

**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**

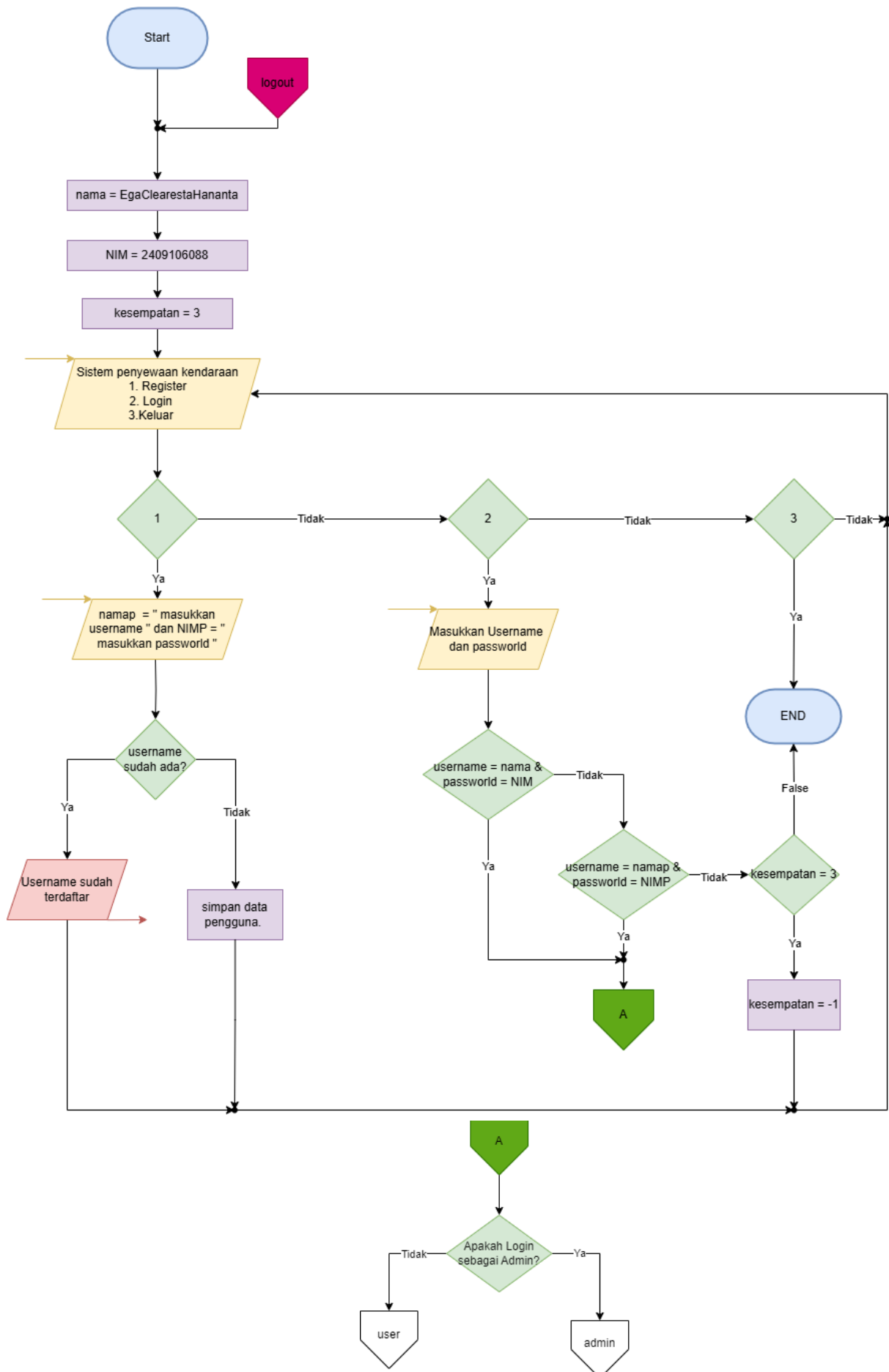


