**V. HASIL DAN PEMBAHASAN**

* 1. **Analisis Sistem**

Berikut merupakan hasil penelitian pada sistem yang diambil dari beberapa jurnal dan *web* yang telah dibaca oleh penulis dengan menggunakan *Framework Laravel* dalam pembuatan sistem kasir pada tempat pencucian mobil Berkat Carwas. Analisis ini digunakan sebagaian dasar bagi tahapan penelitian, sehingga akan diketahui hasil yang dibuat dapat mencapai tujuan yang diinginkan.

* 1. **Pembahasan**

Dalam pembahasan ini penulis akan membahas mengenai hasil analisis sistem yang telah di uji coba. Uji coba ini dilakukan untuk mengetahui apakah sistem dapat berjalan dengan lancar.

* + 1. **Perancangan *Use Case Diagram***

Use Case Diagram atau diagram use case merupakan pemodelan untuk memodelkan kelakuan (behavior) sistem informasi yang dibuat. Use case mendeskripsikan sebuah nteraksi antara satu atau lebih aktor dengan sistem informasi. Use case merupakan suatu bentuk diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna di luar sistem.

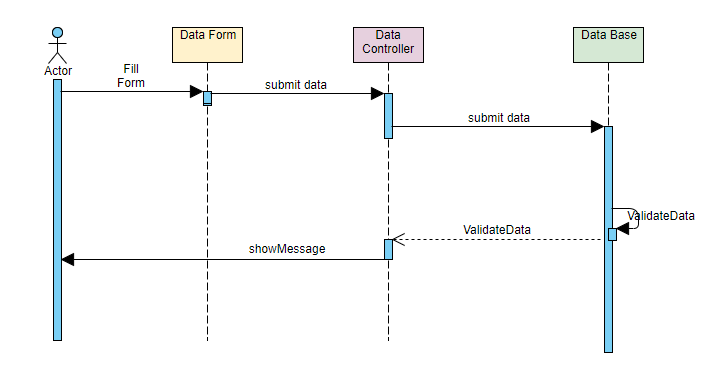
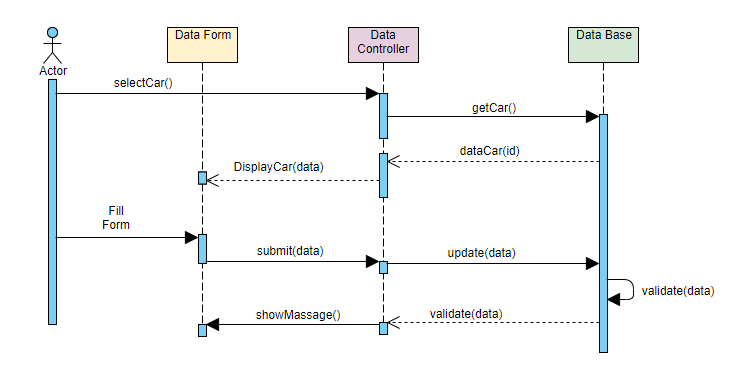
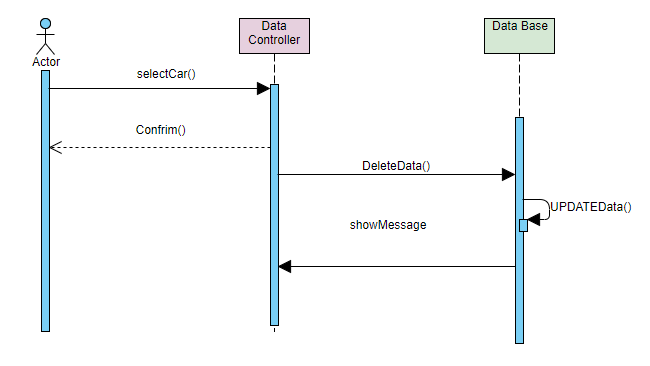
­­**5.2.2 Perancangan *Sequence* Diagram**

**Gambar 5.1 *Use Case Diagram***

Manager

Admin (kasir)

