

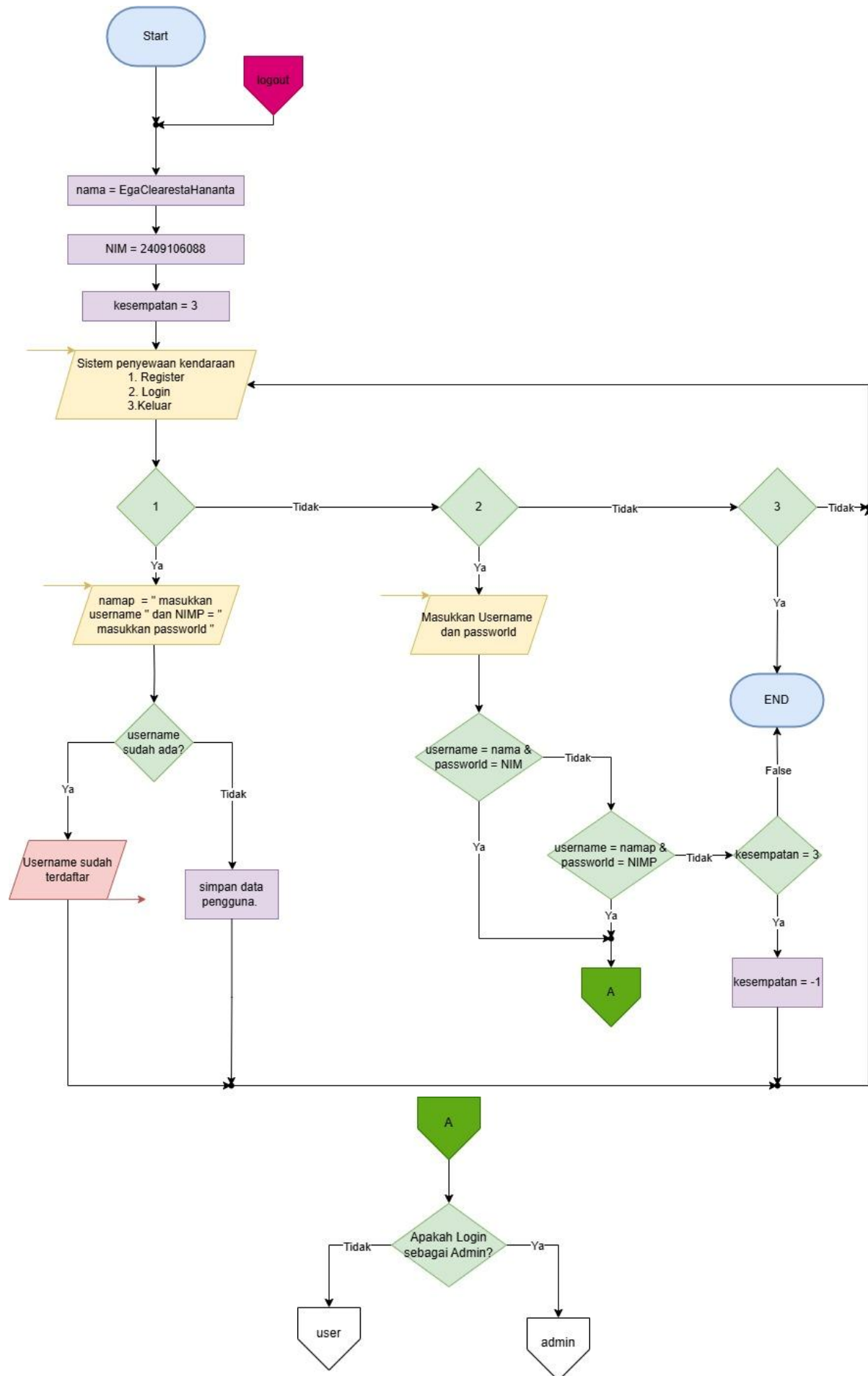
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT

