

BAB III

METODE PENELITIAN

3.1. Desain Penelitian

Untuk menerapkan sistem parkir yang mengintegrasikan metode YOLO untuk mendeteksi plat nomor dan CNN untuk klasifikasi karakter pada plat nomor, diperlukan perencanaan yang terstruktur dan matang. Penelitian ini menggunakan metode pengembangan D&D (*Desain and Development*). Metode ini menghasilkan luaran berupa artefak yang dapat berupa pengembangan alat baru, produk, atau sebuah proses dan menggunakan penelitian dalam prosesnya (Ellis & Levy, 2010). Peneliti menggunakan langkah untuk membuat sistem pada penelitian ini digunakan seperti pada Gambar 3.1.



Gambar 3.1 Langkah Metode Penelitian

3.1.1. Analisis Kebutuhan Sistem

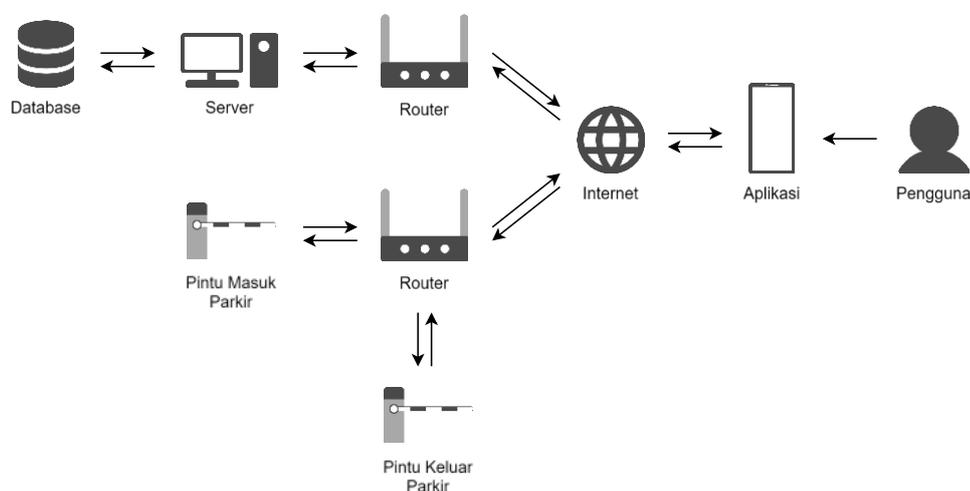
Pada tahap analisis kebutuhan sistem, dilakukan identifikasi terhadap kebutuhan dan fungsionalitas yang harus ada didalam sistem. Sistem ini membutuhkan komponen utama untuk aplikasi Android sebagai pemesanan tempat parkir untuk pengguna, server untuk menyimpan dan mengelola data pada sistem ini, serta perangkat keras yang dilengkapi dengan kamera untuk mendeteksi dan mengklasifikasikan karakter pada plat nomor kendaraan sebagai pintu parkir.

Aplikasi Android yang dibangun akan mengintegrasikan ke internet agar bisa menerima dan memanggil informasi pengguna maupun tempat parkir yang disimpan di server. Fitur utama yang akan ada dalam aplikasi yang dibangun, yaitu pendaftaran akun baru pengguna, *login* akun pengguna yang sudah terdaftar, melihat informasi data pengguna, mengisi saldo pengguna, mengganti data pengguna, pemesanan tempat parkir, pembayaran parkir, pembatalan pemesanan parkir, melihat riwayat parkir, dan keluar dari akun pengguna yang sudah berhasil *login*.

Server harus mampu menyimpan informasi pengguna dan pemesanan, serta menyediakan jalur untuk komunikasi dengan aplikasi dan pintu parkir. Pintu parkir harus mampu untuk mendeteksi plat nomor dan mengenali karakter pada plat nomor. Selain itu, sistem harus dapat membuka pintu parkir secara otomatis berdasarkan hasil deteksi dan klasifikasi plat nomor yang kemudian akan divalidasi ke *database* server.

3.1.2. Perancangan Sistem

Berdasarkan analisis kebutuhan, sistem terdiri dari tiga bagian utama: aplikasi, server, dan pintu parkir dapat dibuat sebuah arsitektur keseluruhan sistem pada Gambar 3.2.



Gambar 3.2 Arsitektur Keseluruhan Sistem

Arsitektur sistem pemesanan parkir berdasarkan plat nomor ini terdiri dari beberapa komponen utama yang terintegrasi, memastikan kelancaran dan keamanan dalam pengelolaan parkir. Pengguna mengakses sistem melalui aplikasi pada perangkat *mobile*, yang berkomunikasi dengan server melalui internet untuk mengirim dan menerima data. Server bertindak sebagai pusat pengolahan data, berinteraksi dengan *database* untuk mengelola informasi pengguna dan transaksi parkir, serta mengontrol akses ke pintu masuk dan keluar parkir menggunakan sistem pengenalan plat nomor. Pintu parkir memeriksa kendaraan berdasarkan plat nomor terdaftar yang telah diverifikasi untuk mengakses dan meninggalkan area parkir, dengan semua data diproses melalui server dan internet.

3.1.2.1. Perancangan Model YOLO dan CNN

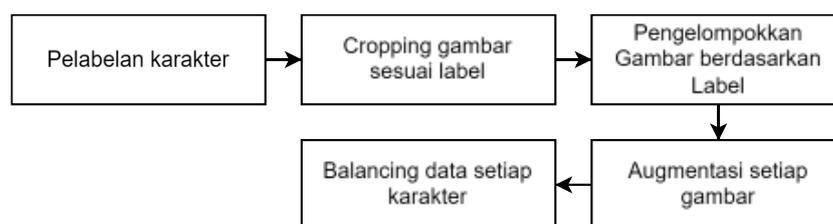
Perancangan model pada penelitian ini melibatkan pelatihan menggunakan model YOLO sebagai mendeteksi plat nomor dan kemudian CNN akan mengekstrak karakter yang berada di plat nomor menjadi sebuah tulisan. Perancangan ini memerlukan beberapa tahapan yaitu:

1. Pengumpulan Data

Data yang akan digunakan untuk penelitian ini dilakukan dengan pengumpulan dataset yang dilakukan dengan membagi data menjadi dua kategori utama, yaitu dataset untuk plat nomor dan dataset untuk karakter pada plat nomor. Pada dataset plat nomor ini diperoleh dari situs open source Roboflow dengan nama “ANPR 2”. Dataset ini berisikan gambar plat nomor Indonesia yang sudah terlabeli. Setiap gambar dalam dataset ini telah melalui anotasi untuk menandai area plat nomor, sehingga memudahkan untuk melakukan pelatihan model. Sedangkan untuk dataset karakter plat nomor, data dikumpulkan dari dua sumber utama. Gambar diperoleh dari tangkapan gambar sendiri menggunakan kamera dari *smartphone*, dan diambil dari foto-foto yang beredar di internet.

2. Pengolahan Data

Setelah pengumpulan data selesai, dataset karakter plat nomor memerlukan proses pengolahan data. Pengolahan data ini diperlukan beberapa tahap pengolahan data seperti pada Gambar 3.3.



Gambar 3.3 Pengolahan Data Karakter Plat Nomor

Tahap pertama yang dilakukan pada pengolahan data karakter plat nomor adalah melakukan label setiap karakter pada plat nomor dalam gambar. Pelabelan dilakukan pada *website* Roboflow yang menghasilkan 36 kelas yang berisi karakter alfanumerik dengan angka 0-9 dan huruf A-Z. Setelah dilakukan pelabelan, dataset akan diolah kedalam berbentuk file zip.

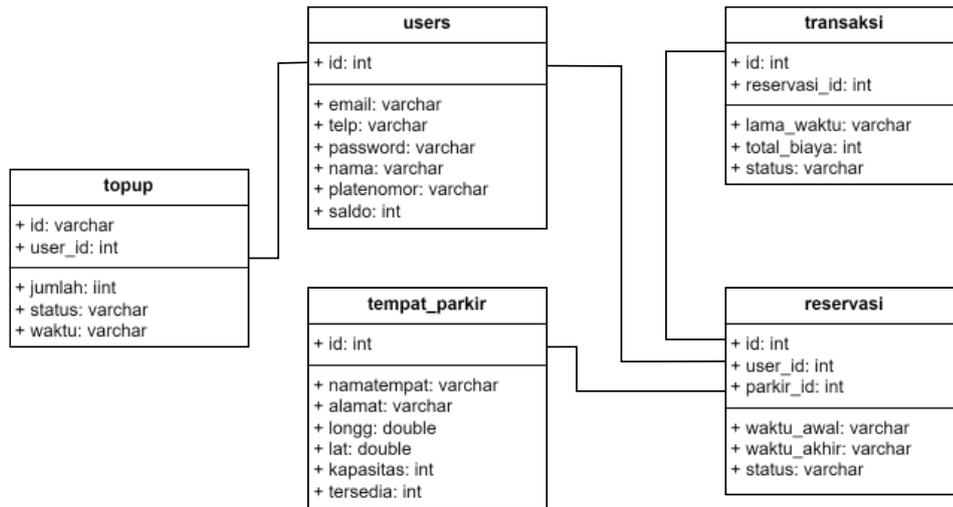
Dari file tersebut akan diekstrak kedalam sebuah folder yang berisi gambar dan menyimpan informasi hasil label setiap gambarnya dalam format *Text Document*. Kemudian dilakukan pengolahan pada setiap karakter pada plat nomor dipotong (*cropping*) dari gambar asli, menghasilkan gambar-gambar kecil yang masing-masing hanya berisi satu karakter disetiap gambarnya. Gambar-gambar ini kemudian dikelompokkan berdasarkan label karakternya.

Untuk meningkatkan jumlah dan variasi data, dilakukan beberapa teknik augmentasi. Teknik augmentasi data digunakan untuk memperkaya dataset dan meningkatkan kemampuan model dalam mengenali berbagai perspektif dari data yang diberikan (Sanjaya & Ayub, 2020). Setiap gambar akan menjalani augmentasi tersebut, yang pada akhirnya menambahkan gambar baru ke dataset untuk setiap gambar asli.

