

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Sistem

Proses pendataan pelat nomor yang dilakukan di negara Indonesia masih dilakukan secara manual, sehingga rentan untuk terjadi kesalahan. Selain rentannya terjadi kesalahan, pendataan secara manual juga tidak efisien apabila kendaraan yang perlu didata berjumlah banyak. Oleh karena itu diperlukan sebuah upaya untuk mengotomatisasi proses pendataan pelat nomor. Untuk menyelesaikan masalah tersebut dibangun sebuah perangkat lunak yang dapat digunakan untuk mendeteksi pelat nomor dan mendatanya secara otomatis. Aplikasi yang akan dibangun merupakan aplikasi berbasis *mobile* bersistem operasi Android. Sistem operasi Android dipilih karena umum digunakan dan perangkatnya yang murah. Data yang didapatkan oleh perangkat lunak ini berupa teks dan juga foto dari pelat nomor yang dideteksi. Data tersebut kemudian akan disimpan dalam tabel di dalam *database* pada *backend* aplikasi. Komunikasi antara aplikasi *frontend* Android dengan *backend* dilakukan menggunakan protokol HTTP.

B. Lingkup Masalah

Perangkan lunak yang dihasilkan berupa aplikasi yang terdiri dari dua bagian, yaitu *frontend* dan *backend*. Bagian *frontend* dibangun menggunakan IDE Android Studio dengan bahasa pemrograman Java. Pada bagian *frontend* digunakan *library* digunakan untuk melakukan pemrosesan citra dari pelat nomor kendaraan sebelum dilakukan proses pengenalan karakter. Untuk proses pengenalan karakter dari citra yang sudah diolah digunakan *library* Tesseract OCR. Bagian *backend* dibangun menggunakan *framework* Laravel untuk membuat REST *server*. Untuk *database* dari *backend* menggunakan MySQL.

Aplikasi Pendeteksi Pelat Nomor Kendaraan dapat mendeteksi karakter pelat nomor kendaraan bermotor pribadi yang ditangkap dari kamera ponsel bersistem operasi Android. Untuk mendeteksi area pelat nomor digunakan metode deteksi tepi *Canny*. Setelah itu *Region of Interest* (ROI) pelat nomor yang terdeteksi akan diekstrak. Pada ROI yang terekstrak kemudian akan dilakukan transformasi perspektif untuk meluruskan rotasi dan perspektif citra. Selanjutnya untuk mempermudah proses pengenalan karakter dilakukan *thresholding* pada citra ROI sehingga dihasilkan citra biner. Citra biner yang dihasilkan kemudian diteruskan ke Tesseract OCR untuk mendapatkan karakter dari pelat nomor. Aplikasi ini dapat mengelola data kendaraan dan juga data hasil deteksi pelat nomor. Data kendaraan seperti merek, jenis, dan nama kendaraan dimasukkan secara manual karena aplikasi ini tidak terintegrasi dengan data Samsat.

Manfaat dari pembangunan aplikasi ini adalah untuk membuat sebuah sistem untuk mengotomatisasi proses pendataan pelat nomor kendaraan yang terjangkau. Aplikasi ini berfungsi untuk melakukan pendataan pelat nomor dengan mengaplikasikan *image processing* dan OCR. Data yang didapatkan pada aplikasi ini akan disimpan pada *database* terpusat sehingga mudah untuk diakses kembali. Dengan pembangunan aplikasi ini proses pendataan pelat nomor kendaraan bermotor menjadi lebih cepat dan akurat. Teknologi dari aplikasi ini bisa diimplementasikan untuk sistem usaha tempat parkir, bengkel, dan logistik.

C. Perspektif Produk

Aplikasi ini dikembangkan untuk membantu proses pendataan dari pelat nomor kendaraan bermotor pada usaha yang bergerak dibidang bengkel, logistik, maupun tempat parkir. Aplikasi ini dikembangkan pada platform *mobile* Android, sehingga lebih terjangkau dan praktis jika dibandingkan dengan kebanyakan sistem ANPR. Aplikasi ini dibuat menggunakan bahasa pemrograman Java dibantu dengan *library* OpenCV dan Tesseract OCR. Aplikasi ini juga memiliki *backend* yang dibangun

menggunakan *framework* Laravel sebagai REST *server* dan MySQL sebagai basis data untuk penyimpanan data.

D. Fungsi Produk

Sistem aplikasi pendeteksi pelat nomor kendaraan memiliki fungsi produk sebagai berikut.

1. Fungsi Login

Fungsi ini digunakan oleh pengguna untuk masuk ke dalam sistem dengan menggunakan *username* dan *password* yang sudah diregistrasikan.

2. Fungsi Registrasi

Fungsi ini digunakan oleh pengguna untuk meregistrasikan dirinya ke dalam sistem. Untuk meregistrasikan diri pengguna perlu memasukkan data nama, *username*, dan *password* yang ingin digunakan.

3. Fungsi Mengelola Data Kendaraan

Fungsi mengelola data kendaraan adalah fungsi yang digunakan untuk melakukan pengelolaan data kendaraan. Fungsi ini terdiri dari:

a. Fungsi Melihat Daftar Kendaraan

Fungsi ini digunakan oleh pengguna untuk melihat daftar kendaraan yang sudah tersedia di dalam sistem. Fungsi ini memungkinkan pengguna untuk melihat *list* data kendaraan dan juga detail dari data kendaraan.

b. Fungsi Mencari Data Kendaraan

Fungsi ini digunakan oleh pengguna untuk mencari data kendaraan yang sudah ada di dalam sistem. Fungsi ini memungkinkan pengguna untuk mencari data kendaraan

berdasarkan nama pemilik, nomor polisi, merek kendaraan, dan tipe kendaraan.

c. Fungsi Menambah Data Kendaraan

Fungsi ini digunakan oleh pengguna untuk menambahkan data kendaraan baru yang belum terdaftar di dalam sistem. Untuk menambahkan data kendaraan baru dibutuhkan nomor polisi, nama pemilik, merek kendaraan, tipe kendaraan, jenis kendaraan, warna kendaraan, dan tahun pembuatan kendaraan.

d. Fungsi Menghapus Data Kendaraan

Fungsi ini digunakan oleh pengguna untuk menghapus data kendaraan yang sudah tersedia di dalam sistem.

e. Fungsi Mengubah Data Kendaraan

Fungsi ini digunakan oleh pengguna untuk melakukan perubahan pada data kendaraan yang sudah ada. Semua data dari kendaraan bisa diubah kecuali untuk nomor polisi kendaraan.

