

BAB II DASAR TEORI

2.1 Tinjauan Pustaka

Setelah meninjau beberapa penelitian dan jurnal yang relevan, diperoleh beberapa referensi yang berkaitan dengan sistem gerbang tol otomatis berbasis IoT. Tinjauan pustaka ini bertujuan untuk mengetahui perkembangan penelitian sebelumnya serta mengidentifikasi celah penelitian (*research gap*) yang masih dapat dikembangkan. Ringkasan penelitian terdahulu disajikan pada **Tabel 2.1**.

Tabel 2. 1 Tinjauan Pustaka

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
1.	Analisis Efektivitas Pembayaran Tol Berbasis <i>Single Lane Free Flow</i> (SLFF) Dibandingkan dengan <i>E-Toll</i> melalui Kinerja Gerbang Tol dan Bekasi Timur	Penelitian ini menganalisis efektivitas pembayaran tol berbasis <i>Single Lane Free Flow</i> (SLFF/Flo) dibandingkan dengan <i>E-Toll</i> melalui kinerja gerbang tol dan persepsi pengguna di Gerbang Tol dan Bekasi Timur	Pembeda utama terletak pada fokus penelitian. Jika penelitian sebelumnya lebih menitikberatkan pada analisis efektivitas sistem SLFF/Flo yang sudah diterapkan di gerbang tol, penelitian ini berfokus pada rancang bangun teknologi RFID sebagai sistem transaksi dan identitas kendaraan, pengolahan data sehingga pengguna dapat kendaraan berbasis melakukan transaksi tanpa Firebase pada prototipe harus berhenti di gerbang tol <i>free flow</i> . Hasil penelitian Dengan penggunaan menunjukkan bahwa Firebase Realtime secara teknis SLFF/Flo Database, sistem yang memiliki waktu pelayanan dibangun dapat

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
		transaksi yang lebih cepat, menyimpan dan yaitu 0,6 detik per mengolah data kendaraan kendaraan, serta proyeksi secara <i>real-time</i> berbasis kapasitas pelayanan yang <i>cloud</i> , sehingga lebih jauh lebih besar fleksibel, mudah diakses, dibandingkan dengan <i>E-</i> dan mendukung <i>Toll</i> . Selain itu, sistem ini sinkronisasi data juga mampu menurunkan transaksi secara intensitas pelayanan serta langsung. Hal ini menjadi waktu tunggu kendaraan. pembeda utama Namun, penerapannya dibandingkan dengan belum efektif secara penelitian sebelumnya menyeluruh karena tingkat yang masih menilai penggunaan masih sangat performa operasional dan rendah, infrastruktur persepsi pengguna pada masih terbatas, serta sistem yang sudah sosialisasi kepada berjalan. masyarakat belum optimal dilakukan. Penelitian ini juga menunjukkan bahwa faktor utama yang memengaruhi pengguna untuk tidak menggunakan SLFF/Flo adalah kurangnya informasi, keterbatasan infrastruktur, dan persepsi manfaat yang belum maksimal [47].	
2.	Purwarupa Sistem Pembayaran	Penelitian ini merancang sistem pembayaran tol	Pembeda utama terletak pada arsitektur <i>database</i> .

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
	Retribusi Jalan Tol Berbasis Teknologi RFID (Primandani & Widodo, 2012)	otomatis berbasis RFID dengan komponen utama berupa <i>RFID reader</i> , <i>tag</i> <i>RFID</i> , sensor PIR, motor servo sebagai penggerak lokal dan terbatas pada portal, serta <i>database</i> satu terminal, penelitian untuk menyimpan data ini transaksi. Sistem bekerja mengimplementasikan dengan membaca identitas kendaraan melalui RFID, Database yang berbasis kemudian mencocokkan <i>cloud</i> . Hal ini data transaksi dengan memungkinkan untuk <i>database</i> untuk sinkronisasi data menentukan tarif sesuai transaksi secara <i>real-time</i> golongan kendaraan. antar gerbang tol yang Setelah proses validasi, berbeda dan aksesibilitas portal akan terbuka secara data bagi pengguna otomatis. Sistem juga melalui jaringan internet mampu menyimpan dan secara universal.	Jika pada penelitian sebelumnya menggunakan Microsoft Access yang bersifat sebagai penggerak lokal dan terbatas pada satu terminal, penelitian ini mengimplementasikan Firebase Realtime Database yang berbasis <i>cloud</i> . Hal ini memungkinkan untuk sinkronisasi data transaksi secara <i>real-time</i> antar gerbang tol yang berbeda dan aksesibilitas data bagi pengguna otomatis. Sistem juga melalui jaringan internet mampu menyimpan dan secara universal.
		menampilkan data transaksi sehingga dapat digunakan untuk keperluan <i>monitoring</i> . Implementasi sistem ini menggunakan perangkat mikrokontroler Arduino Uno R3 dengan logika kendali yang terhubung pada <i>database</i> lokal Microsoft Access. Hasil	

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
		pengujian menunjukkan bahwa sistem mampu mendeteksi kendaraan pada jarak hingga 20 meter [9].	
3.	Implementasi Mikrokontroller Arduino Uno dan <i>Mini-Computer</i> Raspberry Pi 3 (MFRC522), dalam Pengendalian Sistem Gerbang Tol Cerdas (Afiq, dkk, 2019)	Penelitian ini merancang purwarupa gerbang tol cerdas yang tugas akhir yang mengintegrasikan RFID dirancang terletak pada modul media penyimpanan dan komunikasi RFHC-12, pengolahan data. Penelitian sebelumnya menggunakan sensor ultrasonik HC-SR04, webcam, Arduino Uno dan Raspberry Pi 3, sistem penyimpanan dengan Node-RED untuk lokal yang terintegrasi logika kendali. Sistem dengan Raspberry Pi, bekerja dengan sedangkan tugas akhir ini mendeteksi kendaraan menggunakan Firebase melalui sensor ultrasonik, Realtime Database kemudian OBU mengirim berbasis <i>cloud</i> sehingga UID RFID yang diterima data transaksi kendaraan oleh <i>OBU-Reader</i> lewat dapat tersimpan dan <i>link</i> RF, sistem kemudian tersinkronisasi secara memvalidasi UID <i>real-time</i> . Pendekatan ini terhadap <i>database</i> lokal memberikan keunggulan yang mengaktifkan proses dalam hal akses data, pengambilan citra serta pemantauan transaksi, notifikasi pembayaran dan fleksibilitas integrasi melalui Telegram. Setelah sistem pada prototipe validasi berhasil, palang gerbang tol <i>free flow</i> .	Perbedaan utama penelitian ini dengan penelitian sebelumnya yang tugas akhir yang dirancang terletak pada media penyimpanan dan pengolahan data. Penelitian sebelumnya menggunakan sensor ultrasonik HC-SR04, webcam, Arduino Uno dan Raspberry Pi 3, sistem penyimpanan dengan Node-RED untuk lokal yang terintegrasi logika kendali. Sistem dengan Raspberry Pi, bekerja dengan sedangkan tugas akhir ini mendeteksi kendaraan menggunakan Firebase melalui sensor ultrasonik, Realtime Database kemudian OBU mengirim berbasis <i>cloud</i> sehingga UID RFID yang diterima data transaksi kendaraan oleh <i>OBU-Reader</i> lewat dapat tersimpan dan <i>link</i> RF, sistem kemudian tersinkronisasi secara memvalidasi UID <i>real-time</i> . Pendekatan ini terhadap <i>database</i> lokal memberikan keunggulan yang mengaktifkan proses dalam hal akses data, pengambilan citra serta pemantauan transaksi, notifikasi pembayaran dan fleksibilitas integrasi melalui Telegram. Setelah sistem pada prototipe validasi berhasil, palang gerbang tol <i>free flow</i> .

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
		pintu dikendalikan secara otomatis dan transaksi tersimpan untuk keperluan <i>monitoring</i> . Implementasi menggunakan komunikasi serial (SPI untuk MFRC522 dan UART untuk HC-12) serta Node-RED untuk pengaturan blok kontrol, sedangkan <i>database</i> dan notifikasi dikelola secara lokal pada Raspberry Pi [48].	
4.	Rancang Bangun Penelitian Sistem Pajak Tol Otomatis dengan RFID dan Informasi Berbasis Android (Sholihin, dkk, 2017)	Penelitian ini mengintegrasikan dengan modul ESP8266 untuk mentransmisikan data ID pengguna ke server. Sistem ini juga dilengkapi dengan sensor ultrasonik untuk mendeteksi keberadaan objek motor servo sebagai penggerak palang pintu. Inovasi yang ditawarkan adalah integrasi sistem dengan aplikasi Android yang berfungsi untuk memberikan informasi	ini Sistem pada penelitian ini sudah memanfaatkan jaringan internet, namun implementasi masih terbatas pada <i>monitoring</i> melalui aplikasi Android dan belum menggunakan sistem <i>real-time cloud database</i> yang optimal. Selain itu, jarak pembacaan RFID masih terbatas. Penelitian ini akan meningkatkan efisiensi

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
		sisal saldo kepada proses transaksi tanpa pengguna jalan tol. mengubah konsep <i>Database</i> yang digunakan gerbang. dalam sistem ini berbasis <i>web server</i> dengan pengisian ulang saldo melalui antarmuka web [10].	
5.	Rancang Bangun Pintu Gerbang Secara Otomatis Berbasis IoT Menggunakan Aplikasi Blynk (Syaroni, dkk, 2025)	Penelitian ini membahas sistem otomasi pintu rumah yang dikendalikan melalui aplikasi Blynk pada <i>smartphone</i>. Perangkat utama yang digunakan adalah NodeMCU ESP8266 sebagai transmitter yang menghubungkan perintah dari Android ke Arduino Uno. Pengujian fungsionalitas menunjukkan bahwa sistem dapat beroperasi secara efektif pada jarak kendali 1 hingga 29 meter melalui jaringan <i>Wi-Fi</i>. Fokus utama sistem ini adalah memberikan kenyamanan bagi	Perbedaan mendasar terletak pada logika operasional sistem. Penelitian sebelumnya menitikberatkan pada kendali manual jarak jauh (<i>remote control</i>) melalui tombol yang ada di aplikasi. Sedangkan pada penelitian ini, gerbang dikendalikan secara otomatis melalui logika pemrosesan transaksi. Gerbang hanya akan terbuka jika sistem mendeteksi <i>tag RFID</i> yang valid dan saldo di Firebase mencukupi untuk melakukan pembayaran secara otomatis tanpa intervensi manual dari pengguna.

