

ARSITEKTUR SISTEM INFORMASI PENYEWAAN MOBIL BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL

ARSITEKTUR SISTEM: 1.0

Disusun Oleh :

5200411332 - Dwi Muhammad Faza

5200411409 - Pramudea Yohano Firmansyah

5200411416 - Muhammad Ilham Triwibowo

5200411417 - Rusli Pramono

PROGRAM STUDI METODOLOGI DESAIN PERANGKAT LUNAK

PRAKTIK

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

YOGYAKARTA

2021

1	PENDAHULUAN	3
1.1	Latar Belakang	3
1.2	Tujuan.....	3
1.3	Batasan Masalah.....	3
2	PEMODELAN BISNIS.....	4
3	PERMODELAN DATA DAN PROSES	5
3.1	Arsitektur Sistem.....	5
3.2	Use Case.....	6
3.3	Activity Diagram.....	7
3.4	Squence Diagram	8
3.5	Class Diagram	9
4	USER INTERFACE	10
5	PEMBENTUKAN APLIKASI.....	17
6	PENGUJIAN & TURNOVER	26

1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi semakin pesat pada masa kini. Kemajuan ini membuat Teknologi informasi menjadi kebutuhan yang tidak bisa dihiraukan lagi. (Gaol, 2008) Tuntutan bisnis yang menginginkan sesuatu secara lebih praktis dan akurat membuat suatu teknologi informasi menjadi hal yang tak terpisahkan dalam sebuah bisnis. Dengan adanya teknologi informasi, pengolahan informasi maupun data penting atau record menjadi lebih mudah dan dapat diandalkan.

1.2 Tujuan

Dalam usaha rental mobil akan banyak data yang harus dikelola secara sistematis agar dalam pengolahannya kita akan mendapatkan data yang valid. Maka dari itu bisnis rental mobil ini juga diperlukan sebuah sistem. Sistem ini berfungsi untuk memudahkan pegawai dalam pendataan pelanggan dan informasi-informasi penting terkait dengan usaha rental mobil ini.

1.3 Batasan Masalah

- Dalam pendataan data yang mungkin kita kelola memiliki jumlah yang sangat banyak sehingga dalam merekap akan memerlukan waktu dalam menghasilkan data yang valid.
- Saat pelanggan meminjam mobil dengan adanya sistem dapat mempermudah proses peminjamannya.

2 PEMODELAN BISNIS

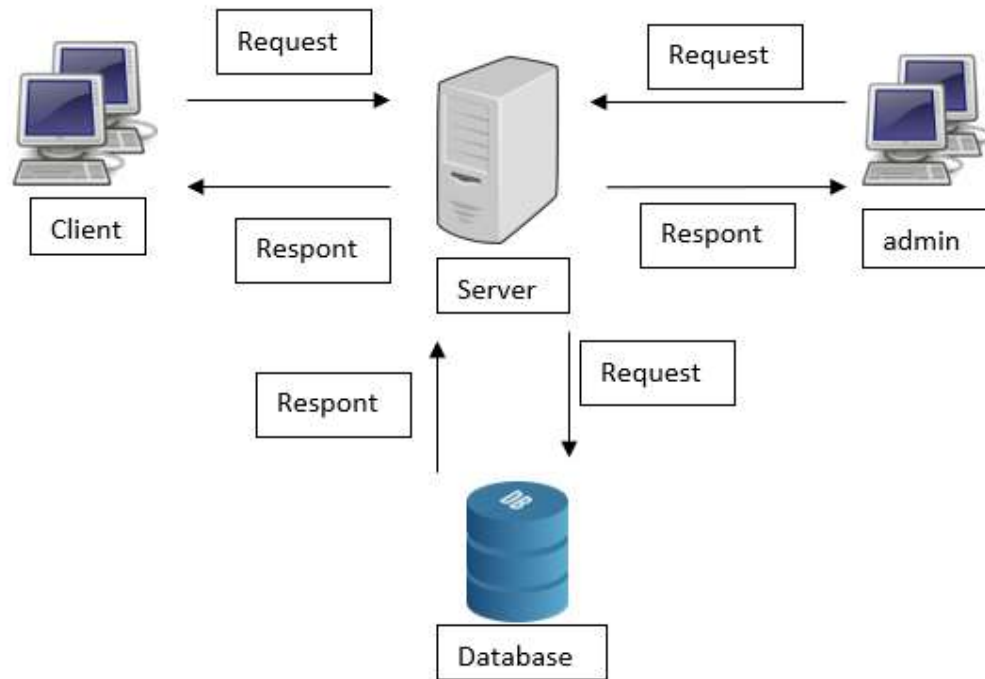
1. Jenis Sistem: Berbasis Website
2. Bahasa Pemrograman yang digunakan: HTML, PHP, CSS, JavaScript, SQL
3. Kebutuhan Hardware:
 - a. Komputer dengan Processor Intel Core I3
 - b. Ram 4GB
 - c. Storage 1TB
 - d. Printer HP 2775
4. Software terdiri dari:
 - a. Sistem Operasi Windows 10
 - b. VSCode v1.62
 - c. xampp
 - d. Microsoft Office dan Excel 365
 - e. Chrome 96.0.4664.45
5. Kebutuhan Input:
 - a. Data Kendaraan
 - b. Data Pelanggan
 - c. Data Transaksi/Peminjaman
6. Kebutuhan Informasi:
 - a. Laporan Data Kendaraan
 - b. Laporan Data Pelanggan
 - c. Laporan Data Transaksi/Peminjaman

3 PERMODELAN DATA DAN PROSES

Model data digunakan untuk menjelaskan bagaimana aplikasi bekerja. Analisis ini dilakukan agar kebutuhan aplikasi berbasis website ini dapat diketahui lebih baik sehingga proses perancangan dan pembangunan sistem dapat berjalan dengan baik.