Setelah augmentasi, dilakukan *balancing* untuk memastikan bahwa setiap kelas karakter memiliki jumlah data yang sama. Proses ini dilakukan dengan mengambil data secara acak untuk setiap kelas karakter, sehingga distribusi data tetap seimbang dan tidak ada kelas yang mendominasi atau terpinggirkan dalam pelatihan model. Jumlah data untuk setiap kelas diratakan berdasarkan jumlah data dari kelas dengan karakter yang paling sedikit, sehingga setiap kelas memiliki jumlah data yang sama sesuai dengan kelas yang paling sedikit tersebut untuk setiap karakternya. Selanjutnya, gambar akan diubah menjadi biner dengan menggunakan OpenCV dan metode *threshold* untuk menghasilkan gambar biner hitam putih.

3.1.2.2. Perancangan Server pada Komputer

Perancangan server dilakukan dengan mempertimbangkan kebutuhan aplikasi dan pintu parkir. Server akan diimplementasikan menggunakan komputer yang berfungsi sebagai server utama. Pada komputer yang dijadikan *server* akan menyimpan *database* menggunakan MySQL. *Database* yang dirancang pada sistem ini seperti pada Gambar 3.4.



Gambar 3.4 Database Class Diagram

Berdasarkan dari Gambar 3.4, *database* akan digunakan sebagai penyimpanan informasi data pengguna dan tempat parkir. Supaya *database* tersebut dapat diakses oleh pengguna melalui aplikasi Android, server harus dihubungkan terlebih dahulu ke internet. Server akan menggunakan metode *tunneling* untuk bisa terhubung ke internet. Untuk menerima permintaan dari pengguna melalui aplikasi Android, server harus menyediakan beberapa *endpoint* API untuk bisa menerima dengan baik dan bisa mengirimkan kembali pesan yang sesuai. Berdasarkan dari analisis kebutuhan dan perancangan aplikasi android, server membutuhkan *endpoint* seperti pada Tabel 3.1 untuk memenuhi kebutuhan server.

Tabel 3.1
Perancangan *Endpoint* API

No	Proses	API	
		Method	endpoint
1	Membuat akun baru pengguna	POST	/signup
2	Masuk melalui akun pengguna	POST	/login
3	Mendapatkan data pengguna	GET	/user/<user_id>
4	Merubah data pengguna	PUT	/user/<user_id>
5	Mengganti <i>password</i> pengguna	PUT	/user/changePassword
6	Melakukan <i>topup</i> saldo pengguna	POST	/user/topUp

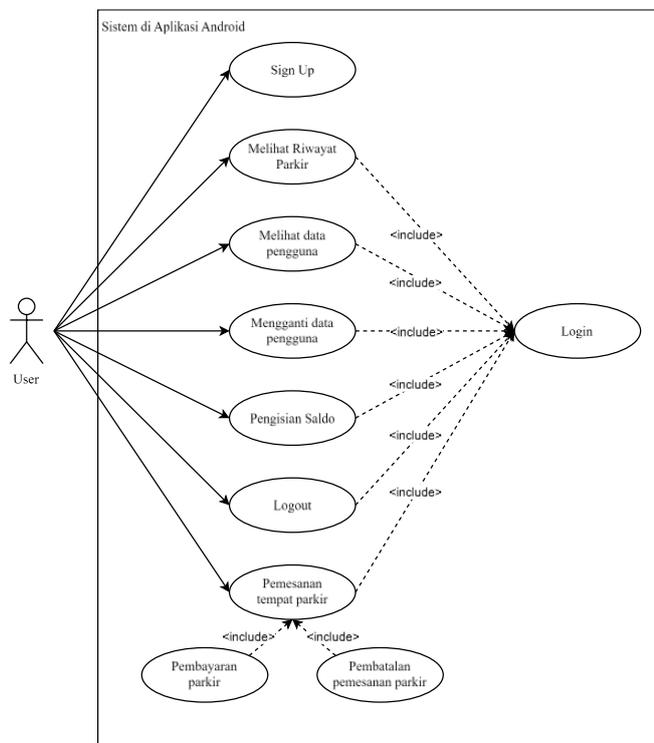
No	Proses	API	
		Method	endpoint
7	Mendapatkan status transaksi topup	GET	/transactions/<user_id>
8	Melihat informasi semua tempat parkir	GET	/tempat_parkir
9	Mendapatkan detail informasi setiap tempat parkir	GET	/tempat_parkir/ <parkir_id>
10	Melakukan pemesanan tempat parkir	POST	/reservasi
11	Membatalkan pemesanan aktif	PUT	/reservasi/cancel <user_id>
12	Mendapatkan status reservasi terbaru pengguna	GET	/reservasi/latest
13	Mendapatkan daftar reservasi aktif pengguna	GET	/reservasi/ongoing/users/ <user_id>
14	Mendapatkan daftar reservasi selesai pengguna	GET	/reservasi/finished/users/ <user_id>
15	Melakukan <i>check in</i> berdasarkan plat nomor	POST	/reservasi/checkIn
16	Melakukan pembayaran parkir	POST	/reservasi/payment
17	Melakukan <i>check out</i> berdasarkan plat nomor	POST	/reservasi/checkOut

Endpoint API pada Tabel 3.1 mencakup fungsi-fungsi utama yang harus didukung oleh server, seperti pendaftaran pengguna, *login*, reservasi tempat parkir, pembayaran, dan lain-lain. *Endpoint* ini dirancang untuk memberikan akses ke *database* yang terstruktur ke berbagai fungsi sistem bagi aplikasi dan alat pintu parkir.

3.1.2.3. Perancangan Aplikasi Android

Setelah menganalisis kebutuhan aplikasi yang ingin dibangun, langkah selanjutnya adalah proses desain aplikasi. Dalam tahap perancangan ini, hasil dari analisis kebutuhan diolah menjadi pemodelan menggunakan *Unified Modeling Language* (UML). *Use case diagram*, salah satu bentuk pemodelan UML, sebuah interaksi antara sistem dan aktor yang memiliki alur tertentu yang akan

diimplementasikan dalam sistem yang sedang dikembangkan (Sutrisno & Karnadi, 2021). *Use case diagram* pada aplikasi yang akan dibangun seperti pada Gambar 3.5.



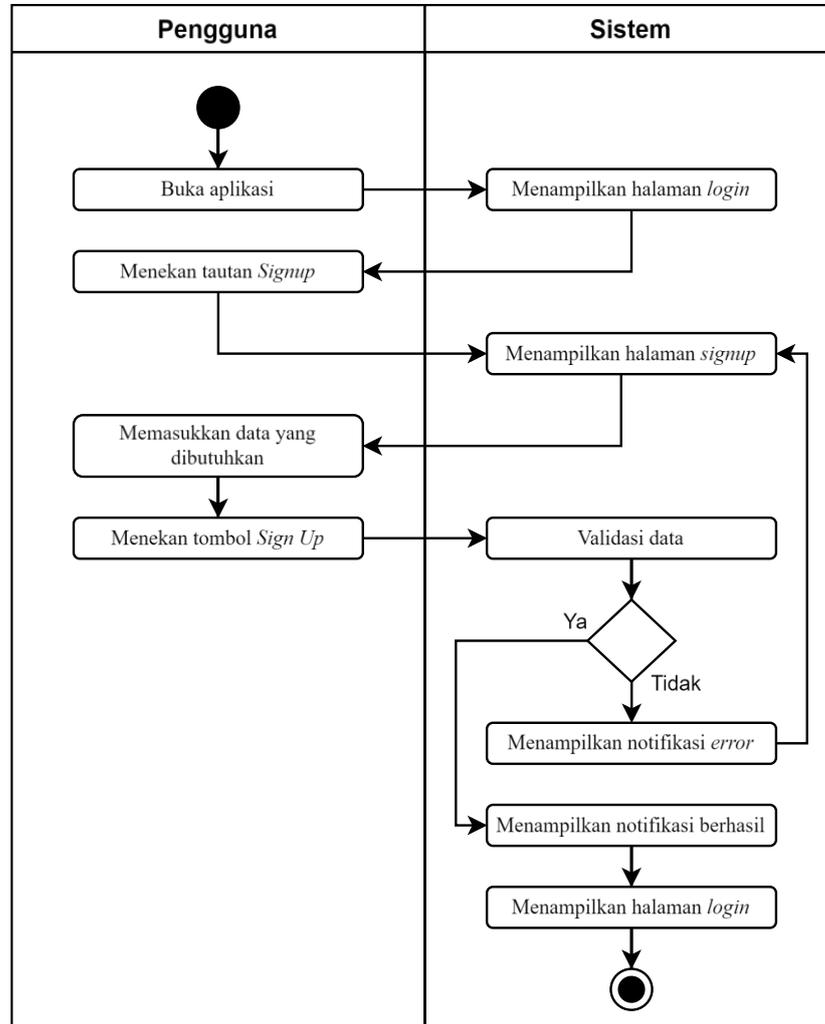
Gambar 3.5 *Use Case Diagram* Aplikasi Pemesanan Parkir

Gambar 3.5 merupakan diagram *use case* yang menggambarkan interaksi pengguna dengan aplikasi Android untuk pemesanan parkir. Diagram ini menampilkan berbagai fungsi utama yang tersedia dalam aplikasi, termasuk *sign up* untuk pendaftaran pengguna baru, *login* yang memungkinkan pengguna mengakses akun pengguna, dan berbagai aktivitas lain yang dapat dilakukan setelah masuk, seperti melihat riwayat parkir, melihat data pengguna, mengganti data pengguna, pengisian saldo, dan *logout*. Fitur-fitur tambahan yang tersedia setelah *login* mencakup pemesanan tempat parkir, pembayaran parkir, dan pembatalan pemesanan parkir. Setiap aksi ini terhubung dengan proses *login* yang menunjukkan bahwa akses ke fitur-fitur tersebut memerlukan pengguna untuk terlebih dahulu masuk ke dalam sistem.

