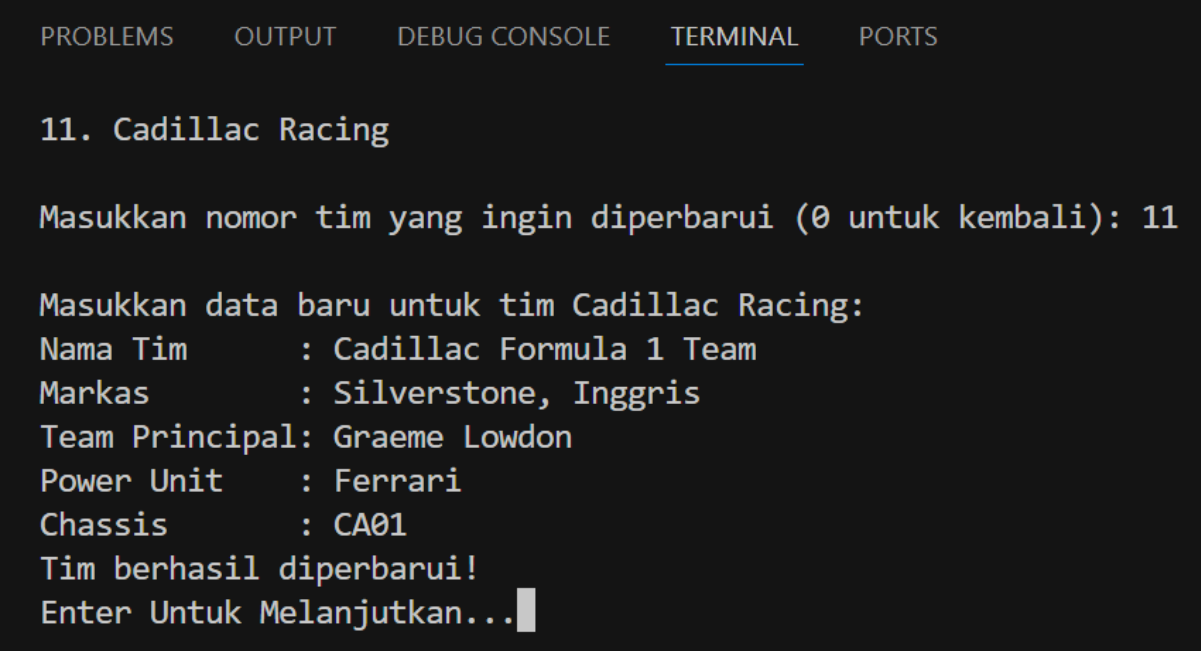


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukkan detail tim baru:
Nama Tim       : Cadillac Racing
Markas         : Indiana, USA
Team Principal : Graeme Lowdon
Power Unit     : Ferrari
Chassis        : CA01
Tim berhasil ditambahkan!
Enter Untuk Melanjutkan...
```

Gambar 4.8 Tambah Tim

I. Update Data Tim



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

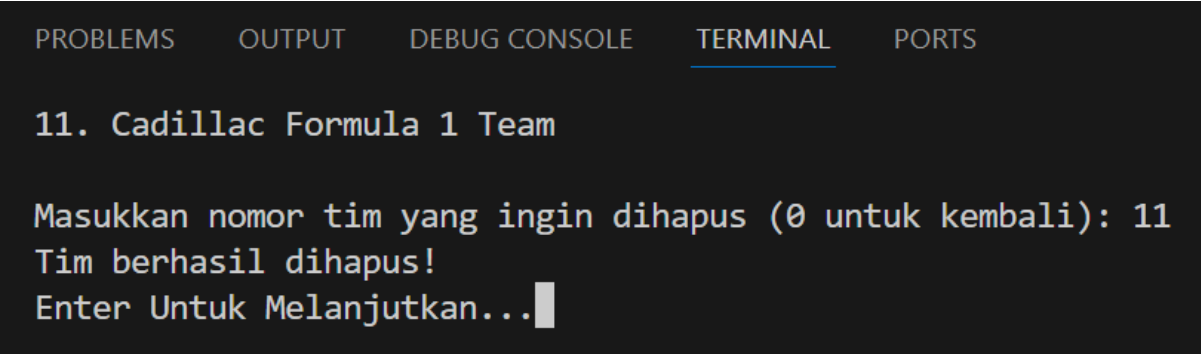
11. Cadillac Racing

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

Masukkan data baru untuk tim Cadillac Racing:
Nama Tim      : Cadillac Formula 1 Team
Markas       : Silverstone, Inggris
Team Principal: Graeme Lowdon
Power Unit    : Ferrari
Chassis       : CA01
Tim berhasil diperbarui!
Enter Untuk Melanjutkan...
```