No.	Judul Penelitian	Keterangan	Perbaikan/Pembeda Utama
		pengguna untuk membuka gerbang tanpa harus turun dari kendaraan [11].	

Sumber: diolah dari literatur [9][10][11][47][48]

Berdasarkan analisis dari penelitian-penelitian sebelumnya yang terdapat pada **Tabel 2.1**, dapat disimpulkan bahwa masih terdapat beberapa keterbatasan, yaitu:

1. Sistem belum terintegrasi secara penuh dengan teknologi *cloud* untuk pengolahan data secara *real-time*.
2. Pengelolaan data transaksi masih belum terstruktur dengan baik sehingga menyulitkan proses *monitoring* dan *tracking*.
3. Integrasi antara proses fisik (deteksi kendaraan dan kontrol gerbang) dengan sistem transaksi belum berjalan secara sinkron.
4. Logika sistem yang digunakan masih sederhana sehingga berpotensi menimbulkan kesalahan dalam proses transaksi.

Oleh karena itu, penelitian ini dikembangkan untuk mengatasi keterbatasan tersebut dengan mengintegrasikan sistem transaksi berbasis IoT menggunakan Firebase serta menerapkan metode *Finite State Machine* (FSM) untuk meningkatkan keandalan logika sistem.

2.2 Kondisi Jalan Tol di Indonesia

Jalan tol di Indonesia dibangun untuk mendukung kelancaran mobilitas masyarakat dan mempercepat distribusi barang maupun jasa. Pertumbuhan jumlah kendaraan bermotor di Indonesia berdampak langsung pada meningkatnya volume lalu lintas di jalan tol, terutama pada ruas yang masih menggunakan sistem transaksi berhenti di gerbang [1]. Pada kondisi titik puncak keramaian, antrean dapat terjadi karena proses identifikasi kendaraan, validasi saldo, dan pembukaan palang masih membutuhkan waktu layanan tertentu. Meskipun hambatan seperti kegagalan

tapping, keterlambatan respons sistem, dan ketidakstabilan perangkat masih menjadi kendala di lingkungan.

Kondisi tersebut menunjukkan bahwa sistem pembayaran tol masih membutuhkan solusi yang lebih efisien, terintegrasi, dan mampu meminimalkan kontak fisik. Oleh karena itu, pengembangan sistem berbasis *free flow* dan otomasi transaksi menjadi relevan sebagai upaya untuk meningkatkan kelancaran arus kendaraan sekaligus meningkatkan kualitas layanan. Untuk memperjelas karakteristik masing-masing metode transaksi yang telah diterapkan di jalan tol Indonesia, perbandingan sistem pembayaran tersebut disajikan pada **Tabel 2.2**.

Tabel 2. 2 Perbandingan Sistem Pembayaran Tol di Indonesia

No.	Sistem	Cara Kerja	Kecepatan Transaksi	Kelebihan	Kekurangan
1.	Manual/ Tunai	Pengendara membayar tunai langsung ke petugas di gerbang tol	Lambat ($\pm 7-8$ detik/ kendaraan)	Tidak butuh perangkat khusus kendaraan	Antrean panjang, di rawan <i>human error</i> , butuh banyak petugas
2.	<i>E-Toll</i>	Pengendara menempelkan kartu uang elektronik ke <i>reader</i> di gerbang	Sedang ($\pm 2-3$ detik/ kendaraan)	Lebih cepat dari sistem tunai, mengurangi kontak fisik dengan petugas	Kendaraan harus berhenti, antrean masih tetap terjadi
3.	SLFF	Identifikasi kendaraan dilakukan otomatis tanpa kartu ditempel, tapi kendaraan	Cepat (Kendaraan tidak berhenti penuh, cukup melambat)	Mengurangi antrean, transaksi tanpa sentuh	Masih perlu lajur khusus, kapasitas terbatas

No.	Sistem	Cara Kerja	Kecepatan Transaksi	Kelebihan	Kekurangan
		tetap melalui satu lajur khusus			
4.	MLFF	Identifikasi lokasi kendaraan dilakukan melalui <i>Global Navigation Satellite System</i> (GNSS) dan aplikasi khusus di <i>smartphone</i> , tarif dihitung otomatis oleh sistem pusat melalui proses <i>map-matching</i> saat kendaraan keluar tol, tanpa kartu maupun kendaraan perlu melambat	Sangat cepat (0 detik, tarif dihitung saat keluar tol)	Menghilangkan antrean total, efisiensi tinggi	Butuh infrastruktur teknologi tinggi, investasi besar, perlu sistem penegakan pelanggaran

Sumber: diolah dari literatur [31][32][39][49][50][51][53]

Berdasarkan **Tabel 2.2**, terlihat bahwa terdapat pergeseran arah pengembangan sistem pembayaran tol di Indonesia, dari sistem yang mengharuskan kendaraan berhenti sepenuhnya menuju sistem transaksi tanpa henti yang lebih efisien dan minim kontak fisik. Adapun perkembangan sistem pembayaran tol di Indonesia dari waktu ke waktu, mulai dari sistem tunai hingga rencana penerapan MLFF, ditunjukkan secara ringkas pada **Gambar 2.1**.



Gambar 2. 1 Perkembangan Sistem Pembayaran Tol di Indonesia

Sumber: [53]

Berdasarkan **Gambar 2.1**, perkembangan tersebut dimulai dari sistem tunai, kemudian beralih ke *e-toll*, dilanjutkan ke *Single Lane Free Flow* (SLFF), dan selanjutnya menuju *Multi Lane Free Flow* (MLFF). Pada sistem tunai, pengemudi melakukan pembayaran secara langsung kepada petugas di gerbang tol sehingga proses transaksi masih berlangsung secara manual dan membutuhkan waktu layanan yang relatif lebih lama. Selanjutnya, pada sistem *e-toll*, pengemudi cukup menempelkan kartu uang elektronik pada *reader* di gerbang tol sehingga transaksi menjadi lebih cepat, kendaraan tetap harus berhenti sehingga antrean masih dapat terjadi pada kondisi volume kendaraan yang tinggi. Setelah itu, sistem SLFF dikembangkan dengan memanfaatkan identifikasi otomatis berbasis RFID pada lajur khusus sehingga kendaraan dapat melintas tanpa perlu menempelkan kartu secara langsung. Tahap berikutnya adalah MLFF, yaitu sistem transaksi tanpa henti yang memungkinkan identifikasi kendaraan dilakukan secara otomatis melalui teknologi berbasis GNSS, RFID, atau kombinasi teknologi lain yang terintegrasi, sehingga kendaraan tidak perlu berhenti maupun berpindah ke lajur khusus saat melakukan transaksi. Perbedaan waktu layanan pada keempat sistem tersebut menunjukkan bahwa semakin minim interaksi fisik yang dibutuhkan dalam proses pembayaran tol, semakin cepat pula waktu pelayanan yang dapat dicapai, sehingga mendorong arah pengembangan sistem pembayaran tol menuju transaksi yang lebih otomatis, efisien, dan bebas hambatan [53].

Salah satu akar masalah pada sistem semiotomatis seperti *e-toll* adalah ketergantungannya pada kontak fisik antara kartu dan *reader*, yang membuat

keberhasilan transaksi sangat dipengaruhi oleh kondisi kartu, jarak *tapping*, dan kestabilan perangkat pembaca di gerbang tol. Untuk mengatasi keterbatasan ini, badan usaha jalan tol di Indonesia mulai mengembangkan sistem transaksi tanpa henti berbasis *Radio Frequency Identification* (RFID) yang dikenal sebagai *Single Lane Free Flow* (SLFF), di mana kendaraan cukup melintas pada lajur khusus tanpa perlu menempelkan kartu, karena identifikasi dilakukan secara otomatis melalui *RFID tag* yang terpasang pada kendaraan. Namun demikian, efektivitas sistem berbasis RFID ini sangat bergantung pada konfigurasi teknis seperti daya pancar atau sudut antena, karena kesalahan konfigurasi dapat menimbulkan gangguan berupa *tailgating* maupun *bouncing* yang berpotensi menyebabkan kegagalan identifikasi kendaraan [50].

Perkembangan lebih lanjut mengarah pada penerapan sistem *Multi Lane Free Flow* (MLFF), yang dirancang agar transaksi tol dapat dilakukan pada kecepatan normal tanpa kendaraan perlu melambat maupun berpindah ke lajur khusus. Kajian mengenai kesiapan penerapan MLFF di Indonesia menunjukkan bahwa keberhasilan sistem ini tidak hanya ditentukan oleh kesiapan teknologi, tetapi juga oleh kesiapan budaya dan perilaku pengguna jalan tol, sehingga pemilihan teknologi yang tepat perlu disesuaikan dengan karakteristik masyarakat Indonesia [51]. Selain itu, evaluasi terhadap sistem pengumpulan tol elektronik yang telah berjalan menunjukkan bahwa peningkatan efisiensi transaksi memerlukan integrasi yang lebih baik antara sistem identifikasi kendaraan, validasi saldo, dan mekanisme pembukaan palang otomatis, agar waktu layanan di gerbang tol dapat ditekan secara signifikan [49][52].

Dengan mempertimbangkan berbagai kendala pada sistem yang telah berjalan serta arah pengembangan menuju transaksi tanpa henti sepenuhnya, penelitian ini berfokus pada perancangan sistem transaksi dan pengolahan data kendaraan berbasis Firebase guna mendukung transisi sistem pembayaran tol di Indonesia ke arah yang lebih efisien, terintegrasi, dan minim hambatan operasional di lapangan.

2.3 Sistem *Single Lane Free Flow* (SLFF)

Transformasi sistem transaksi jalan tol menuju *Single Lane Free Flow* (SLFF) menjadi solusi krusial untuk mengatasi kemacetan dengan meniadakan keharusan kendaraan berhenti di gardu tol [31]. Secara teknis, sistem ini mengintegrasikan teknologi identifikasi otomatis berbasis *Radio Frequency Identification* (RFID) untuk mendeteksi perangkat kendaraan saat melintas. Berbeda dengan sistem konvensional yang memicu antrean akibat reduksi kecepatan, SLFF memungkinkan pemrosesan transaksi berlangsung secara *real-time* dan otomatis tanpa interaksi langsung saat kendaraan tetap melaju [12]. Penerapan sistem serupa juga telah lebih dahulu dilakukan di Jepang melalui *Electronic Toll Collection* (ETC) berbasis RFID, dengan tujuan utama mengurangi kemacetan di sekitar gerbang tol, meningkatkan kenyamanan pengguna, dan menghilangkan kebutuhan transaksi tunai secara langsung. Hal ini menunjukkan bahwa konsep tol tanpa henti telah terbukti efektif sebagai salah satu pendekatan modern untuk mendukung kelancaran arus lalu lintas dan efisiensi pelayanan jalan tol [60][61]. Penerapan konsep serupa turut ditemukan di Inggris melalui sistem *Dartford Crossing* (*Dart Charge*) yang beralih dari gerbang pembayaran manual menuju skema pembayaran tanpa henti (*free-flow charging*) sejak tahun 2014. Evaluasi resmi terhadap sistem tersebut menunjukkan adanya perbaikan signifikan pada waktu tempuh kendaraan serta keberhasilan pengurangan kemacetan di titik penyeberangan tersebut [62].