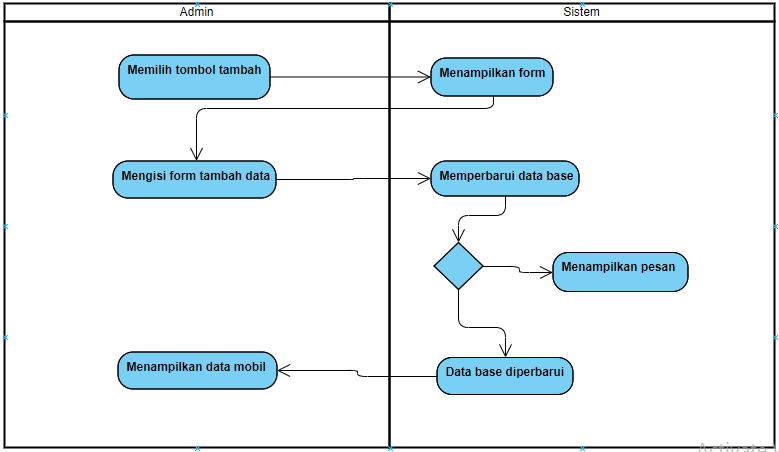
Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripisikan waktu hidup objek dan message yang dikirim dan diterima antar objek. Berdasarkan desain *use case*, terdapat beberapa *use case* yang prosesnya hampir sama satu sama lain. Untuk mempermudah pembahasan proses dalam pembuatan diagram sekuen, berikut ini ringkasan diagram sekuen pada sistem yang dikembangkan:

**5.2.3 Perancangan *Activity Diagram***

**Gambar 5.4 Hapus Data**

**Gambar 5.3 *Update* Data**

**Gambar 5.2 Input Data**

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (decision) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut. Berikut ini diagram aktivitas pada sistem informasi yang dikembangkan:

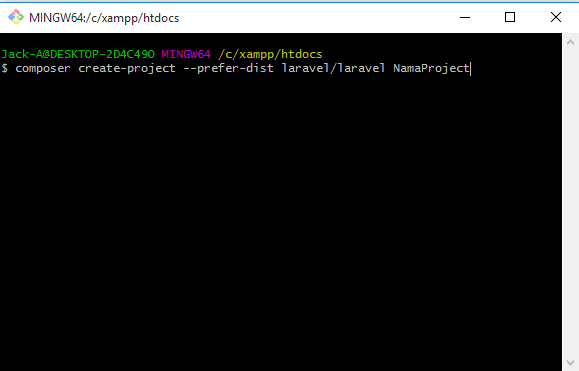
**Gambar 5.4 Diagram Aktivitas**

**5.2.4 Perancangan *Class Diagram***

*Class diagram* menggambarkan struktur sistem dari sisi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas dan mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam aplikasi pencucian mobil Berkat Carwash dapat digambarkan *class diagram* sebagai berikut:

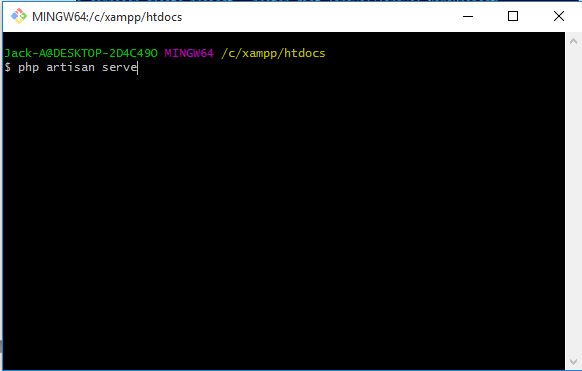
**5.2.5 Rancangan *Interface* menggunakan *Framework Laravel***

1. Langkah pertama yang dilakukan adalah unduh penginstal *Laravel* menggunakan *Composer* dengan cara mengetikkan perintah di *gitbash* “composer create-project –prefer-dist laravel/laravel NamaProject”.