Gambar 4.9 Update Data Tim

J. Hapus Tim



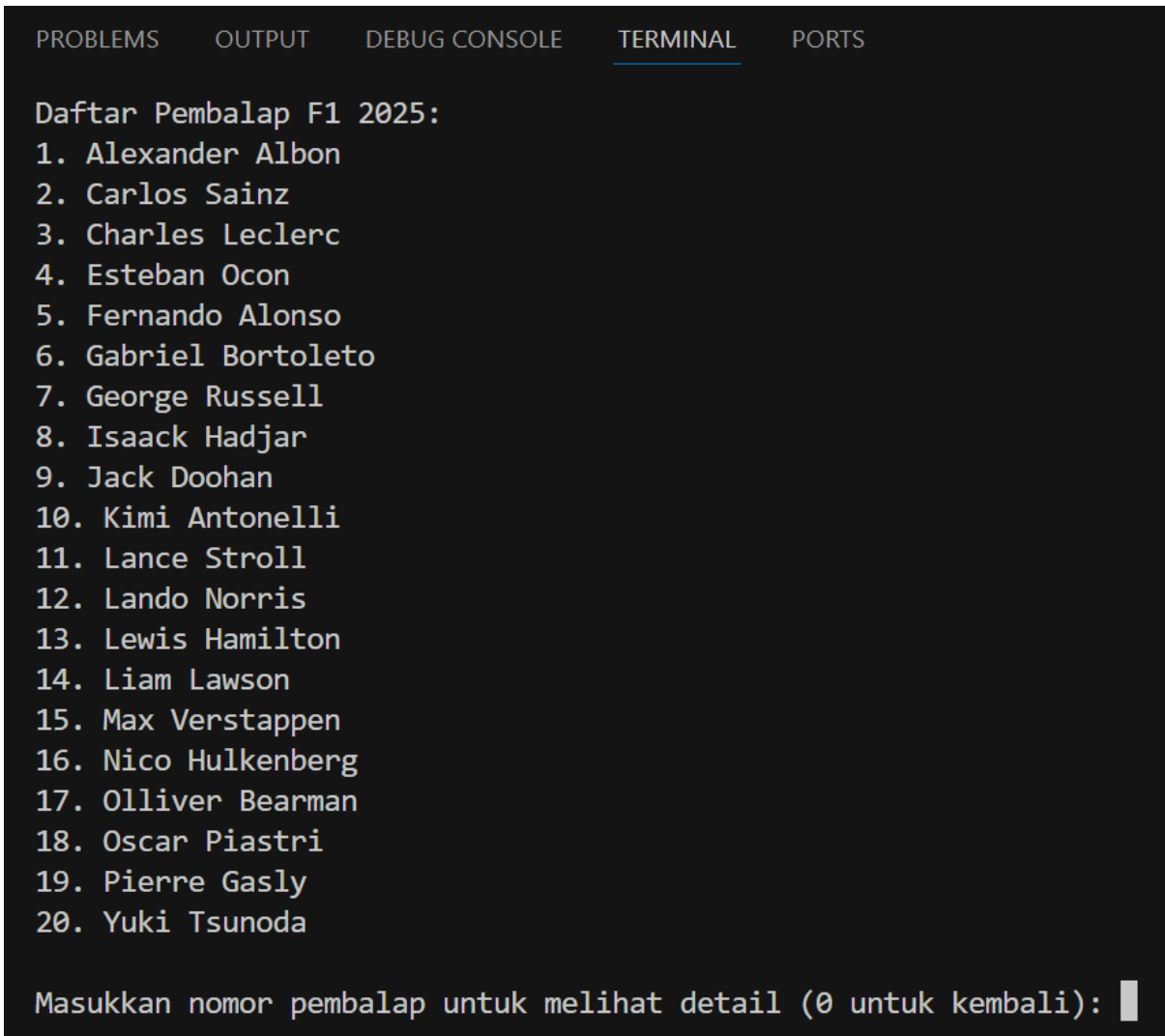
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