Pada sistem SLFF, kendaraan yang telah terdaftar akan terbaca oleh *RFID reader* yang dipasang di area pembacaan. Setelah identitas kendaraan diterima, sistem akan memvalidasi data kendaraan dan saldo yang tersedia. Jika data dinyatakan valid, sistem akan memberikan izin akses sehingga kendaraan dapat terus melintas tanpa perlu berhenti di gerbang tol. Mekanisme ini dirancang untuk mempercepat waktu pelayanan kendaraan sekaligus mengurangi potensi kemacetan di titik transaksi [42]. Ilustrasi penerapan sistem *Single Lane Free Flow* ditunjukkan pada **Gambar 2.2**.



Gambar 2. 2 *Single Lane Free Flow*
Sumber: [42]

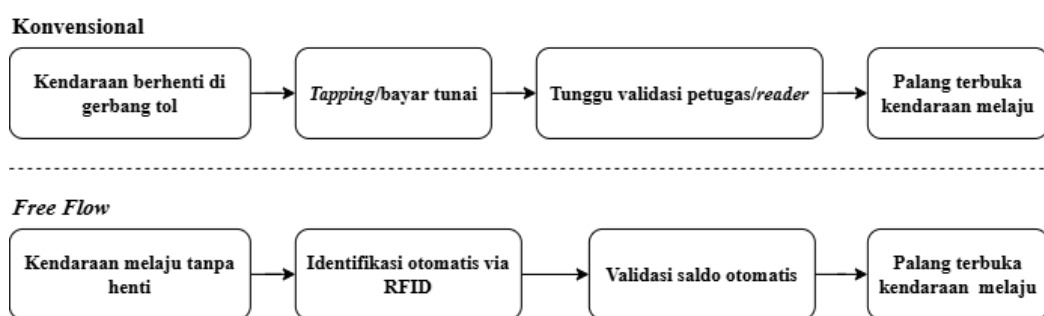
Berdasarkan **Gambar 2.2**, kendaraan dapat melintasi gerbang tol tanpa harus berhenti, sehingga proses transaksi berlangsung lebih cepat dan efisien. Sistem ini memanfaatkan teknologi identifikasi kendaraan dan komunikasi data untuk melakukan transaksi secara otomatis [42]. Akurasi dan kecepatan teknologi ini memberikan dampak signifikan terhadap efisiensi pelayanan, di mana penelitian sebelumnya mencatat bahwa waktu transaksi dapat dikurangi hingga 0,6 detik per kendaraan. Waktu transaksi yang lebih singkat tersebut memberikan dampak positif terhadap efisiensi konsumsi bahan bakar serta penurunan emisi gas buang. Kondisi ini menunjukkan bahwa implementasi SLFF tidak hanya meningkatkan efisiensi operasional jalan tol, tetapi juga berkontribusi untuk mewujudkan sistem transportasi yang lebih berkelanjutan [32][47].

Dengan demikian, efektivitas sistem ini bergantung pada stabilitas infrastruktur, keandalan pembacaan data, serta integrasi antarperangkat yang baik [31][32]. Berdasarkan pertimbangan tersebut, penelitian ini mengembangkan prototipe yang dipadukan dengan penggunaan gerbang fisik sebagai media simulasi. Prototipe tugas akhir ini dirancang menggunakan satu jalur (*single lane*) untuk mempermudah pengamatan alur data dan pengujian respons setiap komponen secara terstruktur, sekaligus tetap merepresentasikan karakteristik operasional sistem pada kondisi nyata [13].

2.3.1 Alur Pembayaran Tol Konvensional dan *Free Flow*

Pada sistem pembayaran tol, alur transaksi secara umum dimulai dari

kedatangan kendaraan, proses identifikasi, validasi data atau saldo, hingga pemberian akses untuk melintas. Pada sistem konvensional, kendaraan harus berhenti terlebih dahulu di gerbang tol untuk melakukan transaksi, sedangkan pada sistem *free flow*, proses tersebut dirancang agar kendaraan tetap bergerak selama identifikasi berlangsung. Perbedaan alur ini menjadi salah satu dasar pengembangan sistem tol modern yang lebih efisien dan minim hambatan operasional [46][49]. Untuk memperjelas mekanisme tersebut, alur pembayaran tol konvensional dan *free flow* ditunjukkan pada **Gambar 2.3**.



Gambar 2.3 Perbandingan Alur Pembayaran Tol Konvensional dan *Free Flow*

Sumber: diolah dari literatur [46][47][49]

Berdasarkan **Gambar 2.3**, sistem pembayaran tol konvensional masih mengharuskan kendaraan berhenti di gerbang untuk melakukan transaksi, sedangkan pada sistem *free flow* kendaraan dapat melintas tanpa berhenti karena identifikasi dan validasi dilakukan secara otomatis. Perbedaan alur tersebut menunjukkan bahwa sistem *free flow* memiliki potensi untuk meningkatkan efisiensi pelayanan dan mengurangi antrean kendaraan di gerbang tol [46][47]. Oleh karena itu, pada penelitian tugas akhir ini digunakan pendekatan *free flow* sebagai dasar pengembangan prototipe agar sistem dapat merepresentasikan proses transaksi tol yang lebih modern, otomatis, dan terintegrasi dengan pengolahan data berbasis *cloud*.

2.4 Internet of Things (IoT)

Internet of Things (IoT) merupakan konsep yang menghubungkan berbagai perangkat fisik ke jaringan internet sehingga memungkinkan pertukaran data secara otomatis dan *real-time* tanpa memerlukan interaksi langsung dari manusia, seperti ditunjukkan pada **Gambar 2.4**.



Gambar 2. 4 *Internet of Things (IoT)*
Sumber: [43]

Berdasarkan **Gambar 2.4**, sistem IoT terdiri atas perangkat fisik yang dilengkapi sensor atau aktuator untuk mengumpulkan data. Data tersebut kemudian dikirim melalui jaringan internet ke *cloud* atau *server* untuk diproses dan disimpan. Karakteristik tersebut menjadikan IoT sebagai dasar pengembangan sistem cerdas yang mengintegrasikan teknologi identifikasi, sensor, dan komunikasi untuk meningkatkan efisiensi di berbagai bidang, termasuk transportasi [14]. Penerapan IoT telah banyak dimanfaatkan untuk mendukung mobilitas cerdas pada ekosistem *smart city* [33]. Selain itu, teknologi ini juga diterapkan dalam sistem manajemen lalu lintas berbasis sensor dan pelacakan kendaraan untuk mendukung pengambilan keputusan berdasarkan data yang diperoleh secara *real-time* [31].

Dalam penelitian ini, IoT berperan sebagai penghubung antara perangkat pada prototipe gerbang tol dan sistem pengolahan data berbasis digital. Melalui mikrokontroler ESP32, sensor, dan modul komunikasi, sistem dapat membaca identitas atau status kendaraan, kemudian mengirimkan data tersebut secara langsung ke *database* berbasis *cloud*. Proses ini mengurangi kebutuhan pencatatan manual sekaligus memungkinkan data diakses dan diperbarui dengan lebih efisien. Penggunaan Firebase sebagai media penyimpanan terpusat mendukung proses sinkronisasi data transaksi kendaraan sehingga informasi dapat diperbarui secara cepat dan digunakan sebagai dasar pengolahan data yang lebih efektif [32].

Penerapan IoT pada tugas akhir ini sejalan dengan konsep transportasi cerdas yang menekankan efisiensi, responsivitas, dan peningkatan kualitas layanan [34].

Meskipun demikian, implementasi IoT masih dipengaruhi oleh beberapa faktor, seperti stabilitas komunikasi antarperangkat, integrasi sistem, dan keamanan data. Oleh karena itu, pada prototipe gerbang tol *free flow* ini, IoT berperan sebagai komponen utama yang menghubungkan proses identifikasi kendaraan, pengiriman data, hingga penyimpanan informasi secara terintegrasi, sehingga seluruh proses transaksi dan pengolahan data dapat berlangsung secara cepat, akurat, dan berkesinambungan.

2.4.1 Komunikasi Data dan Koneksi Wi-Fi

Komunikasi data pada sistem IoT merupakan proses pertukaran informasi antara perangkat *embedded* dan *server* melalui media jaringan. Pada penelitian tugas akhir ini, komunikasi data dilakukan melalui koneksi *Wi-Fi* karena ESP32 telah dilengkapi dengan modul komunikasi nirkabel yang memungkinkan perangkat terhubung langsung ke internet tanpa memerlukan modul tambahan. Dengan adanya koneksi ini, data hasil pembacaan sensor dan identitas kendaraan dapat dikirimkan ke Firebase Realtime Database secara *real-time* [54].

Penggunaan jaringan *Wi-Fi* pada sistem ini memungkinkan ESP32 untuk berinteraksi dengan *cloud database* secara langsung, sehingga data transaksi dapat disimpan dan diperbarui tanpa harus melalui proses penyimpanan lokal terlebih dahulu. Mekanisme ini penting dalam sistem gerbang tol *free flow* karena proses transaksi harus berlangsung cepat dan sinkron dengan status data yang tersimpan di *database*. Selain itu, koneksi *Wi-Fi* juga mendukung pemantauan sistem dari jarak jauh melalui antarmuka yang terhubung ke Firebase.

Namun, kualitas komunikasi data sangat dipengaruhi oleh kestabilan jaringan *Wi-Fi* [55]. Jika koneksi jaringan mengalami gangguan, proses pengiriman data ke *database* dapat tertunda atau gagal. Kondisi ini dapat memengaruhi respons sistem secara keseluruhan, terutama saat ESP32 melakukan pembaruan data transaksi. Oleh karena itu, kestabilan koneksi *Wi-Fi* menjadi salah satu aspek penting yang perlu diperhatikan dalam perancangan dan pengujian sistem berbasis IoT.

