

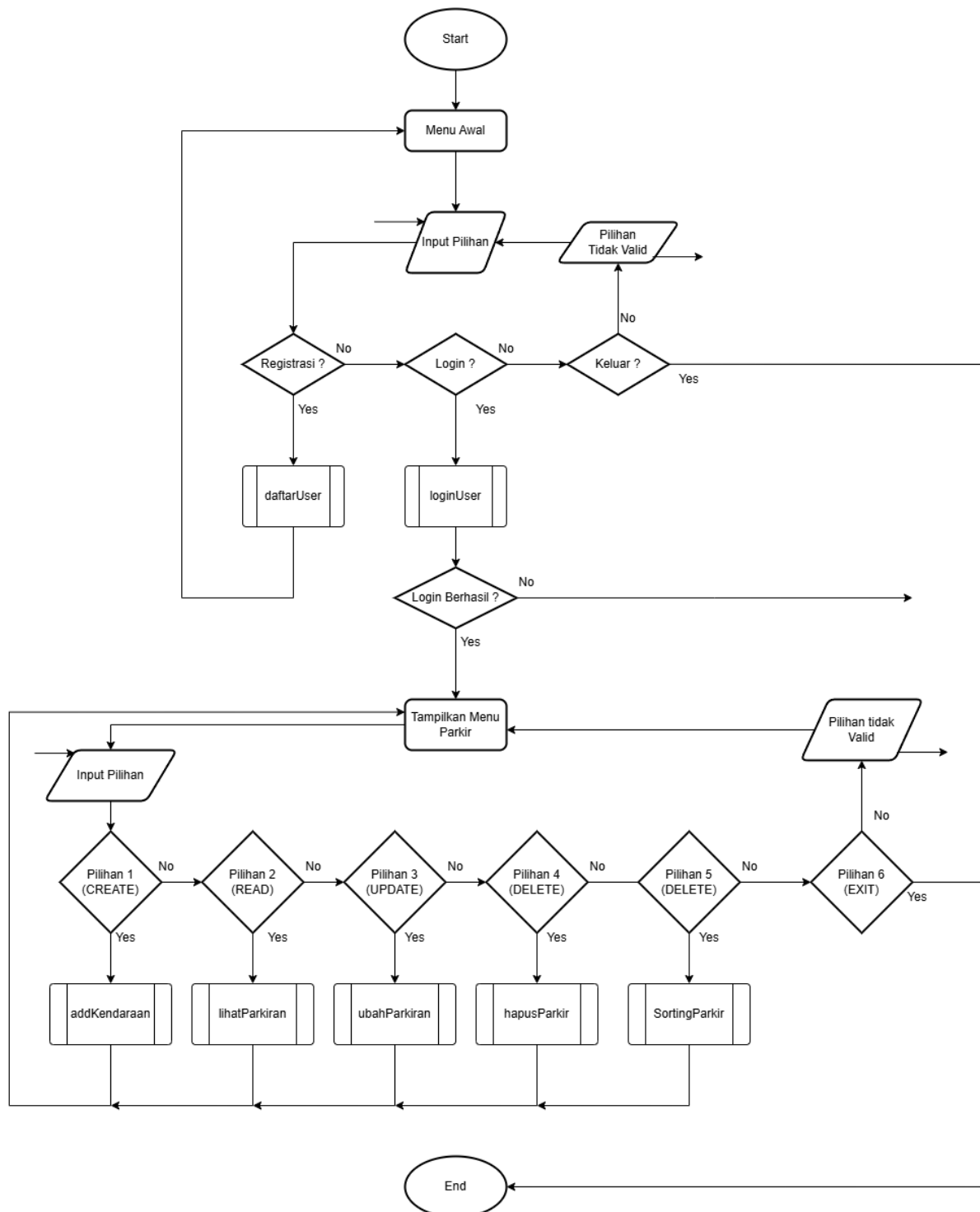
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