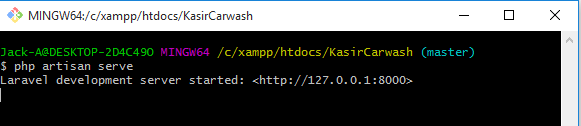
**Gambar 5.1 Penginstalan *Laravel***

1. Selanjutnya jalankan *webserver* *Laravel, Laravel*  mempunyai web*server* sendiri, ketikkan “php artisan serve” untuk menjalankannya.



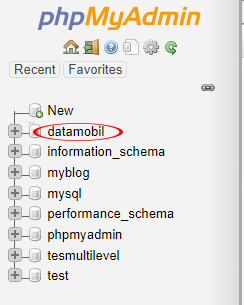
**Gambar 5.2 *webserver***

1. *Coppy*kan <http://127.0.0.1:8000> kedalam url google chrome

**

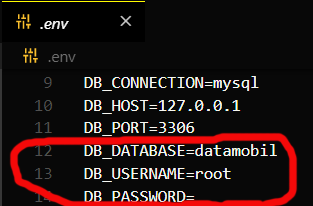
**Gambar 5.3 *webserver***

1. Selanjutnya XAMPP aktif buat database pada <http://localhost/phpmyadmin> dengan nama database “datamobil”.



**Gambar 5.4 *database***

1. Didalam file *.env* kita harus melakukan konfigurasi terlebih dahulu agar tersambung ke database yang kita buat tadi.



**Gambar 5.5 file *.env***

**5.2.1.2 Konstruksi**

1. *Route*

*route*untuk menangani permintaan kita di *url*, kemudian mengarahkan aplikasi ke halaman tertentu. Pada *laravel GET* digunakan untuk menampilkan resource yang bias berupa fungsi atau *controller. Post*  pada laravel digunakan untuk mengarahkan form ketika di *submit*

Route::get('/about', 'PagesController@about');

Route::get('/mobil', 'MobilController@index');

//Project Baru

Route::get('/cars', 'CarsController@index');

Route::get('/cars/create', 'CarsController@create');

Route::get('/cars/{car}', 'CarsController@show');

Route::post('/cars', 'CarsController@store');

Route::delete('/cars/{car}', 'CarsController@destroy');

Route::get('/cars/{car}/edit', 'CarsController@edit');

Route::patch('/cars/{car}', 'CarsController@update');

Auth::routes();

Route::get('/cars', 'CarsController@index')->name('index');

Route::group(['middleware' => 'web'], function(){

    Route::auth();

});

Route::group(['middleware' => ['web', 'auth']], function()

{

    Route::get('/cars', 'CarsController@index');

    Route::get('/', function()

    {

        if (Auth::user()->admin == 1)

        {

            return view ('index');

        }else{

            return view('cars.meneger');

        }

    });

});

Route::get('admin', ['middlewere' => ['web', 'auth', 'admin'], function()

{

    return view('admin/meneger');

}]);

1. *Model*

*Model* merupakan sekumpulan data yang memiliki fungsi-fungsi untuk mengelola suatu tabel pada sebuah database. Struktur pemodelan data pada laravel yakni memiliki fungsi yang terdiri dari tabel, *primary key* dan *fillable*. Di mana ke tiga fungsi tersebut harus di *protected*.

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\SoftDeletes;

class Car extends Model

{

    use SoftDeletes;

    // yang gak boleh di isi

    // protected $guarded  = ['nama', 'nohp', 'merk', 'plat'];

    // yang boleh di isi

    protected $fillable = ['nama', 'nohp', 'merk', 'plat', 'harga'];

}

1. *View*

*View* merupakan file yagn berisi kode yang berfungsi untuk menampilkan data ke dalam browser. Format *view* pada *laravel* menggunakan istilah *blade,* contohya index.blade.php

@extends('layout.main')

@section('title', 'Buku Mobil')

@section('container')