2.4.2 Wi-Fi RSSI

Wi-Fi RSSI (Received Signal Strength Indicator) adalah parameter yang digunakan untuk menunjukkan kekuatan sinyal *Wi-Fi* yang diterima oleh perangkat. Pada jaringan IEEE 802.11, RSSI umum dipakai sebagai indikator kualitas sinyal pada sisi penerima, dan nilainya dinyatakan dalam satuan dBm. Semakin mendekati 0 dBm, sinyal semakin kuat, sedangkan semakin kecil nilainya secara absolut, kualitas sinyal semakin lemah [57].

Dalam sistem berbasis IoT, RSSI berperan penting karena kualitas sinyal *Wi-Fi* memengaruhi kestabilan komunikasi data antara perangkat dan server [56]. Koneksi dengan RSSI yang rendah menyebabkan pengiriman data menjadi lambat, tidak stabil, atau bahkan gagal tersambung ke *database*. Oleh karena itu, parameter RSSI dapat digunakan untuk mengevaluasi performa komunikasi antara ESP32 dan jaringan *Wi-Fi* pada sistem yang terhubung dengan Firebase Realtime Database. Parameter tersebut ditunjukkan pada **Tabel 2.3**.

Tabel 2. 3 Parameter Kekuatan Sinyal Wi-Fi RSSI

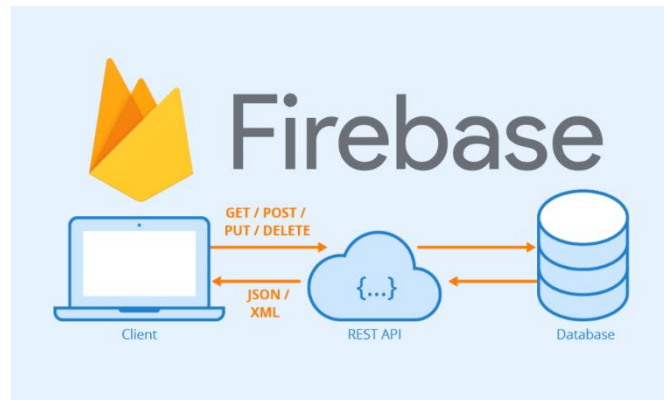
RSSI (dBm)	Kategori	Keterangan
-67 ke atas	Sangat Baik	Koneksi sangat baik dan stabil
-68 s.d. -70	Baik	Koneksi stabil untuk sebagian besar aktivitas
-71 s.d. -80	Cukup	Koneksi mulai kurang stabil
Dibawah -80	Buruk	Berisiko mengalami gangguan koneksi

Sumber: diolah dari literatur [57][58]

Berdasarkan **Tabel 2.3**, semakin kecil nilai RSSI secara absolut, kualitas sinyal *Wi-Fi* semakin baik. Pada penelitian tugas akhir ini, RSSI digunakan sebagai salah satu parameter untuk menganalisis kestabilan koneksi ESP32 selama proses pengiriman data ke Firebase, sehingga hasil pengujian dapat dinilai berdasarkan kondisi jaringan yang digunakan.

2.5 Firebase Realtime Database

Firebase merupakan platform *Backend-as-a-Service* (BaaS) berbasis *cloud* yang dikembangkan oleh Google untuk mendukung pengembangan aplikasi yang membutuhkan pengolahan dan sinkronisasi data secara cepat yang ditunjukkan pada **Gambar 2.5**.



Gambar 2. 5 Firebase
Sumber: [44]

Berdasarkan **Gambar 2.5**, salah satu layanan utamanya adalah Firebase Realtime Database, yang memungkinkan data disimpan, diperbarui, dan disinkronkan secara *real-time* antara perangkat pengguna dan *server*. Kehadiran layanan ini mempermudah proses pengembangan aplikasi karena tidak perlu membangun infrastruktur *backend* secara manual. Melalui mekanisme sinkronisasi otomatis, setiap perubahan data dapat langsung diperbarui pada seluruh perangkat yang terhubung ke jaringan [15].

Dalam ekosistem *Internet of Things* (IoT), Firebase berperan sebagai media penyimpanan dan pengelolaan data berbasis *cloud*. Data yang dikirimkan oleh perangkat, seperti mikrokontroler ESP32, dapat disimpan secara langsung ke *database* sehingga tidak bergantung pada kapasitas penyimpanan lokal perangkat. Selain itu, Firebase mendukung komunikasi dua arah, sehingga tidak hanya menerima data dari perangkat, tetapi juga dapat mengirimkan pembaruan status atau informasi lain secara *real-time* sesuai kebutuhan sistem [16].

Firebase memiliki kemampuan sinkronisasi data dengan latensi yang rendah, mudah diintegrasikan dengan berbagai platform, serta mampu menangani akses data secara bersamaan dari banyak pengguna. Karakteristik tersebut menjadikannya

sesuai untuk aplikasi yang memerlukan respons cepat dan pertukaran data secara *real-time* [15]. Pada penelitian tugas akhir ini, kemampuan tersebut dimanfaatkan untuk mendukung proses pencatatan transaksi pada prototipe gerbang tol *free flow*, sehingga komunikasi antara perangkat pembaca, mikrokontroler ESP32, dan antarmuka pemantauan dapat berlangsung secara terintegrasi [35].

Firebase Realtime Database menyimpan data dalam struktur JSON (*JavaScript Object Notation*), sehingga proses pembacaan, penulisan, dan pembaruan data dapat dilakukan dengan lebih sederhana dibandingkan dengan *database* relasional. Struktur tersebut memudahkan pengelolaan data identitas kendaraan, informasi saldo, dan riwayat transaksi yang terus diperbarui selama sistem beroperasi. Selain itu, penyimpanan berbasis *cloud* membantu menjaga ketersediaan data apabila terjadi gangguan pada perangkat keras di lapangan [15].

Dibandingkan dengan basis data relasional atau Relational Database Management System (RDBMS) yang mengorganisasikan data dalam tabel dengan skema baris dan kolom serta relasi antartabel, Firebase Realtime Database menggunakan struktur data hierarkis berbasis JSON. Struktur tersebut memungkinkan data dikelompokkan ke dalam node yang dapat disesuaikan dengan kebutuhan akses aplikasi, sehingga proses pembacaan dan pembaruan data dapat dilakukan secara lebih sederhana tanpa memerlukan operasi join antartabel. Perancangan struktur Firebase juga perlu mempertimbangkan pola akses data dan penggunaan teknik denormalisasi agar waktu akses dapat dioptimalkan [63]. Pada penelitian ini, data dirancang dengan memisahkan node yang menyimpan kondisi terkini pengguna pada `/users` dari node yang menyimpan riwayat transaksi pada `/transaksi/log`. Node `/users` digunakan untuk menyimpan informasi yang sering diperbarui, seperti saldo dan status akun kendaraan, sedangkan node `/transaksi/log` digunakan sebagai catatan historis yang bersifat *append-only*. Pemisahan tersebut bertujuan untuk mengurangi kompetisi akses terhadap struktur data yang sama dan menyesuaikan model penyimpanan dengan karakteristik transaksi pada sistem MLFF. Meskipun demikian, pembaruan saldo tetap perlu dilakukan menggunakan mekanisme transaksi atau pembaruan atomik untuk menjaga konsistensi ketika terdapat beberapa transaksi yang berlangsung secara bersamaan [64]. Pemilihan

basis data NoSQL pada sistem dengan frekuensi penulisan tinggi harus didasarkan pada kesesuaian antara model data, pola akses, kebutuhan konsistensi, dan beban kerja sistem. Studi mengenai performa NoSQL menunjukkan bahwa tidak ada satu model basis data yang selalu unggul untuk semua skenario karena throughput dan latensi dipengaruhi oleh desain data serta tingkat konsistensi yang diterapkan [65].

Pemanfaatan Firebase pada tugas akhir ini juga didukung oleh berbagai penelitian sebelumnya yang menunjukkan bahwa platform tersebut dapat digunakan untuk mendukung sistem pemantauan dan pencatatan data berbasis IoT secara *real-time*. Berdasarkan hasil penelitian tersebut, Firebase dinilai mampu mendukung proses penyimpanan, sinkronisasi, dan penyajian data secara terintegrasi [16]. Oleh karena itu, pada prototipe gerbang tol *free flow* ini, Firebase berfungsi sebagai media yang menghubungkan proses identifikasi kendaraan dengan penyimpanan dan penyajian informasi transaksi secara terpusat.

Meskipun menawarkan berbagai kemudahan, implementasi Firebase tetap memerlukan perancangan struktur data dan pengaturan keamanan yang sesuai. Pengaturan skema data, pembuatan indeks, serta pengelolaan hak akses baca dan tulis perlu diperhatikan agar proses sinkronisasi tetap berjalan dengan baik, terutama ketika jumlah data yang diproses meningkat. Oleh karena itu, aspek tersebut menjadi salah satu pertimbangan dalam perancangan sistem pada penelitian ini agar proses transaksi dan pengolahan data kendaraan dapat berlangsung secara konsisten dan mendukung pemantauan sistem secara *real-time*.

2.6 Finite State Machine (FSM)

Finite State Machine (FSM) merupakan metode pemodelan perilaku sistem yang didasarkan pada kumpulan keadaan (*state*), masukan (*input*), dan aturan perpindahan (*transition*). Pada suatu waktu, sistem hanya berada pada satu *state* aktif dan akan berpindah ke *state* berikutnya ketika menerima masukan yang sesuai. Dalam perancangan sistem, FSM digunakan untuk mengatur alur kerja agar berjalan secara terstruktur dan sesuai dengan urutan proses yang telah ditentukan [17]. Metode ini sesuai untuk diterapkan pada prototipe gerbang tol *free flow*, karena setiap kondisi operasional, seperti kendaraan terdeteksi, transaksi diproses,

hingga data berhasil disimpan, dapat direpresentasikan sebagai *state* yang berbeda [36].

Pada sistem *embedded*, FSM banyak digunakan untuk mengatur urutan proses yang melibatkan berbagai komponen serta respons terhadap perubahan kondisi. Dengan metode ini, setiap *input* dari sensor dapat diproses sesuai urutan yang telah ditentukan sehingga perilaku sistem menjadi lebih terstruktur dan konsisten [18][19]. Sebagai contoh, ketika sensor mendeteksi kendaraan, sistem akan berpindah dari kondisi *idle* ke proses identifikasi kendaraan, dilanjutkan dengan validasi transaksi, dan pengiriman data ke Firebase. Pembagian proses ke dalam beberapa *state* juga memudahkan pengujian dan pemeliharaan sistem [37].

