

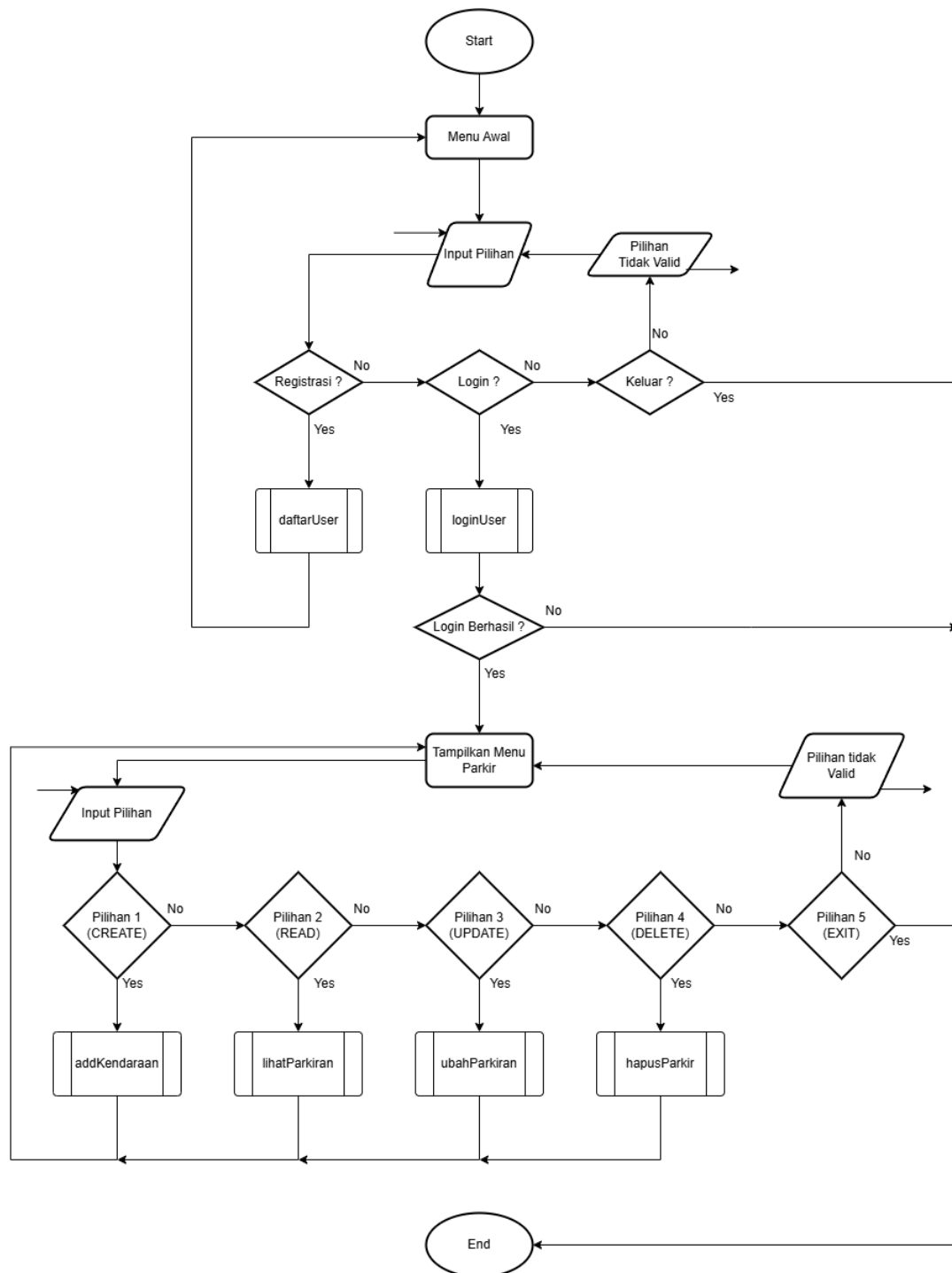
**LAPORAN PRAKTIKUM**  
**POSTTEST 5**  
**ALGORITMA PEMROGRAMAN LANJUT**