4. Fungsi Memindai Pelat Nomor

Fungsi ini digunakan oleh pengguna untuk memindai teks dari pelat nomor kendaraan. Fungsi ini memungkinkan pengguna untuk memindai teks dari pelat nomor kendaraan menggunakan kamera dari *smartphone* secara *real-time*. Setelah terdeteksi apabila pelat nomor sudah terdaftar pada di dalam sistem maka data akan ditambahkan pada data riwayat pemindaian.

5. Fungsi Mengelola Data Riwayat Pemindaian

Fungsi mengelola data riwayat pemindaian adalah fungsi yang digunakan untuk melakukan pengelolaan data kendaraan. Fungsi ini terdiri dari:

a. Fungsi Melihat Daftar Riwayat Pemindaian

Fungsi ini digunakan oleh pengguna untuk melihat daftar riwayat pemindaian yang sudah tersedia di dalam sistem. Fungsi ini memungkinkan pengguna untuk melihat *list* data riwayat pemindaian pelat nomor kendaraan dan juga detailnya.

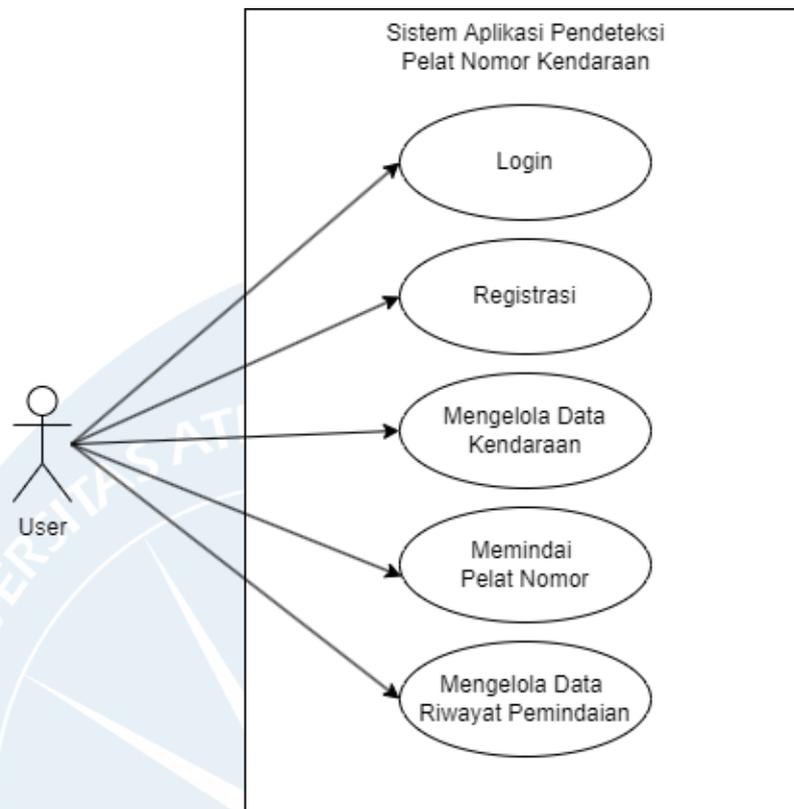
b. Fungsi Mencari Data Riwayat Pemindaian

Fungsi ini digunakan oleh pengguna untuk mencari data riwayat pemindaian yang sudah tersedia di dalam sistem. Fungsi ini memungkinkan pengguna untuk mencari data riwayat pemindaian berdasarkan tanggal pemindaian atau nomor polisi yang terdeteksi.

c. Fungsi Menghapus Data Riwayat Pemindaian

Fungsi ini digunakan oleh pengguna untuk menghapus data riwayat pemindaian yang sudah tersedia di dalam sistem.

Use case diagram adalah diagram yang digunakan untuk merepresentasikan hubungan antara aktor atau pengguna sistem dengan sistem yang digunakan. Aplikasi pendeteksi pelat nomor kendaraan ini terdiri dari lima *use case* yaitu: *login*, registrasi, mengelola data kendaraan, mengelola data riwayat pemindaian, dan memindai pelat nomor. Pada Gambar 4.1 berikut terdapat *use case diagram* dari aplikasi pendeteksi pelat nomor kendaraan.



Gambar 4.1 Use Case Diagram

E. Kebutuhan Antarmuka

1. Antarmuka Pengguna

Sistem aplikasi pendeteksi pelat nomor kendaraan memiliki *form* antarmuka pengguna seperti yang dijelaskan pada Tabel 4.1 berikut.

Tabel 4.1 Tabel *Form* Antarmuka Pengguna

No.	Nama <i>Form</i>	:	Deskripsi Fungsi <i>Form</i>
1.	<i>Form</i> Login	:	Sebagai <i>form</i> yang digunakan oleh pengguna untuk masuk ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> yang sudah terdaftar.

2.	<i>Form</i> Registrasi	:	Sebagai <i>form</i> yang digunakan untuk pengguna baru mendaftar akun baru untuk masuk ke dalam sistem. Data yang perlu dimasukkan adalah nama, <i>username</i> , dan <i>password</i> .
3.	<i>Form</i> Daftar Kendaraan	:	Sebagai <i>form</i> yang digunakan untuk melihat daftar data kendaraan yang sudah terdaftar.
4.	<i>Form</i> Detail Kendaraan	:	Sebagai <i>form</i> yang digunakan untuk melihat detail dari salah satu data kendaraan yang sudah terdaftar. Pada <i>form</i> ini juga terlihat daftar riwayat pemindaian dari kendaraan ini.
5.	<i>Form</i> Tambah Data Kendaraan	:	Sebagai <i>form</i> yang digunakan untuk menambah data kendaraan yang belum dimasukkan ke dalam sistem. Data yang perlu dimasukkan adalah nomor polisi, nama pemilik, jenis kendaraan, gambar, warna kendaraan, merek kendaraan, tipe kendaraan, dan tahun pembuatan kendaraan.
6.	<i>Form</i> Ubah Data Kendaraan	:	Sebagai <i>form</i> yang digunakan untuk mengubah data kendaraan yang sudah terdaftar di dalam sistem. Data yang dapat diubah adalah nama pemilik, jenis kendaraan, gambar, warna kendaraan, merek kendaraan,