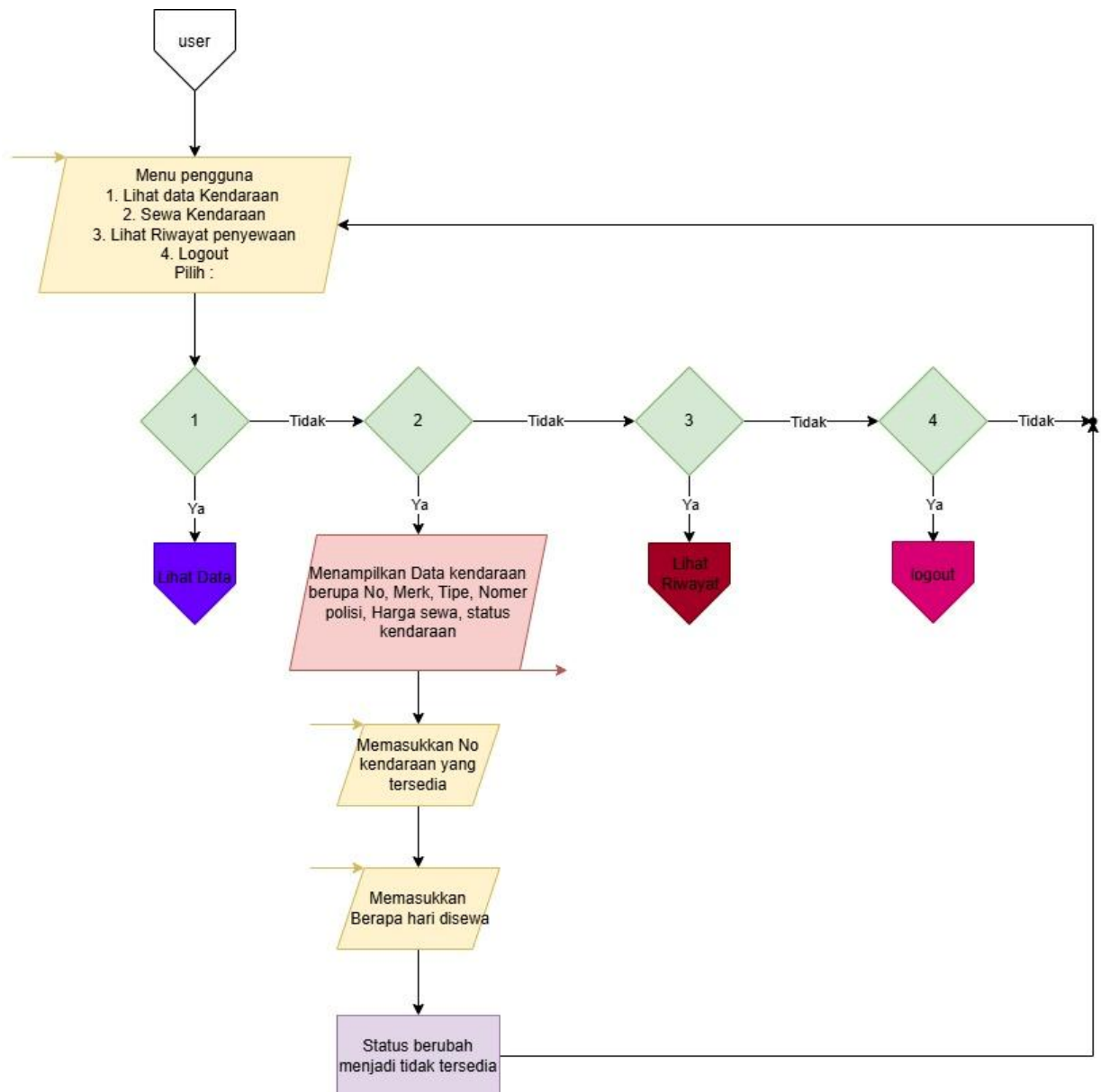
**Disusun oleh:**  
**Ega Clearesta Hananta (2409106088)**  
**Kelas (B2 '24)**