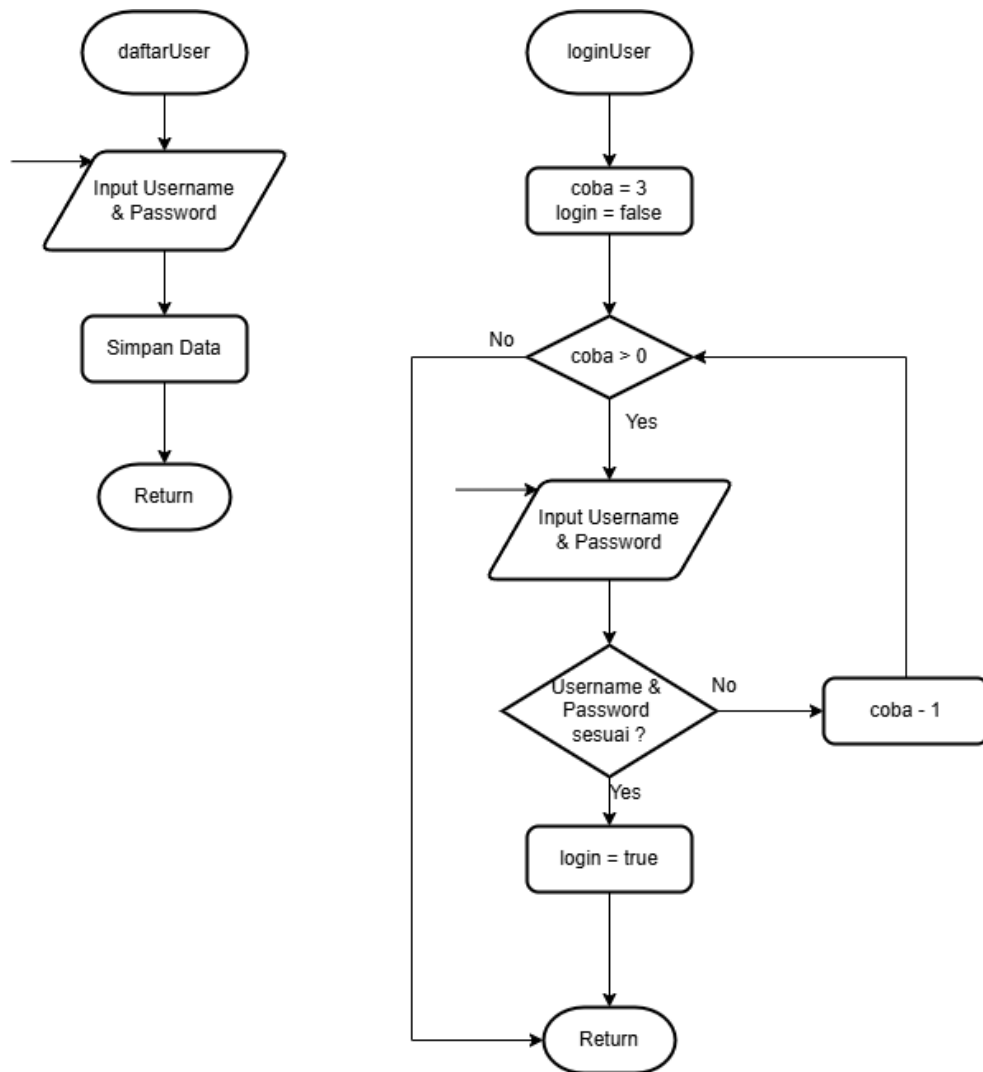
**Disusun oleh:**  
**Dwi Prasetyawan (2409106028)**  
**Kelas (A2 '24)**