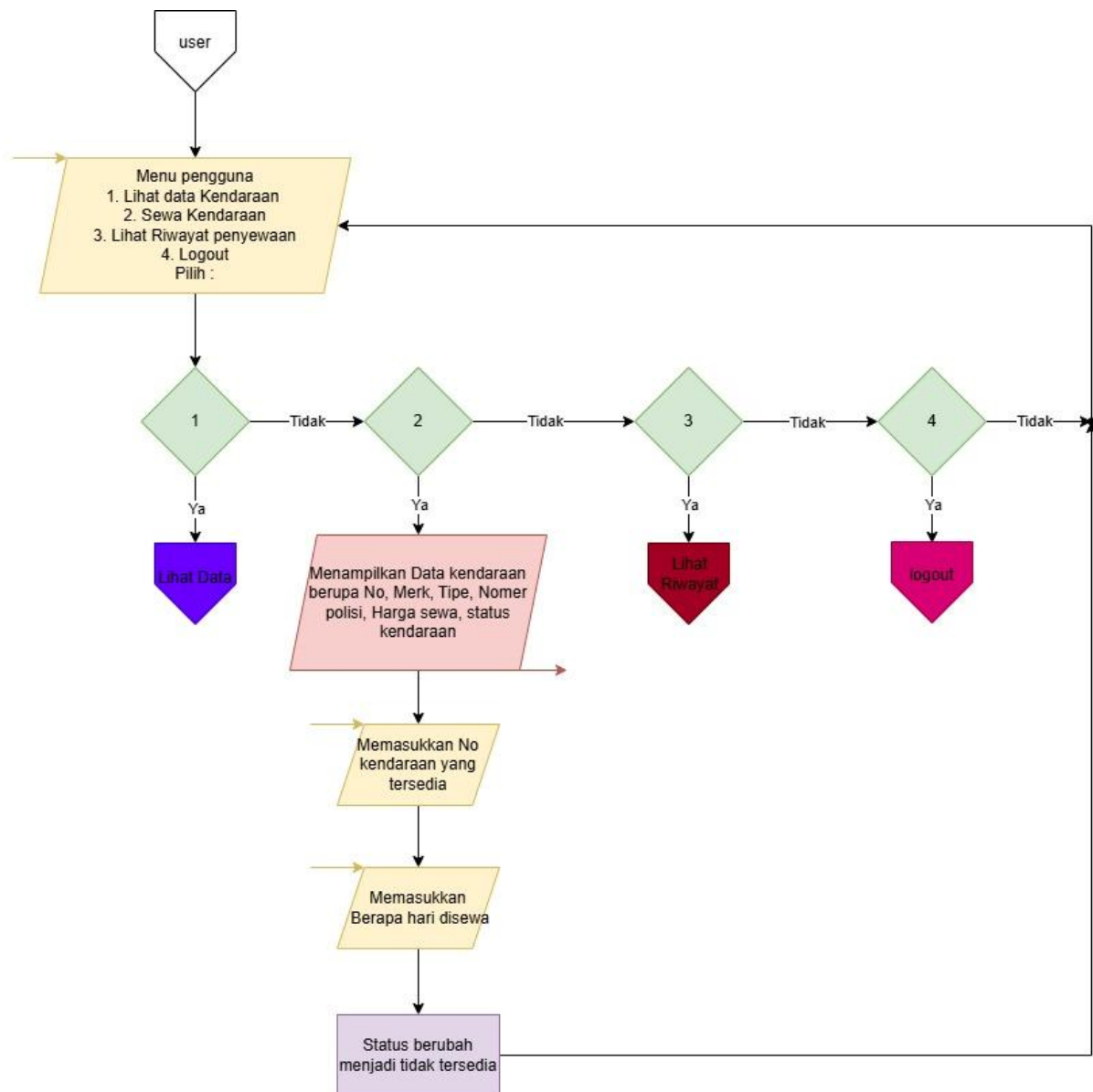


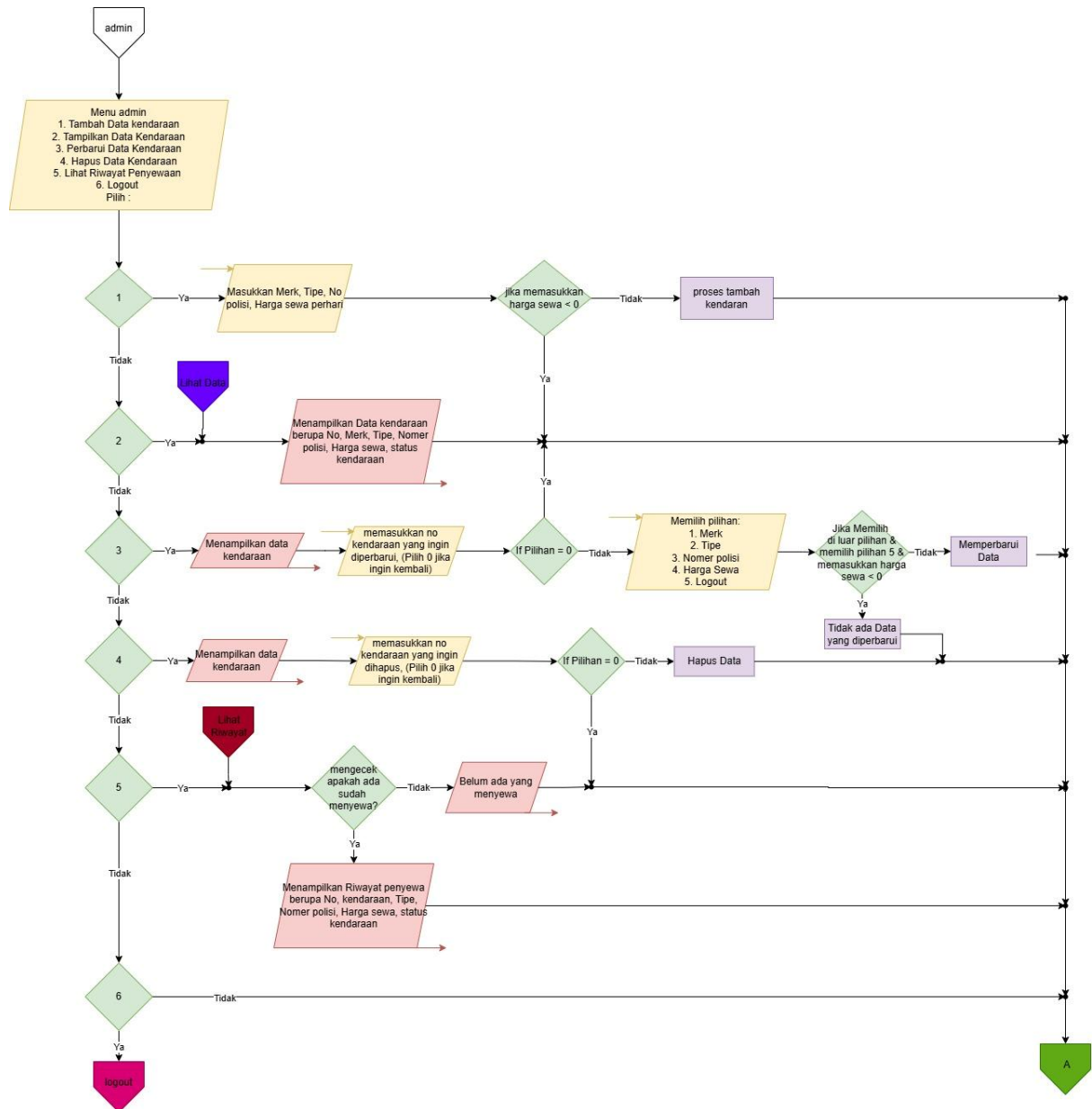
Disusun oleh:
Ega Clearesta Hananta (2409106088)
Kelas (B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







2. Analisis Program

Program ini bertujuan untuk mengelola sistem penyewaan kendaraan dengan menyediakan fitur registrasi, login, serta manajemen data kendaraan bagi admin dan pengguna. Fungsi utamanya meliputi penambahan, pembaruan, penghapusan, dan penyewaan kendaraan, serta pencatatan riwayat penyewaan untuk setiap kendaraan. Admin dapat mengelola seluruh data kendaraan dan melihat riwayat penyewaan, sedangkan pengguna dapat melihat daftar kendaraan, menyewa, dan mengecek riwayat penyewaan mereka. Program ini memastikan validasi input, membatasi akses berdasarkan peran pengguna, dan menyediakan tampilan yang terstruktur untuk kemudahan penggunaan.