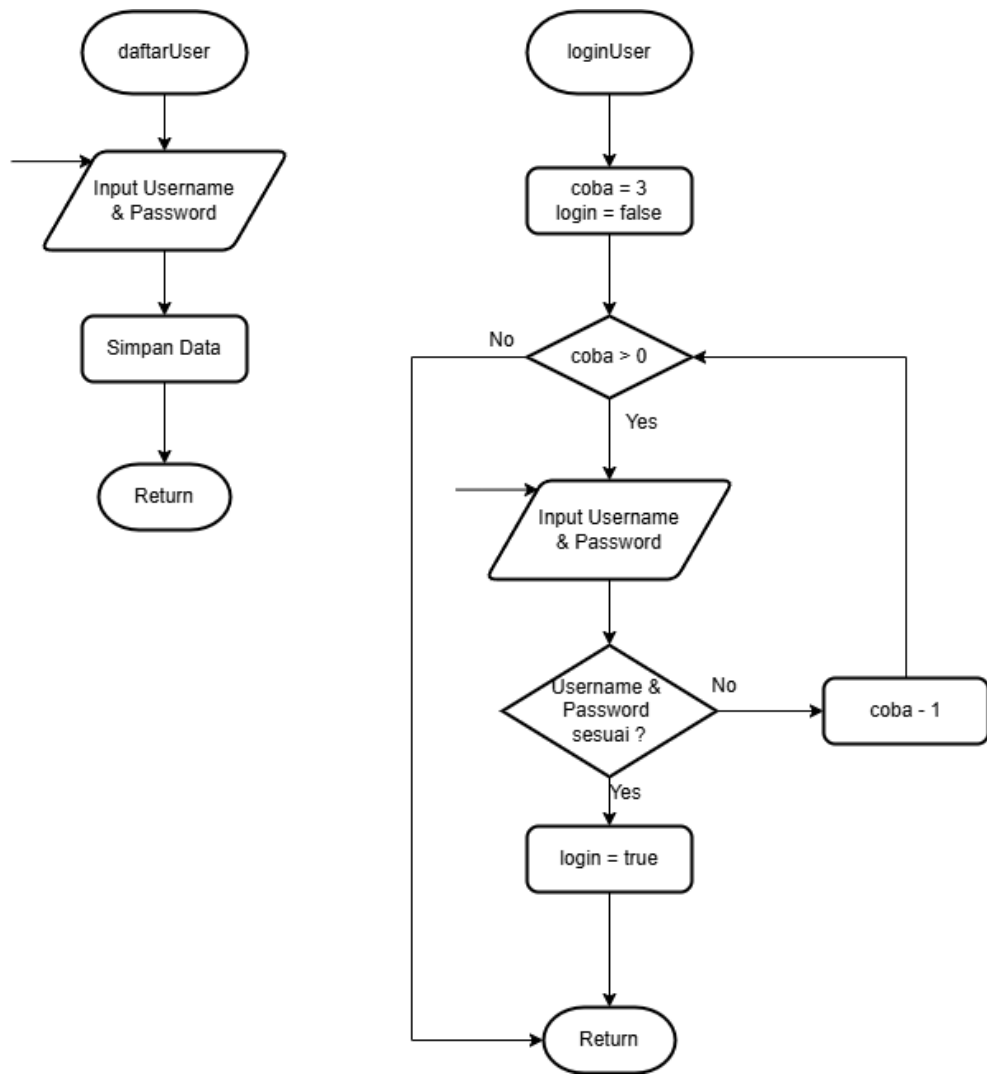
Disusun oleh:
Dwi Prasetyawan (2409106028)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