11. Cadillac Formula 1 Team

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

Gambar 4.10 Hapus Tim

K. Daftar Pembalap



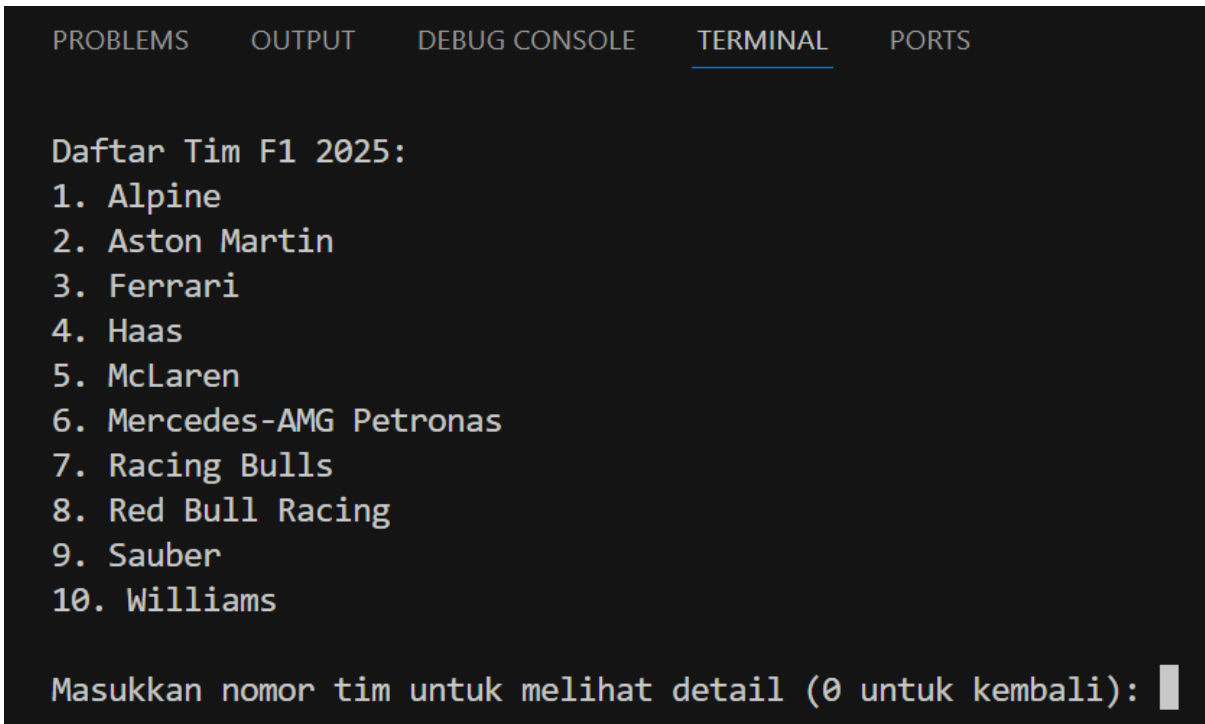
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Daftar Pembalap F1 2025:
1. Alexander Albon
2. Carlos Sainz
3. Charles Leclerc
4. Esteban Ocon
5. Fernando Alonso
6. Gabriel Bortoleto
7. George Russell
8. Isaack Hadjar
9. Jack Doohan
10. Kimi Antonelli
11. Lance Stroll
12. Lando Norris
13. Lewis Hamilton
14. Liam Lawson
15. Max Verstappen
16. Nico Hulkenberg
17. Olliver Bearman
18. Oscar Piastri
19. Pierre Gasly
20. Yuki Tsunoda