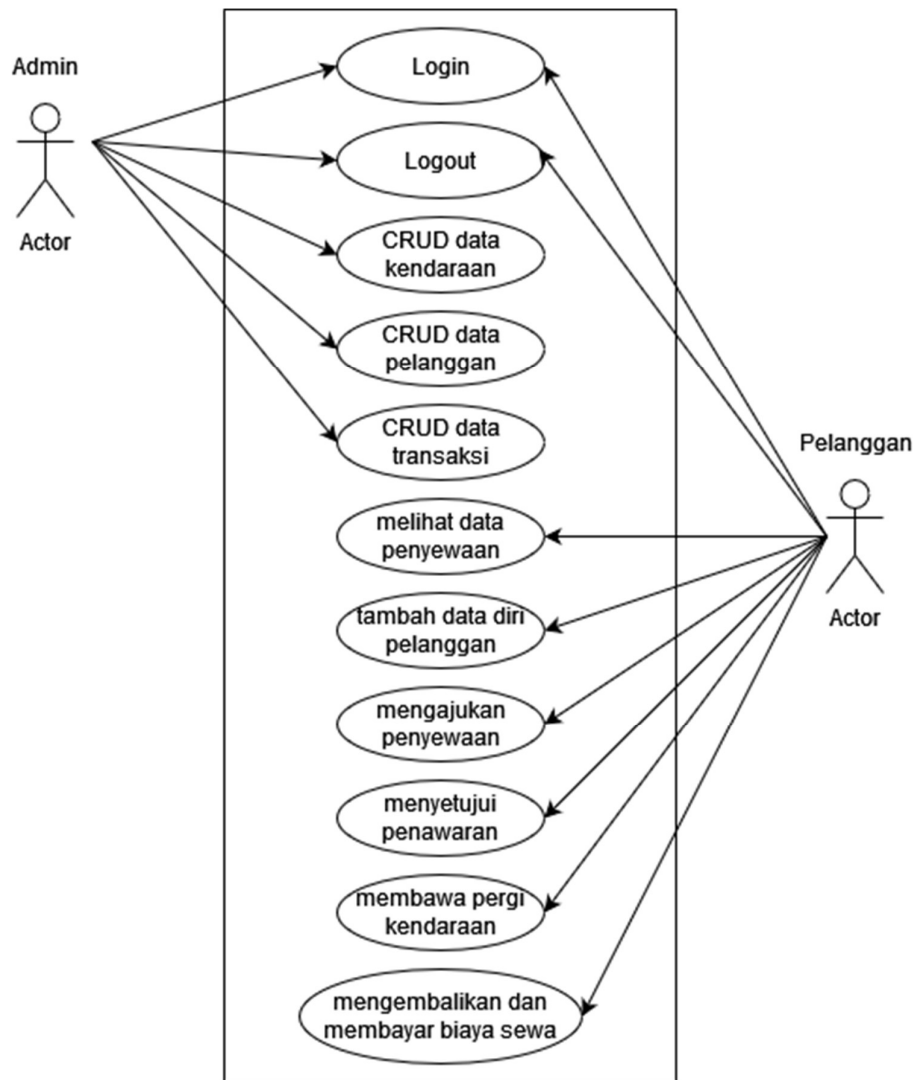
3.1 Arsitektur Sistem

Arsitektur Sistem Web Rental Mobil



Gambar 1 Arsitektur Sistem

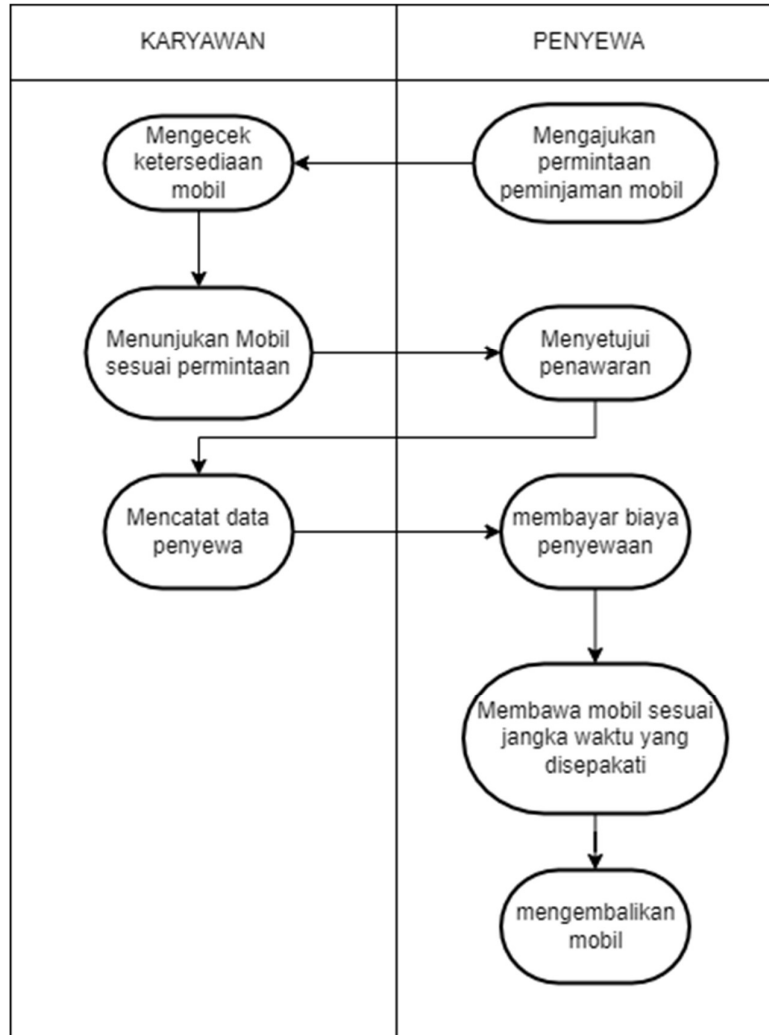
3.2 Use Case



Gambar 2 Use Case Diagram

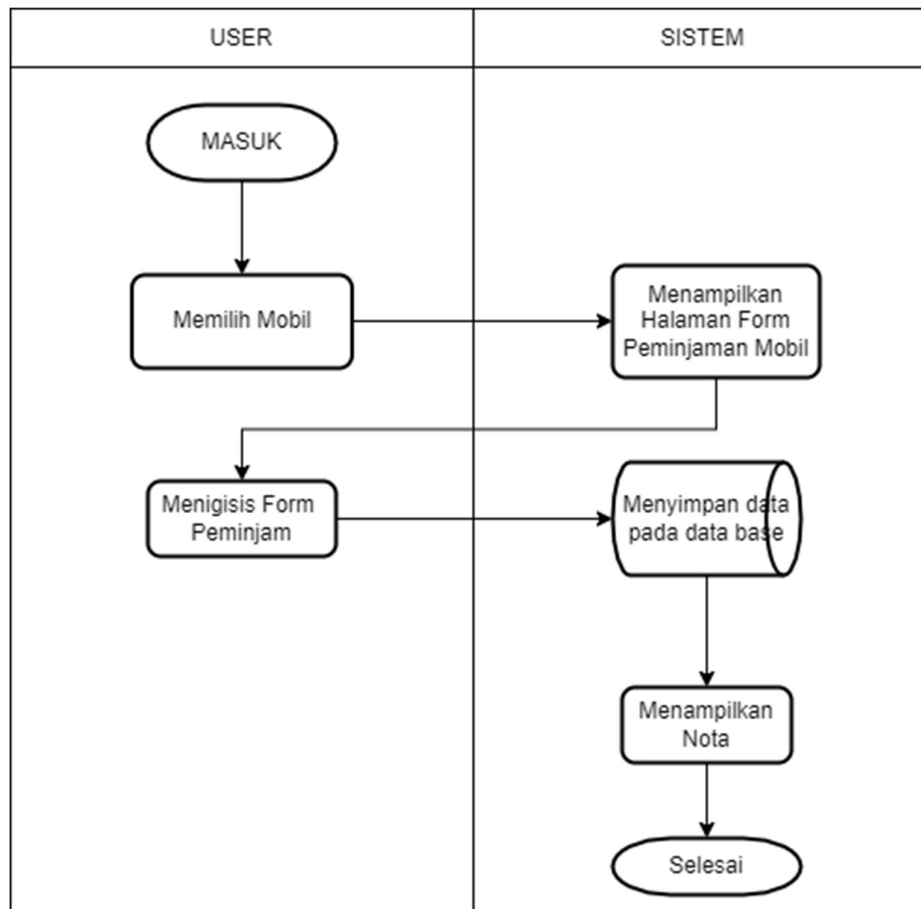