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

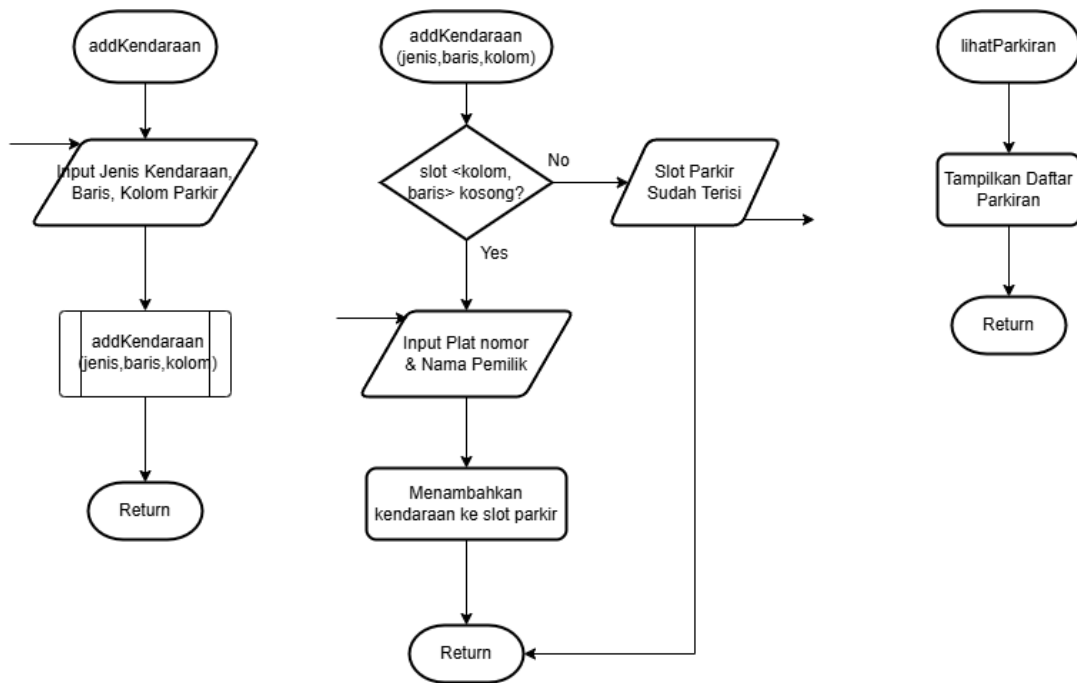
## 1. Flowchart



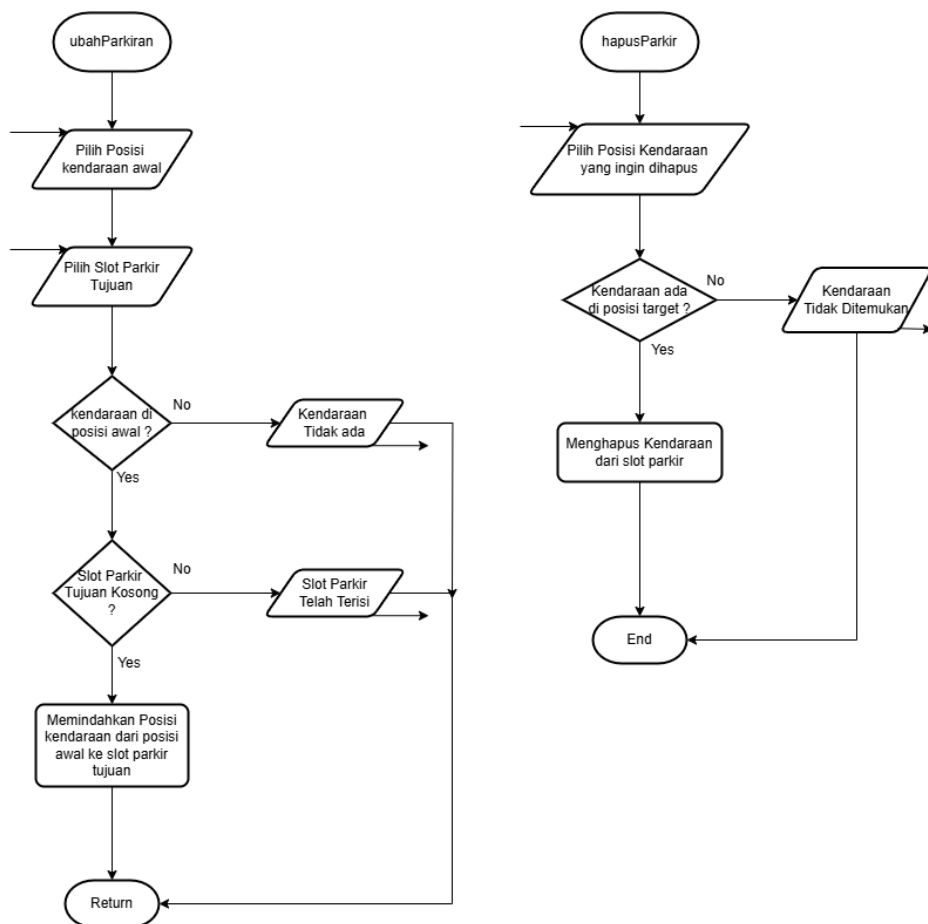
Gambar 1.1 Flowchart Utama



Gambar 1.2 Flowchart Fungsi Daftar dan Login



Gambar 1.3 Flowchart Fungsi Create, Read



Gambar 1.4 Flowchart Fungsi Update, Delete

## 2. Analisis Program

*Program Manajemen Parkir Mall* ini dibuat untuk membantu staf mall dalam mengelola parkir kendaraan mobil dan motor. Program ini memungkinkan staf untuk menambahkan, melihat, memindahkan, dan menghapus kendaraan dari slot parkir dengan sistem berbasis array dua dimensi.