			tipe kendaraan, dan tahun pembuatan kendaraan.
7.	<i>Form</i> Daftar Riwayat Pemindaian	:	Sebagai <i>form</i> yang digunakan untuk melihat daftar riwayat pemindaian pelat nomor yang dilakukan pengguna.
8.	<i>Form</i> Detail Riwayat Pemindaian	:	Sebagai <i>form</i> yang digunakan untuk melihat detail dari riwayat pemindaian seperti gambar yang diambil saat pemindaian dan ROI pelat nomor yang terdeteksi.
9.	<i>Form</i> Pindai Pelat Nomor	:	Sebagai <i>form</i> yang digunakan untuk memindai pelat nomor kendaraan menggunakan kamera dari <i>smartphone</i> pengguna. Ketika pelat nomor terdeteksi pada <i>form</i> ini akan terlihat garis yang mengelilingi pelat nomor. Apabila pengguna mengklik tombol untuk mendeteksi maka karakter yang terdeteksi dari pelat nomor akan dicek ketersediaannya di data kendaraan. Apabila tersedia akan dimasukkan ke dalam riwayat pemindaian.

2. Antarmuka Perangkat Keras

Perangkat keras yang dibutuhkan untuk mengoperasikan sistem aplikasi pendeteksi pelat nomor kendaraan adalah *smartphone* yang menggunakan sistem operasi Android.

3. Antarmuka Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan sistem aplikasi pendeteksi pelat nomor kendaraan di sisi *server* adalah sebagai berikut:

1. Nama : Apache
Sumber : Apache Software Foundation
Deskripsi : Sebagai *web server* untuk menjalankan perangkat lunak sisi server pendeteksi pelat nomor kendaraan.
2. Nama : MySQL
Sumber : Oracle Corporation
Deskripsi : Sebagai *Database Management System* (DBMS) yang digunakan untuk menyimpan data pada server.

Perangkat lunak yang dibutuhkan untuk mengoperasikan sistem aplikasi pendeteksi pelat nomor kendaraan di sisi klien adalah sebagai berikut:

1. Nama : Android
Sumber : Google
Deskripsi : Sebagai sistem operasi untuk menjalankan perangkat lunak sisi klien pendeteksi pelat nomor kendaraan.

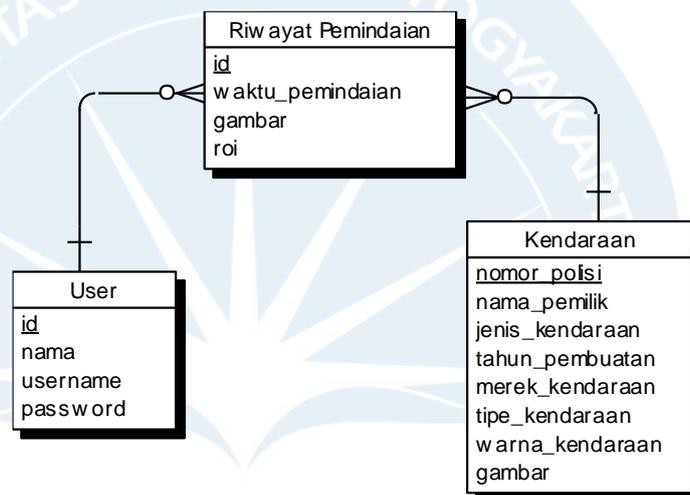
4. Antarmuka Komunikasi

Antarmuka komunikasi yang digunakan oleh sistem ini untuk melakukan komunikasi antara sisi klien dengan sisi server adalah *Application Programming Interface* (API). Setiap permintaan maupun pengiriman data yang dilakukan oleh klien ke *database* akan dilakukan oleh *web server* sebagai penyedia API. Layanan API yang digunakan pada sistem ini adalah *Representational State Transfer* (REST) API yang menggunakan protokol *Hypertext Transfer Protocol* (HTTP).

F. Perancangan

1. Perancangan Data

Pada Gambar 4.2 terdapat sebuah diagram *Entity Relationship Diagram* (ERD) yang merepresentasikan relasi antar entitas atau objek. Selain menunjukkan relasi antar entitas ERD juga menampilkan atribut-atribut dari tiap entitas yang ada pada sistem. Pada sistem ini terdapat tiga buah tabel yang saling berhubungan dan memiliki atribut yang berbeda-beda.



Gambar 4.2 Entity Relationship Diagram

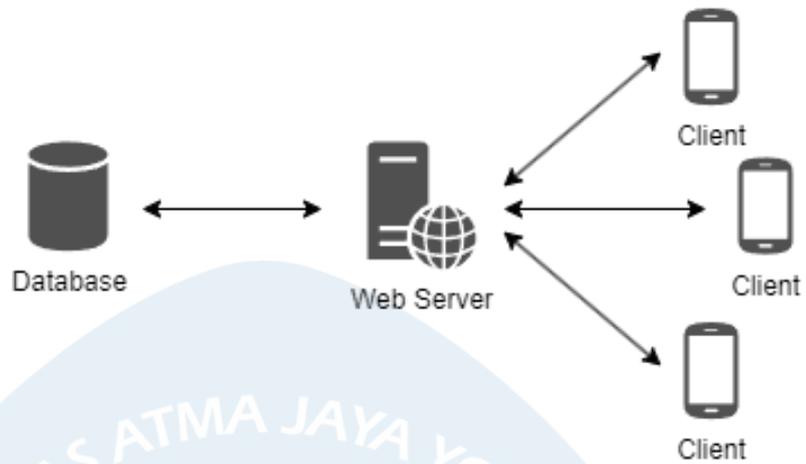
Tabel *user* digunakan untuk menyimpan data pengguna yang berupa nama pengguna, *username*, dan *password*. Data yang ada pada tabel *user* digunakan oleh pengguna untuk masuk ke dalam sistem. Tabel kendaraan digunakan untuk menyimpan detail dari data kendaraan yang sudah dimasukkan ke dalam sistem. Data kendaraan terdiri dari nomor polisi, nama pemilik, gambar, jenis kendaraan, tahun pembuatan, merek kendaraan, tipe kendaraan, dan warna kendaraan. Selanjutnya ada tabel riwayat pemindaian yang digunakan untuk menyimpan hasil pemindaian pelat nomor yang dilakukan menggunakan kamera *smartphone*. Atribut yang disimpan di dalam tabel riwayat pemindaian adalah waktu pemindaian yang berupa waktu dan tanggal ketika pemindaian dilakukan, gambar yang diambil saat

melakukan pemindaian, dan gambar *Region of Interest* (ROI) dari pelat nomor yang terdeteksi. Tabel riwayat pemindaian terhubung dengan tabel *user* dan tabel kendaraan, sehingga tabel ini juga akan menyimpan *id user* yang melakukan pemindaian dan juga nomor polisi kendaraan yang terdeteksi.