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

Gambar 4.11 Lihat Daftar Pembalap

L. Daftar Tim



The image shows a terminal window with a dark background and light-colored text. At the top, there is a navigation bar with five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is underlined), and 'PORTS'. Below the tabs, the text 'Daftar Tim F1 2025:' is displayed. This is followed by a numbered list of ten F1 teams: 1. Alpine, 2. Aston Martin, 3. Ferrari, 4. Haas, 5. McLaren, 6. Mercedes-AMG Petronas, 7. Racing Bulls, 8. Red Bull Racing, 9. Sauber, and 10. Williams. At the bottom of the terminal, there is a prompt 'Masukkan nomor tim untuk melihat detail (0 untuk kembali):' followed by a cursor icon.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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

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

Gambar 4.12 Lihat Daftar Tim

M. Jumlah Podium



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

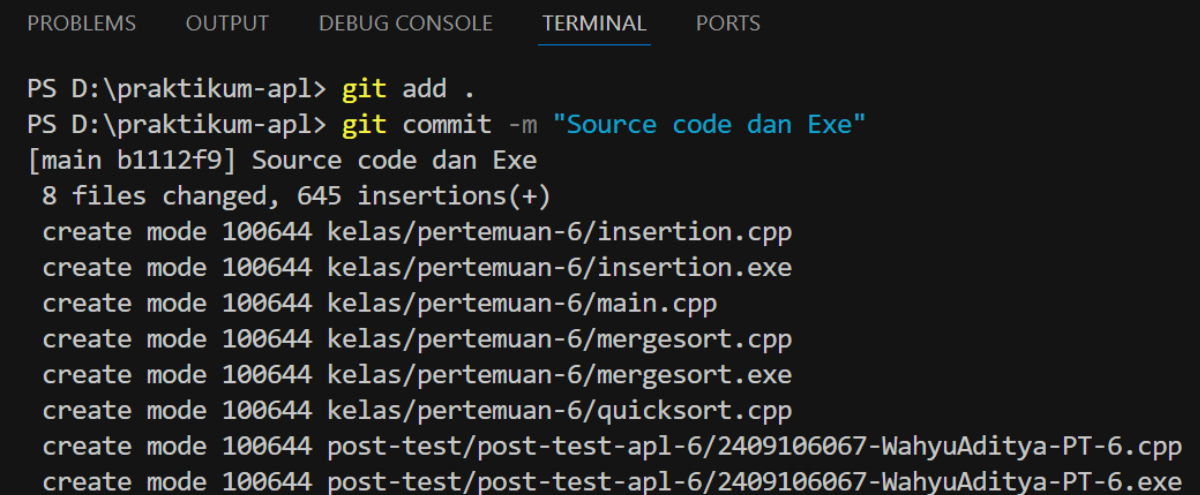
Jumlah Podium Pembalap :
1. Lewis Hamilton (195 podium)
2. Fernando Alonso (104 podium)
3. Max Verstappen (89 podium)
4. Charles Leclerc (27 podium)
5. Carlos Sainz (18 podium)
6. Lando Norris (13 podium)
7. George Russell (10 podium)
8. Pierre Gasly (4 podium)
9. Lance Stroll (3 podium)
10. Oscar Piastri (3 podium)
11. Alexander Albon (2 podium)
12. Esteban Ocon (2 podium)
13. Gabriel Bortoleto (0 podium)
14. Isaack Hadjar (0 podium)
15. Jack Doohan (0 podium)
16. Kimi Antonelli (0 podium)
17. Liam Lawson (0 podium)
18. Nico Hulkenberg (0 podium)
19. Olliver Bearman (0 podium)
20. Yuki Tsunoda (0 podium)