Adapun pembaruan terhadap kode program ini, meliputi :

- Penggunaan pointer berupa address-of pointer dan dereference pointer untuk efisiensi pada pemakaian memori

### 3. Source Code

#### A. Struktur Data dan Deklarasi Fungsi

Fitur ini mencakup struktur data dan deklarasi fungsi yang diperlukan dalam program

#### Source Code:

```
#include <iostream>
using namespace std;

#define MAX_Y 5 // Jumlah baris parkir untuk mobil dan motor
#define MAX_X 5 // Jumlah kolom parkir untuk mobil dan motor
#define MAX_USERS 10 // Jumlah maksimal pengguna yang dapat dibuat

struct Kendaraan {
    string platNomor;
    string pemilik;
};

struct User {
    string username;
    string password;
};

User users[MAX_USERS] = { {"dwi", "028"} };
Kendaraan parkirMobil[MAX_Y][MAX_X];
Kendaraan parkirMotor[MAX_Y][MAX_X];
int jumlahUser = 1;
bool ulang = true;

void daftarUser ();
void loginUser ();
void addKendaraan(string jenis, int Y, int X);
void addKendaraan();
string lihatParkir();
void pindahKendaraan();
void hapusKendaraan(Kendaraan *kendaraan);
void menuParkir();
void mainMenu();
```

Gambar 3.1 Source Code Struktur Data dan Deklarasi Fungsi

## B. Inisiasi program

Fitur ini berfungsi untuk menjalankan program yang telah dibuat, disusun ulang dalam bentuk fungsi dan prosedur agar

### Source Code:

```
int main() {  
    mainMenu();  
    return 0;  
}
```

Gambar 3.2 Source Code Inisiasi Program

## C. Fitur Menu Awal

Fitur ini hadir agar user dapat melakukan registrasi, lalu Login untuk bisa mengakses menu utama

### Source Code:

```
void mainMenu() {  
    int pilihan;  
    while (ulang) {  
        cout << "===== MENU =====" << endl;  
        cout << "1. Register" << endl;  
        cout << "2. Login" << endl;  
        cout << "3. Keluar" << endl;  
        cout << "===== " << endl;  
        cout << "Pilihan: ";  
        cin >> pilihan;  
  
        switch (pilihan) {  
            case 1: daftarUser (); break;  
            case 2: loginUser (); break;  
            case 3: cout << "| Terima kasih telah menggunakan program ini!"  
| " << endl; return;  
            default: cout << "Pilihan tidak valid!\n"; break;  
        }  
    }  
}
```

Gambar 3.3 Source Code Menu Awal

#### D. Fitur Daftar User / Registrasi

Fitur ini digunakan agar user dapat melakukan Login dan mengakses menu utama

##### Source Code:

```
void daftarUser () {
    if (jumlahUser < MAX_USERS) {
        cout << "===== DAFTAR =====" << endl;
        cout << "Masukkan Nama: ";
        cin >> users[jumlahUser].username;
        cout << "Masukkan Password (3 Angka belakang NIM): ";
        cin >> users[jumlahUser].password;
        jumlahUser++;
        cout << "Registrasi berhasil! Silakan Login.\n";
    } else {
        cout << "Pendaftaran penuh!\n";
    }
}
```

Gambar 3.4 Source Code Daftar User / Registrasi

#### E. Fitur Login User

Fitur ini digunakan untuk melakukan verifikasi terhadap akun yang telah dibuat sebelumnya untuk bisa mengakses menu utama