2. Perancangan Arsitektur

a. Overview Sistem

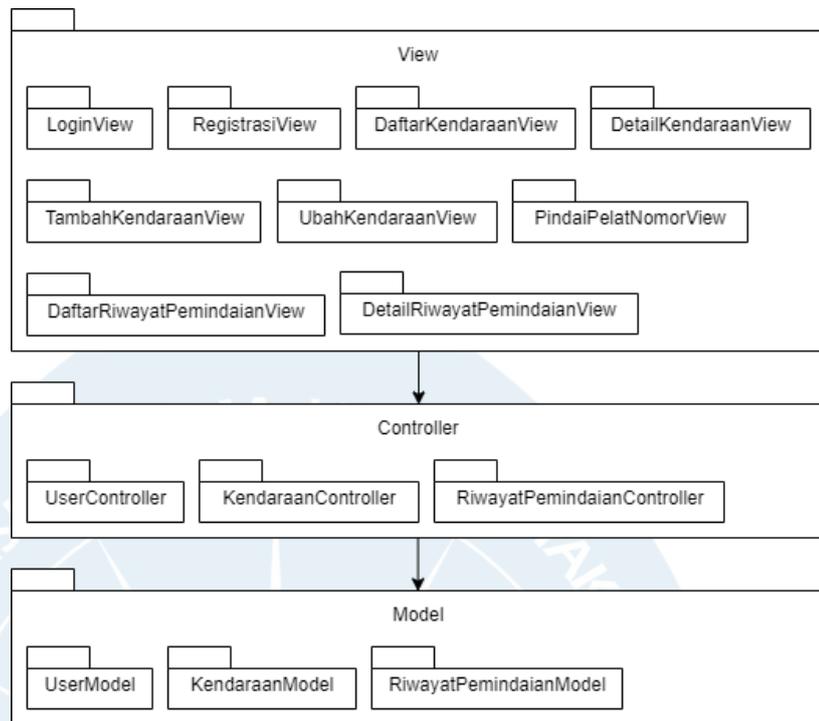
Pada Gambar 4.3 terlihat diagram yang merepresentasikan arsitektur sistem aplikasi pendeteksi pelat nomor kendaraan. Sistem ini terdiri dari dua bagian yaitu sisi klien dan sisi server. Sisi klien sistem berupa sebuah perangkat lunak berbasis Android. Perangkat lunak klien memiliki tanggung jawab untuk memberikan tampilan pada aplikasi. Sisi server dari sistem adalah sebuah *Representational State Transfer* (REST) server, yang berjalan pada *web server*. Web server berkomunikasi dengan aplikasi sisi klien menggunakan protokol *Hypertext Transfer Protocol* (HTTP). Aplikasi klien akan mengirimkan *request* ke *endpoint* REST API yang telah disediakan oleh REST server. REST server kemudian akan menerima *request* dari klien. Sisi server aplikasi kemudian berkomunikasi dengan *database* untuk melakukan pengambilan dan perubahan data sesuai dengan *request* dari klien dan kemudian akan memberikan *response* dalam bentuk *JavaScript Object Notation* (JSON) kembali ke klien. Data *response* berformat JSON ini kemudian akan diolah oleh aplikasi sisi klien untuk disajikan sesuai dengan kebutuhan pengguna. Pada sistem ini *database* yang digunakan adalah MySQL.



Gambar 4.3 Overview Sistem

b. Arsitektur Perangkat Lunak

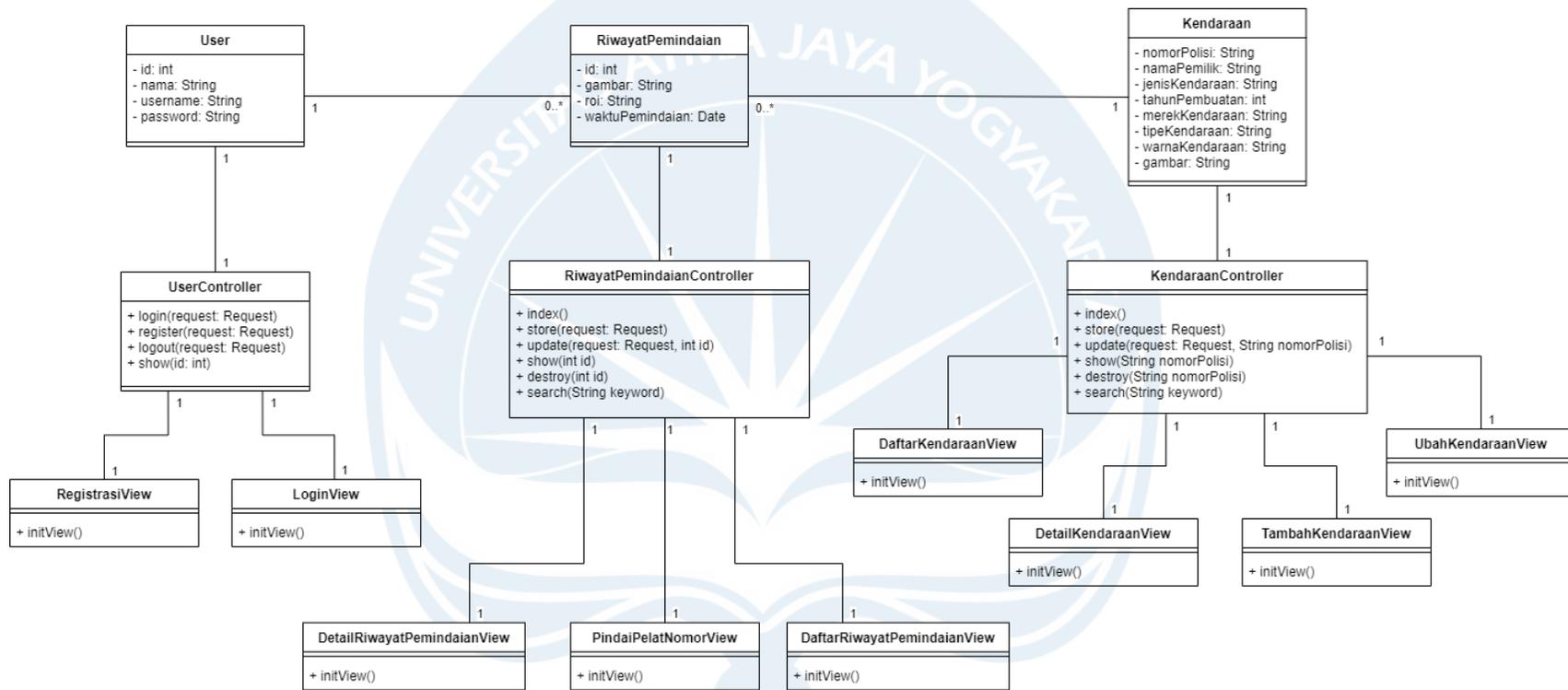
Pada Gambar 4.4 terlihat sebuah *package diagram* yang merepresentasikan arsitektur perancangan perangkat lunak yang digunakan pada sistem aplikasi pendeteksi pelat nomor kendaraan ini. Sistem ini mengimplementasikan arsitektur MVC yang terdiri dari tiga bagian yaitu *Model*, *View*, dan *Controller*. *Model* adalah bagian yang terhubung secara langsung dengan *database* dan bertugas untuk mengelola data. *View* merupakan bagian yang memberikan tampilan yang dapat digunakan pengguna untuk melihat informasi. *Controller* adalah bagian yang bertugas untuk menjadi jembatan penghubung antara *model* dan *view*.