3.3 Activity Diagram

Activity diagram Sistem Bekerja



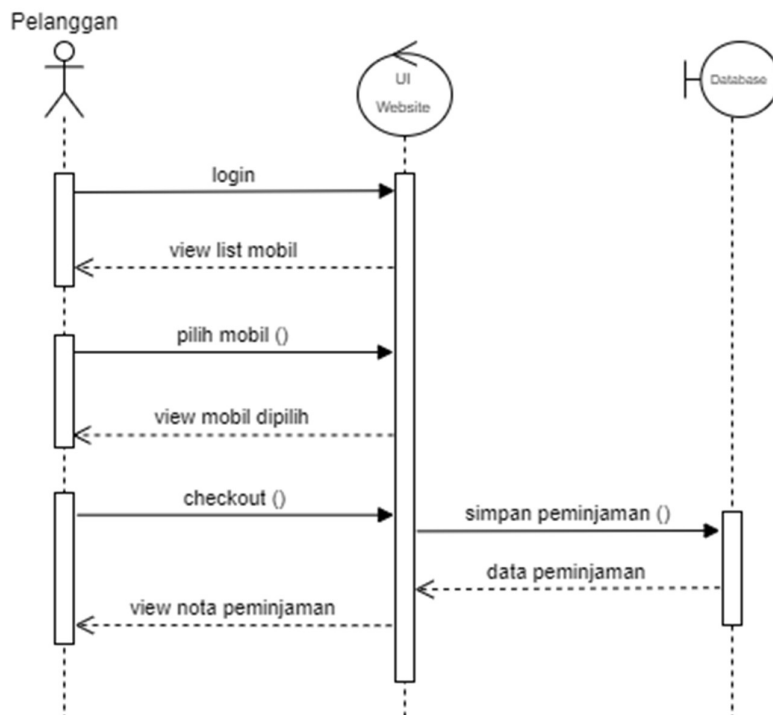
Gambar 3 Activity Diagram Sistem Berjalan

Activity Diagram Proses Peminjaman



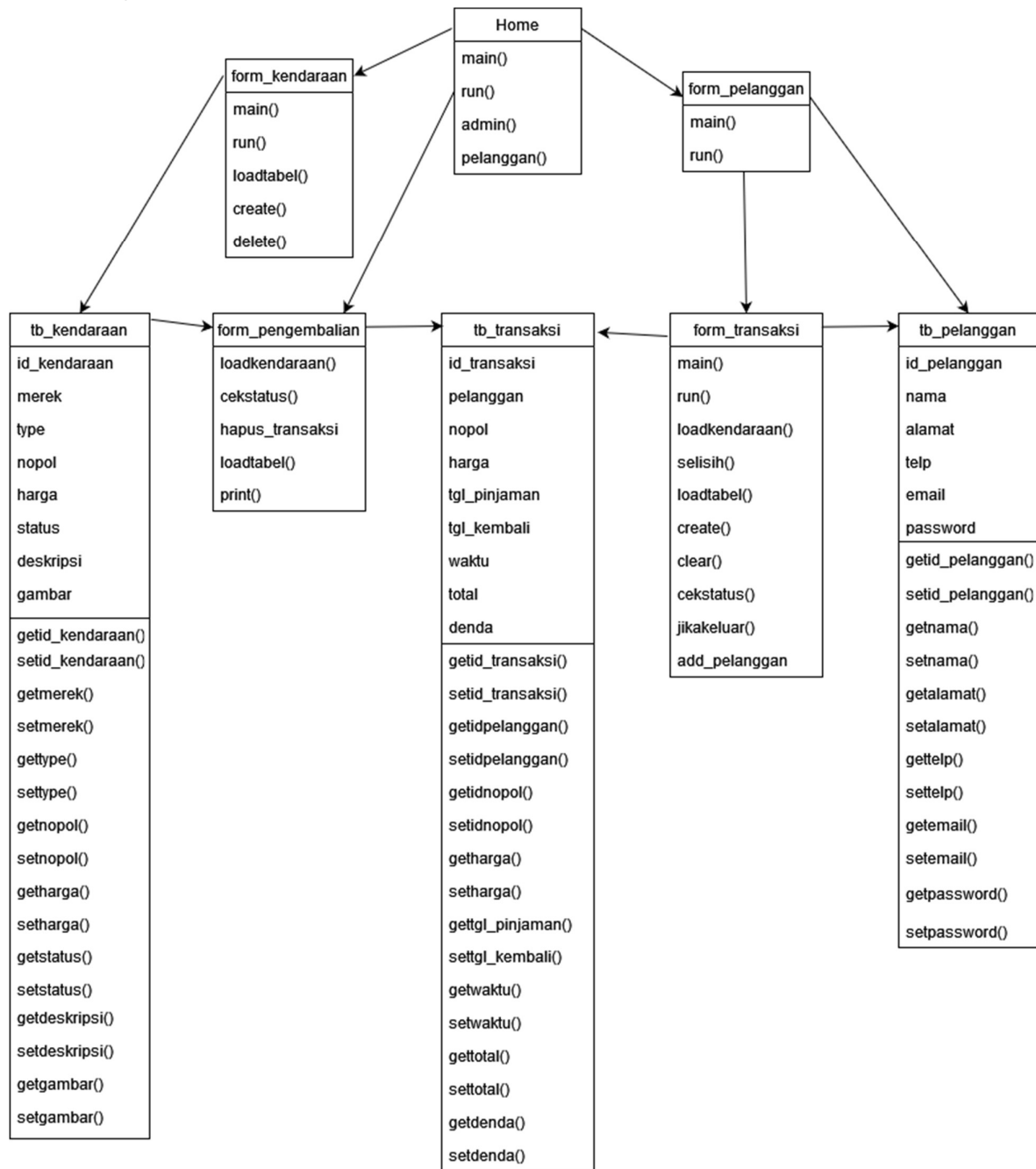
Gambar 4 Activity Diagram Proses Peyewaan