3. Source Code

A. Struct

Struktur RiwayatSewa dan Kendaraan digunakan untuk mengorganisasi data terkait penyewaan kendaraan secara sistematis. Struktur RiwayatSewa menyimpan informasi tentang penyewa, seperti nama penyewa (namaPenyewa) dan durasi sewa (durasi), yang memungkinkan pelacakan aktivitas penyewaan per kendaraan. Sementara itu, struktur Kendaraan berfungsi untuk merepresentasikan detail kendaraan, termasuk merk, tipe, nomor polisi, harga sewa, status ketersediaan (status), serta riwayat penyewaan dalam bentuk array riwayatPenyewa. Dengan adanya atribut jumlahPenyewa, program dapat melacak jumlah total penyewaan untuk setiap kendaraan. Kedua struct ini saling berhubungan untuk mendukung pengelolaan data kendaraan dan riwayat penyewaan secara efisien dalam program.

```
struct RiwayatSewa {
    string namaPenyewa;
    string durasi;
};

struct Kendaraan {
    string merk;
    string tipe;
    string nomor_polisi;
    double harga_sewa;
    string status;
    RiwayatSewa riwayatPenyewa[MAX_SEWA];
    int jumlahPenyewa = 0;
};
```

B. Validasi Input

Fungsi `validasiInputan` digunakan untuk memastikan bahwa input yang dimasukkan oleh pengguna berupa angka integer dan berada dalam rentang yang ditentukan (`min` hingga `max`). Fungsi ini membaca input dari pengguna menggunakan `cin`, lalu memeriksa apakah input valid dengan mengevaluasi apakah terjadi kegagalan input (`cin.fail()`) atau apakah nilai yang dimasukkan berada di luar batas minimum dan maksimum. Jika input tidak valid, fungsi akan membersihkan status error pada `cin` menggunakan `cin.clear()` dan mengabaikan sisa input yang salah menggunakan `cin.ignore()`. Setelah itu, pengguna diminta memasukkan ulang input dengan menampilkan pesan kesalahan dan memanggil kembali fungsi secara rekursif hingga input yang valid diterima. Jika input valid, nilai tersebut dikembalikan ke pemanggil fungsi untuk digunakan dalam program.

```

int validasiInputan(int min, int max) {
    int input;
    cin >> input;
    if (cin.fail() || input < min || input > max) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Input tidak valid! Masukkan angka antara " << min << " dan "
<< max << ": ";
        return validasiInputan(min, max);
    }
    return input;
}

```

C. Fungsi ubahHargaSewa

Fungsi ubahHargaSewa digunakan untuk mengubah nilai variabel harga dengan nilai baru (hargaBaru) secara langsung melalui referensi alamat memori. Parameter pertama, double &harga, menggunakan konsep pass by reference dengan operator &, yang berarti bahwa perubahan yang dilakukan pada parameter harga di dalam fungsi akan memengaruhi nilai asli variabel yang dilewatkan saat pemanggilan fungsi. Dengan demikian, ketika fungsi ini dipanggil dengan memberikan alamat variabel harga sewa kendaraan, nilai variabel tersebut diperbarui menjadi hargaBaru tanpa perlu mengembalikan nilai secara eksplisit. Fungsi ini sangat berguna untuk memastikan bahwa perubahan harga sewa dapat dilakukan secara efisien dan langsung memengaruhi data asli dalam program.

```

void ubahHargaSewa(double &harga, double hargaBaru) {
    harga = hargaBaru;
}

```

D. Fungsi tampilkanHargaSewa

Fungsi tampilkanHargaSewa digunakan untuk menampilkan nilai harga sewa kendaraan dengan menggunakan konsep pointer melalui parameter dereference (*). Parameter double *harga merupakan pointer yang menyimpan alamat memori dari variabel harga sewa, sehingga fungsi dapat mengakses nilai asli dari alamat tersebut menggunakan operator dereference (*). Dalam fungsi ini, nilai yang ditunjuk oleh pointer harga kemudian diformat menggunakan fixed dan setprecision(0) untuk menampilkan harga sebagai bilangan bulat tanpa desimal, dan hasilnya dicetak ke layar dengan format "Harga Sewa: Rp [nilai]". Fungsi ini sangat berguna untuk menampilkan data harga sewa secara langsung dari memori, memastikan bahwa nilai yang ditampilkan selalu sesuai dengan data terkini dalam program.

```

void tampilkanHargaSewa(double *harga) {
    cout << "Harga Sewa: Rp " << fixed << setprecision(0) << *harga << endl;
}

```

```
}
```

E. Fungsi tambah data kendaraan

Fungsi tambah digunakan untuk menambahkan data kendaraan baru ke dalam array `daftarKendaraan` dengan validasi dan pengisian detail secara sistematis. Pertama, fungsi memeriksa apakah jumlah kendaraan sudah mencapai batas maksimal (`MAX_DATA`), jika ya, program memberi pesan bahwa data penuh dan keluar dari fungsi. Jika belum penuh, pengguna diminta menginput informasi kendaraan seperti merk, tipe, nomor polisi, dan harga sewa per hari. Input harga sewa divalidasi untuk memastikan berupa angka positif menggunakan perulangan `do-while`. Setelah semua data diisi, status kendaraan diatur sebagai "Tersedia" dan jumlah penyewa diinisialisasi menjadi 0. Data kendaraan yang baru kemudian disimpan ke dalam array `daftarKendaraan` pada indeks sesuai jumlah kendaraan saat ini, dan variabel `jumlahKendaraan` ditingkatkan untuk mencerminkan penambahan data. Fungsi ini juga menyediakan konfirmasi kepada pengguna bahwa data berhasil ditambahkan sebelum kembali ke menu utama.

```
void tambah(Kendaraan daftarKendaraan[], int &jumlahKendaraan) {
    system("cls");
    if (jumlahKendaraan >= MAX_DATA) {
        cout << "Data kendaraan sudah penuh! Tidak dapat menambah lagi." <<
endl;
        Enter();
        return;
    }

    Kendaraan k;
    cin.ignore();
    cout << "===== Tambah Data Kendaraan
=====\\n";
    cout << "Masukkan Merk Kendaraan: ";
    getline(cin, k.merk);
    cout << "Masukkan Tipe Kendaraan: ";
    getline(cin, k.tipe);
    cout << "Masukkan Nomor Polisi: ";
    getline(cin, k.nomor_polisi);

    do {
        cout << "Masukkan Harga Sewa per Hari: ";
        cin >> k.harga_sewa;
        if (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\\n');
            cout << "Harga sewa tidak valid! Masukkan angka." << endl;
            continue;
        }
    } while (k.harga_sewa <= 0);

    daftarKendaraan[jumlahKendaraan] = k;
    jumlahKendaraan++;
    cout << "Data kendaraan berhasil ditambahkan." << endl;
    Enter();
}
```



```

    }
    if (k.harga_sewa <= 0) {
        cout << "Harga sewa harus lebih dari 0!" << endl;
        continue;
    }
    break;
} while (true);

k.status = "Tersedia";
k.jumlahPenyewa = 0;
daftarKendaraan[jumlahKendaraan] = k;
jumlahKendaraan++;
cout << "Data kendaraan berhasil ditambahkan!" << endl;
Enter();
}