Pada tugas akhir ini, FSM digunakan untuk mengatur tahapan proses transaksi dan pengolahan data kendaraan secara berurutan, mulai dari kondisi *standby*, pembacaan RFID, validasi saldo, hingga sinkronisasi data ke Firebase. Dengan pengaturan alur menggunakan FSM, setiap proses dapat dijalankan sesuai urutannya sehingga membantu mengurangi kemungkinan terjadinya kesalahan logika, seperti pemrosesan transaksi yang berulang atau ketidaksesuaian data antara perangkat dan *database* [38].

2.7 Sistem Transaksi dan Pengolahan Data Kendaraan

Sistem transaksi dan pengolahan data kendaraan merupakan rangkaian proses yang mengatur identifikasi kendaraan, validasi data, pemrosesan pembayaran, hingga penyimpanan informasi transaksi secara terstruktur. Pada sistem konvensional, proses transaksi masih menggunakan kartu elektronik sehingga kendaraan harus berhenti terlebih dahulu di gerbang tol. Kondisi tersebut dapat menyebabkan perlambatan arus lalu lintas dan antrean kendaraan, terutama pada periode dengan volume kendaraan yang tinggi. Akibatnya, efisiensi pelayanan di gerbang tol menurun dan berdampak pada kelancaran arus lalu lintas secara keseluruhan.

Untuk mengatasi permasalahan tersebut, berbagai penelitian telah mengembangkan sistem pembayaran tol berbasis elektronik yang terintegrasi dengan teknologi komunikasi. Hasil penelitian menunjukkan bahwa penerapan

sistem tersebut dapat meningkatkan kecepatan transaksi sekaligus mendukung pengelolaan data kendaraan secara lebih efisien [20]. Selain itu, integrasi *Internet of Things* (IoT) memungkinkan proses transaksi dan pemantauan data dilakukan secara *real-time* melalui sistem pusat [21]. Metode ini mendukung pengembangan sistem transportasi yang lebih cerdas, efisien, dan terintegrasi. Penelitian lain yang menerapkan integrasi RFID, mikrokontroler ESP32, dan *cloud database* pada sistem identifikasi otomatis mengatakan bahwa seluruh data akses kendaraan dapat tercatat secara *real-time* tanpa kegagalan pencatatan, dengan tingkat akurasi validasi yang tinggi. Namun, penelitian tersebut juga mencatat bahwa keandalan pencatatan data semacam ini masih bergantung pada kestabilan koneksi internet pada sisi perangkat [59].

Pada sistem transaksi tol modern, proses yang dilakukan meliputi identifikasi kendaraan, validasi data, pemrosesan pembayaran, hingga pemberian respons sistem. Penerapan teknologi *Single Lane Free Flow* (SLFF) berbasis RFID mendukung proses transaksi tanpa mengharuskan kendaraan berhenti di gerbang tol, sehingga waktu pelayanan dapat dipersingkat dan arus lalu lintas menjadi lebih lancar [13]. Penggunaan RFID juga membantu mempercepat proses identifikasi kendaraan dan mengurangi antrean pada sistem pembayaran konvensional [39]. Melalui pemanfaatan sensor dan mikrokontroler, seluruh proses transaksi dapat dijalankan secara otomatis dan terintegrasi dengan sistem pengolahan data.

Meskipun menawarkan berbagai keunggulan, efektivitas sistem transaksi tol tetap dipengaruhi oleh keandalan proses identifikasi kendaraan, stabilitas komunikasi, serta integrasi antara perangkat keras dan perangkat lunak. Hal ini sejalan dengan temuan [59] bahwa sistem berbasis IoT dan *cloud database* tetap memerlukan koneksi jaringan yang stabil agar proses pencatatan data berjalan secara konsisten. Oleh karena itu, pada tugas akhir ini dikembangkan prototipe gerbang tol *free flow* berbasis IoT dan Firebase untuk mendukung proses identifikasi, transaksi, dan pengolahan data kendaraan secara terintegrasi. Dengan begitu, sistem transaksi tidak hanya berfungsi sebagai media pembayaran, tetapi juga sebagai bagian dari sistem yang menghubungkan identifikasi kendaraan, pengolahan data secara *real-time*, dan pengendalian gerbang secara otomatis.

2.8 ESP32

ESP32 merupakan mikrokontroler berbasis *System on Chip* (SoC) yang dikembangkan oleh Espressif Systems dengan modul *Wi-Fi* dan *Bluetooth* yang telah terintegrasi dalam satu *chip*. Integrasi konektivitas nirkabel tersebut memungkinkan ESP32 terhubung langsung ke jaringan internet tanpa memerlukan perangkat tambahan, sehingga banyak digunakan pada berbagai aplikasi *Internet of Things* (IoT) [22]. Selain mendukung komunikasi data nirkabel, ESP32 juga memiliki konsumsi daya yang relatif rendah dengan tegangan operasi sekitar 2,2 V hingga 3,6 V, sehingga sesuai untuk diterapkan pada sistem *embedded* [23]. Bentuk fisik ESP32 ditunjukkan pada **Gambar 2.6**.



Gambar 2. 6 ESP32
Sumber: [23]

Dari **Gambar 2.6**, ESP32 dilengkapi dengan prosesor *dual-core* yang mampu menjalankan proses pengolahan data dan komunikasi jaringan secara bersamaan. Mikrokontroler ini juga menyediakan berbagai pin *General Purpose Input Output* (GPIO) serta mendukung antarmuka komunikasi, seperti I2C, SPI, dan UART, sehingga dapat diintegrasikan dengan berbagai sensor dan perangkat pendukung lainnya. Karakteristik tersebut menjadikan ESP32 sesuai untuk digunakan pada sistem yang melibatkan banyak komponen dan proses komunikasi data secara bersamaan [22]. Selain itu, ESP32 memiliki spesifikasi yang terdapat pada **Tabel 2.4**.

Tabel 2. 4 Spesifikasi ESP32

Fitur	Spesifikasi
CPU	Dual-core Xtensa 32-bit LX6
Kecepatan	Hingga 240 MHz
Memori Internal	448 KB ROM, 520 KB SRAM
Memori Eksternal	Mendukung SPI Flash & SRAM hingga 16 MB
Konektivitas	Wi-Fi (802.11 b/g/n) & Bluetooth (v4.2 BR/EDR & BLE)
Antarmuka (I/O)	UART, SPI, I2C, I2S, PWM, ADC, DAC, Touch Sensor
Keamanan <i>Hardware</i>	Akselerator AES, SHA, RSA, Flash Encryption, dan RNG

Sumber: [23]

Dengan spesifikasi pada **Tabel 2.4**, pada tugas akhir ini, ESP32 berfungsi sebagai pengendali utama yang mengoordinasikan proses pembacaan data dari sensor, pengendalian aktuator gerbang, pengoperasian indikator status, serta pengiriman data transaksi ke Firebase secara *real-time*. Dengan kemampuan komunikasi nirkabel yang dimilikinya, ESP32 dapat menghubungkan perangkat prototipe dengan *database* berbasis *cloud*, sehingga proses pertukaran dan sinkronisasi data dapat berlangsung secara terintegrasi.

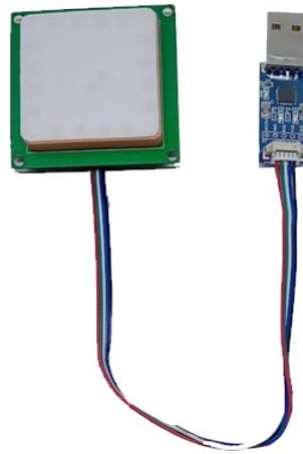
Meskipun memiliki kemampuan pemrosesan dan komunikasi yang baik, kinerja ESP32 tetap dipengaruhi oleh kualitas koneksi jaringan serta konfigurasi sistem yang digunakan [33]. Oleh karena itu, pada penelitian ini ESP32 dimanfaatkan sebagai pusat pengendali yang menghubungkan proses identifikasi kendaraan, pengolahan data, dan komunikasi dengan Firebase, sehingga seluruh proses pada prototipe gerbang tol *free flow* dapat berjalan secara terkoordinasi dan mendukung pengolahan data secara *real-time*.

2.9 RFID

Radio Frequency Identification (RFID) merupakan teknologi identifikasi otomatis yang memanfaatkan gelombang radio untuk membaca dan mentransmisikan data tanpa memerlukan kontak langsung (*line-of-sight*). Teknologi ini terdiri atas tiga komponen utama, yaitu *tag* sebagai media penyimpanan informasi, *reader* sebagai perangkat pembaca, dan sistem pengolah data. Dibandingkan dengan metode identifikasi konvensional, seperti *barcode* atau kartu manual, RFID mampu melakukan proses identifikasi lebih cepat serta mengurangi kemungkinan terjadinya kesalahan dalam proses pembacaan. Karakteristik tersebut menjadikan RFID banyak diterapkan pada berbagai sistem yang memerlukan proses identifikasi secara otomatis, termasuk sistem transaksi di gerbang tol [24].

Berdasarkan sumber dayanya, teknologi RFID dapat dikategorikan menjadi tiga jenis, yaitu RFID pasif, aktif, dan semipasif. RFID pasif tidak memiliki sumber daya internal dan memperoleh energi dari gelombang radio yang dipancarkan oleh *reader*, sehingga berukuran lebih kecil dan lebih murah, namun memiliki jarak baca yang relatif terbatas. RFID aktif memiliki baterai internal yang memungkinkan *tag* memancarkan sinyalnya sendiri secara mandiri, sehingga jarak bacanya jauh lebih besar, sedangkan RFID semipasif menggabungkan keduanya, yaitu memiliki baterai internal untuk mengaktifkan sirkuit *tag*, namun tetap mengandalkan energi dari *reader* untuk mengirimkan data [66].

Pada tugas akhir ini digunakan jenis RFID pasif dengan modul *Ultra High Frequency* (UHF) RFID Reader tipe EL-UHF-RMT01 untuk mendukung proses identifikasi kendaraan pada prototipe gerbang tol *free flow*. Bentuk fisik modul RFID *reader* yang digunakan ditunjukkan pada **Gambar 2.7**.