3.4 Sequence Diagram



Gambar 5 Squence Diagram

3.5 Class Diagram



Gambar 6 Class Diagram

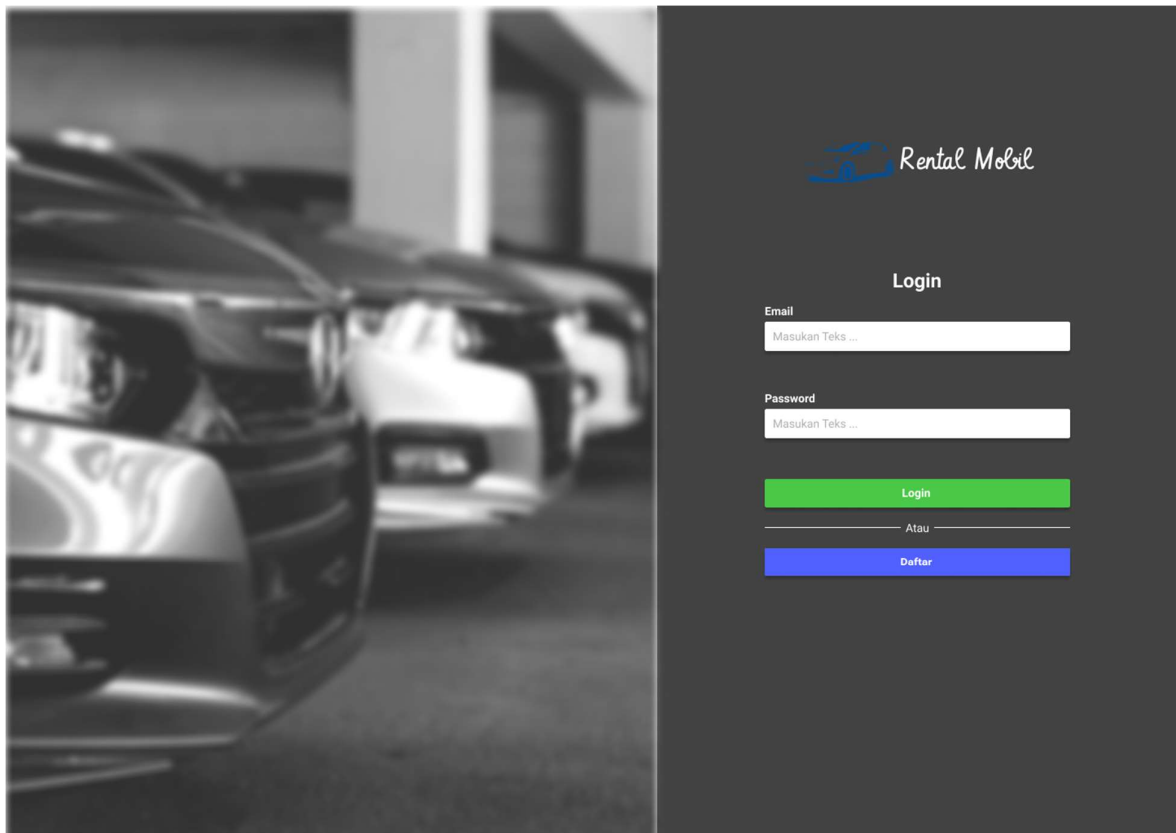
4 USER INTERFACE

Login(Admin)




The image shows a login form for an admin user. At the top center, there is a logo consisting of a blue car icon and the text "Rental Mobil" in a white, cursive font. Below the logo, the word "Login" is centered in a white, sans-serif font. Underneath "Login", there are two input fields. The first is labeled "Username" in a small white font, and the input field itself contains the placeholder text "Masukan Username". The second is labeled "Password" in a small white font, and the input field contains the placeholder text "Masukan Password". Below these two fields is a single, wide green button with the word "Login" in white, centered text.


Login User



The image shows a login form for a regular user. On the left side of the image, there is a black and white photograph of several cars parked in a row, with the focus on the front of the nearest car. On the right side, there is a dark gray background with the "Rental Mobil" logo at the top center. Below the logo, the word "Login" is centered in a white, sans-serif font. Underneath "Login", there are two input fields. The first is labeled "Email" in a small white font, and the input field contains the placeholder text "Masukan Teks ...". The second is labeled "Password" in a small white font, and the input field contains the placeholder text "Masukan Teks ...". Below these two fields is a green button with the word "Login" in white, centered text. Below the green button is a horizontal line, and below the line is a blue button with the word "Daftar" in white, centered text. The word "Atau" is centered above the blue button.

Daftar



 Rental Mobil

Daftar

Nama Lengkap

Masukan Teks ...

Alamat

Masukan Teks ...

No.Telp

Masukan Teks ...

Email

Masukan Teks ...

Password


Masukan Teks ...

Masukan Ulang Password

Masukan Teks ...