Berdasarkan fitur-fitur yang ada pada *use case diagram*, terdapat *activity diagram* yang menunjukkan proses setiap fitur, yaitu:

1. Activity Diagram Signup

Activity diagram ini menggambarkan alur proses pendaftaran pengguna baru seperti pada Gambar 3.6.

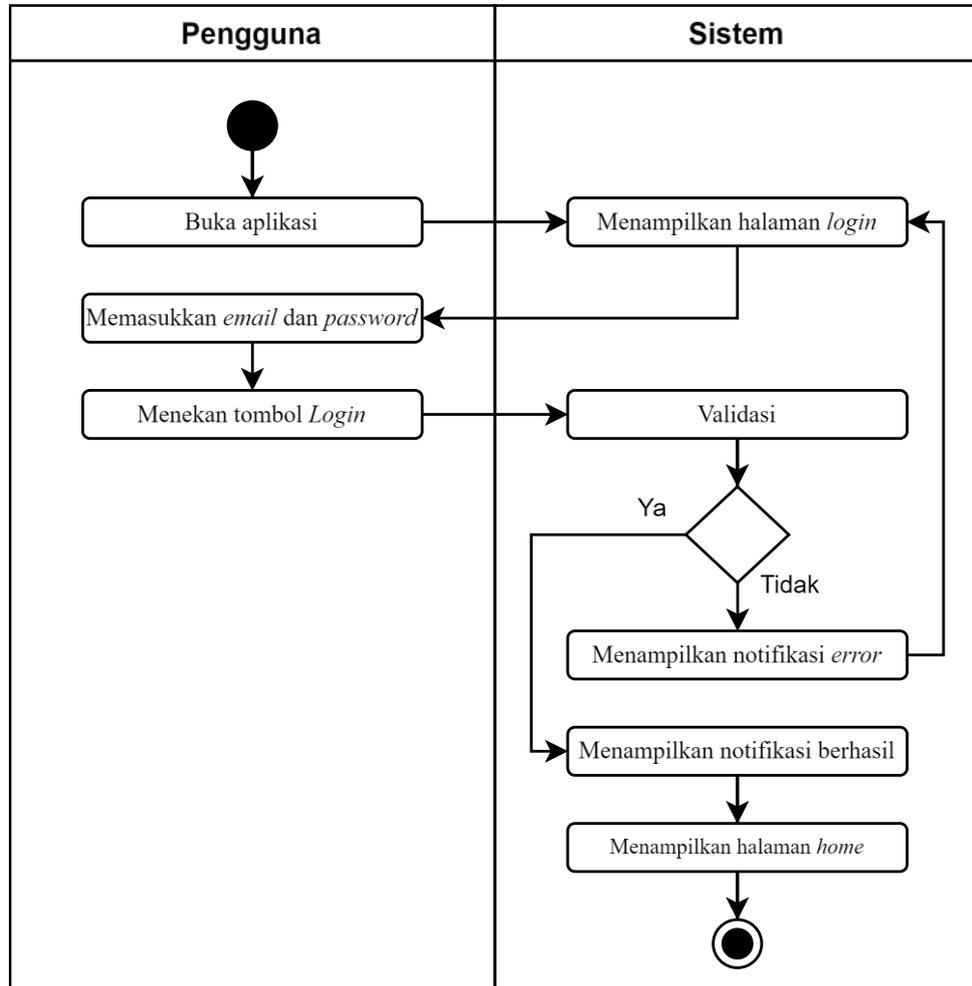


Gambar 3.6 Activity Diagram Signup

Activity diagram pada Gambar 3.6 merupakan proses pendaftaran pengguna dalam sebuah aplikasi. Awalnya, pengguna membuka aplikasi dan diarahkan ke halaman Login. Pengguna dapat menekan tautan Signup yang membawanya ke halaman pendaftaran. Di sana, pengguna memasukkan data yang dibutuhkan dan menekan tombol Sign Up. Data tersebut kemudian divalidasi oleh sistem. Jika validasi berhasil, sistem menampilkan notifikasi keberhasilan dan mengarahkan pengguna kembali ke halaman Login. Jika terjadi kesalahan, sistem akan menampilkan notifikasi *error*.

2. Activity Diagram Login

Activity diagram ini menggambarkan alur proses *login* pengguna ke dalam aplikasi seperti pada Gambar 3.7.

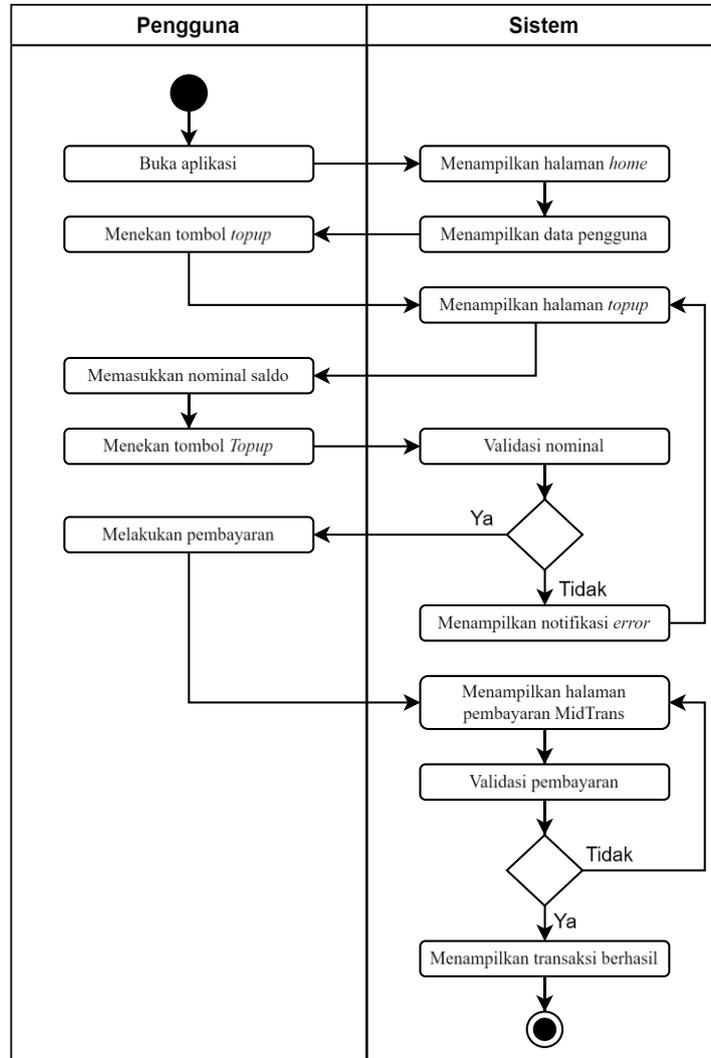


Gambar 3.7 Activity Diagram Login

Activity diagram pada Gambar 3.7 menjelaskan proses login pengguna dalam sebuah aplikasi. Mulai dari membuka aplikasi, pengguna dihadapkan pada halaman Login di mana mereka diminta untuk memasukkan *email* dan *password*. Setelah data dimasukkan, pengguna menekan tombol Login untuk proses validasi. Jika informasi yang dimasukkan benar, sistem akan menampilkan notifikasi berhasil dan pengguna akan diarahkan ke halaman Home aplikasi. Sebaliknya, jika terjadi kesalahan dalam validasi, sistem akan menampilkan notifikasi *error*, dan pengguna tidak akan dipindahkan dari halaman login.

3. Activity Diagram Pengisian Saldo

Activity diagram ini menggambarkan alur proses penambahan saldo akun pengguna baru seperti pada Gambar 3.8.



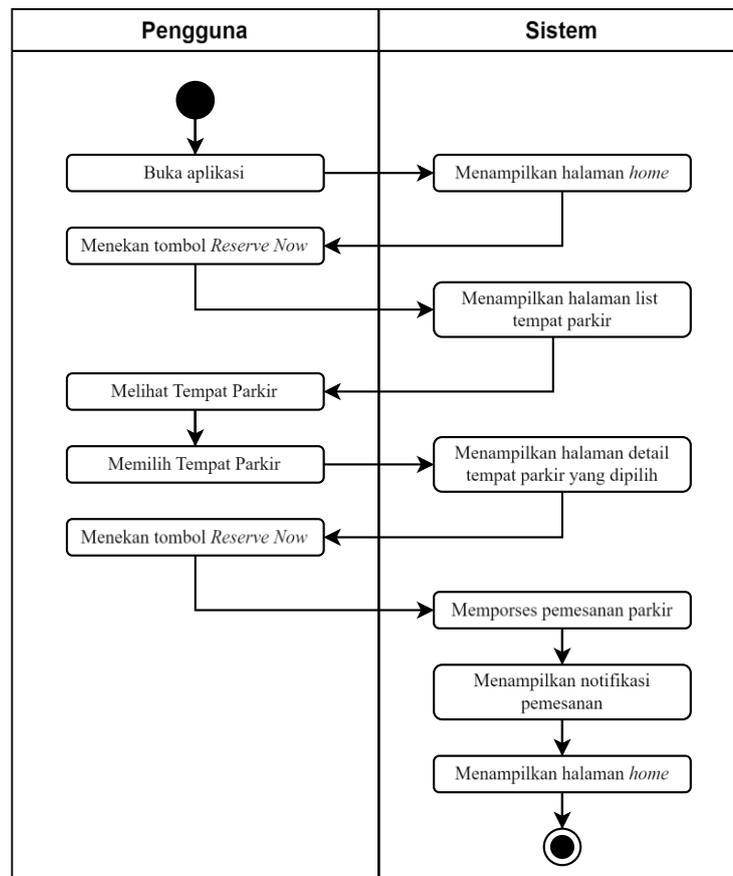
Gambar 3.8 Activity Diagram Pengisian Saldo

Activity diagram pada Gambar 3.8 menjelaskan proses pengisian saldo atau *top up* di aplikasi. Prosesnya dimulai ketika pengguna membuka aplikasi dan langsung menuju halaman Home. Setelah itu, pengguna menekan tombol Top Up yang akan membawa mereka ke halaman Top Up untuk menampilkan data pengguna dan opsi pengisian saldo. Pengguna memasukkan nominal saldo yang diinginkan dan menekan tombol Top Up untuk proses validasi nominal. Jika nominal valid, pengguna akan dibawa ke halaman pembayaran, di mana mereka

melakukan pembayaran menggunakan metode yang tersedia. Sistem melakukan validasi pembayaran, dan jika berhasil, pengguna akan mendapatkan konfirmasi transaksi berhasil. Jika ada kesalahan pada tahap validasi nominal atau pembayaran, sistem akan menampilkan notifikasi *error*.