```

F. Fungsi Melihat Data kendaraan

Fungsi `lihatKendaraan` digunakan untuk menampilkan daftar kendaraan yang tersimpan dalam array `daftarKendaraan` dengan format tabel yang rapi dan terstruktur. Fungsi ini memiliki dua versi: satu untuk menampilkan semua data kendaraan tanpa filter dan satu lagi untuk menampilkan kendaraan berdasarkan status tertentu (misalnya "Tersedia" atau "Disewa"). Pada kedua versi, jika tidak ada data kendaraan yang tersimpan, program akan memberikan pesan bahwa belum ada data. Untuk versi pertama, semua kendaraan ditampilkan dalam tabel dengan kolom seperti nomor urut, merk, tipe, nomor polisi, harga sewa, dan status. Versi kedua menyaring kendaraan berdasarkan parameter status, sehingga hanya kendaraan dengan status yang sesuai yang ditampilkan; jika tidak ada kendaraan dengan status tersebut, program memberikan pesan khusus. Kedua fungsi menggunakan `setw` untuk memastikan lebar kolom tetap konsisten dan mudah dibaca oleh pengguna.

```

void lihatKendaraan(const Kendaraan daftarKendaraan[], int jumlahKendaraan) {
    system("cls");
    if (jumlahKendaraan == 0) {
        cout << "\nBelum ada data kendaraan yang tersimpan!\n";
        Enter();
        return;
    }

    cout << "\n===== Data Kendaraan\n";
    cout << "|-----\n";
    cout << "| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status | \n";
    cout << "|-----\n";
}

```

```

        for (int i = 0; i < jumlahKendaraan; i++) {
            cout << " | " << setw(2) << i + 1 << " | "
                << setw(14) << daftarKendaraan[i].merk << " | "
                << setw(14) << daftarKendaraan[i].tipe << " | "
                << setw(14) << daftarKendaraan[i].nomor_polisi << " | "
                << "Rp " << setw(12) << fixed << setprecision(0) <<
daftarKendaraan[i].harga_sewa << " | "
                << setw(15) << daftarKendaraan[i].status << " | \n";
        }

        cout << " | -----
----- | \n";
        Enter();
    }

void lihatKendaraan(const Kendaraan daftarKendaraan[], int jumlahKendaraan,
string status) {
    system("cls");
    if (jumlahKendaraan == 0) {
        cout << "\nBelum ada data kendaraan yang tersimpan! \n";
        Enter();
        return;
    }

    cout << "\n===== Data Kendaraan (" <<
status << ") ===== \n";
    cout << " | -----
----- | \n";

    cout << " | No |      Merk      |      Tipe      | Nomor Polisi |
Harga Sewa    |      Status      | \n";
    cout << " | -----
----- | \n";

    int count = 0;
    for (int i = 0; i < jumlahKendaraan; i++) {
        if (daftarKendaraan[i].status == status) {
            cout << " | " << setw(2) << ++count << " | "
                << setw(14) << daftarKendaraan[i].merk << " | "
                << setw(14) << daftarKendaraan[i].tipe << " | "
                << setw(14) << daftarKendaraan[i].nomor_polisi << " | "
                << "Rp " << setw(12) << fixed << setprecision(0) <<
daftarKendaraan[i].harga_sewa << " | "
                << setw(15) << daftarKendaraan[i].status << " | \n";
        }
    }

    if (count == 0) {
        cout << " | Tidak ada kendaraan dengan status " << setw(62) << left <<
status << " | \n";
    }
}

```

```

        cout << "|-----\n";
        Enter();
    }

```

G. fungsi Menperbarui data kendaraan

Fungsi `perbarui` digunakan untuk memperbarui data kendaraan yang sudah tersimpan dalam array `daftarKendaraan` dengan memberikan opsi kepada pengguna untuk mengubah salah satu atribut tertentu seperti merk, tipe, nomor polisi, atau harga sewa. Pertama, fungsi menampilkan daftar kendaraan dan meminta pengguna memilih nomor kendaraan yang ingin diperbarui. Setelah dipilih, detail kendaraan tersebut ditampilkan secara rinci untuk memudahkan identifikasi. Pengguna kemudian diminta memilih atribut spesifik yang ingin diperbarui melalui menu pilihan. Fungsi ini menggunakan validasi input untuk memastikan data yang dimasukkan sesuai format, seperti penggunaan `cin.ignore()` dan `getline` untuk input string, serta fungsi pointer seperti `ubahHargaSewa` dan `tampilkanHargaSewa` untuk memperbarui dan menampilkan harga sewa. Proses perulangan memungkinkan pengguna memperbarui beberapa atribut sekaligus tanpa kembali ke menu utama, dan pengguna dapat keluar dari menu pembaruan saat memilih opsi "Kembali ke Menu Utama".

```

void perbarui(Kendaraan daftarKendaraan[], int jumlahKendaraan) {
    system("cls");
    if (jumlahKendaraan == 0) {
        cout << "\nBelum ada data kendaraan untuk diperbarui!\n";
        Enter();
        return;
    }

    lihatKendaraan(daftarKendaraan, jumlahKendaraan);
    cout << "Masukkan nomor kendaraan yang ingin diperbarui: ";
    int nomor = validasiInputan(1, jumlahKendaraan) - 1;
    system("cls");

    cout << "\n===== Detail Kendaraan Terpilih =====\n";
    cout << "|-----\n";
    cout << "| Merk          : " << setw(30) << left <<
daftarKendaraan[nomor].merk << "\n";
    cout << "| Tipe          : " << setw(30) << left <<
daftarKendaraan[nomor].tipe << "\n";
    cout << "| Nomor Polisi : " << setw(30) << left <<
daftarKendaraan[nomor].nomor_polisi << "\n";
    cout << "| Harga Sewa   : Rp " << setw(27) << left << fixed <<
setprecision(2) << daftarKendaraan[nomor].harga_sewa << "\n";
    cout << "| Status       : " << setw(30) << left <<