Daftar

List mobil (Admin)

 Rental Mobil

Home





List Mobil

List Pelanggan

List Peminjaman

List Mobil

Tambah +

#	Image	Deskripsi	Merek	Type	No.Polisi	Harga	Status	Id Kendaraan	Option
1		Text	Text	Text	Text	Rp. ###.###	Tersedia	Text	Edit Hapus
2		Text	Text	Text	Text	Rp. ###.###	Dipinjam	Text	Edit Hapus
3		Text	Text	Text	Text	Rp. ###.###	Dipinjam	Text	Edit Hapus
4		Text	Text	Text	Text	Rp. ###.###	Tersedia	Text	Edit Hapus

List mobil(User)

Daftar Mobil Tersedia



Type Mobil
Merek Mobil
Deskripsi

Mulai dari
Rp. ###.### /hari

Pilih



Type Mobil
Merek Mobil
Deskripsi

Mulai dari
Rp. ###.### /hari

Pilih



Type Mobil
Merek Mobil
Deskripsi

Mulai dari
Rp. ###.### /hari

Pilih



Type Mobil
Merek Mobil
Deskripsi

Mulai dari
Rp. ###.### /hari

Pilih

Tambah List Mobil

☒ List Mobil

☐ List Pelanggan

☐ List Peminjaman

Tambah Mobil

Gambar

Pilih Gambar

Merek

Masukan Teks ...

Type

Masukan Teks ...

Nomor Polisi

Masukan Teks ...

Harga Sewa


Masukan Teks ...

Deskripsi

Masukan Teks ...

Tambah +

Update list mobil



List Mobil

List Pelanggan

List Peminjaman

Home

Edit Mobil - Id_mobil

Gambar

Pilih Gambar

Merek

Masukan Teks ...

Type

Masukan Teks ...

Nomor Polisi

Masukan Teks ...

Harga Sewa

Masukan Teks ...

Deskripsi

Masukan Teks ...

Simpan

View Mobil



Home

About

Logout



Type Mobil

Merek Mobil

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean scelerisque sollicitudin ex, ut sollicitudin ex. Praesent a mattis neque, eget ultricies ligula. Phasellus varius elit ac interdum volutpat. Proin ac lacinia sapien. Vestibulum id mauris vel mi lacinia convallis eu pharetra nulla. Vestibulum pretium mollis dui a imperdiet.

Tanggal Pinjam

Masukan Teks ...

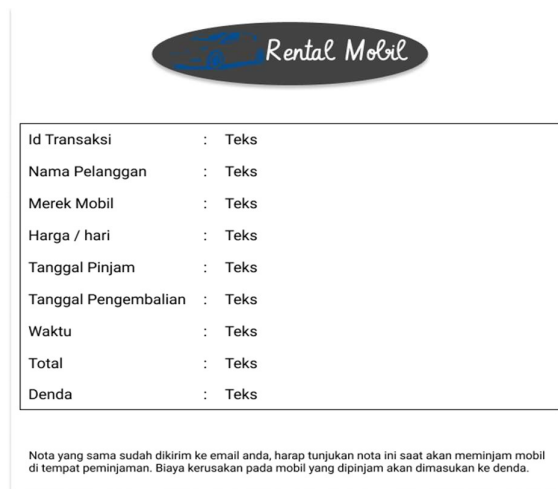
Tanggal Kembali

Masukan Teks ...

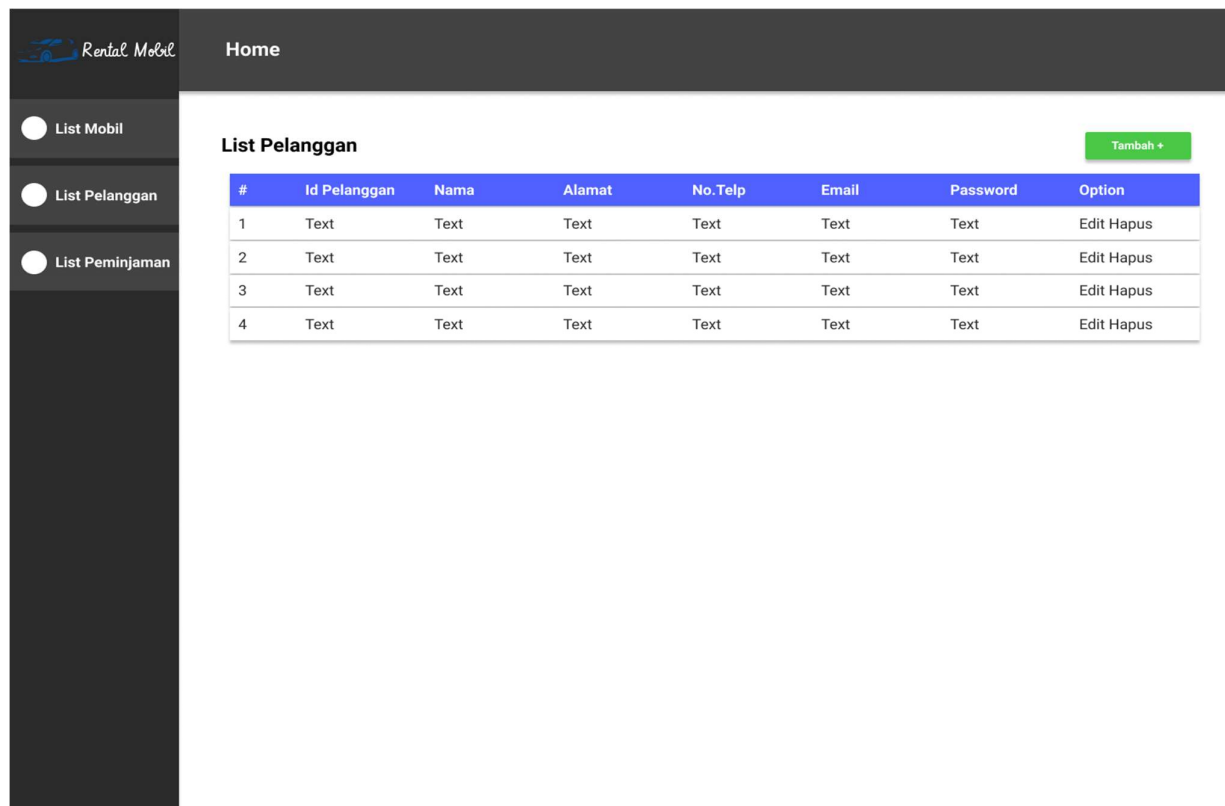
Total : Rp. ###.###

Pesan


Nota



List Pelanggan(Admin)



Update Pelanggan



List Mobil

List Pelanggan

List Peminjaman

Home

Edit data Pelanggan - Id Pelanggan

Nama

Masukan Teks ...

No. Telp

Masukan Teks ...

Email

Masukan Teks ...

Password


Masukan Teks ...

Alamat

Masukan Teks ...