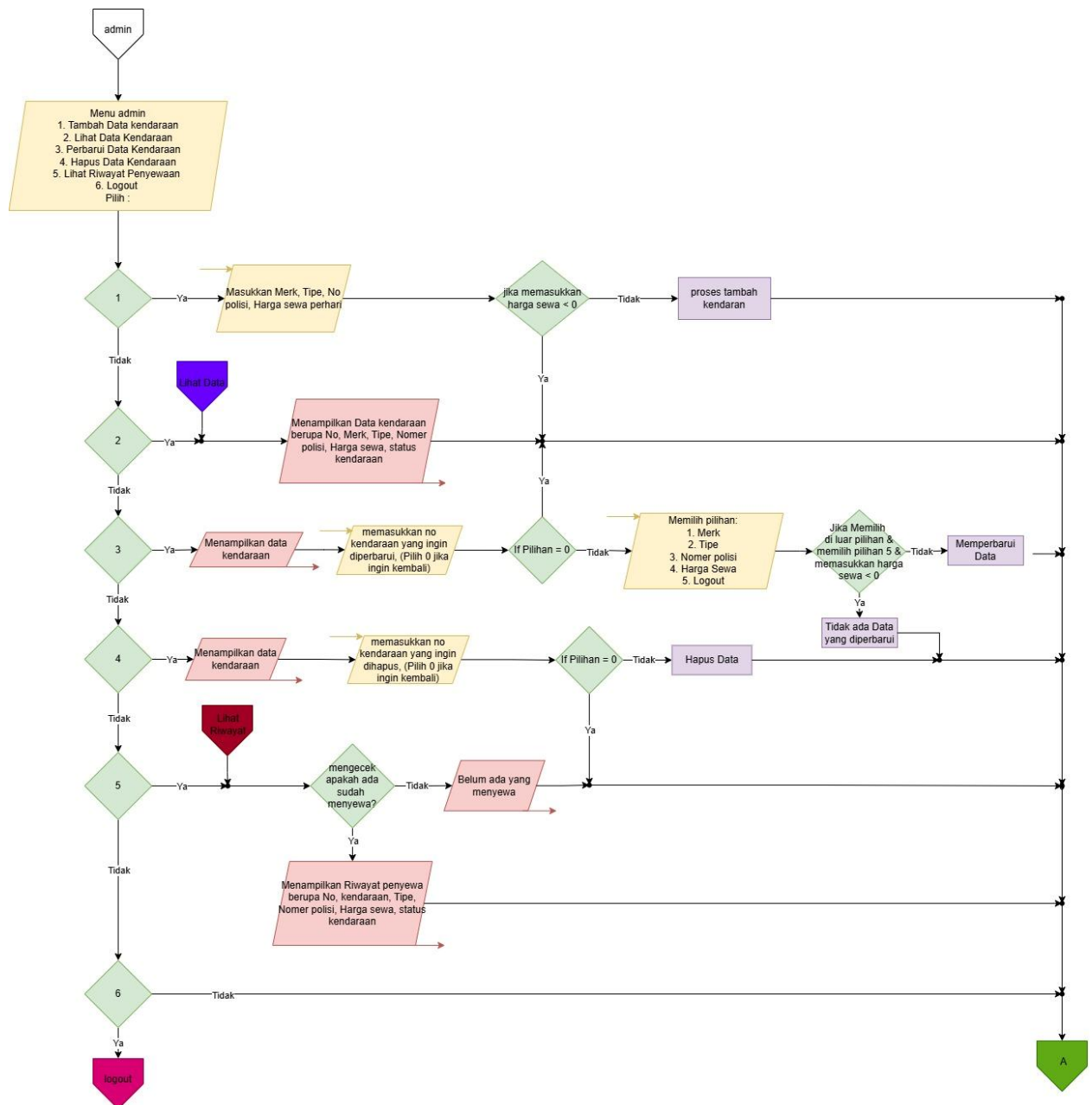
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**

**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 digunakan untuk merepresentasikan data penyewaan kendaraan oleh pengguna, yang terdiri dari nama penyewa dan durasi penyewaan. Struktur ini dimasukkan sebagai bagian dari struct Kendaraan, yang menyimpan informasi lengkap mengenai kendaraan, seperti merk, tipe, nomor polisi, harga sewa per hari, dan status kendaraan (misalnya "Tersedia" atau "Disewa"). Selain itu, setiap objek Kendaraan memiliki array riwayatPenyewa untuk mencatat riwayat penyewaan berdasarkan data RiwayatSewa, dan variabel jumlahPenyewa untuk menghitung berapa kali kendaraan telah disewa. Struktur ini memungkinkan sistem menyimpan dan mengelola data kendaraan beserta histori penyewaannya secara terorganisir.

```
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. Fungsi Validasi Input

Fungsi validasiInputan(int min, int max) digunakan untuk memastikan bahwa input dari pengguna berupa angka dan berada dalam rentang yang ditentukan. Jika input tidak sesuai baik karena bukan angka atau berada di luar batas minimum dan maksimum maka fungsi akan membersihkan error pada input (cin), menampilkan pesan kesalahan, dan memanggil dirinya sendiri secara rekursif hingga mendapatkan input yang valid. Fungsi ini sangat berguna untuk menjaga kestabilan program dari input yang tidak sesuai.

```
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 Enter ()

Fungsi Enter() digunakan untuk menghentikan sementara jalannya program dan menunggu pengguna menekan tombol Enter sebelum melanjutkan. Fungsi ini pertama-tama membersihkan buffer input menggunakan cin.ignore() untuk menghindari karakter sisa yang mungkin mengganggu input berikutnya, lalu menampilkan pesan "Tekan Enter untuk melanjutkan..." dan menggunakan cin.get() untuk menunggu input dari pengguna. Fungsi ini sering dipakai setelah menampilkan pesan atau hasil proses agar pengguna punya waktu untuk membaca sebelum program berlanjut.

```

void Enter() {
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << "Tekan Enter untuk melanjutkan...";
    cin.get();
}