```

```

daftarKendaraan[nomor].status << "\n";
    cout << "|-----|\n";

    bool kembaliKeMenu = false;
    do {
        cout << "\n===== Menu Perbarui Data =====\n";
        cout << "|-----|\n";
        cout << "| No |          Pilihan Menu          |\n";
        cout << "|----|-----|\n";
        cout << "| 1 | Perbarui Merk                  |\n";
        cout << "| 2 | Perbarui Tipe                  |\n";
        cout << "| 3 | Perbarui Nomor Polisi          |\n";
        cout << "| 4 | Perbarui Harga Sewa            |\n";
        cout << "| 5 | Kembali ke Menu Utama          |\n";
        cout << "|-----|\n";
        cout << "Pilih opsi: ";
        int pilihan = validasiInputan(1, 5);

        switch (pilihan) {
            case 1: {
                cout << "\n----- Perbarui Merk -----|\n";
                cout << "Merk Lama      : " << daftarKendaraan[nomor].merk <<
endl;

                cout << "Masukkan Merk Baru: ";
                cin.ignore();
                getline(cin, daftarKendaraan[nomor].merk);
                cout << "\nMerk kendaraan berhasil diperbarui!\n";
                break;
            }
            case 2: {
                cout << "\n----- Perbarui Tipe -----|\n";
                cout << "Tipe Lama      : " << daftarKendaraan[nomor].tipe <<
endl;

                cout << "Masukkan Tipe Baru: ";
                cin.ignore();
                getline(cin, daftarKendaraan[nomor].tipe);
                cout << "\nTipe kendaraan berhasil diperbarui!\n";
                break;
            }
            case 3: {
                cout << "\n----- Perbarui Nomor Polisi -----|\n";
                cout << "Nomor Polisi Lama: " <<
daftarKendaraan[nomor].nomor_polisi << endl;
                cout << "Masukkan Nomor Polisi Baru: ";
                cin.ignore();
                getline(cin, daftarKendaraan[nomor].nomor_polisi);
                cout << "\nNomor polisi kendaraan berhasil diperbarui!\n";
                break;
            }
            case 4: {
                cout << "\n----- Perbarui Harga Sewa -----|\n";

```

```

        cout << "Harga Sewa Lama: Rp" << fixed << setprecision(2) <<
daftarKendaraan[nomor].harga_sewa << endl;
        cout << "Masukkan Harga Sewa Baru: Rp";
        double hargaBaru;
        cin >> hargaBaru;
        ubahHargaSewa(daftarKendaraan[nomor].harga_sewa, hargaBaru);
        tampilkanHargaSewa(&daftarKendaraan[nomor].harga_sewa);
        cout << "\nHarga sewa kendaraan berhasil diperbarui!\n";
        break;
    }
    case 5: {
        kembaliKeMenu = true;
        break;
    }
}

if (!kembaliKeMenu) {
    Enter();
}
} while (!kembaliKeMenu);

cout << "\nKembali ke menu admin...\n";
Enter();
}

```

H. Fungsi Menghapus Data Kendaraan

Fungsi hapus digunakan untuk menghapus data kendaraan dari array daftarKendaraan berdasarkan nomor urut kendaraan yang dipilih oleh pengguna. Pertama, fungsi memeriksa apakah ada data kendaraan yang tersimpan; jika tidak ada, program memberikan pesan bahwa belum ada data dan keluar dari fungsi. Jika ada data, daftar kendaraan ditampilkan menggunakan fungsi lihatKendaraan, dan pengguna diminta memasukkan nomor kendaraan yang ingin dihapus. Jika pengguna memilih untuk kembali (memasukkan 0), fungsi akan langsung keluar. Setelah nomor kendaraan dipilih, detail kendaraan tersebut ditampilkan untuk konfirmasi penghapusan. Jika pengguna mengonfirmasi dengan memasukkan 'y' atau 'Y', data kendaraan dihapus dengan menggeser elemen array ke posisi sebelumnya untuk menutupi indeks yang dihapus, lalu jumlah kendaraan dikurangi. Jika pengguna membatalkan penghapusan, fungsi akan memberikan pesan pembatalan tanpa mengubah data. Proses ini memastikan data kendaraan dikelola secara aman dan terstruktur.

```

void hapus(Kendaraan daftarKendaraan[], int &jumlahKendaraan) {
    system("cls");
    if (jumlahKendaraan == 0) {
        cout << "\n|=====|\n";
        cout << "|    Belum ada data kendaraan yang tersimpan!    |\n";
    }
}

```

```

        cout << "|=====|\n";
        Enter();
        return;
    }

    lihatKendaraan(daftarKendaraan, jumlahKendaraan);
    cout << "\n|=====|\n";
    cout << "|          Hapus Data Kendaraan          |\n";
    cout << "|=====|\n";
    cout << "Masukkan nomor kendaraan yang ingin dihapus (0 untuk kembali):
";

    int nomor = validasiInputan(0, jumlahKendaraan);

    if (nomor == 0) {
        system("cls");
        cout << "\n|=====|\n";
        cout << "|          Kembali ke menu utama...          |\n";
        cout << "|=====|\n";
        Enter();
        return;
    }

    nomor--;

    cout << "\n===== Detail Kendaraan yang Akan Dihapus =====\n";
    cout << "|-----|\n";
    cout << "| Merk          : " << setw(35) << left <<
daftarKendaraan[nomor].merk << "|\n";
    cout << "| Tipe          : " << setw(35) << left <<
daftarKendaraan[nomor].tipe << "|\n";
    cout << "| Nomor Polisi : " << setw(35) << left <<
daftarKendaraan[nomor].nomor_polisi << "|\n";
    cout << "| Harga Sewa    : Rp " << setw(32) << left << fixed <<
setprecision(2) << daftarKendaraan[nomor].harga_sewa << "|\n";
    cout << "| Status        : " << setw(35) << left <<
daftarKendaraan[nomor].status << "|\n";
    cout << "|-----|\n";

    cout << "\nApakah Anda yakin ingin menghapus data ini? (y/n): ";
    char konfirmasi;
    cin >> konfirmasi;

    if (konfirmasi == 'y' || konfirmasi == 'Y') {
        for (int i = nomor; i < jumlahKendaraan - 1; i++) {
            daftarKendaraan[i] = daftarKendaraan[i + 1];
        }
        jumlahKendaraan--;

        cout << "\n|=====|\n";
        cout << "|          Data kendaraan berhasil dihapus!          |\n";
        cout << "|=====|\n";
    }

```

```

    } else {
        cout << "\n|=====|\n";
        cout << "|          Penghapusan data dibatalkan!          |\n";
        cout << "|=====|\n";
    }

    Enter();
}