4. Activity Diagram Pemesanan Tempat Parkir

Activity diagram ini menggambarkan alur proses pemesanan parkir pengguna seperti pada Gambar 3.9.



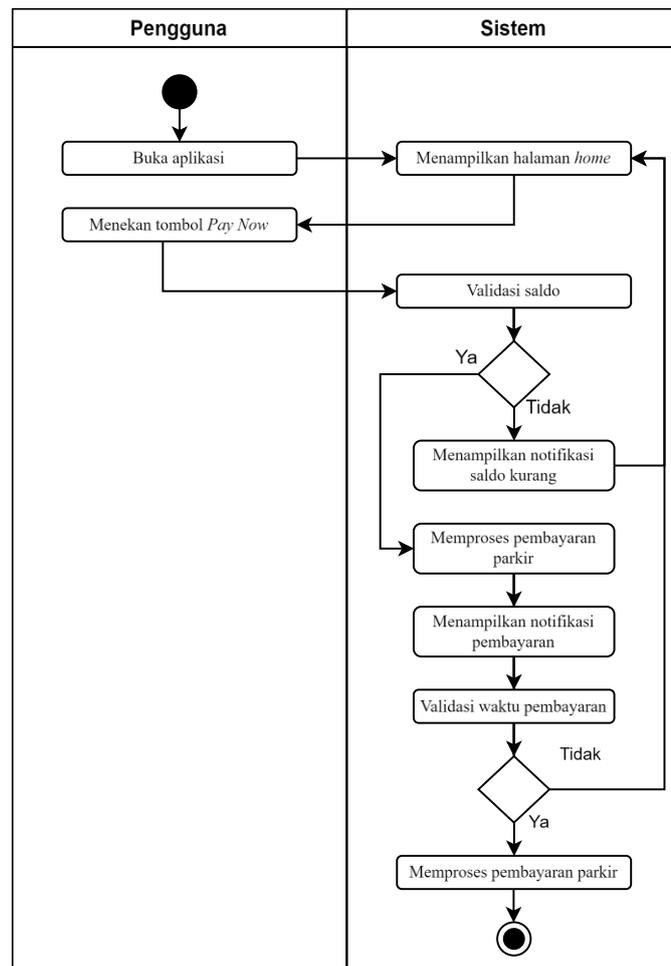
Gambar 3.9 Activity Diagram Pemesanan Tempat Parkir

Activity diagram pada Gambar 3.9 menggambarkan proses pemesanan tempat parkir dalam aplikasi. Berawal dari pengguna yang membuka aplikasi dan masuk ke halaman Home. Selanjutnya, pengguna menekan tombol Reserve Now yang mengarahkan ke halaman yang menampilkan daftar tempat parkir. Pengguna kemudian memilih tempat parkir yang diinginkan dan sistem akan menampilkan halaman detail

tempat parkir tersebut. Setelah memilih tempat parkir, pengguna menekan kembali tombol Reserve Now untuk memulai proses pemesanan. Sistem kemudian memproses pemesanan, jika berhasil, menampilkan notifikasi bahwa pemesanan telah berhasil. Akhirnya, sistem akan mengarahkan pengguna kembali ke halaman Home.

5. Activity Diagram Pembayaran Parkir

Activity diagram ini menggambarkan alur proses pembayaran dari pemesanan parkir pengguna seperti pada Gambar 3.10.



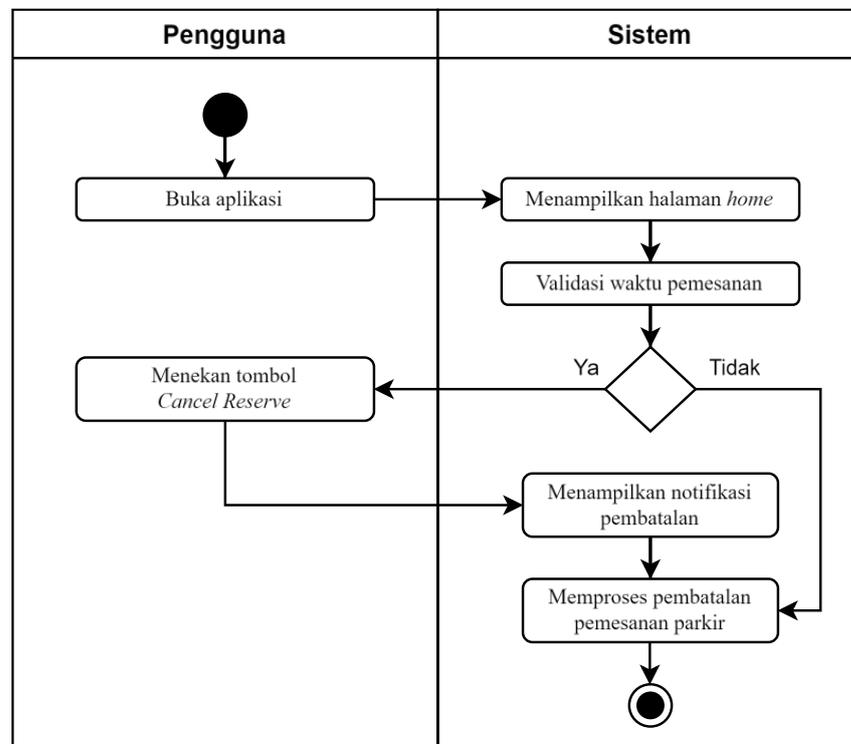
Gambar 3.10 Activity Diagram Pembayaran Parkir

Activity diagram pada Gambar 3.10 menggambarkan proses pembayaran parkir melalui aplikasi. Pengguna memulai dengan membuka aplikasi dan menavigasi ke halaman Home. Setelah itu, pengguna memilih opsi Pay Now untuk memulai proses pembayaran. Sistem pertama-tama melakukan validasi saldo pengguna. Jika saldo tidak mencukupi, sistem

akan menampilkan notifikasi yang menyatakan saldo tidak cukup. Jika saldo cukup, sistem melanjutkan dengan memproses pembayaran parkir dan menampilkan notifikasi bahwa pembayaran telah berhasil. Selanjutnya, sistem melakukan validasi terhadap waktu pembayaran, memastikan transaksi dilakukan dalam waktu yang sesuai. Jika validasi ini berhasil, sistem secara resmi menyelesaikan proses pembayaran parkir dan membawa pengguna kembali ke halaman utama.

6. Activity Diagram Pembatalan Pemesanan Parkir

Activity diagram ini menggambarkan alur proses pembatalan dari pemesanan parkir pengguna seperti pada Gambar 3.11.



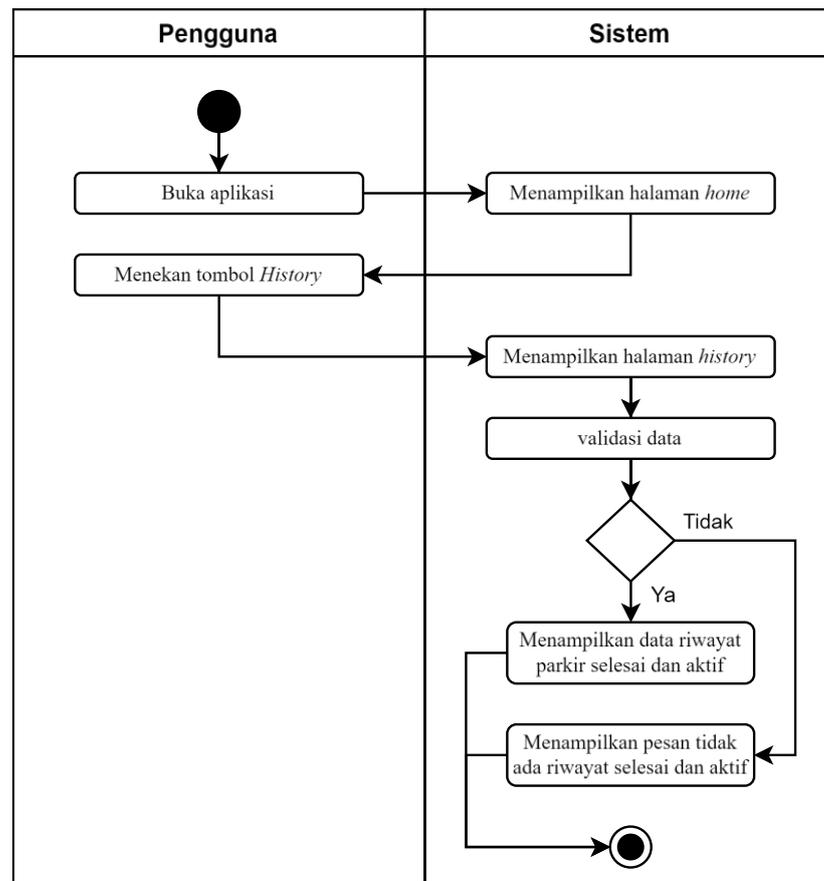
Gambar 3.11 Activity Diagram Pembatalan Pemesanan Parkir

Activity diagram pada Gambar 3.11 menggambarkan proses pembatalan pemesanan tempat parkir melalui aplikasi. Pengguna membuka aplikasi dan diarahkan ke halaman utama. Dari sini, pengguna dapat memilih untuk membatalkan reservasi yang telah dibuat dengan menekan tombol Cancel Reserve. Sistem kemudian melakukan validasi terhadap waktu pemesanan untuk menentukan apakah pembatalan masih dapat dilakukan. Jika memenuhi kriteria, sistem akan memproses

pembatalan pemesanan dan menampilkan notifikasi berhasil kepada pengguna. Jika tidak, sistem mungkin menampilkan pemberitahuan bahwa pemesanan tidak dapat dibatalkan karena melampaui batas waktu yang ditentukan oleh sistem.