```

### D. Fungsi Menu Admin atau pengguna

Fungsi Menu(bool isAdmin) digunakan untuk menampilkan tampilan menu utama berdasarkan peran pengguna. Jika parameter isAdmin bernilai true, maka akan ditampilkan menu khusus admin yang mencakup fitur manajemen data kendaraan dan riwayat penyewaan. Sebaliknya, jika isAdmin bernilai false, maka ditampilkan menu untuk pengguna biasa yang hanya memiliki akses untuk melihat data kendaraan, melakukan penyewaan, melihat riwayat pribadi, dan logout. Fungsi ini mempermudah pemisahan akses berdasarkan hak pengguna dalam sistem penyewaan kendaraan.

```

void Menu(bool isAdmin) {
    if (isAdmin) {
        cout << "|=====|\n";
        cout << "| Sistem Pendataan Penyewaan Kendaraan |\n";
        cout << "|====|=====|\n";
        cout << "| No | Menu |\n";
        cout << "|====|=====|\n";
        cout << "| 1 | Tambah Data Kendaraan |\n";
        cout << "| 2 | Lihat Data Kendaraan |\n";
        cout << "| 3 | Perbarui Data Kendaraan |\n";
        cout << "| 4 | Hapus Data Kendaraan |\n";
        cout << "| 5 | Lihat Riwayat Penyewaan |\n";
    }
}

```

```

        cout << "| 6 | Logout                                |\n";
        cout << "|====|=====|\n";
    } else {
        cout << "|=====|\n";
        cout << "| Sistem Pendataan Penyewaan Kendaraan |\n";
        cout << "|====|=====|\n";
        cout << "| No | Menu                                |\n";
        cout << "|====|=====|\n";
        cout << "| 1 | Lihat Data Kendaraan                |\n";
        cout << "| 2 | Sewa Kendaraan                      |\n";
        cout << "| 3 | Lihat Riwayat Penyewaan Anda        |\n";
        cout << "| 4 | Logout                              |\n";
        cout << "|====|=====|\n";
    }
}
}

```

#### E. Fungsi Tambah Kendaraan

Fungsi tambah() digunakan oleh admin untuk menambahkan data kendaraan baru ke dalam sistem penyewaan. Fungsi ini terlebih dahulu mengecek apakah jumlah data kendaraan telah mencapai batas maksimum (MAX\_DATA). Jika belum penuh, program akan meminta input dari pengguna berupa merk, tipe, nomor polisi, dan harga sewa per hari. Input harga sewa divalidasi agar harus berupa angka dan lebih dari 0. Setelah semua data valid, status kendaraan akan otomatis diatur menjadi "Tersedia" dan jumlah penyewa diinisialisasi ke nol. Data kendaraan baru kemudian disimpan ke array dan jumlah kendaraan ditambahkan. Di akhir, program menampilkan notifikasi bahwa data berhasil ditambahkan dan menunggu pengguna menekan Enter sebelum melanjutkan.

```

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

    Kendaraan k;
    cin.ignore();
    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;
    } while (k.harga_sewa <= 0);
}

```



```

        if (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "\nHarga sewa tidak valid! Masukkan angka.\n";
            continue;
        }

        if (k.harga_sewa <= 0) {
            cout << "\nHarga sewa harus lebih dari 0!\n";
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            continue;
        }

        break;
    } while (true);

    k.status = "Tersedia";
    k.jumlahPenyewa = 0;

    daftarKendaraan[jumlahKendaraan] = k;
    jumlahKendaraan++;

    cout << "\nData kendaraan berhasil ditambahkan!\n";
    Enter();
}

```

#### F. Fungsi Lihat Data Kendaraan untuk Admin

Fungsi lihatKendaraan() digunakan untuk menampilkan seluruh data kendaraan yang tersimpan dalam sistem. Jika belum ada data kendaraan (`jumlahKendaraan == 0`), program akan menampilkan pesan bahwa data masih kosong dan kembali ke menu setelah menekan Enter. Jika ada data, program mencetak daftar kendaraan dalam bentuk tabel yang rapi, lengkap dengan informasi nomor urut, merk, tipe, nomor polisi, harga sewa, dan status kendaraan. Fungsi ini memanfaatkan `setw()` dan `setprecision()` agar tampilan tabel terlihat rapi dan mudah dibaca oleh pengguna. Setelah semua data ditampilkan, fungsi akan menunggu pengguna menekan Enter sebelum melanjutkan ke proses berikutnya.

```

void lihatKendaraan(const Kendaraan daftarKendaraan[], int jumlahKendaraan) {
    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(15) << daftarKendaraan[i].merk << " | "
                << setw(15) << daftarKendaraan[i].tipe << " | "
                << setw(15) << daftarKendaraan[i].nomor_polisi << " | "
                << "Rp. " << setw(11) << fixed << setprecision(0) <<
daftarKendaraan[i].harga_sewa << " | "
                << setw(10) << daftarKendaraan[i].status << "      |\n";
        }
        cout << "|-----|
-----|\n";
        Enter();
    }
}