    <div class="container">

        <div class="row">

            <div class="col-6">

                <h1 class="mt-3">Data Mobil</h1>

<a href="/cars/create" class="btn btn-primary my-3">Tambah Data</a>

                @if (session('status'))

                    <div class="alert alert-success">

                        {{ session('status') }}

                    </div>

                @endif

                <ul class="list-group">

                    @foreach ($cars as $car)

                        <li class="list-group-item d-flex justify-content-between align-items-center">

                            {{$car->merk}} <br>

                            {{$car->plat}}

                            <a href="/cars/{{ $car->id }}" class="btn btn-info"> <h6> detail </h6></a>

                        </li>

                    @endforeach

                </ul>

            </div>

        </div>

    </div>

@endsection

                            <a href="/cars/{{ $car->id }}" class="btn btn-info"> <h6> detail </h6></a>

                        </li>

                    @endforeach

                </ul>

            </div>

        </div>

    </div>

@endsection

</div

</div>

1. *Controller*

*Controller* digunakan untuk megatur hubungan antara *model* dan *view*, *controller* menerima *request* dan data dari *user* kemudian menentukan apa yang akan di proses. Berikut adalah *controller* untuk menampilkan data mobil.

public function index()

    {

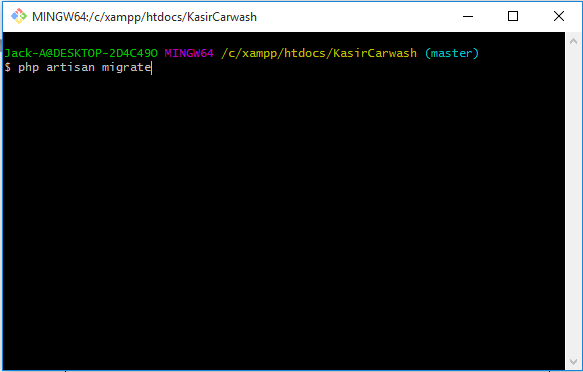
        $cars = Car::all();

        return view ('cars.index', compact('cars'));

    }

1. Database

* Migration

Laravel memberikan fitur *migrate* membuat table pada *database.* Migration berfungsi sebagai blueprint *database* atau dapat diistilahkan sebagai penyedia sistem *control* untuk skema *database.*

* Struktur table dengan *migrasi*

Laravel memberikan fungsi untuk membuat tipe kolom pada table seperti increments, string, varchar, int, char, timestamps.

class CreateCarsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('cars', function (Blueprint $table) {

            $table->increments('id');

            $table->string('merk');

            $table->varchar('plat', 20)->unique;

            $table->int('harga', 20)->unique;

            $table->string('nama');

            $table->char('nohp', 13)->unique;

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

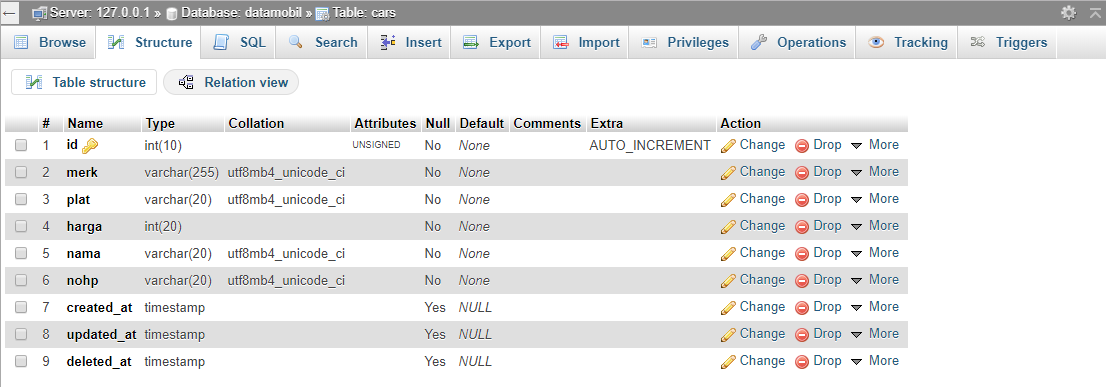
        Schema::dropIfExists('cars');