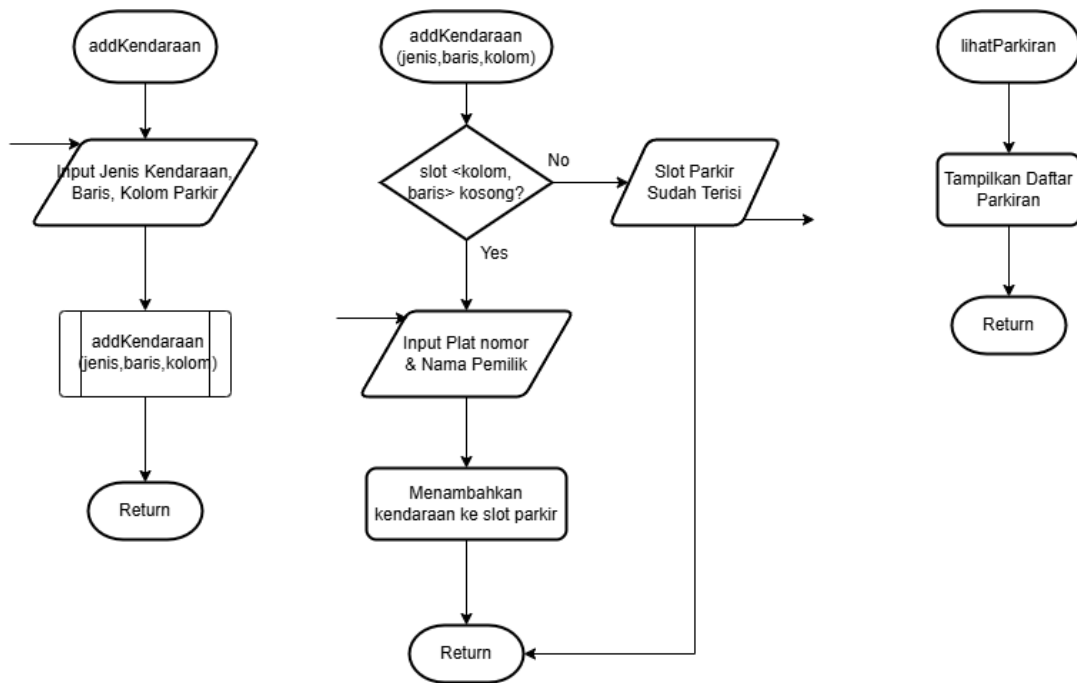
1. Flowchart



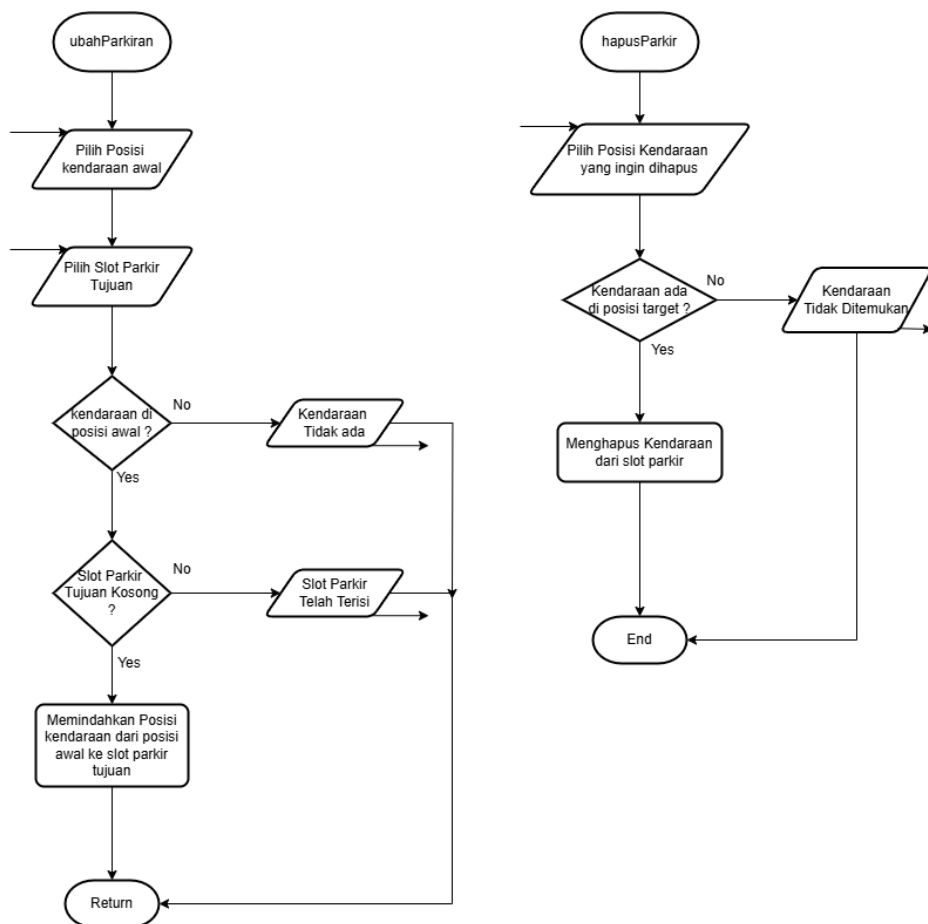
Gambar 1.1 Flowchart Utama



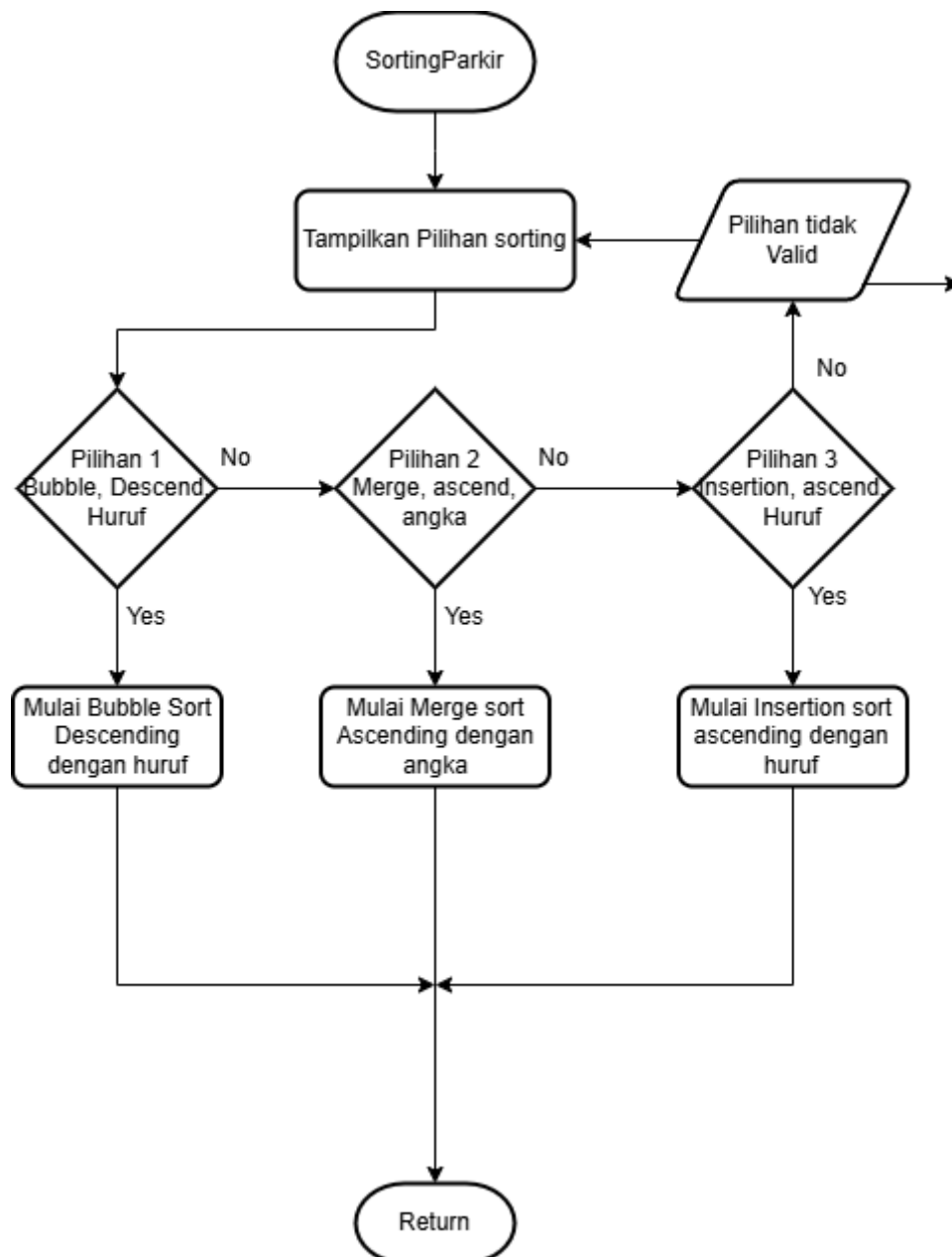
Gambar 1.2 Flowchart Fungsi Daftar dan Login



Gambar 1.3 Flowchart Fungsi Create, Read



Gambar 1.4 Flowchart Fungsi Update, Delete



Gambar 1.5 Flowchart Fungsi Sorting

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 :

- Tambahkan Fitur baru, yakni Sortir untuk menyortir parkir sesuai dengan angka pada plat nomor atau pemiliknya. Sistem sortir ini menggunakan 3 jenis sorting yang berbeda, yakni :
 1. **Bubble Sort** terhadap pemilik kendaraan menggunakan huruf, yang dilakukan secara Descending.
 2. **Merge Sort** terhadap plat nomor kendaraan menggunakan angka, yang dilakukan secara Ascending.
 3. **Insertion Sort** terhadap pemilik kendaraan menggunakan huruf, yang dilakukan secara Ascending.