Enter Untuk Melanjutkan...
```

Gambar 4.13 Jumlah Podium Pembalap

5. Git

Tahap pertama jika ingin mengupload file menggunakan git adalah menginsiasi git pada repository lokal. Karena git sudah diinisiasi pada posttest sebelumnya, maka langsung saja git

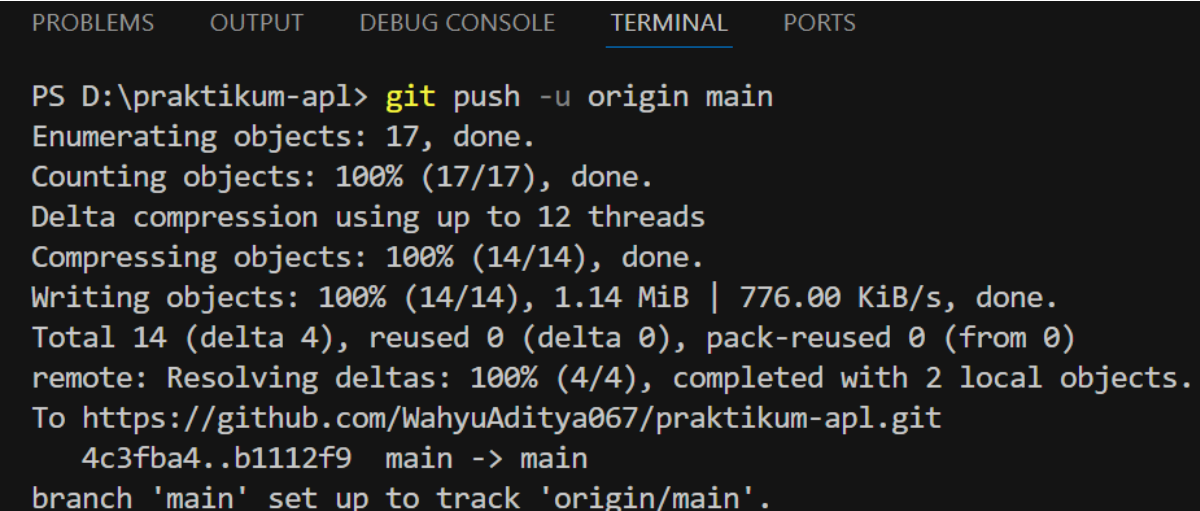
add untuk menambahkan file yang ingin diupload. Tambahkan juga pesan commit untuk memudahkan mengetahui perubahan yang dilakukan.

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows the following commands and output:

```
PS D:\praktikum-apl> git add .
PS D:\praktikum-apl> git commit -m "Source code dan Exe"
[main b1112f9] Source code dan Exe
 8 files changed, 645 insertions(+)
 create mode 100644 kelas/pertemuan-6/insertion.cpp
 create mode 100644 kelas/pertemuan-6/insertion.exe
 create mode 100644 kelas/pertemuan-6/main.cpp
 create mode 100644 kelas/pertemuan-6/mergesort.cpp
 create mode 100644 kelas/pertemuan-6/mergesort.exe
 create mode 100644 kelas/pertemuan-6/quicksort.cpp
 create mode 100644 post-test/post-test-apl-6/2409106067-WahyuAditya-PT-6.cpp
 create mode 100644 post-test/post-test-apl-6/2409106067-WahyuAditya-PT-6.exe
```

Gambar 5.1 Git add dan Commit

Karena git juga 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”.

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows the following command and output:

```
PS D:\praktikum-apl> git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.14 MiB | 776.00 KiB/s, done.
Total 14 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/WahyuAditya067/praktikum-apl.git
 4c3fba4..b1112f9  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5.2 Git Push