    }

}

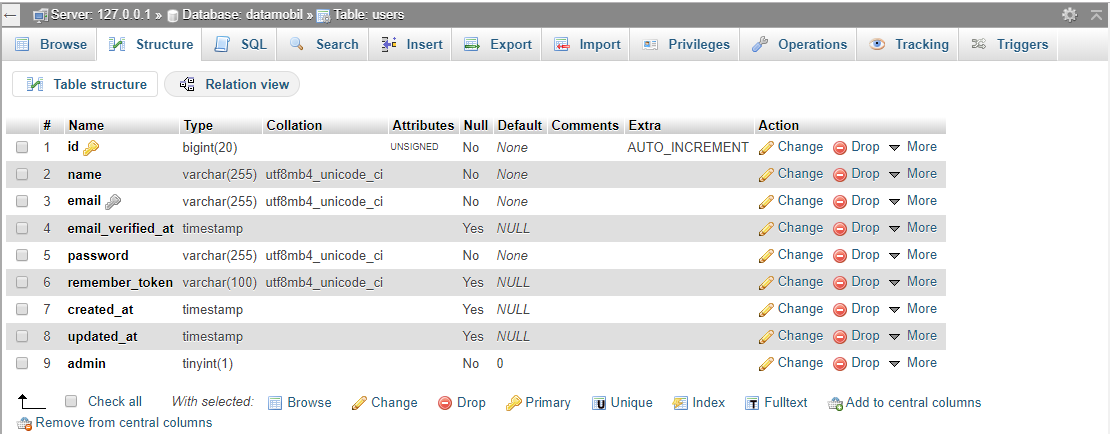
Keuntungan dengan menggunakan fitur migrate pada laravel 5 memberikan kemudahan jika terjadi hilang struktur table atau rusak dan tidak memiliki backup, website Carwash memilliki backup pada folder /*database*/*migration.* Berikut adalah struktur table pada *database* *website* Carwash yang dibuat menggunakan fitur *migrate:*

* + - 1. Tabel cars

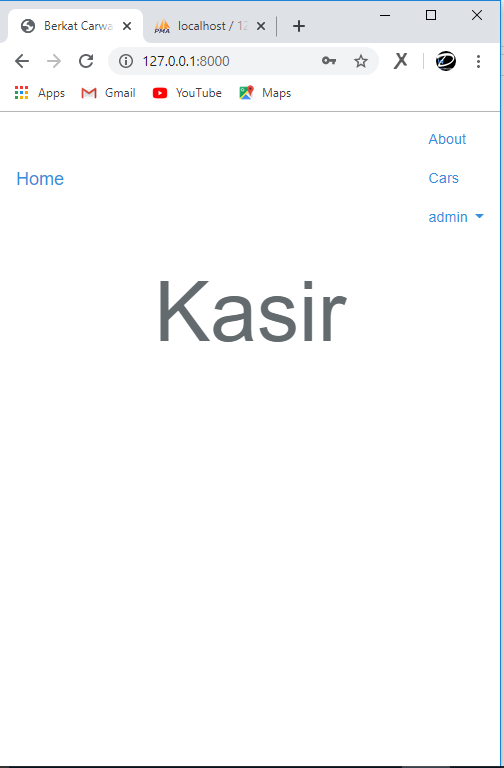
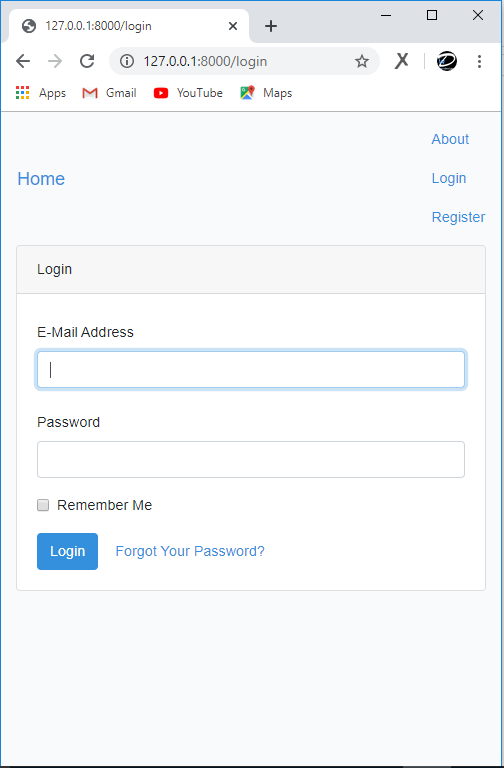
Tabel cars memilik field id, merk, plat, harga, nama, nohp, created\_at, updated\_at, deleted\_at.