##### Source Code:

```
void loginUser () {
    int coba = 3;
    string username, password;
    bool loggedIn = false;

    while (coba > 0 && !loggedIn) {
        cout << "===== LOGIN =====" << endl;
        cout << "Masukkan Nama: ";
        cin >> username;
        cout << "Masukkan Password: ";
        cin >> password;

        for (int i = 0; i < jumlahUser; i++) {
            if (users[i].username == username && users[i].password ==
password) {
                loggedIn = true;
                cout << "\nLogin Berhasil! Selamat datang, " << username <<
"! \n";
            }
        }
    }
}
```



```

        menuParkir();
        break;
    }
}

if (!LoggedIn) {
    coba--;
    cout << "\nLogin Gagal! Nama atau password salah." << endl;
    cout << "Percobaan tersisa: " << coba << endl << endl;
}
}
if (coba == 0) {
    cout << "Terlalu banyak percobaan, Keluar dari program . . .\n";
    ulang = false;
    return;
}
}
}

```

Gambar 3.5 Source Code Login

#### F. Menu Parkir / Menu Utama

Fitur ini adalah menu Utama dalam program ini, terdapat 5 fitur yang dapat digunakan oleh user, yaitu Tambah Kendaraan (Create), Tampilkan Slot Parkiran (Read), Ubah Kendaraan di Slot (Update), dan Hapus Kendaraan dari Slot (Delete). Serta fitur keluar untuk keluar dari menu Utama.

##### Source Code:

```

void menuParkir() {
    int pilihan;
    do {
        cout << "\n===== " << endl;
        cout << "| Menu Manajemen Parkir Mall" << endl;
        cout << "| 1. Tambah Kendaraan ke Slot" << endl;
        cout << "| 2. Tampilkan Slot Parkir" << endl;
        cout << "| 3. Ubah Kendaraan di Slot" << endl;
        cout << "| 4. Hapus Kendaraan dari Slot" << endl;
        cout << "| 5. Keluar" << endl;
        cout << "===== " << endl;
        cout << "Pilihan: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1: addKendaraan(); break;
            case 2: cout << lihatParkir(); break;
            case 3: pindahKendaraan(); break;
        }
    } while (pilihan != 5);
}

```

```

        case 4: hapusKendaraan(); break;
        case 5: cout << "| Terima kasih telah menggunakan program ini!
|" << endl; break;
        default: cout << "Pilihan tidak valid!\n"; break;
    }
} while (pilihan != 5);
}

```

Gambar 3.6 Source Code Menu Utama

## G. Fitur Tambah Kendaraan

Fitur ini digunakan untuk menambahkan kendaraan ke slot parkir dan menganggap slot yang telah ditempati sebagai terisi

### Source Code:

```

void addKendaraan() {
    string jenis;
    int Y, X;
    cout << "Masukkan jenis kendaraan (mobil/motor): ";
    cin >> jenis;
    cout << "Masukkan baris parkir (1-" << MAX_Y << "): ";
    cin >> Y;
    cout << "Masukkan kolom parkir (1-" << MAX_X << "): ";
    cin >> X;

    addKendaraan(jenis, Y, X);
}

```

Gambar 3.7 Source Code Tambah Kendaraan (bagian 1)

```

void addKendaraan(string jenis, int Y, int X) {
    if (Y > 0 && Y <= MAX_Y && X > 0 && X <= MAX_X) {
        if (jenis == "mobil" && parkirMobil[Y - 1][X - 1].platNomor == "")
        {
            cout << "Masukkan Plat Nomor: ";
            cin >> parkirMobil[Y - 1][X - 1].platNomor;
            cout << "Masukkan Nama Pemilik: ";
            cin >> parkirMobil[Y - 1][X - 1].pemilik;
        } else if (jenis == "motor" && parkirMotor[Y - 1][X - 1].platNomor
== "") {
            cout << "Masukkan Plat Nomor: ";
            cin >> parkirMotor[Y - 1][X - 1].platNomor;
            cout << "Masukkan Nama Pemilik: ";
            cin >> parkirMotor[Y - 1][X - 1].pemilik;
        } else {
            cout << "Slot sudah terisi atau jenis kendaraan tidak valid!\n";

```

```

    }
} else {
    cout << "Nomor slot tidak valid!\n";
}
}
}

```

Gambar 3.8 Source Code Tambah Kendaraan (bagian 2)

## H. Fitur Menampilkan Parkiran

Fitur ini digunakan untuk menambahkan kendaraan ke slot parkir dan menganggap slot yang ditempati sebagai terisi