Gambar 4.4 Package Diagram

c. Class Diagram

Pada Gambar 4.5 terlihat sebuah *class diagram* dari sistem aplikasi pendeteksi pelat nomor kendaraan. *Class diagram* menggambarkan struktur, atribut, *method*, dan hubungan dari kelas yang digunakan pada sistem.



Gambar 4.5 Class Diagram

3. Perancangan Antarmuka

d. Halaman *Login*

Pada Gambar 4.6 terlihat rancangan tampilan dari halaman *login*. Halaman *login* ini akan muncul ketika aplikasi pertama kali dibuka dan pengguna belum melakukan *login*. Untuk masuk ke dalam sistem pengguna diminta untuk memasukkan *username* dan *password* dari akun yang sudah terdaftar lalu klik tombol “Login”. Apabila belum *login* di tampilan ini juga terdapat tombol yang bertuliskan “Belum punya akun? Daftar” yang ketika diklik akan membuka halaman registrasi.



Gambar 4.6 Rancangan Tampilan Halaman *Login*

e. Halaman Registrasi

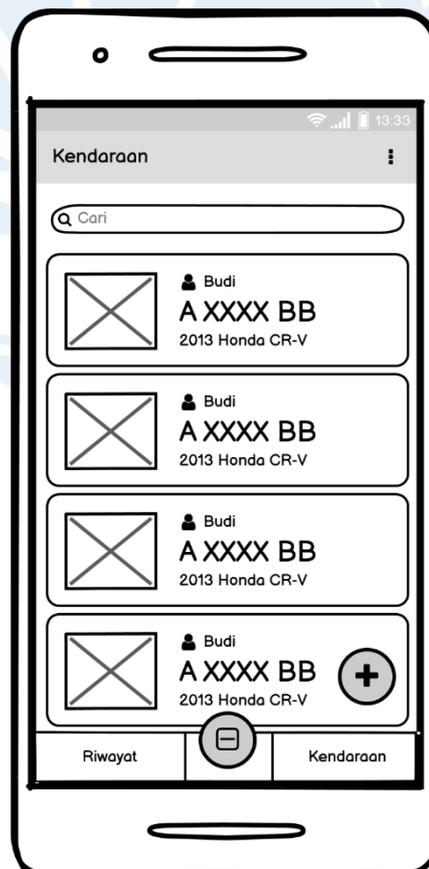
Pada Gambar 4.7 terlihat rancangan tampilan dari halaman registrasi. Halaman registrasi ini digunakan untuk mendaftarkan kredensial dari akun baru. Untuk mendaftarkan akun baru pengguna diminta untuk memasukkan *username* yang masih belum dipakai, nama lengkap, dan *password*. Setelah semua *text field* terisi pengguna dapat menekan tombol “Daftar” untuk mendaftarkan akun barunya. Pada tampilan ini juga terdapat tombol yang bertuliskan “Sudah punya akun? Login” untuk membuka tampilan *login*.



Gambar 4.7 Rancangan Tampilan Halaman Registrasi

f. Halaman Daftar Kendaraan

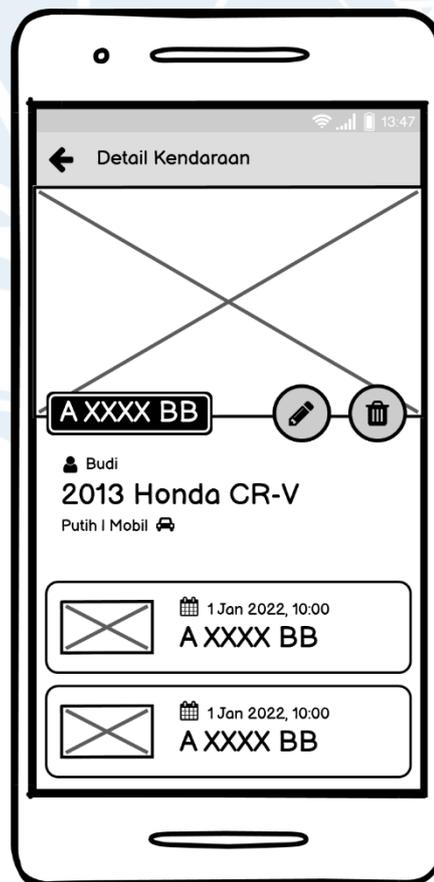
Pada Gambar 4.8 terlihat rancangan tampilan dari halaman daftar kendaraan. Halaman ini digunakan untuk melihat daftar data kendaraan yang sudah terdaftar pada sistem. Atribut dari kendaraan yang ditampilkan pada tampilan ini adalah nama pemilik, nomor polisi, tahun pembuatan kendaraan, merek kendaraan, dan tipe kendaraan. Pada tampilan ini terdapat sebuah *search box* yang digunakan untuk memfilter data kendaraan berdasarkan masukkan pengguna. Bila pengguna menekan salah satu item data kendaraan maka tampilan detail kendaraan akan terbuka. Pada tampilan ini juga terdapat tombol dengan *icon* “+” yang akan membuka tampilan tambah kendaraan ketika diklik.