Simpan

List Peminjaman (Admin)

Rental Mobil

List Mobil

List Pelanggan


List Peminjaman

Home

List Peminjaman

#	Id Peminjaman	Nama Pelanggan	No. Polisi	Harga	Tgl Pinjam	Tgl Kembali	Total	Option
1	Text	Text	Text	Text	Text	Text	Text	Edit Hapus
2	Text	Text	Text	Text	Text	Text	Text	Edit Hapus
3	Text	Text	Text	Text	Text	Text	Text	Edit Hapus
4	Text	Text	Text	Text	Text	Text	Text	Edit Hapus

Update peminjaman

 Rental Mobil

● List Mobil

● List Pelanggan

● List Peminjaman

Home

Edit Peminjaman - Id_peminjaman

Id Pelanggan - Nama Pelanggan

Masukan Teks ...

Id Mobil - Nomor Polisi

Masukan Teks ...

Tanggal Pinjam

Masukan Teks ...

Tanggal Kembali

Masukan Teks ...

Simpan

5 PEMBENTUKAN APLIKASI

- View List Pelanggan

```
public function index()
{
    $data['title'] = "List Pelanggan";
    $data['menu'] = 3;
    $users = DB::table('tb_pelanggan')->get()->toArray();
    $data['users'] = json_decode(json_encode($users), true);
    $data['no'] = 1;
    return view('users.index', $data);
}
```

Pada function index ini dibuat array dengan variable \$data yang kemudian di isi dengan title, menu, list user yang ada di database dan no. Lalu akan di return berupa view bersamaan dengan \$data yang kemudian pada view tersebut menampilkan list pelanggan yang ada.

- View Edit Pelanggan

```
public function edit($id)
{
    $data['pelanggan'] = User::find($id);
    $data['title'] = "Edit Pelanggan ".$id;
    $data['menu'] = 3;

    return view('users.edit', $data);
}
```

Pada function edit ini dibutuhkan memiliki parameter berupa \$id yang nantinya akan digunakan pada \$data['pelanggan'] untuk mencari data pelanggan berdasarkan id tersebut. kemudian pada \$data diberikan title dan menu. Setelah itu akan di return view bersamaan dengan \$data yang nantinya akan menampilkan pelanggan yang memiliki id yang sesuai dengan id yang ada pada parameter yang nantinya akan di edit datanya pada form edit.

- Update Pelanggan

```
public function update(Request $request, $id)
{
    $validate = $request->validate([
        'nama' => 'required',
        'telp' => 'nullable',
        'alamat' => 'nullable',
        'email' => 'required',
        'password_new' => 'required',
    ]);

    $data['nama'] = $request['nama'];
    $data['telp'] = $request['telp'];
    $data['alamat'] = $request['alamat'];
    $data['email'] = $request['email'];
    if($request['password_old'] != $request['password_new']){
        $data['password'] = bcrypt($request['password_new']);
    }else{
        $data['password'] = $request['password_old'];
    }

    $update = User::find($id)->update($data);
    return redirect()->route('pelanggan.index');
}
```

Pada function update ini dibutuhkan 2 parameter yaitu \$request yang berupa value yang sudah di isi dari form edit sebelumnya dan parameter \$id yang berupa id. Pada \$validate ini digunakan untuk memvalidasi agar data yang di isikan sesuai dengan ketentuan yang ada di array validasi ini dan jika

tidak sesuai maka akan dikembalikan ke view edit. Kemudian pada if statement password di cek apakah password terjadi perubahan atau tidak, jika ya maka password akan disimpan jika tidak maka akan tetap menggunakan password lama. Kemudian pada \$update dicari pelanggan berdasarkan parameter id yang kemudian datanya di update berdasarkan \$data atau data yang baru saja dimasukan. Lalu akan dikembalikan ke view list pelanggan.

- Delete Pelanggan

```
public function destroy($id)
{
    $data = User::find($id);
    $data->delete();
}
```

Pada function destroy hanya membutuhkan 1 parameter yaitu berupa id yang nantinya akan digunakan untuk mencari pelanggan berdasarkan id lalu dihapus dengan menggunakan delete().

- View List Mobil

```
public function index()
{
    $data['title'] = "List Mobil";
    $data['menu'] = 1;
    $cars = DB::table('tb_kendaraan')->get()->toArray();
    $data['cars'] = json_decode(json_encode($cars), true);
    $data['no'] = 1;
    return view('car.index', $data);
}
```

Pada function index terdapat 2 array, array \$cars yang berisi data mobil yang ada dan array \$data yang berisikan title, menu, \$cars, dan no. yang kemudian akan di return ke view bersamaan dengan array \$data.

- View Form Add Mobil

```
public function create()
{
    $data['title'] = "Add New Cars";
    $data['menu'] = 1;

    return view('car.create', $data);
}
```

Pada function create terdapat array \$data yang berisikan title dan menu, yang kemudian akan di return view form tambah mobil beserta array \$data.

- Add Mobil

```
public function store(Request $request)
{
    $validate = $request->validate([
        'merek' => 'required',
        'nopol' => 'required',
        'harga' => 'required',
        'type' => 'required',
        'deskripsi' => 'required',
        'gambar' => 'required'
    ]);

    if ($request->hasFile('gambar')) {
        $image_name = Image::make($request->file('gambar'))->resize(320, 240, function ($constraint) {
            $constraint->aspectRatio();
        });
        $location = $request->file('gambar')->store('public/uploads');
        $fileName = md5($location . microtime());
        $extension = '.' . explode("/", $image_name->mime())[1];
        Storage::put('public/mobil/' . $fileName . $extension, $image_name->encode());
    }

    $data['merek'] = $request['merek'];
    $data['nopol'] = $request['nopol'];
    $data['harga'] = $request['harga'];
    $data['type'] = $request['type'];
    $data['deskripsi'] = $request['deskripsi'];
    $data['gambar'] = $fileName . $extension;

    $insert = Car::create($data);

    return redirect()->route('car.index');
}
```

Pada function store hanya membutuhkan parameter \$request yaitu berupa value dari form add mobil yang kemudian akan divalidasi menggunakan \$validate. Jika data memenuhi kriteria pada \$validate maka data akan disimpan lalu di return ke view list mobil dan jika tidak maka akan dikembalikan ke view add mobil. Pada If statement di cek apakah ada gambar atau tidak yang digunakan untuk menyimpan gambar yang diupload ke server local.

- View Edit Mobil

```
public function edit($id)
{
    $data['car'] = Car::find($id);
    $data['title'] = "Edit Mobil " . $id;
    $data['menu'] = 1;

    return view('car.edit', $data);
}
```

Pada function edit ini dibutuhkan memiliki parameter berupa \$id yang nantinya akan digunakan pada \$data['car'] untuk mencari data mobil berdasarkan id tersebut. kemudian pada \$data diberikan title dan menu. Setelah itu akan di return view bersamaan dengan \$data yang nantinya akan menampilkan mobil yang memiliki id yang sesuai dengan id yang ada pada parameter yang nantinya akan di edit datanya pada form edit.