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();
void lihatParkir();
void pindahKendaraan();
void hapusKendaraan();
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. Sorting Kendaraan" << endl;
        cout << "| 6. Keluar" << endl;
        cout << "===== " << endl;
        cout << "Pilihan: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                addKendaraan(); break;

```

```

        case 2:
            cout << lihatParkir(); break;
        case 3:
            pindahKendaraan(); break;
        case 4:
            hapusKendaraan(); break;
        case 5: {
            int sortChoice;
            cout << "Pilih metode sorting:\n";
            cout << "\n===== " << endl;
            cout << "1. Sorting Huruf Pemilik (Bubble, Descending)\n";
            cout << "2. Sorting Angka Plat Nomor (Merge, Ascending)\n";
            cout << "3. Sorting Huruf Pemilik (Insertion, Ascending)\n";
            cout << "\n===== " << endl;
            cout << "Pilihan: ";
            cin >> sortChoice;

            if (sortChoice == 1) {
                for (int i = 0; i < MAX_Y; i++) {
                    BubbleSort(parkiranMobil[i], MAX_X);
                }
                cout << "Sorting huruf selesai.\n";
            } else if (sortChoice == 2) {
                for (int i = 0; i < MAX_Y; i++) {
                    mergeSort(parkiranMobil[i], 0, MAX_X - 1);
                }
                cout << "Sorting angka selesai.\n";
            } else if (sortChoice == 3) {
                for (int i = 0; i < MAX_Y; i++) {
                    InsertionSort(parkiranMobil[i], MAX_X);
                }
                cout << "Sorting pemilik selesai.\n";
            } else {
                cout << "Pilihan tidak valid!\n";
            }
            break;
        }
        case 6:
            cout << "| Terima kasih telah menggunakan program ini! |" <<
endl; break;
        default:
            cout << "Pilihan tidak valid!\n"; break;
    }
} while (pilihan != 6);
}

```

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:

```
void LihatParkir() {
    cout << "\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 == "") {
                cout << "[Kosong] ";
            } else {
                cout << "[" << parkiranMobil[i][j].platNomor << " - " <<
parkiranMobil[i][j].pemilik << "] ";
            }
        }
        cout << endl;
    }
    cout << "\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 == "") {
                cout << "[Kosong] ";
            } else {
                cout << "[" << parkiranMotor[i][j].platNomor << " - " <<
parkiranMotor[i][j].pemilik << "] ";
            }
        }
        cout << endl;
    }
}
```

Gambar 3.9 Source Code Lihat Parkir

I. Mengubah Posisi Kendaraan

Fitur ini mengubah posisi kendaraan ke slot parkir kosong yang dituju

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) {
    if (jenis == "mobil") {
        if (parkiranMobil[asal_Y - 1][asal_X - 1].platNomor != "" &&
            parkiranMobil[tujuan_Y - 1][tujuan_X - 1].platNomor == "") {
            parkiranMobil[tujuan_Y - 1][tujuan_X - 1] =
parkiranMobil[asal_Y - 1][asal_X - 1]; //pemindahan posisi parkir
            parkiranMobil[asal_Y - 1][asal_X - 1] = {}; //kosongkan
posisi awal
            cout << "Kendaraan berhasil dipindahkan.\n";
        } else {
            cout << "Slot asal kosong atau slot tujuan sudah terisi!\n";
        }
    } else if (jenis == "motor") {
        if (parkiranMotor[asal_Y - 1][asal_X - 1].platNomor != "" &&
            parkiranMotor[tujuan_Y - 1][tujuan_X - 1].platNomor == "") {
            parkiranMotor[tujuan_Y - 1][tujuan_X - 1] =
parkiranMotor[asal_Y - 1][asal_X - 1]; //pemindahan posisi parkir
            parkiranMotor[asal_Y - 1][asal_X - 1] = {}; //kosongkan
posisi awal
            cout << "Kendaraan berhasil dipindahkan.\n";
        } else {
            cout << "Slot asal kosong atau slot tujuan sudah terisi!\n";
        }
    } else {
        cout << "Jenis kendaraan tidak valid!\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

Source Code:

```

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 != "")
    {
        parkirMobil[Y - 1][X - 1].platNomor = "";
        parkirMobil[Y - 1][X - 1].pemilik = "";
    } else if (jenis == "motor" && parkirMotor[Y - 1][X -
1].platNomor != "") {
        parkirMotor[Y - 1][X - 1].platNomor = "";
        parkirMotor[Y - 1][X - 1].pemilik = "";
    } else {
        cout << "Slot kosong atau jenis kendaraan tidak valid!\n";
    }
} else {
    cout << "Nomor slot tidak valid!\n";
}
}
}

```

Gambar 3.11 Source Code Hapus Kendaraan

K. Fitur Bubble Sort

Fitur ini melakukan pengurutan (sorting) melalui metode Bubble Sort, dengan menggunakan Huruf sebagai dasar pengurutan, dilakukan secara Descending (Dari yang terbesar hingga ke yang terkecil), menggunakan variabel pemilik kendaraan sebagai isi yang akan diurutkan.

Source Code:

```

// Bubble sort, Gunakan Huruf, Secara Descending, Variabel Pemilik Kendaraan
void BubbleSort(Kendaraan parkir[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (parkir[j].pemilik < parkir[j + 1].pemilik) { //
Mengurutkan berdasarkan pemilik
                Kendaraan temp = parkir[j];
                parkir[j] = parkir[j + 1];
                parkir[j + 1] = temp;
            }
        }
    }
}

```

Gambar 3.12 Source Code Bubble Sort (Descending)

L. Fitur Merge Sort

Fitur pengurutan yang menggunakan metode Merge Sort, dengan menggunakan angka sebagai dasar pengurutan, dilakukan secara Ascending (Dari kecil ke besar), menggunakan variabel Plat Nomor Kendaraan untuk diurutkan. Fungsi AmbilNomor disini berfungsi untuk memisahkan antara huruf dan angka, dan mengambil angka pada variabel plat nomor tersebut, jika tidak ada angka, maka hanya mengembalikan 0.

Source Code:

```
int AmbilNomor(const string& platNomor) { //Digunakan untuk Merge Sort
    karena harus memisahkan angka dari huruf
    string numberStr;
    for (char c : platNomor) {
        if (isdigit(c)) {
            numberStr += c;
        }
    }
    return numberStr.empty() ? 0 : stoi(numberStr); // Kembalikan 0 jika
    tidak ada angka, stoi untuk mengubah ke integer ('s'tring 'to' 'i'nteger)
}
// Merge Sort, Gunakan Angka, Secara Ascending, Variabel Plat Nomor
Kendaraan
void merge(Kendaraan arr[], int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1; // sub_array kiri
    int n2 = right - mid;    // sub_array kanan

    // Array static untuk sub-array
    Kendaraan L[MAX_X];
    Kendaraan R[MAX_X];

    // Menyalin data ke sub-array L dan R
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];
    i = 0;
    j = 0;
    k = left;
    // Gabung sub-array L dan R
    while (i < n1 && j < n2) {
        if (AmbilNomor(L[i].platNomor) <= AmbilNomor(R[j].platNomor)) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    // simpan sisa elemen dari L jika ada
    while (i < n1) {
        arr[k] = L[i];
```



```

        i++;
        k++;
    }
    // simpan sisa elemen dari R jika ada
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}
// Fungsi Rekursif Merge Sort
void mergeSort(Kendaraan arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

```

Gambar 3.13 Source Code Merge Sort (Ascending)

M. Fitur Insertion Sort

Fitur pengurutan yang menggunakan metode Insertion Sort, dengan menggunakan Huruf sebagai dasar pengurutan, dilakukan secara Ascending (Dari kecil hingga ke besar), menggunakan variabel pemilik kendaraan sebagai isi yang akan diurutkan.

Source Code:

```

// Insertion Sort, Gunakan Huruf, Secara Ascending, Variabel Pemilik
Kendaraan
void InsertionSort(Kendaraan parkirannya[], int size) {
    for (int i = 1; i < size; i++) {
        Kendaraan key = parkirannya[i];
        int j = i - 1;

        while (j >= 0 && parkirannya[j].pemilik > key.pemilik) { // Urutkan
            berdasarkan pemilik
            parkirannya[j + 1] = parkirannya[j];
            j = j - 1;
        }
        parkirannya[j + 1] = key;
    }
}

```

Gambar 3.14 Source Code Insertion Sort (Ascending)

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!
```