Gambar 2. 7 UHF RFID Reader EL-UHF-RMT01
Sumber: [45]

Dari **Gambar 2.7**, pemilihan modul UHF didasarkan pada kemampuannya membaca *tag* pada jarak yang lebih jauh dibandingkan dengan RFID frekuensi rendah serta mendukung proses identifikasi kendaraan yang sedang bergerak. Pemilihan ini juga didasarkan pada perbandingan karakteristik antarrentang frekuensi RFID. RFID *Low Frequency* (LF) bekerja pada rentang 125-134 kHz dengan jarak baca kurang dari 10 cm; RFID *High Frequency* (HF) bekerja pada frekuensi 13,56 MHz dengan jarak baca hingga 1 meter, sedangkan RFID *Ultra High Frequency* (UHF) bekerja pada rentang 860-960 MHz dengan jarak baca yang dapat mencapai lebih dari 10 meter [67]. Perbandingan ini memperkuat alasan pemilihan RFID UHF pada tugas akhir ini, karena jarak baca yang jauh lebih sesuai untuk mendeteksi kendaraan yang bergerak pada gerbang tol dibandingkan dengan RFID LF maupun HF. Hasil penelitian Al-Mulla menunjukkan bahwa teknologi UHF RFID banyak digunakan dalam sistem transportasi cerdas (*Intelligent Transportation Systems*) karena mampu mendukung proses identifikasi kendaraan secara cepat di lingkungan dengan mobilitas tinggi [25]. Pada tugas akhir ini, modul UHF RFID Reader EL-UHF-RMT01 yang digunakan memiliki spesifikasi yang ditunjukkan pada **Tabel 2.5**.

Tabel 2. 5 Spesifikasi UHF RFID Reader EL-UHF-RMT01

Parameter	Spesifikasi
Protokol	EPC Gen2 (EPC ISO18000-6C)
Frekuensi	865-928 MHz
Jarak Baca	1.5 meter (tergantung <i>tag</i> dan lingkungan) dengan pembacaan melebar
Kecepatan Baca	> 50 tag / detik
Tegangan Operasi	DC 3.5V – 5V
Arus Kerja	180mA @ 3.5V (Output 26 dBm) / 110mA @ 3.5V (Output 18 dBm)
<i>Antenna Power Output</i>	0 - 26 dBm (bisa diatur)
Antarmuka	TTL 3.3V (5 pin wafer) & USB to TTL (untuk uji coba PC)

Sumber: [45]

Dengan spesifikasi pada **Tabel 2.5**, secara teknis modul EL-UHF-RMT01 mendukung komunikasi TTL UART 3,3 V sehingga dapat diintegrasikan dengan mikrokontroler ESP32 tanpa memerlukan penyesuaian level tegangan. Modul ini juga memiliki daya pancar (*output power*) yang dapat dikonfigurasi untuk menyesuaikan kebutuhan pembacaan tag. Pada tugas akhir ini, EL-UHF-RMT01 digunakan sebagai perangkat pembaca identitas kendaraan yang mengirimkan data hasil pembacaan ke ESP32 untuk diproses lebih lanjut. Selanjutnya, data tersebut diteruskan ke Firebase sebagai dasar proses validasi transaksi dan pencatatan data kendaraan secara *real-time*.

2.10 Sensor Ultrasonik HY-SRF05

Sensor ultrasonik HY-SRF05 merupakan sensor pengukur jarak non-kontak yang bekerja dengan memancarkan gelombang ultrasonik, kemudian menerima pantulan gelombang dari objek di depannya. Jarak objek dihitung berdasarkan selang waktu antara sinyal yang dipancarkan dan sinyal pantul yang diterima. Secara matematis, jarak objek dihitung menggunakan persamaan berikut:

$$Jarak (cm) = \frac{t \times v}{2} \quad (2.1)$$

dengan:

- t = selang waktu antara pemancaran sinyal pada pin *trigger* dan penerimaan sinyal pantul pada pin *echo* (dalam mikrodetik).
- v = kecepatan rambat gelombang suara di udara, yaitu sekitar 0,0343 cm/ μ s.

Pembagian dengan angka 2 dilakukan karena selang waktu t yang terukur merupakan waktu tempuh gelombang secara bolak-balik, yaitu dari sensor menuju objek kemudian kembali ke sensor [40]. Metode tersebut memungkinkan sensor melakukan pengukuran jarak secara cepat sehingga banyak digunakan pada sistem deteksi objek. Bentuk fisik sensor ultrasonik HY-SRF05 ditunjukkan pada **Gambar 2.8**.



Gambar 2. 8 Sensor Ultrasonik HY-SRF05
Sumber: [40]

Dari **Gambar 2.8**, sensor ultrasonik HY-SRF05 dapat diintegrasikan dengan mikrokontroler ESP32 melalui pin *trigger* dan *echo*, sehingga proses komunikasi antara sensor dan mikrokontroler dapat dilakukan dengan sederhana. Pin *trigger* berfungsi menginisiasi pemancaran gelombang ultrasonik dari sensor, sedangkan pin *echo* berfungsi mendeteksi durasi pantulan gelombang tersebut untuk selanjutnya diproses oleh mikrokontroler menjadi nilai jarak menggunakan persamaan yang telah dijelaskan sebelumnya. Sensor ultrasonik juga banyak digunakan dalam berbagai sistem otomatisasi dan deteksi objek karena mudah diimplementasikan serta memiliki kemampuan pengukuran yang memadai untuk aplikasi jarak dekat hingga menengah. Pada tugas akhir ini, sensor ultrasonik HY-SRF05 memiliki spesifikasi yang ditunjukkan pada **Tabel 2.6**.

Tabel 2. 6 Spesifikasi Sensor Ultrasonik HY-SRF05

Fitur	Spesifikasi
Jangkauan Pengukuran	2 – 400 cm
Akurasi Pengukuran	Hingga 3 mm
Sudut Deteksi	Lebar
Frekuensi Kerja	40 kHz
Tegangan Operasi	<i>Input</i> tingkat TTL
Mode Operasi	2-Pin Mode: Menggunakan pin Trigger & Echo terpisah. 1-Pin Mode: Pin <i>Mode</i> dihubungkan ke Ground (GND), sehingga satu pin GPIO berfungsi sebagai Trigger sekaligus Echo.

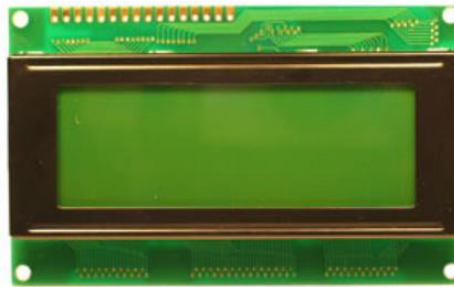
Sumber: [40]

Berdasarkan **Tabel 2.6**, HY-SRF05 memiliki rentang pengukuran sekitar 2 cm hingga 4 m, resolusi hingga 0,3 cm, serta frekuensi kerja 40 kHz. Karakteristik tersebut mendukung penggunaannya pada prototipe gerbang tol *free flow* untuk mendeteksi keberadaan kendaraan pada area pembacaan, sehingga proses transaksi dapat dijalankan secara otomatis [40].

Pada tugas akhir ini, sensor ultrasonik digunakan untuk mendeteksi keberadaan kendaraan sebelum proses identifikasi dilakukan. Informasi hasil pembacaan sensor digunakan sebagai masukan bagi ESP32 untuk menjalankan tahapan berikutnya, seperti pembacaan RFID, pengiriman data ke Firebase, dan pengendalian aktuator gerbang. Dengan demikian, sensor ultrasonik berfungsi sebagai pendeteksi awal yang membantu mengatur urutan proses transaksi pada sistem. Hasil pengukuran sensor masih dapat dipengaruhi oleh beberapa faktor, seperti sudut pantulan objek, karakteristik permukaan kendaraan, dan kondisi lingkungan. Oleh karena itu, pada penelitian ini HY-SRF05 digunakan sebagai sensor pendukung yang bekerja bersama RFID dan logika kendali sistem untuk mendukung proses deteksi kendaraan sebelum transaksi dilakukan.

2.11 LCD I2C 20X4

Liquid Crystal Display (LCD) 20x4 merupakan perangkat keluaran (*output*) yang digunakan untuk menampilkan informasi berupa teks dan angka pada sistem *embedded*. Tampilan bentuk fisik dari LCD 20×4 ditunjukkan pada **Gambar 2.9**.



Gambar 2. 9 LCD 20x4

Sumber: [27]

Dari **Gambar 2.9**, modul ini memiliki kapasitas tampilan sebanyak 20 kolom dan 4 baris, sehingga mampu menampilkan hingga 80 karakter dalam satu layar. Dibandingkan dengan LCD 16×2, LCD 20×4 menyediakan ruang tampilan yang lebih luas sehingga dapat menampilkan lebih banyak informasi secara bersamaan [26]. Selain itu, spesifikasi lainnya dapat dilihat pada **Tabel 2.7**.

Tabel 2. 7 Spesifikasi LCD 20X4

Fitur	Spesifikasi
Tegangan Operasi	+5V (toleransi $\pm 5\%$)
Arus Supply	2.5 mA (Tipikal) / 4.0 mA (Maksimal)
Format Karakter	5 x 8 Dots dengan Kursor
Kontroler Integrasi	HD44780 atau setara (equivalent)

Sumber: [27]

Berdasarkan pada **Tabel 2.7**, LCD 20×4 umumnya menggunakan *driver controller* HD44780 dan dapat diintegrasikan dengan modul antarmuka I2C berbasis PCF8574. Penggunaan modul I2C memungkinkan komunikasi dilakukan melalui jalur serial sehingga jumlah pin mikrokontroler yang digunakan menjadi lebih sedikit. Dengan konfigurasi tersebut, LCD hanya memerlukan 4 jalur utama, yaitu VCC, GND, SDA, dan SCL, sehingga pin *input/output* pada ESP32 dapat dimanfaatkan untuk perangkat lain [27][28].

Komunikasi I2C menggunakan alamat (*address*) yang berbeda pada setiap perangkat yang terhubung pada satu jalur bus, sehingga proses integrasi beberapa perangkat menjadi lebih sederhana [28]. Dari sisi perangkat lunak, LCD dapat dikendalikan menggunakan pustaka (*library*) seperti LiquidCrystal_I2C, yang mempermudah proses penulisan dan pengaturan tampilan karakter. Pada penelitian tugas akhir ini, LCD 20×4 digunakan sebagai media untuk menampilkan informasi yang dihasilkan oleh sistem secara langsung.

Pada prototipe gerbang tol *free flow* yang dikembangkan ini, LCD I2C 20×4 digunakan untuk menampilkan informasi seperti nomor *RFID tag*, status saldo, waktu akses, serta hasil validasi transaksi dari Firebase. Meskipun implementasinya relatif sederhana, penggunaan LCD tetap memerlukan pengaturan catu daya, konfigurasi alamat I2C, dan penyesuaian kontras agar tampilan dapat terbaca dengan baik. Dengan demikian, LCD berfungsi sebagai media informasi yang membantu pengguna memantau proses transaksi dan status sistem secara *real-time*.