**Gambar 4.8 Rancangan Tampilan Halaman Daftar
Kendaraan**

g. Halaman Detail Kendaraan

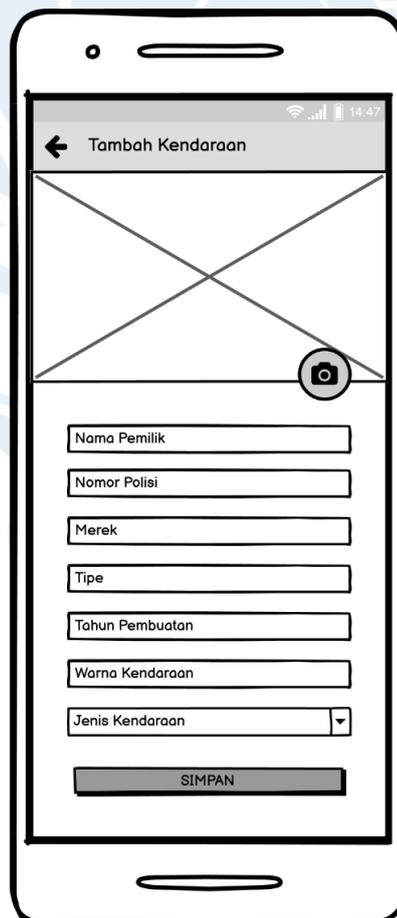
Pada Gambar 4.9 terlihat rancangan tampilan dari halaman detail kendaraan. Halaman ini digunakan untuk melihat detail data dari salah satu data kendaraan yang sudah dipilih. Pada halaman ini akan ditampilkan gambar kendaraan, nomor polisi, nama pemilik, tahun pembuatan, merek kendaraan, tipe kendaraan, warna kendaraan, dan jenis kendaraan. Di bagian bawah kanan gambar juga terdapat dua buah tombol yaitu tombol edit yang akan membuka tampilan Ubah Kendaraan dan juga tombol hapus untuk menghapus data kendaraan. Selain data dari kendaraan tampilan ini juga menampilkan riwayat pemindaian dari kendaraan ini.



**Gambar 4.9 Rancangan Tampilan Halaman Detail
Kendaraan**

h. Halaman Tambah Kendaraan

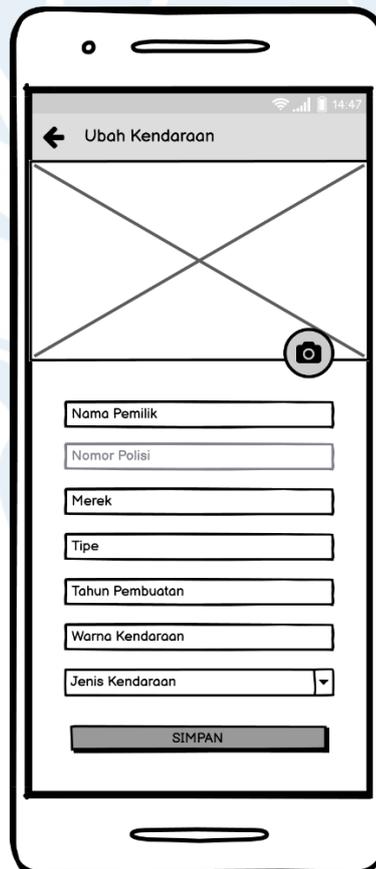
Pada Gambar 4.10 terlihat rancangan tampilan dari halaman tambah kendaraan. Pada halaman ini terdapat enam buah masukkan *text* yaitu untuk memasukkan nama pemilik, nomor polisi, merek kendaraan, tipe kendaraan, tahun pembuatan dan warna kendaraan. Selain itu juga terdapat satu buah *combo box* untuk memilih jenis kendaraan. Pada tampilan ini terdapat tombol kamera untuk menambahkan gambar untuk kendaraan, dan apabila ditambahkan maka gambar akan ditampilkan. Untuk menyimpan data kendaraan terdapat tombol yang berada pada bagian paling bawah tampilan.



Gambar 4.10 Rancangan Tampilan Halaman Tambah Kendaraan

i. Halaman Ubah Kendaraan

Pada Gambar 4.11 terlihat rancangan tampilan dari halaman ubah kendaraan. Pada halaman ini terdapat enam buah masukkan *text* yaitu untuk memasukkan nama pemilik, nomor polisi, merek kendaraan, tipe kendaraan, tahun pembuatan dan warna kendaraan. Tetapi untuk masukkan *text* nomor polisi berbeda dengan tampilan tambah kendaraan, tidak dapat diubah. Selain itu juga terdapat satu buah *combo box* untuk memilih jenis kendaraan. Pada tampilan ini terdapat tombol kamera untuk menambahkan atau mengganti gambar untuk kendaraan, dan apabila ditambahkan maka gambar akan ditampilkan. Untuk menyimpan data kendaraan terdapat tombol yang berada pada bagian paling bawah tampilan.