```

#### G. Fungsi lihat data kendaraan untuk pengguna

Fungsi lihatKendaraan() digunakan untuk menampilkan seluruh data kendaraan yang tersimpan dalam sistem. Jika belum ada data kendaraan (`jumlahKendaraan == 0`), program akan menampilkan pesan bahwa data masih kosong dan kembali ke menu setelah menekan Enter. Jika ada data, program mencetak daftar kendaraan dalam bentuk tabel yang rapi, lengkap dengan informasi nomor urut, merk, tipe, nomor polisi, harga sewa, dan status kendaraan. Fungsi ini memanfaatkan `setw()` dan `setprecision()` agar tampilan tabel terlihat rapi dan mudah dibaca oleh pengguna. Setelah semua data ditampilkan, fungsi akan menunggu pengguna menekan Enter sebelum melanjutkan ke proses berikutnya.

```

void lihatKendaraan(const Kendaraan daftarKendaraan[], int jumlahKendaraan,
string status) {
    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(15) << daftarKendaraan[i].merk << " | "
            << setw(15) << daftarKendaraan[i].tipe << " | "
            << setw(15) << daftarKendaraan[i].nomor_polisi << " | "
            << "Rp. " << setw(11) << fixed << setprecision(0) <<
daftarKendaraan[i].harga_sewa << " | "
            << setw(10) << daftarKendaraan[i].status << "      | \n";
    }
}

if (count == 0) {
    cout << "| Tidak ada kendaraan dengan status " << status << " saat
ini.                               | \n";
}
cout << "|-----| \n";
-----| \n";
Enter();
}

```

#### H. Fungsi Sewa Kendaraan untuk pengguna

Fungsi sewa() digunakan oleh pengguna untuk menyewa kendaraan dari sistem. Pertama, fungsi akan mengecek apakah ada data kendaraan yang tersedia. Jika ada, maka akan ditampilkan seluruh daftar kendaraan. Pengguna diminta memilih kendaraan berdasarkan nomor urut, dan input tersebut divalidasi agar sesuai dengan rentang. Setelah dipilih, program akan memeriksa apakah kendaraan tersebut berstatus "Tersedia". Jika tidak, penyewaan dibatalkan. Jika tersedia, pengguna diminta memasukkan durasi sewa, lalu data penyewa dan durasinya disimpan dalam array riwayat penyewa kendaraan tersebut, dan status kendaraan diubah menjadi "Disewa". Fungsi ini memastikan hanya kendaraan yang tersedia saja yang dapat disewa dan mencatat informasi penyewaan secara otomatis.

```

void sewa(Kendaraan daftarKendaraan[], int jumlahKendaraan, string nama) {
    if (jumlahKendaraan == 0) {
        cout << "\nTidak ada kendaraan yang tersedia untuk disewa.\n";
        Enter();
        return;
    }
    lihatKendaraan(daftarKendaraan, jumlahKendaraan);
    cout << "Masukkan nomor kendaraan yang ingin disewa: ";
    int nomor = validasiInputan(1, jumlahKendaraan) - 1;

    if (daftarKendaraan[nomor].status != "Tersedia") {
        cout << "\nKendaraan ini tidak tersedia untuk disewa.\n";
        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 << "\nKendaraan berhasil disewa.\n";
    Enter();
}

```

## I. Fungsi Lihat Riwayat untuk Pengguna

Fungsi `lihatRiwayatUser()` digunakan untuk menampilkan riwayat penyewaan kendaraan yang dilakukan oleh pengguna tertentu, berdasarkan nama yang diberikan. Fungsi ini akan mencari data riwayat dalam setiap kendaraan dan mencocokkannya dengan nama pengguna. Jika ditemukan, akan ditampilkan dalam bentuk tabel yang berisi informasi merk, tipe, dan durasi penyewaan. Variabel `adaRiwayat` digunakan sebagai penanda untuk mengecek apakah pengguna memang pernah menyewa kendaraan. Jika tidak ditemukan riwayat sama sekali, maka akan ditampilkan pesan khusus bahwa pengguna belum memiliki riwayat penyewaan. Setelah selesai menampilkan data, program akan menunggu pengguna menekan Enter sebelum kembali ke menu.

```

void lihatRiwayatUser(const Kendaraan daftarKendaraan[], int jumlahKendaraan,
string nama) {
    cout << "|=====|\n";
    cout << "|                Riwayat Penyewaan Anda                |\n";
    cout << "|=====|\n";
    cout << "| No |      Merk      |      Tipe      | Durasi (hari) |\n";
    cout << "|=====|\n";

    bool adaRiwayat = false;

    for (int i = 0; i < jumlahKendaraan; i++) {
        for (int j = 0; j < daftarKendaraan[i].jumlahPenyewa; j++) {
            if (daftarKendaraan[i].riwayatPenyewa[j].namaPenyewa == nama) {
                cout << "| " << setw(2) << i + 1 << " | "
                    << setw(15) << daftarKendaraan[i].merk << " | "
                    << setw(15) << daftarKendaraan[i].tipe << " | "
                    << setw(15) <<
                daftarKendaraan[i].riwayatPenyewa[j].durasi << " |\n";
            }
        }
    }
}

```

```

        adaRiwayat = true;
    }
}

if (!adaRiwayat) {
    cout << "| Tidak ada riwayat penyewaan untuk Anda.
|\\n";
}

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