2.12 Motor Servo MG996R

Motor servo MG996R merupakan aktuator yang banyak digunakan dalam sistem kendali otomatis karena mampu mengatur posisi sudut berdasarkan sinyal kendali yang diberikan. Servo ini bekerja menggunakan sinyal *Pulse Width Modulation* (PWM) dari mikrokontroler, di mana lebar pulsa menentukan posisi sudut poros servo. Bentuk fisik motor servo MG996R ditunjukkan pada **Gambar 2.10**.



Gambar 2. 10 Motor Servo MG996R
Sumber: [30]

Dilihat dari **Gambar 2.10**, motor servo MG996R merupakan servo digital bertorsi tinggi yang menggunakan sistem roda gigi logam (*metal gear*) sehingga memiliki ketahanan yang lebih baik terhadap beban mekanis dibandingkan dengan servo standar [29]. Pada tugas akhir ini, motor servo MG996R digunakan karena spesifikasinya sesuai dengan yang ditunjukkan pada **Tabel 2.8**.

Tabel 2. 8 Spesifikasi Motor Servo MG996R

Fitur	Spesifikasi
Sudut Rotasi	$\pm 120^\circ$ (60° ke masing-masing arah)
Tegangan Operasi	4.8 V sampai 7.2 V
<i>Stall Torque</i>	9.4 kgf·cm (pada tegangan 4.8 V) 11 kgf·cm (pada tegangan 6 V)
Kecepatan Operasi	0.17 detik / 60° (pada tegangan 4.8 V) 0.14 detik / 60° (pada tegangan 6 V)
Arus Kerja	500 mA - 900 mA (pada tegangan 6 V)
Arus Macet	2.5 A (pada tegangan 6 V)

Sumber: [30]

Berdasarkan **Tabel 2.7**, servo ini bekerja pada rentang tegangan 4,8 V hingga 7,2 V, memiliki torsi sekitar 9,4 kg·cm hingga 11 kg·cm, serta mampu berputar hingga 180° [30]. Motor servo MG996R dilengkapi dengan rangkaian kendali internal yang berfungsi untuk menyesuaikan posisi poros sesuai dengan sinyal

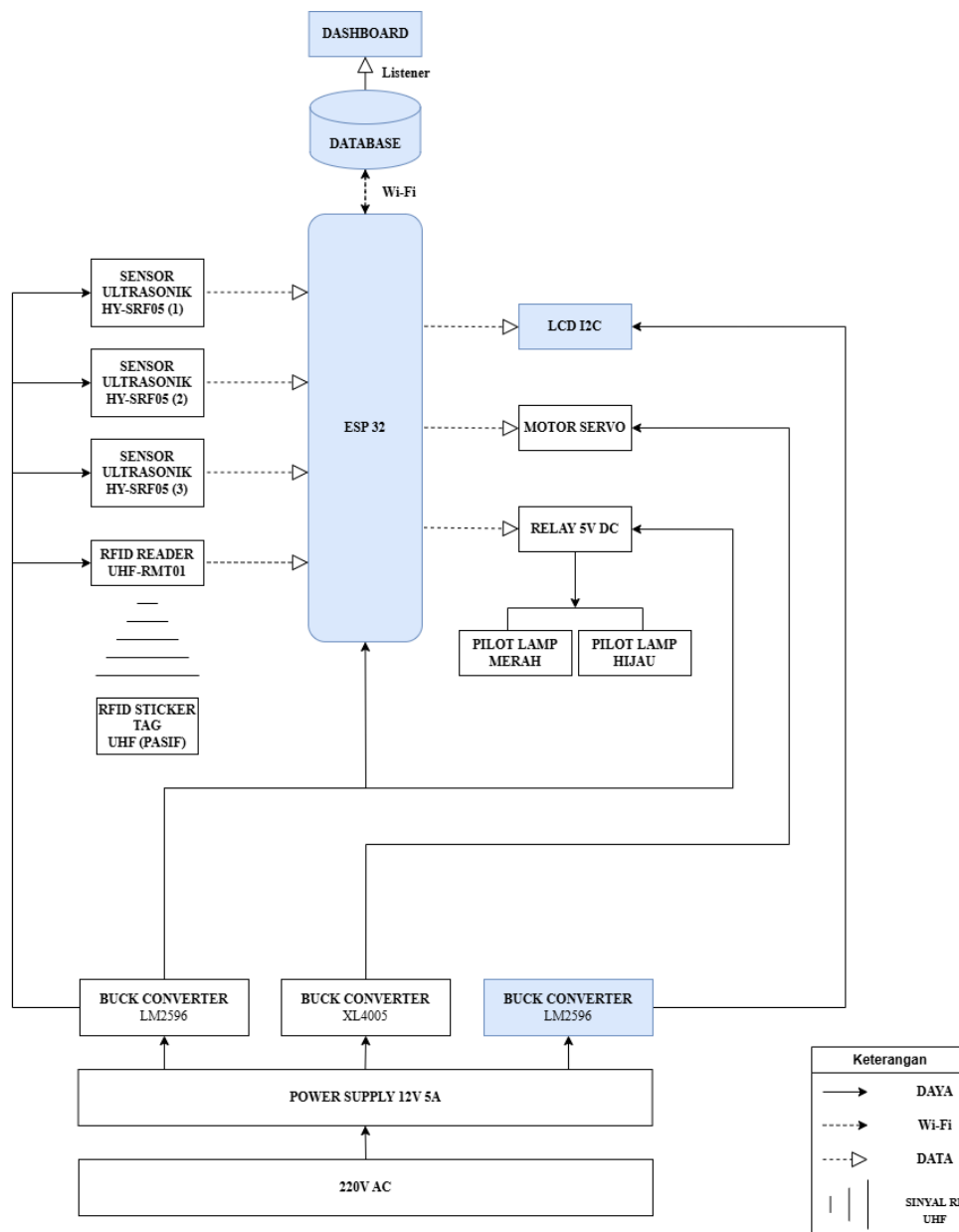
PWM yang diterima. Pada tugas akhir ini, servo digunakan sebagai aktuator untuk menggerakkan palang pada prototipe gerbang tol *free flow*. Servo dikendalikan oleh ESP32 untuk membuka palang setelah proses transaksi dinyatakan berhasil, kemudian mengembalikannya ke posisi semula setelah kendaraan melewati area gerbang. Kemampuan servo dalam melakukan pengaturan posisi secara berulang menjadikannya sesuai untuk mendukung mekanisme gerbang otomatis [41].

Meskipun mudah diintegrasikan dengan mikrokontroler, penggunaan MG996R tetap memerlukan perhatian terhadap kebutuhan catu daya. Berdasarkan *datasheet*, servo ini dapat menarik arus yang cukup besar saat mulai bergerak atau saat menahan beban, sehingga berpotensi menyebabkan penurunan tegangan jika menggunakan sumber daya yang sama dengan mikrokontroler. Oleh karena itu, pada tugas akhir ini motor servo MG996R menggunakan catu daya eksternal yang terpisah dari ESP32. Pengaturan tersebut bertujuan agar proses pengendalian servo dan pengolahan data pada mikrokontroler dapat berjalan stabil selama sistem beroperasi.

BAB III METODE PENELITIAN

3.1 Diagram Blok

Perancangan alat ini membutuhkan skema yang nantinya akan mempermudah alur sistem alat. Skema alat tersebut ditunjukkan pada **Gambar 3.1**.



Gambar 3. 1 Diagram Blok Sistem
Sumber: Hasil Perancangan Penulis

Berdasarkan **Gambar 3.1** yang telah dibuat, sistem ini terdiri dari beberapa bagian utama, yaitu *input* (sensor dan RFID), proses (ESP32), *output* (aktuator dan *display*), serta koneksi ke *database* melalui jaringan *Wi-Fi*.

1. *Power supply* 12V 5A berfungsi untuk menyediakan tegangan yang stabil bagi seluruh rangkaian, yang kemudian diturunkan melalui *buck converter* menjadi tegangan sesuai kebutuhan masing-masing komponen seperti ESP32, sensor ultrasonik, RFID reader, LCD, relay, dan motor servo.
2. *Buck converter* LM2596 dan XL4005 berfungsi untuk menurunkan tegangan 12V dari *power supply* menjadi level tegangan yang sesuai (misalnya 5V) agar aman digunakan oleh mikrokontroler dan modul-modul elektronik lainnya.
3. Mikrokontroler ESP32 berfungsi sebagai pusat pengolahan data (*processing unit*) yang mengatur seluruh jalannya sistem, mulai dari pembacaan data sensor, RFID, hingga pengendalian LCD, motor servo, dan *relay* pada sistem gerbang tol berbasis IoT.
4. Sensor ultrasonik HY-SRF05 (1), (2), dan (3) digunakan untuk mendeteksi keberadaan dan pergerakan kendaraan secara berurutan, sehingga sistem dapat mengenali kondisi kendaraan yang mendekat, berada di area pembacaan, dan melintas keluar dari gerbang tol *free flow*.
5. RFID reader UHF-RMT01 berfungsi untuk membaca identitas kendaraan secara otomatis melalui RFID *tag*, yang menjadi dasar dalam proses identifikasi kendaraan yang masuk ke area gerbang tol.
6. RFID *sticker tag* UHF (pasif) berfungsi sebagai identitas unik kendaraan yang menyimpan data pengguna, sehingga memungkinkan proses transaksi dilakukan secara otomatis tanpa interaksi manual.
7. LCD I2C berfungsi untuk menampilkan informasi transaksi kepada pengguna, seperti status kendaraan terdeteksi, hasil pembacaan RFID, validasi data, serta status saldo dan transaksi.
8. Motor servo berfungsi sebagai aktuator untuk membuka dan menutup palang gerbang tol secara otomatis berdasarkan hasil verifikasi RFID dari ESP32.

9. Relay 5V DC berfungsi sebagai saklar elektronik yang mengendalikan nyala/matinya *pilot lamp* sebagai indikator visual status transaksi/gerbang.
10. *Pilot lamp* merah dan hijau berfungsi sebagai indikator visual, di mana lampu merah menandakan transaksi gagal/kendaraan tidak dikenali dan lampu hijau menandakan transaksi berhasil/kendaraan diizinkan melintas.
11. ESP32 memanfaatkan koneksi *Wi-Fi* untuk terhubung dengan *database* Firebase, yang digunakan sebagai media penyimpanan dan pengolahan data transaksi secara *real-time*.
12. *Database* Firebase berfungsi untuk menyimpan data kendaraan, riwayat transaksi, serta hasil pembacaan RFID, sehingga sistem dapat melakukan *monitoring* dan pengolahan data secara terpusat.