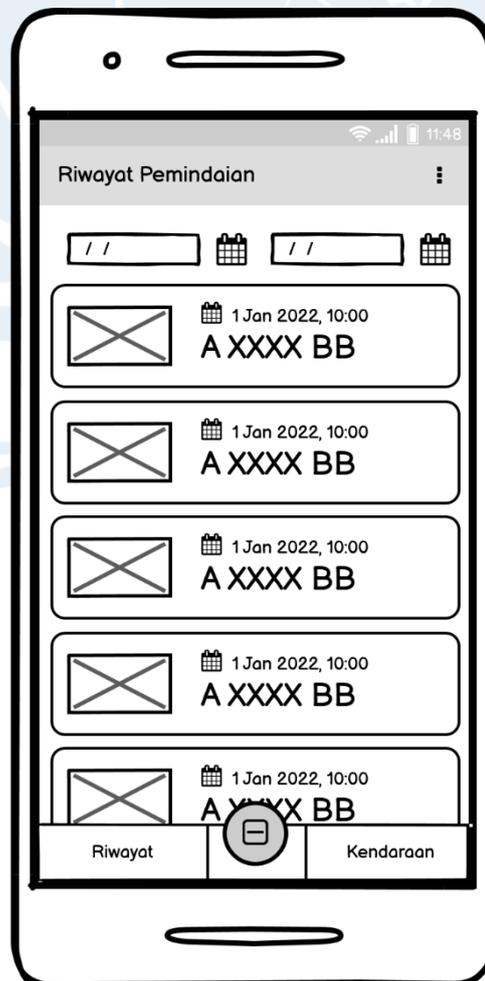


The image shows a mobile application interface for updating vehicle information. The screen is titled "Ubah Kendaraan" (Change Vehicle) and features a back arrow on the left. The main content area is a form with the following fields: "Nama Pemilik" (Owner Name), "Nomor Polisi" (License Plate Number), "Merek" (Brand), "Tipe" (Type), "Tahun Pembuatan" (Year of Manufacture), "Warna Kendaraan" (Vehicle Color), and "Jenis Kendaraan" (Vehicle Type) which is a dropdown menu. A camera icon is located to the right of the form fields. At the bottom of the form is a "SIMPAN" (Save) button. The background of the screen is a light blue circular logo with the text "UNIVERSITAS ANTA KUGIARANTA".

Gambar 4.11 Rancangan Tampilan Halaman Ubah Kendaraan

j. Halaman Daftar Riwayat Pemindaian

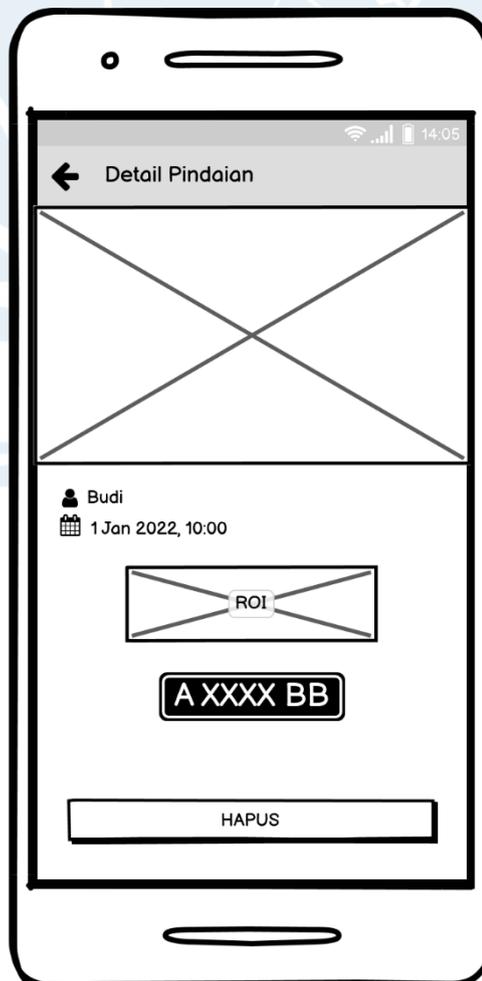
Pada Gambar 4.12 terlihat rancangan tampilan dari halaman daftar riwayat pemindaian. Halaman ini digunakan untuk melihat daftar data riwayat pemindaian yang sudah terdaftar pada sistem. Atribut yang ditampilkan pada tampilan ini adalah waktu pemindaian, nomor polisi yang terdeteksi, dan gambar ROI pelat nomor. Pada tampilan ini terdapat sebuah masukkan *range* tanggal untuk memfilter data kendaraan berdasarkan waktu pemindaian. Bila pengguna menekan salah satu item data riwayat pemindaian maka tampilan detail riwayat pemindaian akan terbuka.



Gambar 4.12 Rancangan Tampilan Halaman Daftar Riwayat Pemindaian

k. Halaman Detail Riwayat Pemindaian

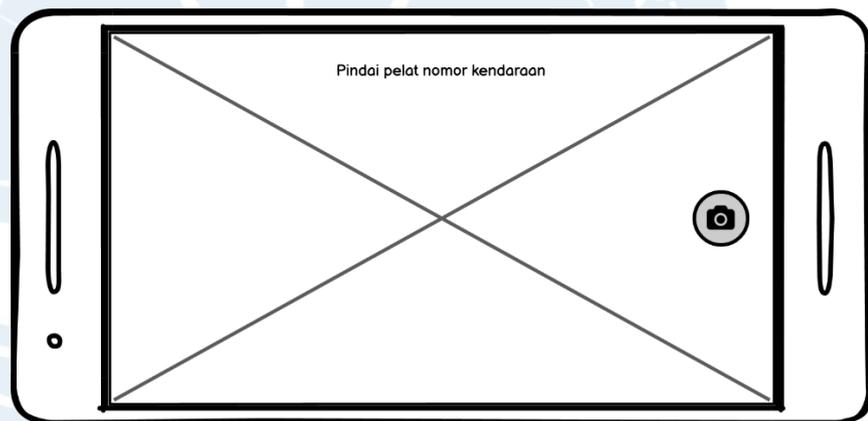
Pada Gambar 4.13 terlihat rancangan tampilan dari halaman detail riwayat pemindaian. Halaman ini digunakan untuk melihat detail data dari salah satu data riwayat pemindaian yang sudah dipilih. Pada halaman ini akan ditampilkan gambar yang ditangkap ketika melakukan pemindaian, gambar ROI dari pelat nomor, nomor polisi yang terdeteksi, nama pengguna yang melakukan pemindaian, dan waktu pemindaian. Di bagian bawah tampilan juga terdapat sebuah tombol untuk menghapus data riwayat pemindaian.