```

I. Fungsi sewa kendaraan

Fungsi sewa digunakan untuk menyewa kendaraan dengan memperbarui status kendaraan menjadi "Disewa" serta mencatat riwayat penyewaan. Pertama, fungsi memeriksa apakah ada data kendaraan yang tersedia; jika tidak ada, pengguna diberi pesan bahwa tidak ada kendaraan untuk disewa. Jika ada, daftar kendaraan ditampilkan menggunakan fungsi lihatKendaraan, dan pengguna diminta memilih nomor kendaraan yang ingin disewa. Sebelum melanjutkan, fungsi memastikan bahwa kendaraan tersebut berstatus "Tersedia". Jika tersedia, pengguna diminta memasukkan durasi sewa dalam hari, lalu status kendaraan diubah menjadi "Disewa", dan data penyewa (nama dan durasi) disimpan ke dalam array riwayatPenyewa pada indeks sesuai jumlah penyewaan sebelumnya. Fungsi juga menambahkan jumlah penyewaan kendaraan tersebut sebelum memberikan konfirmasi bahwa kendaraan berhasil disewa.

```

void sewa(Kendaraan daftarKendaraan[], int jumlahKendaraan, string nama) {
    system("cls");
    if (jumlahKendaraan == 0) {
        cout << "Belum ada data kendaraan yang tersedia untuk disewa!" <<
endl;
        Enter();
        return;
    }

    lihatKendaraan(daftarKendaraan, jumlahKendaraan);
    cout << "Masukkan nomor kendaraan yang ingin disewa: ";
    int nomor = validasiInputan(1, jumlahKendaraan) - 1;

    if (daftarKendaraan[nomor].status != "Tersedia") {
        cout << "Kendaraan ini tidak tersedia untuk disewa!" << endl;
        Enter();
        return;
    }

    cout << "Masukkan durasi sewa (dalam hari): ";
    string durasi;
    cin >> durasi;

    daftarKendaraan[nomor].status = "Disewa";
}

```

```

daftarKendaraan[nomor].riwayatPenyewa[daftarKendaraan[nomor].jumlahPenyewa].namaPenyewa = nama;

daftarKendaraan[nomor].riwayatPenyewa[daftarKendaraan[nomor].jumlahPenyewa].durasi = durasi;
daftarKendaraan[nomor].jumlahPenyewa++;

cout << "Kendaraan berhasil disewa!" << endl;
Enter();
}

```

J. Fungsi lihat Riwayat

Fungsi `lihatRiwayatUser` digunakan untuk menampilkan riwayat penyewaan kendaraan, baik untuk pengguna biasa maupun admin. Jika pengguna adalah admin (`isAdmin = true`), fungsi akan menampilkan riwayat penyewaan semua pengguna; jika tidak, hanya riwayat penyewaan pengguna tertentu yang sesuai dengan parameter `nama`. Fungsi ini mengiterasi array `daftarKendaraan` dan mencetak detail riwayat penyewaan seperti nama penyewa, merk, tipe, durasi sewa, dan status kendaraan dalam format tabel yang terstruktur. Jika tidak ada riwayat penyewaan yang sesuai, fungsi akan menampilkan pesan bahwa belum ada riwayat. Selain itu, jika ada riwayat, total jumlah riwayat akan ditampilkan di akhir tabel. Format output menggunakan `setw` untuk memastikan kolom terlihat rapi dan mudah dibaca.

```

void lihatRiwayatUser(const Kendaraan daftarKendaraan[], int jumlahKendaraan,
string nama, bool isAdmin = false) {
    system("cls");
    cout <<
"\n|=====|\n";
    cout << "|                Riwayat Penyewaan " << (isAdmin ? "Semua
User" : "Anda") << "                |\n";
    cout <<
"|=====|\n";
    cout << "| No |  Penyewa  |   Merk   |   Tipe   |  Durasi  | Status
|\n";
    cout << "|-----|
--|\n";

    bool adaRiwayat = false;
    int count = 0;

    for (int i = 0; i < jumlahKendaraan; i++) {
        for (int j = 0; j < daftarKendaraan[i].jumlahPenyewa; j++) {
            if (isAdmin || daftarKendaraan[i].riwayatPenyewa[j].namaPenyewa
== nama) {
                cout << "| " << setw(2) << ++count << " | "
<< setw(9) <<

```



```

daftarKendaraan[i].riwayatPenyewa[j].namaPenyewa << " | "
                << setw(11) << daftarKendaraan[i].merk << " | "
                << setw(11) << daftarKendaraan[i].tipe << " | "
                << setw(8) <<
daftarKendaraan[i].riwayatPenyewa[j].durasi + " Hari" << " | "
                << setw(7) << daftarKendaraan[i].status << " |\n";
        adaRiwayat = true;
    }
}

if (!adaRiwayat) {
    cout << "|
\n";
    cout << "|                      Belum ada riwayat penyewaan!
\n";
    cout << "|
\n";
}

cout <<
"|=====|\n";
if (adaRiwayat) {
    cout << "| Total Riwayat: " << setw(47) << left << count << " |\n";
    cout <<
"|=====|\n";
}
Enter();
}

```

K. Fungsi login

Fungsi `loginUser` digunakan untuk memverifikasi identitas pengguna yang mencoba masuk ke sistem dengan memeriksa kombinasi username dan password. Fungsi ini memberikan maksimal tiga percobaan login, dan pada setiap percobaan, pengguna diminta memasukkan username dan password. Jika username adalah "EgaClearestaHananta" dengan password "2409106088", pengguna akan login sebagai admin (`isAdmin = true`). Jika username dan password cocok dengan data yang tersimpan dalam map pengguna, pengguna dianggap berhasil login sebagai pengguna biasa (`isAdmin = false`). Jika gagal, pengguna diberi pesan kesalahan dan dapat mencoba kembali hingga tiga kali. Jika semua percobaan gagal, program akan memberikan pesan bahwa login dibatalkan dan berhenti. Fungsi ini juga menggunakan `Enter()` untuk memberikan jeda agar pengguna dapat membaca pesan sebelum melanjutkan.

```

bool loginUser(map<string, string>& pengguna, string& nama, string& password,

```

```

bool& isAdmin) {
    int percobaan = 0;
    while (percobaan < 3) {
        system("cls");
        cout << "|=====|\n";
        cout << "|          Login ke Sistem          |\n";
        cout << "|=====|\n";
        cout << "Masukkan username: ";
        cin >> nama;
        cout << "Masukkan password: ";
        cin >> password;

        if (nama == "EgaClearestaHananta" && password == "2409106088") {
            isAdmin = true;
            cout << "Login sebagai admin berhasil!\n";
            return true;
        } else if (pengguna.find(nama) != pengguna.end() && pengguna[nama] ==
password) {
            isAdmin = false;
            cout << "Login berhasil!\n";
            return true;
        } else {
            percobaan++;
            cout << "username atau password salah! Percobaan ke-" <<
percobaan << " dari 3.\n";
            Enter();
        }
    }

    cout << "Anda telah salah memasukkan username atau password sebanyak 3
kali. Program berhenti.\n";
    Enter();
    return false;
}