### Source Code:

```

string LihatParkir() {
    string parkir;

    parkir += "\nTampilan Parkiran Mobil:\n";
    for (int i = 0; i < MAX_Y; i++) {
        for (int j = 0; j < MAX_X; j++) {
            if (parkiranMobil[i][j].platNomor == "") {
                parkir += "[Kosong] ";
            } else {
                parkir += "[" + parkirMobil[i][j].platNomor + " - " +
                parkirMobil[i][j].pemilik + "]" ";
            }
        }
        parkir += "\n";
    }

    parkir += "\nTampilan Parkiran Motor:\n";
    for (int i = 0; i < MAX_Y; i++) {
        for (int j = 0; j < MAX_X; j++) {
            if (parkiranMotor[i][j].platNomor == "") {
                parkir += "[Kosong] ";
            } else {
                parkir += "[" + parkirMotor[i][j].platNomor + " - " +
                parkirMotor[i][j].pemilik + "]" ";
            }
        }
        parkir += "\n";
    }

    return parkir;
}

```

Gambar 3.9 Source Code Lihat Parkir

## I. Mengubah Posisi Kendaraan

Fitur ini mengubah posisi kendaraan ke slot parkir kosong yang dituju. Menggunakan dereference pointer (\*) untuk mengakses nilai yang disimpan di alamat memori. Lalu, menggunakan address-of pointer (&) untuk mengambil secara langsung dengan lokasi memori nya.

### Source Code:

```
void pindahKendaraan() {
    int asal_Y, asal_X, tujuan_Y, tujuan_X;
    string jenis;
    cout << "Masukkan jenis kendaraan yang ingin dipindahkan (mobil/motor): ";
    cin >> jenis;
    cout << "Masukkan baris asal parkir: ";
    cin >> asal_Y;
    cout << "Masukkan kolom asal parkir: ";
    cin >> asal_X;
    cout << "Masukkan baris tujuan parkir: ";
    cin >> tujuan_Y;
    cout << "Masukkan kolom tujuan parkir: ";
    cin >> tujuan_X;

    if (asal_Y > 0 && asal_Y <= MAX_Y && asal_X > 0 && asal_X <= MAX_X &&
        tujuan_Y > 0 && tujuan_Y <= MAX_Y && tujuan_X > 0 && tujuan_X <=
MAX_X) {

        // Pointer untuk slot asal dan tujuan
        Kendaraan* asalKendaraan = &parkiranMobil[asal_Y - 1][asal_X - 1];
        Kendaraan* tujuanKendaraan = &parkiranMobil[tujuan_Y - 1][tujuan_X -
1];

        if (jenis == "mobil") {
            asalKendaraan = &parkiranMobil[asal_Y - 1][asal_X - 1];
            tujuanKendaraan = &parkiranMobil[tujuan_Y - 1][tujuan_X - 1];
        } else if (jenis == "motor") {
            asalKendaraan = &parkiranMotor[asal_Y - 1][asal_X - 1];
            tujuanKendaraan = &parkiranMotor[tujuan_Y - 1][tujuan_X - 1];
        } else {
```

```

        cout << "Jenis kendaraan tidak valid!\n";
        return;
    }
    // Periksa ketersediaan slot
    if (asalKendaraan->platNomor != "" && tujuanKendaraan->platNomor ==
"" )
    {
        // Pemindahan posisi parkir
        *tujuanKendaraan = *asalKendaraan; // Salin data dengan pointer
        asalKendaraan->platNomor = ""; // Kosongkan posisi awal
        asalKendaraan->pemilik = ""; // Kosongkan pemilik
        cout << "Kendaraan berhasil dipindahkan.\n";
    } else {
        cout << "Slot asal kosong atau slot tujuan sudah terisi!\n";
    }
} else {
    cout << "Nomor slot tidak valid!\n";
}
}
}

```

Gambar 3.10 Source Code Ubah Posisi Parkir

## J. Menghapus Kendaraan

Fitur ini menghapus kendaraan dari slot parkir dan menganggap slot parkir sebagai kosong. Menggunakan pointer untuk menghemat memori dan meningkatkan efisiensi program.