Gambar 4.1 Contoh Daftar dan Login

```
Pilihan: 1
Masukkan jenis kendaraan (mobil/motor): mobil
Masukkan baris parkir (1-5): 1
Masukkan kolom parkir (1-5): 1
Masukkan Plat Nomor: S5903KC
Masukkan Nama Pemilik: Nana

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====
Pilihan: 2

Tampilan Parkiran Mobil:
[S5903KC - Nana] [KT5504LM - Heri] [KT7735JG - Cindy] [Kosong] [B43770S - Ridwan]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
```

Gambar 4.2 Menambahkan Kendaraan dan Menampilkan Kendaraan

```

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

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====

Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [KT5504LM - Heri] [KT7735JG - Cindy] [Kosong] [B43770S - Ridwan]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

```

Gambar 4.3 Contoh Mengubah Posisi parkir

```

Pilihan: 4
Masukkan jenis kendaraan (mobil/motor): mobil
Masukkan baris parkir: 1
Masukkan kolom parkir: 1
Kendaraan berhasil dihapus.

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====

Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [KT5504LM - Heri] [KT7735JG - Cindy] [Kosong] [B43770S - Ridwan]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

```

Gambar 4.4 Contoh Hapus Kendaraan

```

Pilihan: 5
Pilih metode sorting:

=====
1. Sorting Huruf Pemilik (Bubble, Descending)
2. Sorting Angka Plat Nomor (Merge, Ascending)
3. Sorting Huruf Pemilik (Insertion, Ascending)

=====
Pilihan: 1
Sorting huruf selesai.

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====
Pilihan: 2

Tampilan Parkiran Mobil:
[B43770S - Ridwan] [KT5504LM - Heri] [KT7735JG - Cindy] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

```

Gambar 4.5 Contoh Bubble Sort Descending

```

Pilihan: 5
Pilih metode sorting:

=====
1. Sorting Huruf Pemilik (Bubble, Descending)
2. Sorting Angka Plat Nomor (Merge, Ascending)
3. Sorting Huruf Pemilik (Insertion, Ascending)

=====
Pilihan: 2
Sorting angka selesai.

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====
Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [B43770S - Ridwan] [KT5504LM - Heri] [KT7735JG - Cindy]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

```

Gambar 4.6 Contoh Merge Sort Ascending

```

Pilihan: 5
Pilih metode sorting:

=====
1. Sorting Huruf Pemilik (Bubble, Descending)
2. Sorting Angka Plat Nomor (Merge, Ascending)
3. Sorting Huruf Pemilik (Insertion, Ascending)

=====
Pilihan: 3
Sorting pemilik selesai.

=====
| Menu Manajemen Parkir Mall
| 1. Tambah Kendaraan ke Slot
| 2. Tampilkan Slot Parkir
| 3. Ubah Kendaraan di Slot
| 4. Hapus Kendaraan dari Slot
| 5. Sorting Kendaraan
| 6. Keluar
=====
Pilihan: 2

Tampilan Parkiran Mobil:
[Kosong] [Kosong] [KT7735JG - Cindy] [KT5504LM - Heri] [B43770S - Ridwan]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]
[Kosong] [Kosong] [Kosong] [Kosong] [Kosong]

```

Gambar 4.7 Contoh Insertion Sort Ascending

5. Langkah-Langkah Git pada VSCode

```
PS C:\Users\acer\Documents\GitHub\praktikum-apl> git add post-test/post-test-apl-6/2409106028-DwiPrasetyawan-PT-6.cpp
PS C:\Users\acer\Documents\GitHub\praktikum-apl> git commit -m "Finish Post Test 6"
[main 3ebb908] Finish Post Test 6
 1 file changed, 400 insertions(+)
 create mode 100644 post-test/post-test-apl-6/2409106028-DwiPrasetyawan-PT-6.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), 3.28 KiB | 1.64 MiB/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
 52edd31..3ebb908  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-6/2409106028-DwiPrasetyawan-PT-6.cpp`
Perintah ini digunakan untuk menambahkan file tertentu untuk dimasukkan dalam staging area agar siap di commit. *Post-test/post-test-apl-6/* dikenal sebagai direktori yang mengarahkan git ke file yang dituju
2. `git commit -m "Finish Post Test 6"`
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