```

### J Fungsi Lihat Riwayat untuk Admin

Fungsi `lihatRiwayatUser()` digunakan untuk menampilkan riwayat penyewaan kendaraan yang dilakukan oleh pengguna tertentu, berdasarkan nama yang diberikan. Fungsi ini akan mencari data riwayat dalam setiap kendaraan dan mencocokkannya dengan nama pengguna. Jika ditemukan, akan ditampilkan dalam bentuk tabel yang berisi informasi merk, tipe, dan durasi penyewaan. Variabel `adaRiwayat` digunakan sebagai penanda untuk mengecek apakah pengguna memang pernah menyewa kendaraan. Jika tidak ditemukan riwayat sama sekali, maka akan ditampilkan pesan khusus bahwa pengguna belum memiliki riwayat penyewaan. Setelah selesai menampilkan data, program akan menunggu pengguna menekan Enter sebelum kembali ke menu.

```

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

    cout <<
    "\\n=====|\\n";
    cout << "| No |      Merk      |      Tipe      |  Penyewa      |
Durasi (hari) |\\n";
    cout <<
    "\\n=====|\\n";

    bool adaRiwayat = false;
    for (int i = 0; i < jumlahKendaraan; i++) {
        for (int j = 0; j < daftarKendaraan[i].jumlahPenyewa; j++) {
            cout << "| " << setw(2) << i + 1 << " | "
                << setw(15) << daftarKendaraan[i].merk << " | "

```

```

                << setw(15) << daftarKendaraan[i].tipe << " | "
                << setw(15) <<
daftarKendaraan[i].riwayatPenyewa[j].namaPenyewa << " | "
                << setw(15) << daftarKendaraan[i].riwayatPenyewa[j].durasi
<< " |\n";
                adaRiwayat = true;
            }
        }

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

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

```

#### K. Fungsi perbarui data kendaraan

Fungsi perbarui() digunakan oleh admin untuk memperbarui data kendaraan yang telah disimpan dalam sistem. Fungsi ini terlebih dahulu menampilkan seluruh data kendaraan, kemudian meminta pengguna memilih kendaraan yang ingin diperbarui berdasarkan nomor urut. Jika pengguna memilih 0, maka proses dibatalkan dan kembali ke menu admin. Setelah kendaraan dipilih, akan muncul submenu yang memungkinkan admin untuk memperbarui bagian tertentu dari data, yaitu merk, tipe, nomor polisi, atau harga sewa. Input divalidasi, terutama untuk harga sewa yang harus berupa angka positif. Proses pembaruan dilakukan secara interaktif dalam loop hingga admin memilih untuk kembali. Fungsi ini membantu memastikan data kendaraan tetap dapat diperbarui dengan mudah dan aman saat terjadi perubahan informasi.

```

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

    lihatKendaraan(daftarKendaraan, jumlahKendaraan);

    cout << "Masukkan nomor kendaraan yang ingin diperbarui (atau 0 untuk
kembali): ";
    int nomor = validasiInputan(0, jumlahKendaraan);

```

```

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

nomor--;

bool kembaliKeMenu = false;
do {
    cout << "\n===== \n";
    cout << "| Pilih Data yang Ingin Diperbarui | \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 | \n";
    cout << "===== \n";
    cout << "Pilih opsi: ";
    int pilihan = validasiInputan(1, 5);

    switch (pilihan) {
        case 1: {
            cout << "Masukkan Merk Kendaraan Baru: ";
            cin.ignore();
            getline(cin, daftarKendaraan[nomor].merk);
            cout << "Merk kendaraan berhasil diperbarui!\n";
            break;
        }
        case 2: {
            cout << "Masukkan Tipe Kendaraan Baru: ";
            cin.ignore();
            getline(cin, daftarKendaraan[nomor].tipe);
            cout << "Tipe kendaraan berhasil diperbarui!\n";
            break;
        }
        case 3: {
            cout << "Masukkan Nomor Polisi Baru: ";
            cin.ignore();
            getline(cin, daftarKendaraan[nomor].nomor_polisi);
            cout << "Nomor polisi kendaraan berhasil diperbarui!\n";
            break;
        }
        case 4: {
            cout << "Masukkan Harga Sewa Baru: ";
            cin >> daftarKendaraan[nomor].harga_sewa;
            if (cin.fail() || daftarKendaraan[nomor].harga_sewa <= 0) {
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                cout << "\nHarga sewa tidak valid! Perubahan

```

```

dibatalkan.\n";
    } else {
        cout << "Harga sewa kendaraan berhasil diperbarui!\n";
    }
    break;
}
case 5: {
    kembaliKeMenu = true;
    break;
}
default: {
    cout << "Pilihan tidak valid!\n";
    break;
}
}

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

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

```

#### L. Hapus Data Kendaraan

Fungsi hapus() digunakan oleh admin untuk menghapus data kendaraan dari sistem.