7. Activity Diagram Melihat Riwayat Parkir

Activity diagram ini menggambarkan alur proses melihat riwayat pemesanan parkir pengguna yang sedang berlangsung ataupun yang sudah selesai seperti pada Gambar 3.12.



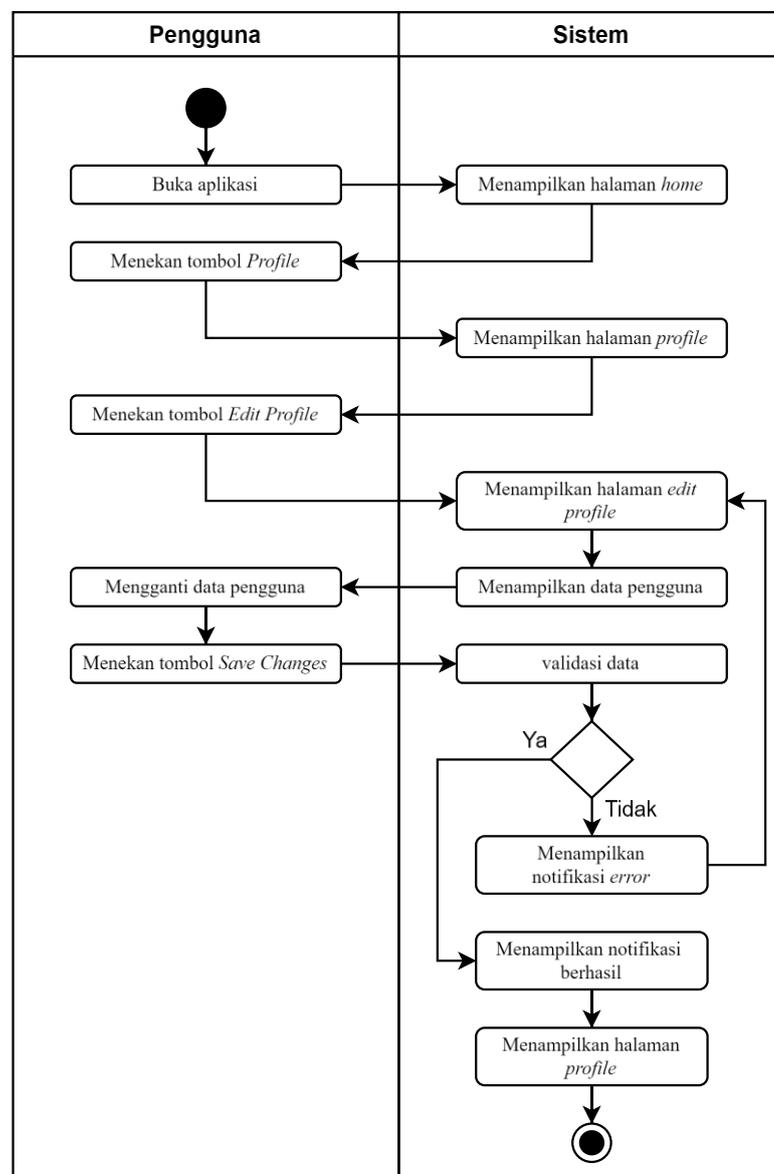
Gambar 3.12 Activity Diagram Melihat Riwayat Parkir

Activity diagram pada Gambar 3.12 menggambarkan alur interaksi pengguna dengan fitur History pada aplikasi. Proses ini dimulai saat pengguna membuka aplikasi dan diarahkan ke halaman Home. Pengguna kemudian menekan tombol History, yang mengarahkan mereka ke halaman riwayat parkir. Sistem akan memvalidasi data yang tersedia, jika ada data riwayat parkir yang aktif atau yang telah selesai, data tersebut

akan ditampilkan kepada pengguna. Jika tidak ada riwayat parkir yang tersedia, sistem akan menampilkan pesan yang menyatakan bahwa tidak ada riwayat parkir yang tersedia. Ini memastikan bahwa pengguna diberikan informasi yang relevan mengenai aktivitas parkir mereka sebelumnya.

8. Activity Diagram Mengganti Data Pengguna

Activity diagram ini menggambarkan alur proses perubahan informasi pada data pengguna seperti pada Gambar 3.13.

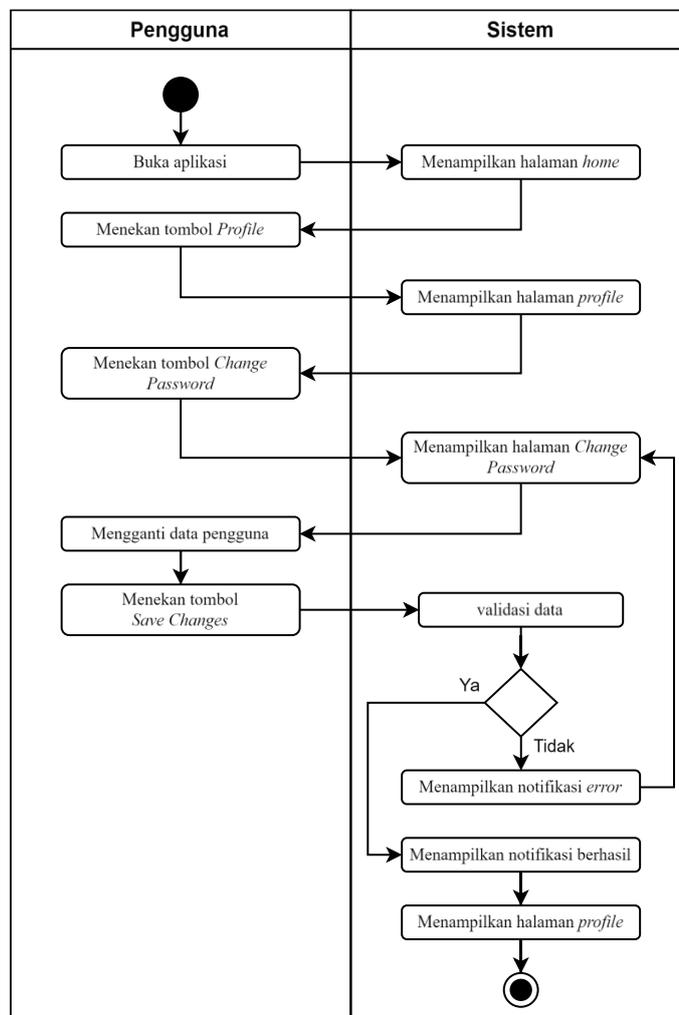


Gambar 3.13 Activity Diagram Mengganti Data Pengguna

Activity diagram pada Gambar 3.13 menggambarkan proses pengeditan profil pengguna dalam aplikasi. Setelah membuka aplikasi dan mengakses halaman Profil, pengguna bisa memilih untuk mengedit profilnya. Setelah melakukan perubahan, sistem akan memvalidasi data tersebut. Jika data valid, pengguna akan mendapatkan notifikasi sukses dan kembali ke halaman Profil yang diperbarui. Jika tidak, sistem akan menampilkan notifikasi *error*, meminta pengguna untuk mengoreksi data sebelum menyimpan kembali.

9. *Activity Diagram* Mengganti Password

Activity diagram ini menggambarkan alur proses pendaftaran pengguna baru seperti pada .

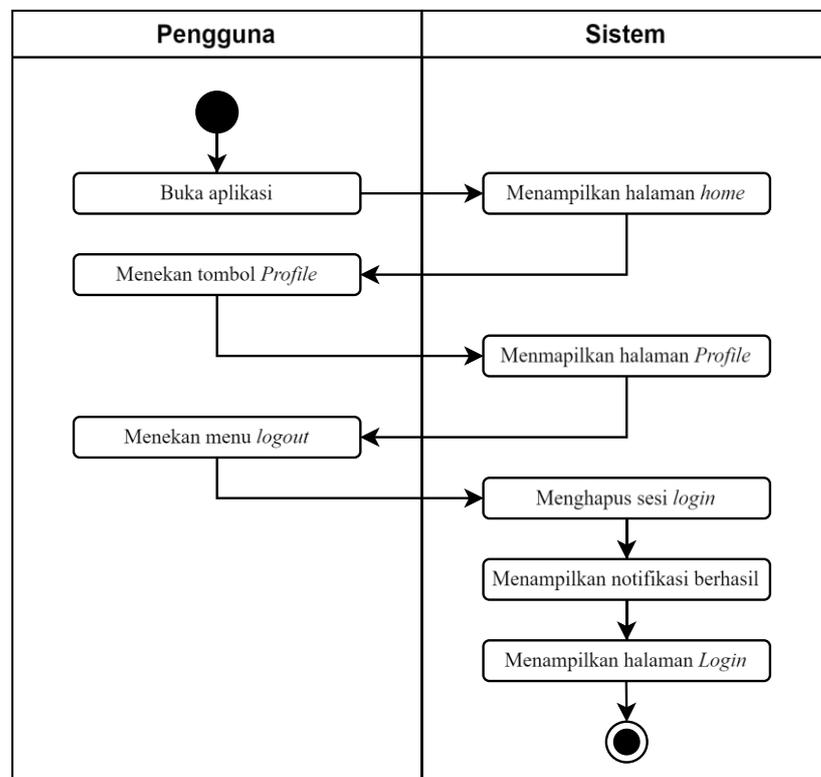


Gambar 3.14 *Activity Diagram* Mengganti Password

Activity diagram pada Gambar 3.14 menggambarkan proses penggantian *password* pengguna dalam aplikasi. Pengguna membuka aplikasi, mengakses halaman Profil, dan memilih opsi untuk mengubah *password*. Setelah memasukkan data baru dan menekan Save Changes, sistem melakukan validasi. Jika validasi berhasil, notifikasi sukses ditampilkan dan pengguna kembali ke halaman profil. Jika tidak, notifikasi *error* akan muncul.