1. Tabel users

Tabel users memiliki field, id, name, email, email\_verified\_at, password, remember\_token, created\_at, updated\_at, admin

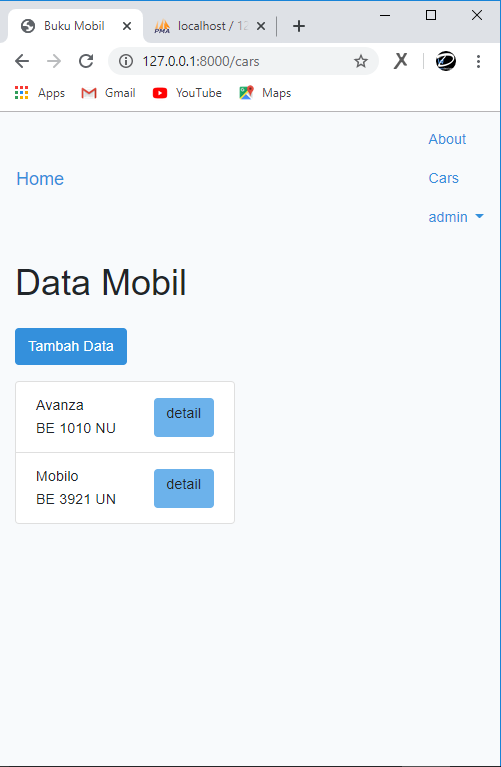
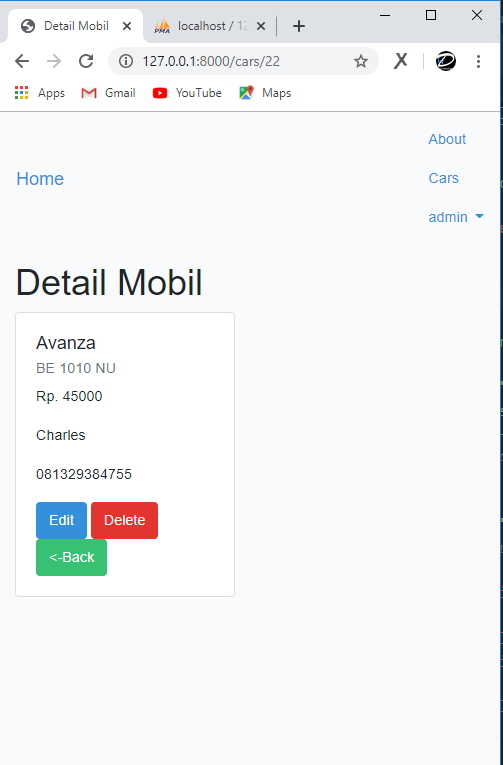


* + 1. **Rancangan *Interface* menggunakan *Framework Laravel***
       - 1. Ini adalah halaman-halaman untuk kasir yang berisi:

****

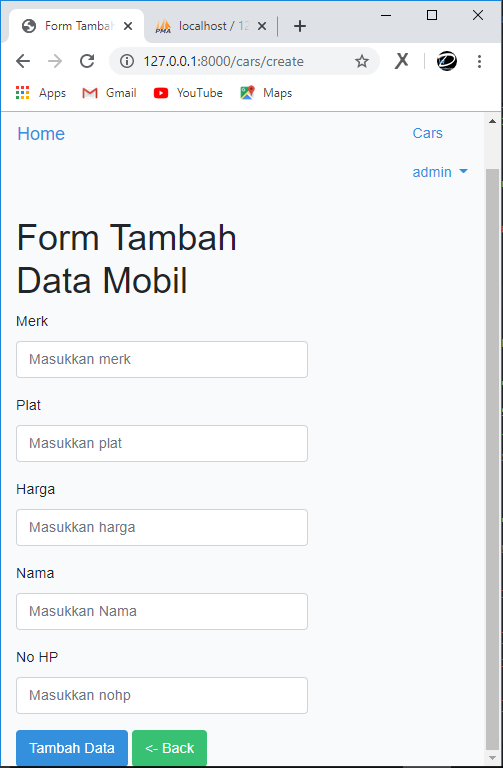
**Gambar 5.2 Halaman Kasir**

**Gambar 5.1 Login**

****

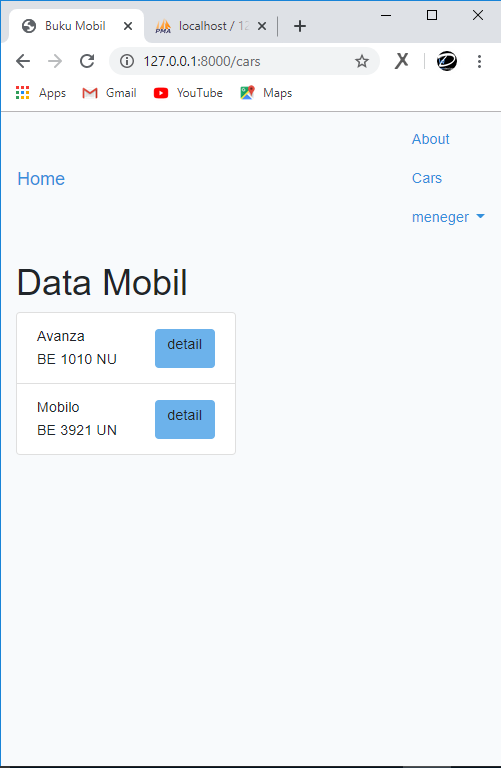
**Gambar 5.4 Halaman Detail Data Mobil**

**Gambar 5.3 Halaman View Data Mobil**

****

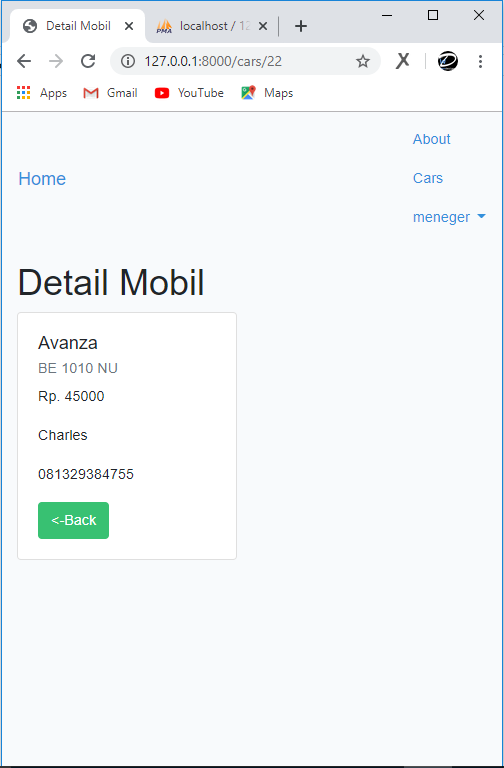
**Gambar 5.5 Halaman Input Data Mobil**

* 1. Ini adalah halaman-halaman untuk kasir yang berisi:

****

**Gambar 5.7 Halaman Data Mobil**

**Gambar 5.6 Halaman Meneger**

****

**Gambar 5.8 Halaman Detail Data Mobil**