Pertama-tama, fungsi akan mengecek apakah terdapat data kendaraan yang tersimpan. Jika ada, maka data kendaraan akan ditampilkan, lalu admin diminta untuk memilih kendaraan mana yang ingin dihapus berdasarkan nomor urut. Apabila admin memilih angka 0, maka proses akan dibatalkan dan kembali ke menu. Jika kendaraan dipilih, data tersebut akan dihapus dengan cara menggeser seluruh data setelahnya satu posisi ke atas dalam array, lalu mengurangi jumlah total kendaraan (jumlahKendaraan). Setelah penghapusan berhasil, program akan menampilkan pesan konfirmasi dan menunggu pengguna menekan Enter sebelum kembali ke menu admin.

```

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

    lihatKendaraan(daftarKendaraan, jumlahKendaraan);

    cout << "Masukkan nomor kendaraan yang ingin dihapus (atau 0 untuk

```



```

kembali): ";
    int nomor = validasiInputan(0, jumlahKendaraan);

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

    nomor--;

    for (int i = nomor; i < jumlahKendaraan - 1; i++) {
        daftarKendaraan[i] = daftarKendaraan[i + 1];
    }
    jumlahKendaraan--;

    cout << "\nData kendaraan berhasil dihapus!\n";
    Enter();
}

```

#### M. Fungsi Register untuk pengguna baru

Fungsi `registerUser()` digunakan untuk mendaftarkan akun pengguna baru ke dalam sistem dengan menyimpan username dan password ke dalam struktur data map. Pertama, program meminta input username, lalu memeriksa apakah username tersebut sudah ada dalam map. Jika sudah terdaftar, pengguna akan diminta untuk menggunakan nama lain dan proses registrasi dibatalkan. Jika belum terdaftar, pengguna akan diminta memasukkan password, kemudian data tersebut disimpan ke dalam map pengguna. Setelah berhasil, program akan menampilkan pesan konfirmasi bahwa registrasi telah sukses dan menunggu pengguna menekan Enter sebelum melanjutkan.

```

void registerUser(map<string, string>& pengguna) {
    string nama, password;

    cout << "Masukkan username baru: ";
    cin >> nama;

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

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

    pengguna[nama] = password;
}

```

```

    cout << "Registrasi berhasil!\n";
    Enter();
}

```

## N. Fungsi Login

Fungsi `loginUser()` digunakan untuk melakukan proses autentikasi pengguna, baik sebagai admin maupun user biasa. Fungsi ini menerima input username dan password, lalu mencocokkannya dengan data yang ada. Jika username dan password sesuai dengan akun admin bawaan (dalam hal ini "EgaClearestaHananta" dan "2409106088"), maka pengguna diidentifikasi sebagai admin. Jika tidak, fungsi akan mengecek keberadaan username dan kecocokan password pada map pengguna untuk login sebagai user biasa. Pengguna hanya diberikan maksimal tiga kali percobaan login. Jika ketiga percobaan gagal, program akan menampilkan pesan bahwa login gagal dan menghentikan proses lebih lanjut. Fungsi ini mengembalikan `true` jika login berhasil, dan `false` jika gagal.

```

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;
    }
}

```

## O. 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 login = false, isAdmin = false;

    while (true) {
        do {
            system("cls");
            cout << "|=====|\n";
            cout << "| Sistem Pendataan Penyewaan Kendaraan |\n";
            cout << "|====|=====|\n";
            cout << "| No | Menu |\n";
            cout << "|====|=====|\n";
            cout << "| 1 | Register |\n";
            cout << "| 2 | Login |\n";
            cout << "| 3 | Keluar |\n";
            cout << "|====|=====|\n";
            cout << "Pilih menu: ";
            int pilihan = validasiInputan(1, 3);

            switch (pilihan) {
                case 1:
                    registerUser(pengguna);
                    break;
                case 2:

```

```

        login = loginUser(pengguna, nama, password, isAdmin);
        break;
    case 3:
        keluar();
        return 0;
    }
} while (!login);

do {
    system("cls");
    Menu(isAdmin);
    cout << "Pilih menu: ";
    int pilihan = validasiInputan(1, isAdmin ? 6 : 4);

    if (isAdmin) {
        switch (pilihan) {
            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:
                lihatRiwayat(daftarKendaraan, jumlahKendaraan);
break;

            case 6:
                login = false;
                cout << "Anda telah logout. Kembali ke menu
awal...\n";

                Enter();
                break;
        }
    } else {
        switch (pilihan) {
            case 1:
                lihatKendaraan(daftarKendaraan, jumlahKendaraan);
break;

            case 2:
                sewa(daftarKendaraan, jumlahKendaraan, nama); break;
            case 3:
                lihatRiwayatUser(daftarKendaraan, jumlahKendaraan,
nama); break;

            case 4:
                login = false;
                cout << "Anda telah logout. Kembali ke menu
awal...\n";