10. Activity Diagram Logout

Activity diagram ini menggambarkan alur proses *logout* akun pengguna pada aplikasi seperti pada .



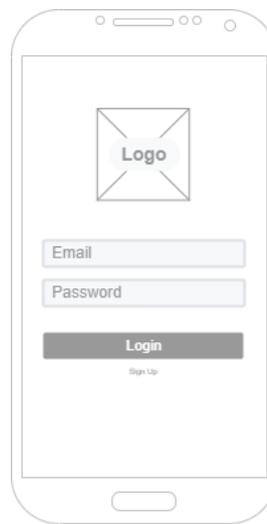
Gambar 3.15 Activity Diagram Logout

Activity diagram pada Gambar 3.15 menunjukkan proses *logout* pengguna dari aplikasi. Pengguna membuka aplikasi, mengakses halaman Profil, dan memilih opsi *logout*. Sistem kemudian menghapus sesi *login*, memberikan notifikasi bahwa *logout* berhasil, dan mengarahkan pengguna kembali ke halaman Login.

Berdasarkan hasil dari perancangan aplikasi Android dengan menggunakan pemodelan UML, dibutuhkan desain *user interface* untuk mendukung berbagai fungsionalitas dalam aplikasi. Halaman-halaman tersebut termasuk:

1. Desain *User Interface* Login

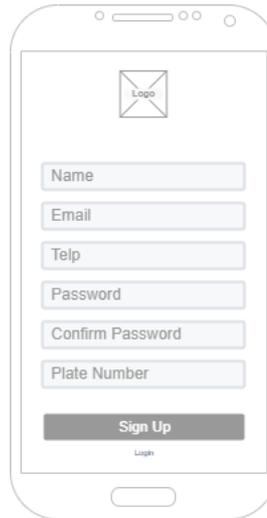
Halaman Login merupakan tempat bagi pengguna untuk memasukkan *email* dan kata sandi akun pengguna yang sudah terdaftar agar bisa masuk ke dalam aplikasi. Halaman ini dirancang seperti pada Gambar 3.16.



Gambar 3.16 Desain *User Interface* Login

2. Desain *User Interface* Signup

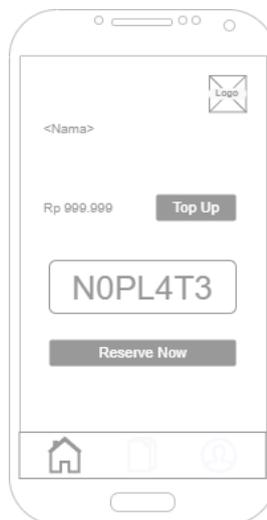
Halaman Signup merupakan tempat bagi pengguna baru untuk membuat akun dengan memasukkan informasi yang diperlukan agar terdaftar ke dalam sistem. Halaman ini dirancang seperti pada Gambar 3.17.



Gambar 3.17 Desain *User Interface* Signup

3. Desain *User Interface* Home

Halaman *Home* adalah halaman utama setelah pengguna berhasil login. Halaman ini menampilkan ringkasan informasi pengguna dan navigasi utama aplikasi. Tampilan halaman Home dirancang seperti pada Gambar 3.18.



Gambar 3.18 Desain *User Interface* Home

4. Desain *User Interface* Topup

Halaman *Topup* memungkinkan pengguna untuk menambah saldo akun pengguna dengan menggunakan metode pembayaran Midtrans. Tampilan halaman *Topup* dirancang seperti pada Gambar 3.19.



Gambar 3.19 Desain *User Interface* Topup

5. Desain *User Interface* List Tempat Parkir

Halaman List Tempat Parkir menampilkan daftar tempat parkir dan menampilkan informasi dari setiap tempat parkir. Tampilan halaman List Tempat Parkir dirancang seperti pada Gambar 3.20.



Gambar 3.20 Desain *User Interface* List Tempat Parkir

6. Desain *User Interface* Detail Tempat Parkir

Halaman Detail Tempat Parkir menampilkan informasi lebih lanjut mengenai tempat parkir yang dipilih dan menampilkan lokasi tempat

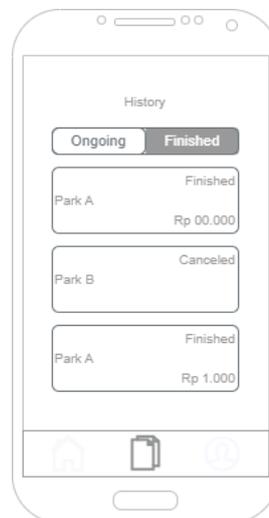
parkir. Tampilan halaman Detail Tempat Parkir dirancang seperti pada Gambar 3.21.



Gambar 3.21 Desain *User Interface* Detail Tempat Parkir

7. Desain *User Interface* History

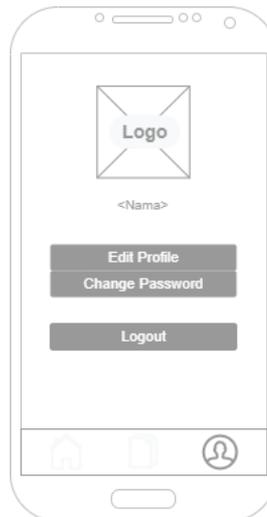
Halaman *history* menampilkan riwayat pemesanan pengguna yang sedang berlangsung maupun yang sudah selesai. Tampilan halaman *History* dirancang seperti pada Gambar 3.22.



Gambar 3.22 Desain *User Interface* History

8. Desain *User Interface* Profile

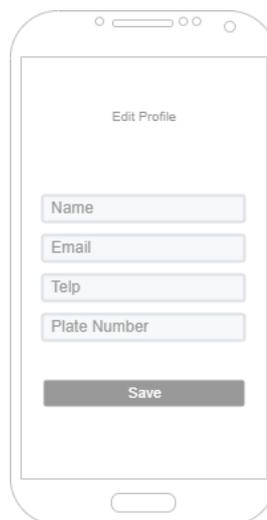
Halaman Profile terdapat beberapa tombol yang memungkinkan pengguna untuk memilih beberapa menu dan *logout* dari akun pengguna. Tampilan halaman Profile dirancang seperti pada Gambar 3.23.



Gambar 3.23 Desain *User Interface* Profile

9. Desain *User Interface* Edit Data Pengguna

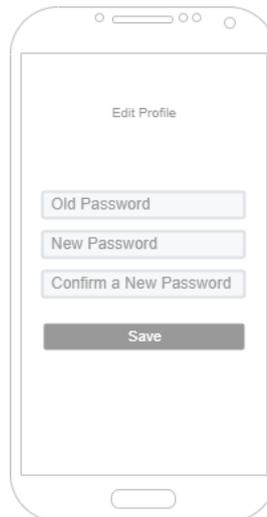
Halaman Edit Data Pengguna memungkinkan pengguna untuk mengedit informasi pribadi mereka. Tampilan halaman Edit Data Pengguna dirancang seperti pada Gambar 3.24.



Gambar 3.24 Desain *User Interface* Edit Data Pengguna

10. Desain *User Interface* Change Password

Halaman *Change Password* memungkinkan pengguna untuk mengubah kata sandi akun pengguna. Tampilan halaman *Change Password* dirancang seperti pada Gambar 3.25.



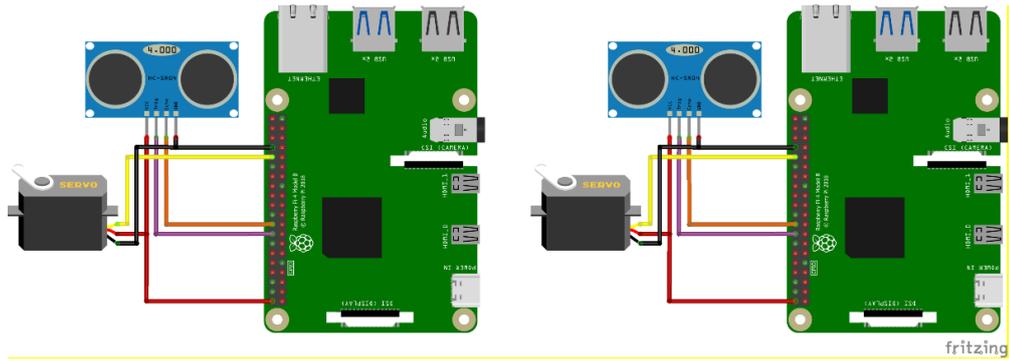
Gambar 3.25 Desain *User Interface* Change Password

3.1.2.4. Perancangan Prototipe Pintu Masuk dan Keluar Parkir

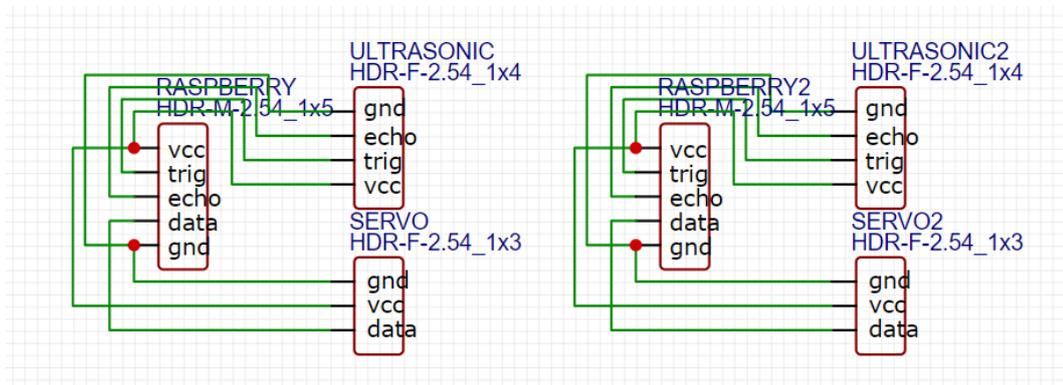
Perangkat keras dalam sistem ini menggunakan Raspberry Pi sebagai unit pemrosesan utama di pintu parkir masuk maupun keluar. Raspberry Pi bertugas menjalankan model deteksi objek, mengekstraksi tulisan pada plat nomor, serta berkomunikasi dengan server. Perangkat keras ini dibuat sebagai alat pengujian yang terintegrasi dengan beberapa komponen utama seperti kamera, servo, dan sensor ultrasonik.