### Source Code:

```

void hapusKendaraan(Kendaraan *kendaraan) { // parameter dereference
    if (kendaraan->platNomor != "") {
        kendaraan->platNomor = "";
        kendaraan->pemilik = "";
        cout << "Kendaraan berhasil dihapus.\n";
    } else {
        cout << "Kendaraan tidak ditemukan!\n";
    }
}

void hapusKendaraan() {
    int Y, X;
    string jenis;
    cout << "Masukkan jenis kendaraan (mobil/motor): ";
    cin >> jenis;
    cout << "Masukkan baris parkir: ";
    cin >> Y;
    cout << "Masukkan kolom parkir: ";
}

```

```

cin >> X;

if (Y > 0 && Y <= MAX_Y && X > 0 && X <= MAX_X) {
    if (jenis == "mobil" && parkirMobil[Y - 1][X - 1].platNomor != "")
    {
        hapusKendaraan(&parkirMobil[Y - 1][X - 1]); // Menggunakan
address-of
    } else if (jenis == "motor" && parkirMotor[Y - 1][X -
1].platNomor != "") {
        hapusKendaraan(&parkirMotor[Y - 1][X - 1]); // Menggunakan
address-of
    } else {
        cout << "Slot kosong atau jenis kendaraan tidak valid!\n";
    }
} else {
    cout << "Nomor slot tidak valid!\n";
}
}

```

Gambar 3.11 Source Code Hapus Kendaraan

#### 4. Hasil Output

```
===== MENU =====
1. Register
2. Login
3. Keluar
=====
Pilihan: 1
===== DAFTAR =====
Masukkan Nama: Dwi
Masukkan Password (3 Angka belakang NIM): 028
Registrasi berhasil! Silakan login.
===== MENU =====
1. Register
2. Login
3. Keluar
=====
Pilihan: 2
===== LOGIN =====
Masukkan Nama: Dwi
Masukkan Password: 028

Login Berhasil! Selamat datang, Dwi!

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Keluar
=====
Pilihan: 1
Masukkan jenis kendaraan (mobil/motor): motor
Masukkan baris parkir (1-5): 2
Masukkan kolom parkir (1-5): 1
Masukkan Plat Nomor: KT7732JH
Masukkan Nama Pemilik: Andi
```

Gambar 4.1 Contoh Daftar, Login, dan Menambahkan Kendaraan

```

Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[KT7732JH - Andi] [Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Keluar
=====

Pilihan: 3
Masukkan jenis kendaraan yang ingin dipindahkan (mobil/motor): motor
Masukkan baris asal parkir: 2
Masukkan kolom asal parkir: 1
Masukkan baris tujuan parkir: 1
Masukkan kolom tujuan parkir: 4
Kendaraan berhasil dipindahkan.

```

Gambar 4.2 Menampilkan Parkiran dan Mengubah Posisi Kendaraan



```

Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [Kosong] [KT7732JH - Andi] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Keluar
=====

Pilihan: 4
Masukkan jenis kendaraan (mobil/motor): motor
Masukkan baris parkiran: 1
Masukkan kolom parkiran: 4

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Keluar
=====

Pilihan: 5
| Terima kasih telah menggunakan program ini! |

```

Gambar 4.3 Menampilkan Perubahan, Menghapus Kendaraan, dan Keluar Program

## 5. Langkah-Langkah Git pada VSCode

```
PS C:\Users\acer\Documents\GitHub\praktikum-apl> git add post-test/post-test-apl-5/2409106028-DwiPrasetyawan-PT-5.cpp
• PS C:\Users\acer\Documents\GitHub\praktikum-apl> git commit -m "Finish Post Test 5"
[main 6b1b589] Finish Post Test 5
 1 file changed, 280 insertions(+)
 create mode 100644 post-test/post-test-apl-5/2409106028-DwiPrasetyawan-PT-5.cpp
• PS C:\Users\acer\Documents\GitHub\praktikum-apl> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.31 KiB | 790.00 KiB/s, done.
Total 5 (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/KzDwi/praktikum-apl.git
 6d2c4df..6b1b589  main -> main
branch 'main' set up to track 'origin/main'.
○ PS C:\Users\acer\Documents\GitHub\praktikum-apl> □
```

Gambar 5.1 Pemakaian Git melalui Terminal VSCode

1. `git add post-test/post-test-apl-5/2409106028-DwiPrasetyawan-PT-5.cpp`  
Perintah ini digunakan untuk menambahkan file tertentu untuk dimasukkan dalam staging area agar siap di commit. *Post-test/post-test-apl-5/* dikenal sebagai direktori yang mengarahkan git ke file yang dituju
2. `git commit -m "Finish Post Test 5"`  
Perintah ini digunakan untuk menyimpan snapshot perubahan beserta pesan sebagai deskripsi perubahan
3. `git push -u origin main`  
Perintah ini digunakan untuk mengunggah kode dari repository lokal ke repository GitHub