                Enter();
                break;
        }
    }
} while (true);

```

```
        }  
    }  
    } while (login);  
}  
  
return 0;  
}
```

#### 4. Uji Coba dan Hasil Output

```
|=====|
| Sistem Pendataan Penyewaan Kendaraan |
|=====|
| No | Menu |
|=====|
| 1 | Register |
| 2 | Login |
| 3 | Keluar |
|=====|
Pilih menu: 2
|=====|
| 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 | Lihat Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|
Pilih menu: 1
Masukkan Merk Kendaraan: Avanza
Masukkan Tipe Kendaraan: Honda
Masukkan Nomor Polisi: KT 4070 BJ
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 | Lihat Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|
Pilih menu: 2

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

Gambar 4 3 Fitur Melihat Data Kendaraan

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

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 4070 BJ | Rp. 200000 | Tersedia |
|-----|
Tekan Enter untuk melanjutkan...
Masukkan nomor kendaraan yang ingin diperbarui (atau 0 untuk kembali): 1

=====
| Pilih Data yang Ingin Diperbarui |
|=====|
| 1. Perbarui Merk |
| 2. Perbarui Tipe |
| 3. Perbarui Nomor Polisi |
| 4. Perbarui Harga Sewa |
| 5. Kembali |
|=====|
Pilih opsi: 1
Masukkan Merk Kendaraan Baru: Terios
Merk kendaraan berhasil diperbarui!
```

Gambar 4 4 Fitur Memperbarui Data Kendaraan

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

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 4070 | Rp. 400000 | Tersedia |
|-----|

Masukkan nomor kendaraan yang akan dihapus: 1
Apakah Anda yakin ingin menghapus kendaraan Avanza Honda (KT 4070)? (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 | Lihat Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
|=====|
Pilih menu: 5

Belum ada data kendaraan yang tersimpan!
Tekan Enter untuk melanjutkan...
```

Gambar 4 6 Melihat Riwayat Penyewa Tetapi belum ada yang menyewa



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

Anda telah logout sebagai Admin.
Tekan Enter untuk kembali ke menu awal...
=====
| Sistem Penyewaan Kendaraan |
=====
| 1. Register |
| 2. Login |
| 3. Keluar |
=====
Pilih menu: 
```

Gambar 4 7 Fitur Logout di Menu utama admin

```
=====
| Menu Register |
=====
| Masukkan Username: RT |
| Masukkan Password: 1 |

Registrasi berhasil! Silakan login.
Tekan Enter untuk melanjutkan...
=====
| Sistem Penyewaan Kendaraan |
=====
| 1. Register |
| 2. Login |
| 3. Keluar |
=====
Pilih menu: 2

=====
| Menu Login |
=====
| Masukkan Username: RT |
| Masukkan Password: 1 |
=====

Login berhasil sebagai Pengguna! Tekan Enter untuk melanjutkan...
```

Gambar 4 8 Melakukan Registrasi dan login sebagai pengguna

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

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 4080 | Rp. 200000 | Tersedia |
|-----|
Tekan Enter untuk melanjutkan...
```

Gambar 4 9 Melihat Data Kendaraan

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

===== Data Kendaraan =====
|-----|
| No | Merk | Tipe | Nomor Polisi | Harga Sewa | Status |
|-----|
| 1 | Avanza | Honda | KT 4060 HZ | 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 10 Menyewa Kendaraan

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

=====
|-----|
| Riwayat Penyewaan Anda |
|-----|
| No | Merk | Tipe | Durasi (hari) |
|-----|
| 1 | Avanza | Honda | 10 |
|-----|
Tekan Enter untuk melanjutkan...
```

Gambar 4 11 Melihat Riwayat Sendiri

```
=====
| Sistem Pendataan Penyewaan Kendaraan |
=====
| No | Menu |
=====
| 1 | Tambah Data Kendaraan |
| 2 | Lihat Data Kendaraan |
| 3 | Perbarui Data Kendaraan |
| 4 | Hapus Data Kendaraan |
| 5 | Lihat Riwayat Penyewaan |
| 6 | Logout |
=====
Pilih menu: 5
=====
| No | Merk | Tipe | Penyewa | Durasi (hari) |
=====
| 1 | Avanza | Honda | E | 10 |
=====
Tekan Enter untuk melanjutkan...
```

Gambar 4 12 Melihat Riwayat Penyewa ketika sudah ada penyewa

## 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> 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> git commit -m "Initial commit"
[main 49fa645] Initial commit
```

Gambar 5 2 git commit

### 5.3 git push

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

```
PS C:\Users\EGA\OneDrive\Dokumen\Praktikum-Apl> git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 1.63 MiB | 932.00 KiB/s, done.
Total 13 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/egaxxxx/praktikum_apl.git
   5a61728..49fa645  main -> main
branch 'main' set up to track 'origin/main'.
```

Gambar 5 3 git push