Kamera akan menggunakan sebuah *webcam* yang berfungsi untuk menangkap gambar plat nomor kendaraan. Gambar tersebut akan diproses oleh Raspberry Pi untuk pendeteksian plat dan pengenalan karakter. Servo digunakan untuk membuka palang pintu parkir, memberikan akses kendaraan untuk masuk atau keluar. Sensor ultrasonik berfungsi sebagai pendeteksi keberadaan kendaraan di pintu parkir. Untuk memberikan gambaran yang lebih jelas mengenai koneksi antar komponen, wiring diagram disajikan pada Gambar 3.26, yang memperlihatkan hubungan antara Raspberry Pi, *webcam*, servo, dan ultrasonik.

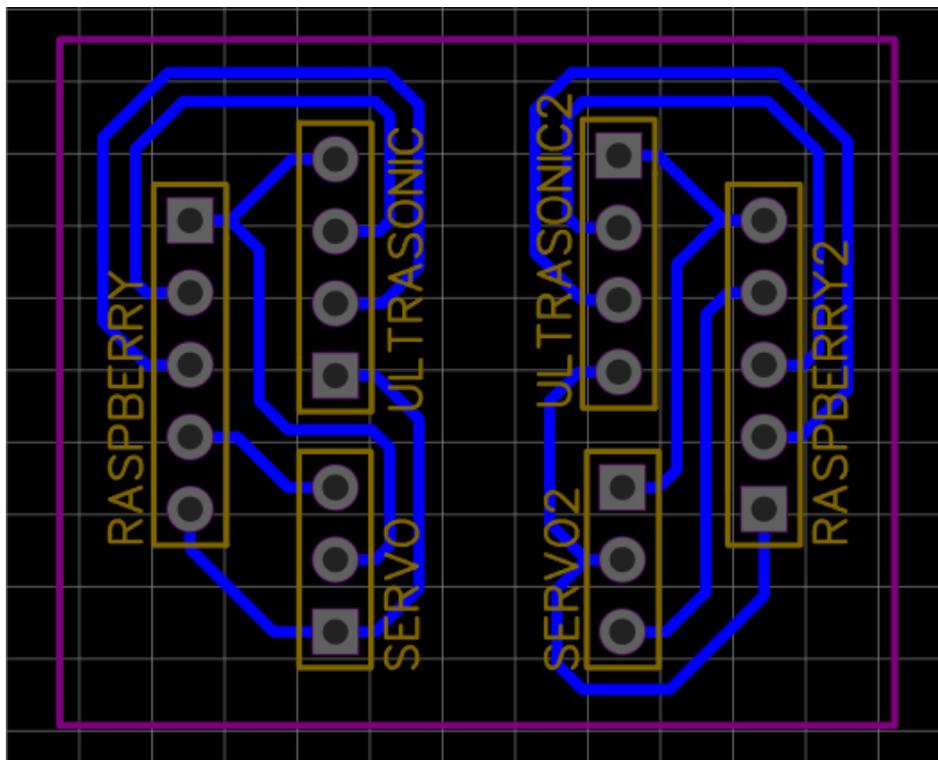
Selain *wiring diagram*, *Printed Circuit Board* (PCB) digunakan untuk menyederhanakan rangkaian dan merapikan koneksi antar komponen. Penggunaan PCB ini akan dijelaskan lebih lanjut melalui *schematic diagram* pada Gambar 3.27 dan desain PCB pada Gambar 3.28.



Gambar 3.26 Wiring Diagram Prototipe Pintu Masuk dan Pintu Keluar Pakir

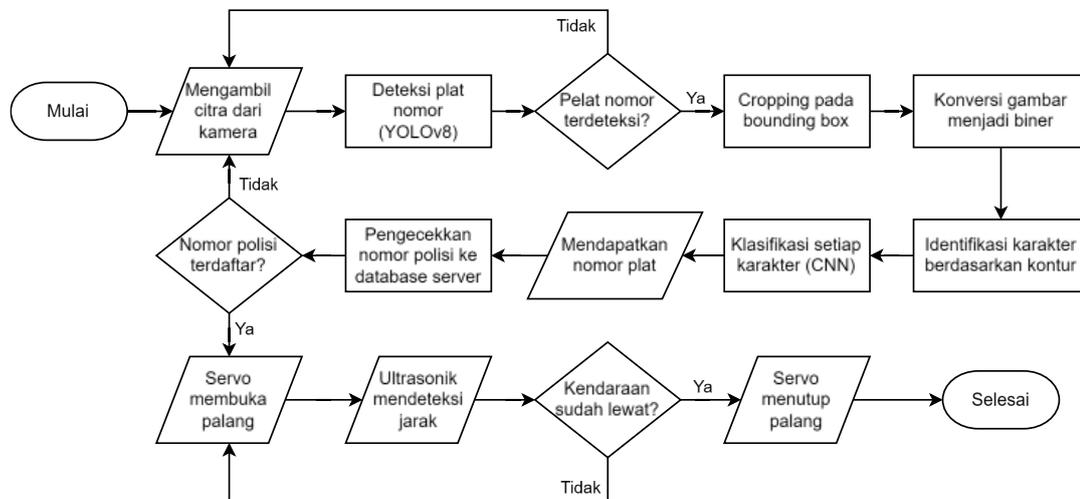


Gambar 3.27 Schematic Diagram PCB



Gambar 3.28 Desain Layout PCB

Berdasarkan penjelasan di atas, alur kerja pintu parkir dapat dijelaskan melalui *flowchart* pada Gambar 3.29.



Gambar 3.29 *Flowchart* Pintu Parkir

Secara keseluruhan, proses di pintu masuk dan keluar parkir berjalan dengan prinsip yang sama. Awalnya, kamera bertugas mendeteksi keberadaan plat nomor kendaraan. Setelah deteksi, sistem akan menunggu beberapa detik untuk mengambil gambar, memastikan gambar tersebut jelas. Langkah berikutnya adalah melakukan *cropping* pada *bounding box* yang mengandung plat nomor tersebut.

Selanjutnya, sistem melakukan proses *thresholding* untuk mengubah gambar menjadi biner hitam putih, yang memudahkan dalam mengidentifikasi karakter mencari kontur berdasarkan lebar dan tinggi elemen yang berhasil di-*threshold*. Setelah karakter teridentifikasi, tahap berikutnya adalah klasifikasi karakter menggunakan model CNN yang telah dilatih terhadap setiap karakter yang terdeteksi.

Ketika semua karakter telah terklasifikasi dan digabungkan menjadi sebuah plat nomor, sistem kemudian akan memeriksa keabsahan plat nomor tersebut di server dengan melakukan proses pengiriman API. Jika plat nomor dinyatakan valid, gerbang akan terbuka. Gerbang akan menutup kembali setelah kendaraan melintas, yang dikonfirmasi melalui deteksi sensor ultrasonik.

3.1.3. Pengembangan Sistem

Pada tahap pengembangan sistem, dilakukan implementasi dari perancangan yang sudah dibuat. Pengembangan model melibatkan pelatihan model yang sudah dijelaskan pada perancangan model. Model YOLO menggunakan arsitektur YOLOv8n sebagai model awal yang akan dilatih dengan menggunakan *dataset* yang sudah disiapkan. Sementara itu, CNN digunakan untuk pengenalan karakter pada plat nomor. Pelatihan model dilakukan dengan melakukan proses pelatihan model dengan bantuan CUDA dari GPU, menggunakan *framework* Jupyter dengan bahasa pemrograman Python pada Microsoft Visual Studio Code sebagai IDE.

Pengembangan meliputi pembuatan Android dengan menggunakan Android Studio sebagai IDE. Fitur-fitur yang sudah dijelaskan pada perancangan aplikasi akan diimplementasikan sesuai dengan desain UML dan desain UI yang telah dibuat.

Pada bagian server, pengembangan dilakukan menggunakan *micro framework* Flask dengan bahasa pemrograman Python. Server mengimplementasikan *endpoint* API yang telah dirancang menggunakan protokol HTTP/HTTPS dengan arsitektur RESTful API. Untuk menangani permintaan dari aplikasi Android dan pintu parkir, serta mengelola *database* MySQL, server menggunakan fitur *tunneling* yang disediakan oleh Cloudflare. Ini memastikan bahwa server dapat terhubung dengan internet dan dapat diakses oleh pengguna aplikasi Android dan perangkat pintu parkir.

Di sisi perangkat keras sebagai pintu parkir, Raspberry Pi diprogram untuk menjalankan model yang telah disimpan dari pelatihan model YOLO dan CNN. Pembuatan program menggunakan bahasa pemrograman Python pada Geany. Ultrasonik dan servo akan dipasangkan ke Raspberry Pi.

3.1.4. Pengujian Sistem

Pengujian sistem menggunakan metode *black box*, yang berfokus pada keluaran sistem tanpa mempertimbangkan proses internal dan implementasi kode secara mendetail (Fikri dkk., 2024). Pengujian sistem dibagi menjadi beberapa bagian dari pengujian.