- Update Mobil

```
public function update(Request $request, $id)
{
    $validate = $request->validate([
        'merek' => 'required',
        'nopol' => 'required',
        'harga' => 'required',
        'type' => 'required',
        'deskripsi' => 'required',
        'gambar' => 'nullable'
    ]);

    if ($request->hasFile('gambar')) {
        $image_name = Image::make($request->file('gambar'))->resize(320, 240, function ($constraint) {
            $constraint->aspectRatio();
        });
        $location = $request->file('gambar')->store('public/uploads');
        $fileName = md5($location . microtime());
        $extension = '.' . explode("/", $image_name->mime())[1];
        Storage::put('public/mobil/' . $fileName . $extension, $image_name->encode());
        $data['gambar'] = $fileName . $extension;
    }

    $data['merek'] = $request['merek'];
    $data['nopol'] = $request['nopol'];
    $data['harga'] = $request['harga'];
    $data['type'] = $request['type'];
    $data['deskripsi'] = $request['deskripsi'];

    $update = Car::find($id)->update($data);
    return redirect()->route('car.index');
}
```

Pada function update terdapat 2 parameter yaitu \$request yang berupa value dari form edit data mobil dan \$id yang berupa id mobil. Yang kemudian nantinya akan divalidasi pada \$validate. Lalu pada if statement di cek apakah ada gambar yang di tambahkan atau tidak, jika ada maka gambar mobil tersebut akan diganti jika tidak maka gambar mobil tidak akan diganti. Lalu pada \$update dicari mobil berdasarkan \$id yang kemudian datanya di update dengan \$data. Setelah itu akan di return ke view list mobil.

- Delete Mobil

```
public function destroy($id)
{
    DB::table('tb_kendaraan')->where('id_kendaraan', '=', $id)->delete();
}
```

Pada function destroy terdapat 1 parameter berupa \$id yang digunakan untuk mencari mobil berdasarkan parameter \$id yang kemudian akan dihapus.

- View List Transaksi

```
public function index(){
    $data['menu'] = 2;
    $data['title'] = 'List Peminjaman';
    $transaksi = Returns::All();
    $data['no'] = 1;
    return view('returns.index', ['data'=>$data, 'transaksi'=>$transaksi]);
}
```

Pada function index terdapat 2 array, array \$transaksi yang berisi data transaksi yang ada dan array \$data yang berisikan title, menu, dan no. yang kemudian akan di return ke view bersamaan dengan array \$data dan \$transaksi.

- View Edit Transaksi

```
public function edit($id)
{
    $data['peminjaman'] = Returns::find($id);
    $data['title'] = "Edit Peminjaman ".$id;
    $data['menu'] = 2;

    return view('returns.edit', $data);
}
```

Pada function edit ini terdapat parameter berupa \$id yang nantinya akan digunakan pada \$data['peminjaman'] untuk mencari data transaksi berdasarkan id tersebut. kemudian pada \$data diberikan title dan menu. Setelah itu akan di return view bersamaan dengan \$data yang nantinya akan menampilkan transaksi yang memiliki id yang sesuai dengan id yang ada pada parameter yang nantinya akan di edit datanya pada view form edit.

- Update Transaksi

```
public function update(Request $request, $id)
{
    $validate = $request->validate([
        'harga' => 'required',
        'tgl_pinjam' => 'required',
        'tgl_kembali' => 'required',
        'waktu' => 'nullable',
        'denda' => 'nullable',
        'total' => 'nullable'
    ]);

    $data['harga'] = $request['harga'];
    $data['tgl_pinjam'] = $request['tgl_pinjam'];
    $data['tgl_kembali'] = $request['tgl_kembali'];
    $data['waktu'] = $request['waktu'];
    $date_now = date('d-m-Y\TH:i:s');
    $siswa_waktu = (int) round((strtotime($date_now)-strtotime($data['tgl_kembali']))/3600);

    if ($siswa_waktu > 0){
        $data['denda'] = (int)((($data['harga']/100)*2)*$siswa_waktu);
    }else{
        $data['denda'] = 0;
    }

    $data['total'] = $data['denda']+$data['harga'];
    if ($request['status'] == 'selesai'){
        $data['status'] = $request['status'];
        $status['status'] = 'tersedia';
        $update = Car::where('nopol', $request['nopol'])->update($status);
    }
    $update = Returns::find($id)->update($data);

    return redirect()->route('returns.index');
}
```

Pada function update terdapat 2 parameter yaitu \$request yaitu berupa value yang sudah di isikan pada view form edit dan \$id yaitu berupa id transaksi. Yang kemudian akan divalidasi pada \$validate. Jika data yang divalidasi sudah memenuhi kriteria maka akan disimpan dan dikembalikan ke view list transaksi, jika tidak sesuai maka akan dikembalikan ke view edit transaksi. Pada \$siswa_waktu dihitung berapa jam yang sudah di lewati oleh peminjam, jika jam sudah melewati batas waktu peminjaman maka pada if statement akan di hitung denda nya, jika belum melewati batas waktu maka denda akan dihitung 0. Kemudian pada form edit transaksi jika status di isi selesai maka akan mengupdate data mobil dan mengubah status mobil ke tersedia, jika tidak maka status mobil akan tetap dipinjam.

- Delete Transaksi

```
public function destroy($id)
{
    $data = Returns::find($id);
    $status['status'] = 'tersedia';
    $update = Car::where('id_kendaraan', $data['id_kendaraan'])->update($status);
    $data->delete();
}
```

Pada function destroy terdapat parameter \$id yang berupa id transaksi yang nantinya akan digunakan untuk mencari transaksi berdasarkan parameter tersebut. lalu pada \$update dicari mobil berdasarkan

id_kendaraan yang ada pada \$data transaksi yang kemudian statusnya diubah ke tersedia lalu transaksi pun dihapus.

- Pelanggan - View Profile

```
public function index($id)
{
    $data_user = User::find($id)->where('id_pelanggan', $id)->get()[0];
    $menu = 2;
    $title = 'Profile - '.$data_user->nama;
    return view('pages.user.index', compact('title', 'menu', 'data_user'));
}
```

Pada index terdapat parameter \$id yang berupa id pelanggan. Pada \$data_user dicari pelanggan berdasarkan parameter tersebut yang kemudian akan return pada view.