Gambar 4.13 Rancangan Tampilan Halaman Detail Riwayat Pemindaian

1. Halaman Pindai Pelat Nomor

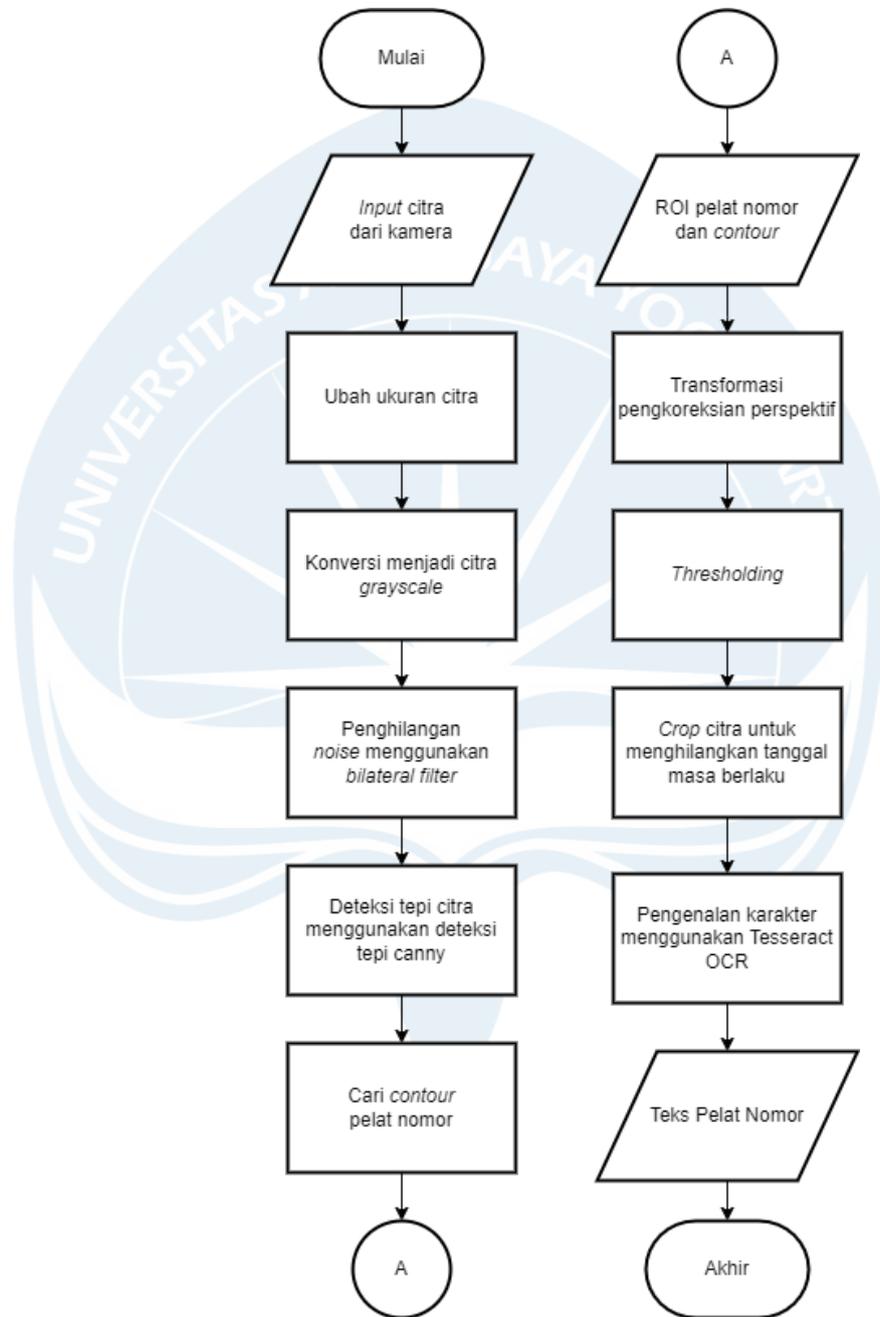
Pada Gambar 4.14 terlihat rancangan tampilan dari halaman pindai pelat nomor. Halaman ini digunakan oleh pengguna untuk memindai pelat nomor kendaraan menggunakan kamera dari *smartphone*. Pada tampilan ini lokasi pelat nomor dideteksi secara *real-time* dan terdapat tombol ber-*icon* kamera yang dapat ditekan untuk mengenali teks pelat nomor yang terdeteksi. Setelah teks dari pelat nomor sudah terbaca kemudian teks tersebut akan dicocokkan dengan data kendaraan dalam sistem. Apabila data kendaraan sudah terdaftar maka riwayat pemindaian baru akan ditambahkan. Apabila belum maka pengguna akan diminta untuk mendaftarkan data kendaraan.



Gambar 4.14 Rancangan Tampilan Halaman Pindai Pelat Nomor

4. Perancangan Proses Deteksi Pelat Nomor

Pada Gambar 4.15 merupakan *flowchart* atau diagram alir dari proses deteksi pelat nomor kendaraan yang digunakan pada sistem ini.



Gambar 4.15 *Flowchart* Proses Deteksi Pelat Nomor

Masukkan dari sistem merupakan citra yang diambil secara *realtime* dari kamera perangkat. Untuk meningkatkan performa sistem

citra yang diambil ukurannya akan diubah menjadi setengah ukuran citra asli. Citra yang sudah diubah ukurannya kemudian akan dikonversi menjadi citra *grayscale* yang sebelumnya merupakan citra warna. Selanjutnya akan dilakukan proses penghalusan citra untuk menghilangkan *noise* pada citra menggunakan metode *bilateral filter*. *Bilateral filter* dipilih karena metode ini dapat menghilangkan *noise* citra tanpa mengganggu tepi (*edge-preserving*). Selanjutnya tepi pada citra akan dideteksi menggunakan metode deteksi tepi Canny. Selanjutnya dilakukan pencarian *contour* yang kemungkinan merupakan pelat nomor. Pencarian ini dilakukan berdasarkan ciri-ciri dari pelat nomor yang memiliki empat sudut. Proses selanjutnya apabila *contour* yang berkemungkinan merupakan pelat nomor ditemukan, daerah ini akan dijadikan sebagai ROI yang kemudian akan dipotong dari citra aslinya.

Region of Interest dari pelat nomor yang sudah didapatkan dari proses sebelumnya kemudian akan diproses untuk persiapan OCR. Proses pertama adalah untuk memperbaiki perspektif dari objek pelat nomor, proses ini dilakukan supaya hasil dari OCR bisa tetap akurat walaupun citra diambil persis dari depan ataupun miring. Selanjutnya akan dilakukan *thresholding* untuk mengubah citra *grayscale* menjadi citra biner. Pemrosesan yang terakhir adalah untuk memotong bagian tanggal yang ada di bagian bawah pelat nomor sehingga tidak mengganggu proses pembacaan teks pelat nomor. Setelah citra diproses kemudian citra akan diserahkan ke Tesseract OCR yang kemudian akan memberikan teks yang terbaca dari citra.