3.1.4.1. Pengujian Model YOLO dan CNN

Pengujian model melibatkan model YOLO dan CNN yang telah dikembangkan. Penulis mengumpulkan sebanyak 20 foto kendaraan mobil dengan plat berwarna putih dan jenis tulisan terbaru sebagai data uji. Fokus utama pengujian ini adalah kemampuan model YOLO untuk mendeteksi plat nomor pada kendaraan. Kemampuan deteksi model akan diukur dengan menggunakan *output* variabel *conf*, yang merupakan hasil dari *confidence score*. Nilai *confidence score* ini akan dikonversi menjadi persentase, yang menunjukkan *confidence* sebuah model dalam mendeteksi plat nomor yang berhasil terdeteksi dengan menggunakan persamaan (5).

$$\text{Persentase Confidence Score (PCS)} = \text{confidence score} \times 100\% \quad (5)$$

Setelah karakter berhasil teridentifikasi, tahap selanjutnya melibatkan klasifikasi menggunakan model CNN yang telah dilatih khusus untuk mengenali setiap karakter yang terdeteksi. Hasil pengenalan ini kemudian dibandingkan dengan plat nomor asli untuk menghitung persentase akurasi. Perhitungan akurasi untuk menghitung persentase karakter yang benar dilakukan dengan menggunakan persamaan (6).

$$\text{Akurasi Karakter (AK)} = \frac{\text{Jumlah Karakter yang Terdeteksi Benar}}{\text{Jumlah Karakter yang Sebenarnya}} \times 100\% \quad (6)$$

Hasil akurasi untuk pengenalan karakter pada plat nomor dihitung dengan membandingkan jumlah karakter yang berhasil dideteksi oleh model dengan jumlah total karakter yang sebenarnya. Hasil dari perhitungan akurasi ini kemudian akan dibandingkan dengan akurasi yang diperoleh menggunakan persamaan yang sama. Selanjutnya, akurasi karakter dari model CNN akan dibandingkan dengan akurasi Tesseract-OCR yang mana telah digunakan pada penelitian sebelumnya untuk mengevaluasi keefektifan pendekatan yang diusulkan.

3.1.4.2. Pengujian API

Pengujian API dilakukan dengan mengevaluasi setiap *endpoint* untuk memastikan bahwa dapat menerima *request* dan mengirimkan *response* secara efektif. Pengujian dilakukan dengan menggunakan aplikasi Postman, dengan melibatkan simulasi berbagai *request* ke API menggunakan *endpoint* dan *methode* yang sesuai. Tujuan utama adalah untuk memverifikasi bahwa setiap *endpoint*

berfungsi sesuai dengan spesifikasi teknis dan kebutuhan sistem yang telah dirancang, menjamin integritas dan keandalan API.

3.1.4.3. Pengujian Fitur Aplikasi Android

Pengujian ini meliputi semua fitur utama dari aplikasi Android yang dikembangkan. Setiap fitur diuji secara menyeluruh untuk memastikan fungsionalitas yang dapat berjalan sebagaimana mestinya.

3.1.4.4. Pengujian Integrasi Sistem

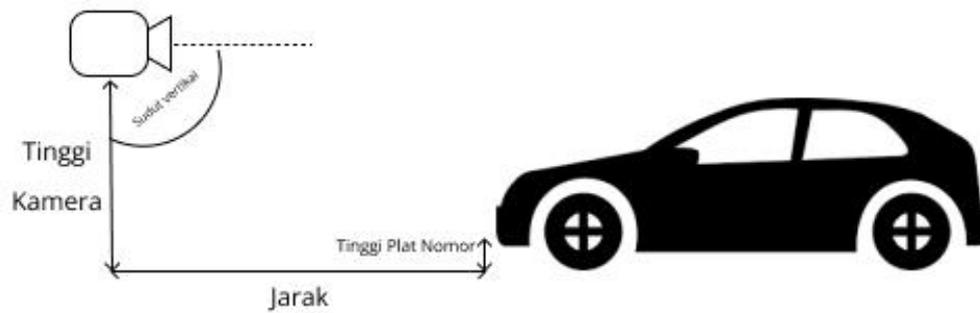
Pengujian integrasi sistem dilakukan dengan beberapa skenario pengujian untuk mengetahui keberhasilan sistem dalam berbagai kondisi operasional, meliputi semua komponen sistem dari perangkat keras ke perangkat lunak, untuk memverifikasi bahwa semua bagian bekerja.

3.1.4.5. Pengujian Performansi Model Terhadap Tinggi, Sudut, dan Jarak

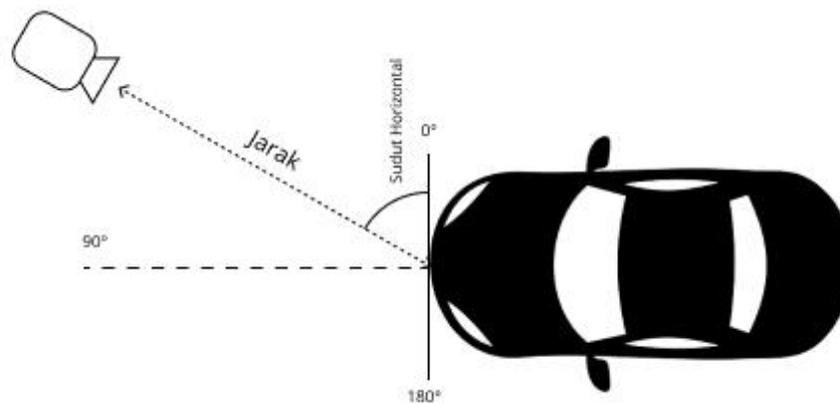
Pengujian ini berfokus pada menentukan konfigurasi optimal untuk ketinggian, sudut, dan jarak kamera. Proses pengujian dimulai dengan pengambilan data pada satu mobil dengan menggunakan *webcam* yang diatur pada berbagai variabel, dengan menentukan estimasi ketinggian, sudut, dan jarak, seperti yang ditunjukkan pada Gambar 3.30 dan Gambar 3.31.

Kamera diuji pada ketinggian kamera 50 cm, 100 cm, dan 150 cm dan mengatur sudut horizontal pada 90, 70, dan 50 derajat relatif terhadap plat nomor. Selain itu, kamera juga diuji pada jarak berbeda, mulai dari 50 cm, 100 cm, 150 cm, 200 cm, 250 cm, hingga 300 cm, untuk mengidentifikasi posisi yang paling efisien dalam mendeteksi plat nomor secara jelas dan akurat yang nantinya akan dijadikan sebagai acuan untuk penempatan kamera pada pintu parkir.

Setelah pengambilan data uji selesai, proses selanjutnya adalah pendeteksian plat nomor dan klasifikasi karakter pada foto yang diambil untuk setiap variabel menggunakan model YOLO dan CNN. Model YOLO akan menghasilkan *confidence score* dalam bentuk persentase untuk menilai deteksi plat nomor seperti pada persamaan (5), sementara model CNN akan menghitung akurasi klasifikasi karakter yang terdeteksi, juga dalam bentuk persentase, seperti yang dijelaskan pada persamaan (6).



Gambar 3.30 Ilustrasi Pengujian Performasi Model Terhadap Tinggi, Sudut, dan Jarak dari Sisi Samping



Gambar 3.31 Ilustrasi Pengujian Performasi Model Terhadap Tinggi, Sudut, dan Jarak dari Sisi Atas

3.1.5. Evaluasi Sistem

Evaluasi keseluruhan dari sistem yang dikembangkan melibatkan analisis menyeluruh dari berbagai aspek teknis dan operasional untuk memastikan bahwa sistem memenuhi semua persyaratan yang diharapkan dan beroperasi dengan baik. Proses ini mencakup evaluasi dari fungsionalitas sistem, keandalannya dalam kondisi operasional yang beragam, kemudahan penggunaan dari antarmuka, serta kemampuan integrasinya dengan sistem lain yang relevan. Hal ini dilakukan untuk memastikan bahwa setiap komponen sistem bekerja harmonis dan dapat memberikan pengalaman pengguna yang optimal. Secara khusus, untuk model yang terintegrasi, termasuk YOLO untuk deteksi plat nomor dievaluasi menggunakan *precision*, *recall*, *F1-Score*, *mAP50*, dan *mAP50-95* sedangkan untuk CNN dievaluasi menggunakan *accuracy*, *precision*, *recall*, dan *F1-Score*.

3.1.6. Penulisan Laporan

Setelah melakukan tahap sebelumnya, langkah selanjutnya adalah penulisan laporan. Tahap ini krusial karena memungkinkan peneliti untuk mendokumentasikan secara sistematis seluruh proses penelitian, dari pengumpulan data hingga analisis hasil. Laporan yang dihasilkan dalam bentuk skripsi ini akan mencakup secara rinci semua fase penelitian, mulai dari konsepsi awal hingga pengujian dan evaluasi hasil.

3.2. Spesifikasi Pengembangan Penelitian

Pengembangan model, aplikasi Android, dan server dilakukan pada satu komputer yang sama dengan spesifikasi seperti pada Tabel 3.2.

Tabel 3.2
Spesifikasi Kebutuhan Pengembangan Sistem

Jenis	Nama	Detail
<i>Hardware</i>	<i>Processor</i>	Intel i5 12400F
	<i>Memory</i>	16GB DDR4
	<i>Graphic Card</i>	RTX 3060 12GB
	<i>Storage</i>	SSD NVMe 512GB
<i>Software</i>	<i>Operating System</i>	Windows 11
	Visual Studio Code	Versi 1.90.2
	Python	Versi 3.10.11
	Android Studio	Versi 17.0.6
<i>Library Python</i>	Ultralytics	Versi 8.2.45
	TensorFlow	Versi 2.16.1
<i>Library Android</i>	Retrofit	Versi 2.9.0
	Midtrans	Versi 2.0.0
	Google Maps	Versi 18.2.0

Pengembangan pintu parkir untuk pengujian membutuhkan *software* maupun *hardware* seperti pada Tabel 3.3.

Tabel 3.3
Spesifikasi Kebutuhan Pengembangan Raspberry Pi

Jenis	Nama	Detail
<i>Hardware</i>	Raspberry Pi 4	RAM 4GB LPDDR4
		32GB Storage
	Ultrasonik	HC-SR04
	Servo	SG90
<i>Software</i>	<i>Operating System</i>	Raspberry Pi OS
	Geany	Versi 3.24.38
<i>Library Python</i>	Ultralytics	Versi 8.2.45
	TensorFlow	Versi 2.16.1
	GPIOzero	Versi 1.78