- Pelanggan - Update Profile

```
public function update(Request $request, $id)
{
    $validate = $request->validate([
        'nama' => 'required',
        'telp' => 'nullable',
        'alamat' => 'nullable',
        'email' => 'required',
        'password_new' => 'required',
    ]);

    $data['nama'] = $request['nama'];
    $data['telp'] = $request['telp'];
    $data['alamat'] = $request['alamat'];
    $data['email'] = $request['email'];
    if($request['password_old'] != $request['password_new']){
        $data['password'] = bcrypt($request['password_new']);
    }else{
        $data['password'] = $request['password_old'];
    }

    $update = User::find($id)->update($data);
    return redirect()->route('profile', $id);
}
```

Pada function update terdapat 2 parameter yaitu \$request yaitu berupa value dari form view profile dan \$id yaitu berupa id pelanggan tersebut. Yang kemudian divalidasi pada \$validate jika sesuai kriteria maka data akan disimpan lalu di return view profile dan jika tidak sesuai maka akan di return ke view profile tanpa disimpan. Pada if statement juga di cek apakah password mengalami perubahan atau tidak, jika ya maka password akan diubah.

- Pelanggan - View List Mobil

```
public function index()
{
    $title = "Home";
    $data = Car::Where('status', '=', 'tersedia')->get();
    $menu = 0;
    return view('pages.dashboard', compact('title', 'menu', 'data'));
}
```

Pada function index dibuat array \$data yang isinya list mobil yang memiliki status tersedia yang kemudian akan ditampilkan pada view home pada halaman pelanggan.

- Pelanggan - View Mobil

```
public function viewMobil($id)
{
    $title = "Pesan Mobil";
    $mobil = Car::Where('id_kendaraan', '=', $id)->get();
    $menu = 0;
    return view('pages.view_mobil', compact('title', 'menu', 'mobil'));
}
```

Pada function viewMobil terdapat parameter \$id yang berupa id dari mobil yang dipilih, yang kemudian pada \$mobil digunakan untuk mencari mobil berdasarkan parameter tersebut. lalu ditampilkan pada view mobil pada halaman pelanggan.

- Pelanggan - Add Transaksi

```
public function pesan(Request $request)
{
    $validate = $request->validate([
        'id_transaksi' => 'nullable',
        'nopol' => 'nullable',
        'harga' => 'required',
        'tgl_pinjam' => 'required',
        'tgl_kembali' => 'required',
        'waktu' => 'required',
        'denda' => 'nullable',
        'total' => 'required'
    ]);
    $id = DB::table('tb_transaksi')->orderBy('id_transaksi', 'desc')->select('id_transaksi')->first()->id_transaksi;
    $data['id_transaksi'] = 'PM'.sprintf("%04d", (int)str_replace('PM', '', $id)+1);
    $data['id_pelanggan'] = Auth::user()->id_pelanggan;
    $data['id_kendaraan'] = (int)$request['id_mobil'];
    $data['harga'] = (int)$request['harga'];
    $data['tgl_pinjam'] = $request['tgl_pinjam'];
    $data['tgl_kembali'] = $request['tgl_kembali'];
    $data['total'] = (int)$request['total'];
    $data['waktu'] = (int)$request['waktu'];
    if ($data['waktu'] < 0){
        $data['denda'] = int((( $data['harga']/100)*2)*$data['waktu']);
    }else{
        $data['denda'] = 0;
    }

    $data['total'] = $data['denda']+$data['total'];
    $data['status'] = 'dipinjam';
    $insert = Returns::create($data);
    $status['status'] = 'dipinjam';
    $update = Car::find($data['id_kendaraan'])->update($status);
    return redirect()->route('nota', $data['id_transaksi']);
}
```

Pada function pesan terdapat paramter \$request yang berupa value dari form yang terdapat di view mobil pada halaman pelanggan yang kemudian akan divalidasi datanya dengan \$validate. Jika data

sesuai dengan validasi maka akan dilanjutkan ke proses berikutnya jika tidak maka akan kembali ke halaman view mobil pada pelanggan. Kemudian pada \$id di cek data transaksi terakhir dan diambil value integer nya lalu ditambahkan 1 yang nantinya akan digunakan untuk menjadi id transaksi tersebut. kemudian pada \$insert dibuatlah transaksi berdasarkan \$data yang tadi sudah di isikan dan disimpan. Lalu pada \$update di update data mobil berdasarkan \$data['id_kendaraan'] dan mengubah status mobil tersebut ke dipinjam. Setelah itu akan di return ke halaman nota dengan argumen \$data['id_transaksi.'].

- Pelanggan - view nota

```
public function nota($id)
{
    $title = "Nota - ".$id;
    $menu = 0;
    $data = Returns::find($id)->where('id_transaksi', $id)->get();
    return view('pages.nota', compact('data', 'title', 'menu'));
}
```

Pada view nota terdapat parameter \$id yang berisikan id transaksi yang sudah dibuat pada function pesan tadi, yang kemudian akan menampilkan data transaksi yang tadi dibuat dengan return view halaman nota.

6 PENGUJIAN & TURNOVER

Saran:

1. Saat registrasi dapat diberikan NIK jadi bisa digunakan untuk melacak pelanggan agar terhindar dari pencurian.

Pertanyaan & Tanggapan:

Pertanyaan	Tanggapan
Untuk page login apakah jadi 1?	tidak, nantinya akan ada 2 login page yaitu login page khusus admin dan khusus pelanggan
Semisal ada tanggal peminjaman dan tanggal kembali, untuk menentukan denda nya itu bagaimana?	untuk menentukan denda menggunakan perhitungan di backend, untuk perhitungannya $\text{waktu} = (\text{tanggal} + \text{jam pengembalian}) - (\text{tanggal} + \text{jam saat mobil sudah divertifikasi kembali})$ jika $\text{waktu} < 0$ maka $\text{denda} = ((\text{waktu} \times -1)/12) \times ((\text{harga}/100) \times 2)$ selain itu maka denda 0 karena mengembalikannya tepat waktu atau tidak melewati waktu deadline
Data apa saja yang diminta saat user registrasi?	untuk saat ini hanya memasukan nama, email dan password. untuk mengisi alamat dan no.telp dapat dilakukan di halaman profile.
Apakah saat menyewa sudah termasuk drivernya?	tidak, karena ini masuk ke penyewaan mobil personal jadi tidak menyediakan penyewaan driver.