```

L. Fungsi register

Fungsi `registerUser` digunakan untuk mendaftarkan akun pengguna baru dengan memasukkan username dan password, serta menyimpannya ke dalam struktur data `map<string, string>` bernama `pengguna`. Pertama, fungsi membersihkan layar menggunakan `system("cls")` dan menampilkan menu pendaftaran. Pengguna diminta memasukkan username baru, lalu fungsi memeriksa apakah username tersebut sudah terdaftar di dalam `map` menggunakan metode `find`. Jika username sudah ada, pengguna diberi pesan bahwa username telah digunakan dan diminta mencoba username lain. Jika username belum terdaftar, pengguna diminta memasukkan password, dan pasangan username-password disimpan ke dalam `map`. Setelah berhasil, pengguna mendapat konfirmasi bahwa registrasi berhasil dan diminta untuk login agar dapat

mengakses sistem. Fungsi ini memastikan tidak ada duplikasi username dalam sistem.

```
void registerUser(map<string, string>& pengguna) {
    system("cls");
    string nama, password;

    cout << "\n|=====|\n";
    cout << "|          Register Akun          |\n";
    cout << "|=====|\n";
    cout << "Masukkan username baru: ";
    cin >> nama;

    if (pengguna.find(nama) != pengguna.end()) {
        cout << "\n|=====|\n";
        cout << "|      Username sudah terdaftar!      |\n";
        cout << "|      Silahkan gunakan username lain      |\n";
        cout << "|=====|\n";
        Enter();
        return;
    }

    cout << "Masukkan password baru: ";
    cin >> password;

    pengguna[nama] = password;

    cout << "\n|=====|\n";
    cout << "|          Registrasi Berhasil!          |\n";
    cout << "|      Silahkan login untuk masuk      |\n";
    cout << "|=====|\n";
    Enter();
}
```

M. Alur Program

Fungsi main() merupakan pusat kendali dari seluruh program sistem penyewaan kendaraan. Program dimulai dengan inisialisasi array kendaraan, variabel jumlah kendaraan, dan map untuk menyimpan data pengguna. Program berjalan dalam dua loop utama: pertama adalah loop menu awal, yang memungkinkan pengguna untuk melakukan registrasi, login, atau keluar dari program. Setelah login berhasil, program masuk ke loop kedua yaitu menu utama, yang menampilkan opsi berbeda tergantung apakah pengguna masuk sebagai admin atau pengguna biasa. Admin memiliki akses penuh untuk mengelola data kendaraan dan melihat riwayat penyewaan, sedangkan pengguna biasa hanya bisa melihat kendaraan, menyewa, dan melihat riwayat miliknya. Program ini akan terus berjalan hingga pengguna memilih keluar, sehingga tetap responsif terhadap input pengguna selama program aktif.

```

int main() {
    Kendaraan daftarKendaraan[MAX_DATA];
    int jumlahKendaraan = 0;
    map<string, string> pengguna;
    string nama, password;
    bool isAdmin = false;
    bool login = false;

    pengguna["user1"] = "password1";
    pengguna["user2"] = "password2";

    while (true) {
        system("cls");
        cout << "|=====|\n";
        cout << "| Sistem Pendataan Penyewaan Kendaraan |\n";
        cout << "|====|=====|\n";
        cout << "| 1 | Login |\n";
        cout << "| 2 | Register |\n";
        cout << "| 3 | Keluar |\n";
        cout << "|====|=====|\n";
        cout << "Pilihan: ";

        int pilihan = validasiInputan(1, 3);

        switch (pilihan) {
            case 1:
                login = loginUser(pengguna, nama, password, isAdmin);
                break;
            case 2:
                registerUser(pengguna);
                continue;
            case 3:
                keluar();
                return 0;
        }

        if (login) {
            string namaPengguna = nama;
            int pilihanMenu;
            bool loginState = true;
            do {
                Menu(isAdmin);
                cout << "Pilihan: ";
                pilihanMenu = validasiInputan(1, isAdmin ? 6 : 4);

                if (isAdmin) {
                    switch (pilihanMenu) {
                        case 1:
                            tambah(daftarKendaraan, jumlahKendaraan);
                            break;
                        case 2:

```

```

        lihatKendaraan(daftarKendaraan, jumlahKendaraan);
        break;
    case 3:
        perbarui(daftarKendaraan, jumlahKendaraan);
        break;
    case 4:
        hapus(daftarKendaraan, jumlahKendaraan);
        break;
    case 5:
        lihatRiwayatUser(daftarKendaraan,
jumlahKendaraan, namaPegguna, isAdmin);
        break;
    case 6:
        loginState = false;
        cout << "Anda telah logout. Kembali ke menu
awal..." << endl;

        Enter();
        break;
    }
} else {
    switch (pilihanMenu) {
        case 1:
            lihatKendaraan(daftarKendaraan, jumlahKendaraan);
            break;
        case 2:
            sewa(daftarKendaraan, jumlahKendaraan,
namaPegguna);

            break;
        case 3:
            lihatRiwayatUser(daftarKendaraan,
jumlahKendaraan, namaPegguna);
            break;
        case 4:
            loginState = false;
            cout << "Anda telah logout. Kembali ke menu
awal..." << endl;

            Enter();
            break;
    }
}
} while (loginState);
}
}

return 0;
}

```

4. Uji Coba dan Hasil Output

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
=====
| 1 | Login |
| 2 | Register |
| 3 | Keluar |
=====
Pilihan: 1
=====
| Login ke Sistem |
=====
Masukkan username: EgaClearestaHananta
Masukkan password: 2409106088
Login sebagai admin berhasil!
```

Gambar 4.1 Login Sebagai Admin

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
=====
| No | Menu |
=====
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
=====
Pilihan: 1
===== Tambah Data Kendaraan =====
Masukkan Merk Kendaraan: Avanza
Masukkan Tipe Kendaraan: Honda
Masukkan Nomor Polisi: KT 4070
Masukkan Harga Sewa per Hari: 200000
Data kendaraan berhasil ditambahkan!
Tekan Enter untuk melanjutkan...
```

Gambar 4 2 Fitur Tambah Data Kendaraan

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
=====
| No | Menu |
=====
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
=====
Pilihan: 2
===== Data Kendaraan =====
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|-----|-----|-----|-----|-----|
| 1 | Avanza | Honda | KT 4070 | Rp 200000 | Tersedia |
|-----|-----|-----|-----|-----|-----|
Tekan Enter untuk melanjutkan...
```

Gambar 4 3 Fitur melihat Data Kendaraan

```
=====
| No | Menu |
|=====|
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|

Pilihan: 3

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 4070 | Rp 200000 | Tersedia |
|-----|

Tekan Enter untuk melanjutkan...
Masukkan nomor kendaraan yang ingin diperbarui: 1

===== Detail Kendaraan Terpilih =====
|-----|
| Merk : Avanza |
| Tipe : Honda |
| Nomor Polisi : KT 4070 |
| Harga Sewa : Rp 200000.00 |
| Status : Tersedia |
|-----|

===== Menu Perbarui Data =====
|-----|
| No | Pilihan Menu |
|-----|
| 1 | Perbarui Merk |
| 2 | Perbarui Tipe |
| 3 | Perbarui Nomor Polisi |
| 4 | Perbarui Harga Sewa |
| 5 | Kembali ke Menu Utama |
|-----|

Pilih opsi: 1

----- Perbarui Merk -----
Merk Lama : Avanza
Masukkan Merk Baru: Terios

Merk kendaraan berhasil diperbarui!
```

Gambar 4 4 Fitur Memperbarui Data Kendaraan

```

=====
| Sistem Pendataan Penyewaan Kendaraan |
=====
| No | Menu |
=====
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
=====
Pilihan: 4

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Terios | Honda | KT 4070 | Rp 200000 | Tersedia |
|-----|
Tekan Enter untuk melanjutkan...

|-----|
| Hapus Data Kendaraan |
|-----|
Masukkan nomor kendaraan yang ingin dihapus (0 untuk kembali): 1

===== Detail Kendaraan yang Akan Dihapus =====
|-----|
| Merk : Terios |
| Tipe : Honda |
| Nomor Polisi : KT 4070 |
| Harga Sewa : Rp 200000.00 |
| Status : Tersedia |
|-----|

Apakah Anda yakin ingin menghapus data ini? (y/n): y

|-----|
| Data kendaraan berhasil dihapus! |
|-----|
Tekan Enter untuk melanjutkan...

```

Gambar 4 5 Fitur Menghapus Data Kendaraan


```
=====|
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|====|=====|
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|
```

Pilihan: 5

```
|=====|
|                               |
|          Riwayat Penyewaan Semua User          |
|=====|
| No | Penyewa | Merk | Tipe | Durasi | Status |
|-----|
|
|          Belum ada riwayat penyewaan!          |
|=====|
```

Tekan Enter untuk melanjutkan...|

Gambar 4 6 fitur Melihat Riwayat penyewa tetapi belum ada penyewa

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|=====|
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|

Pilihan: 6
Anda telah logout. Kembali ke menu awal...
Tekan Enter untuk melanjutkan...

=====
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| 1 | Login |
| 2 | Register |
| 3 | Keluar |
|=====|
```

Gambar 4 7 Fitur logout di menu admin

```

=====
|                                     |
|                               Register Akun                               |
|                                     |
|=====|
Masukkan username baru: R
Masukkan password baru: 123

|=====|
|                               Registrasi Berhasil!                       |
|                               Silahkan login untuk masuk                 |
|=====|
Tekan Enter untuk melanjutkan...

```

Gambar 4 8 melakukan registrasi

```

=====
|                                     |
|                               Login ke Sistem                             |
|                                     |
|=====|
Masukkan username: R
Masukkan password: 123
Login berhasil!

|=====|
| Sistem Pendataan Penyewaan Kendaraan                                     |
|=====|
| No | Menu                               |
|=====|
| 1 | Lihat Data Kendaraan               |
| 2 | Sewa Kendaraan                     |
| 3 | Lihat Riwayat Penyewaan Anda       |
| 4 | Logout                             |
|=====|
Pilihan:

```

Gambar 4 9 Login sebagai Pengguna

```

=====
| Sistem Pendataan Penyewaan Kendaraan                                     |
|=====|
| No | Menu                               |
|=====|
| 1 | Lihat Data Kendaraan               |
| 2 | Sewa Kendaraan                     |
| 3 | Lihat Riwayat Penyewaan Anda       |
| 4 | Logout                             |
|=====|
Pilihan: 1

===== Data Kendaraan =====
|-----|
| No | Merk      | Tipe   | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza    | Honda  | KT 3080      | Rp 200000  | Tersedia |
|-----|
Tekan Enter untuk melanjutkan...

```

Gambar 4 10 Melihat Data Kendaraan

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|=====|
| 1 | Lihat Data Kendaraan |
| 2 | Sewa Kendaraan |
| 3 | Lihat Riwayat Penyewaan Anda |
| 4 | Logout |
|=====|
Pilihan: 2

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 3080 | Rp 200000 | Tersedia |
|-----|
Tekan Enter untuk melanjutkan...
Masukkan nomor kendaraan yang ingin disewa: 1
Masukkan durasi sewa (dalam hari): 10
Kendaraan berhasil disewa!
Tekan Enter untuk melanjutkan...|
```

Gambar 4 11 Fitur sewa kendaraan

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|=====|
| 1 | Lihat Data Kendaraan |
| 2 | Sewa Kendaraan |
| 3 | Lihat Riwayat Penyewaan Anda |
| 4 | Logout |
|=====|
Pilihan: 3

|=====|
| Riwayat Penyewaan Anda |
|=====|
| No | Penyewa | Merk | Tipe | Durasi | Status |
|-----|
| 1 | R | Avanza | Honda | 10 Hari | Disewa |
|-----|
| Total Riwayat: 1 |
|=====|
Tekan Enter untuk melanjutkan...|
```

Gambar 4 12 Melihat Riwayat Sendiri

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|=====|
| 1 | Tambah Data Kendaraan |
| 2 | Tampilkan Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|
Pilihan: 5

|=====|
|                               Riwayat Penyewaan Semua User |
|=====|
| No | Penyewa | Merk | Tipe | Durasi | Status |
|-----|-----|-----|-----|-----|-----|
| 1 | R | Avanza | Honda | 10 Hari | Disewa |
|-----|-----|-----|-----|-----|
| Total Riwayat: 1 |
|=====|
Tekan Enter untuk melanjutkan...
```

Gambar 4 13 Melihat Riwayat Penyewa ketika sudah ada yang sewa

5. Langkah-Langkah Git

5.1 git add

Perintah git add digunakan untuk menambahkan file apa saja sebelum dilakukannya commit.

```
PS C:\Users\EGA\OneDrive\Dokumen\Praktikum-Apl\post-test\post-test-5> Git add .
```

Gambar 5 1 git add

5.2 git commit

Perintah git commit digunakan untuk menyimpan perubahan yang telah di tambahkan ke dalam repository git.

```
PS C:\Users\EGA\OneDrive\Dokumen\Praktikum-Apl\post-test\post-test-5> git commit -m "finish posttest 5"
```

Gambar 5 2 git commit

5.3 git push

perintah git push digunakan untuk mengunggahnya perubahan dari repository lokal ke github.

```
PS C:\Users\EGA\OneDrive\Dokumen\Praktikum-Apl\post-test\post-test-5> git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 697.99 KiB | 6.23 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/egaxxxx/praktikum_apl.git
d52b3aa..09399c5 main -> main
PS C:\Users\EGA\OneDrive\Dokumen\Praktikum-Apl\post-test\post-test-5>
```

Gambar 5 3 git push