Secara keseluruhan, sistem bekerja dengan cara mendeteksi kendaraan menggunakan sensor, membaca identitas melalui RFID, memproses data di ESP32, kemudian mengirimkan data ke Firebase untuk disimpan dan diolah, serta memberikan respons *output* secara langsung sebagai bagian dari sistem transaksi tol berbasis IoT yang berjalan secara otomatis dan *real-time*. Dengan adanya diagram blok ini, hubungan antarkomponen dalam sistem dapat terlihat dengan jelas sehingga mempermudah proses perancangan dan implementasi sistem.

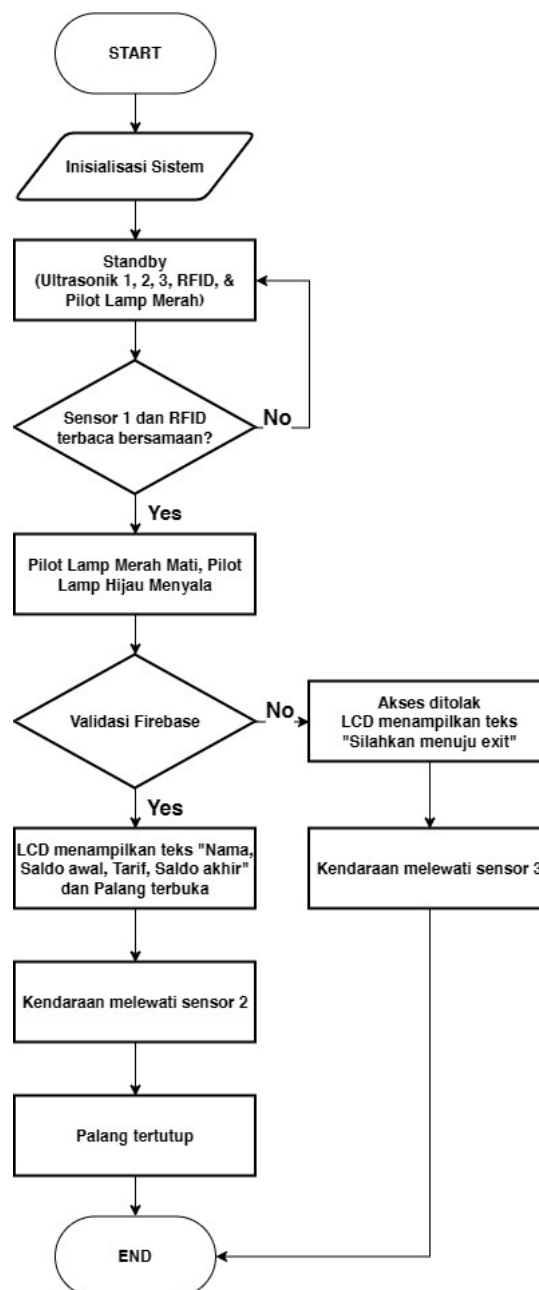
3.2 Flowchart

Pada tahap ini, selain menggunakan diagram blok, penulis juga menyusun *flowchart* untuk menggambarkan alur kerja sistem secara lebih rinci. *Flowchart* ini digunakan untuk mempermudah pemahaman urutan proses yang terjadi pada sistem, mulai dari kondisi awal hingga sistem kembali ke kondisi *standby*.

Dalam penelitian tugas akhir ini, *flowchart* dibagi menjadi dua bagian, yaitu *flowchart* keseluruhan sistem dan *flowchart* khusus untuk sistem transaksi dan pengolahan data berbasis IoT menggunakan Firebase. Pembagian ini dilakukan agar alur sistem dapat dijelaskan secara lebih terstruktur, di mana *flowchart* pertama lebih menekankan proses fisik sistem, sedangkan *flowchart* kedua berfokus pada pengolahan data dan transaksi.

3.2.1 Sistem Keseluruhan

Flowchart sistem secara keseluruhan menjelaskan proses kinerja sistem yang terstruktur, dimulai dari tahap awal hingga tahap akhir. *Flowchart* sistem keseluruhan ini ditampilkan pada **Gambar 3.2**.



Gambar 3. 2 *Flowchart* Keseluruhan Sistem Gerbang Tol *Free Flow*
Sumber: Hasil Perancangan Penulis

Berdasarkan **Gambar 3.2**, sistem dimulai dari proses inisialisasi, di mana ESP32 mengaktifkan seluruh komponen seperti sensor ultrasonik, RFID *reader*,

LCD, dan lampu indikator. Setelah itu, sistem masuk ke kondisi *standby*, yaitu kondisi menunggu kendaraan yang akan melintas. Pada kondisi ini, sensor dan *RFID reader* sudah aktif, sedangkan lampu merah menyala sebagai indikator bahwa sistem belum memproses kendaraan.

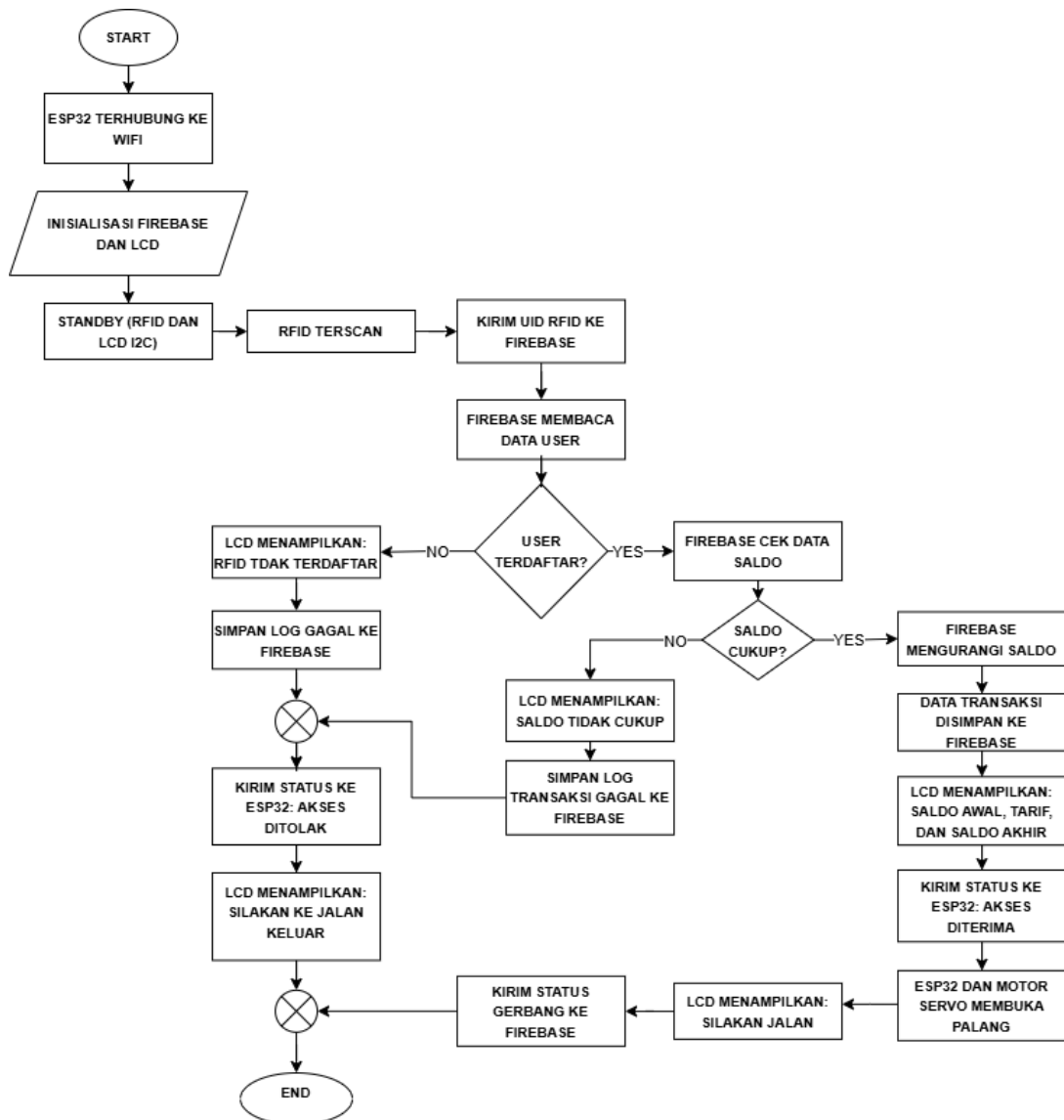
Ketika kendaraan terdeteksi oleh sensor ultrasonik pertama, sistem akan mengaktifkan *RFID reader* untuk membaca *tag* kendaraan. Hasil pembacaan RFID kemudian divalidasi untuk menentukan apakah *tag* tersebut terdaftar atau tidak. Jika *tag* tidak terdaftar, sistem akan menolak akses dengan menyalakan lampu merah dan menampilkan pesan “akses ditolak” pada LCD. Sistem kemudian menunggu hingga kendaraan keluar dari area deteksi sebelum kembali ke kondisi *standby*.

Apabila *tag* terdaftar, sistem akan melanjutkan ke proses pengecekan saldo. Jika saldo tidak mencukupi, sistem juga akan menolak akses dengan menampilkan informasi saldo tidak cukup dan menyalakan lampu merah. Namun, jika saldo mencukupi, sistem akan memberikan akses dengan menyalakan lampu hijau, menampilkan informasi saldo pada LCD, serta membuka palang gerbang menggunakan motor servo.

Selanjutnya, sensor ultrasonik kedua digunakan untuk memastikan kendaraan telah melewati gerbang. Selama kendaraan masih terdeteksi, palang akan tetap terbuka. Setelah kendaraan tidak lagi terdeteksi, palang akan menutup kembali dan sistem akan kembali ke kondisi semula. Proses ini akan terus berulang untuk setiap kendaraan yang melintas.

3.2.2 Sistem Transaksi dan Pengolahan Data Kendaraan

Flowchart sistem transaksi dan pengolahan data menjelaskan alur data kendaraan yang diproses dan disimpan menggunakan Firebase sebagai bagian dari sistem IoT. *Flowchart* ini menjadi bagian penting karena menunjukkan bagaimana sistem tidak hanya bekerja secara lokal, tetapi juga terhubung dengan *database* secara *real-time*. *Flowchart* sistem keseluruhan ini ditunjukkan pada **Gambar 3.3**.



Gambar 3. 3 Flowchart Sistem Transaksi dan Pengolahan Data Kendaraan
Sumber: Hasil Perancangan Penulis

Berdasarkan **Gambar 3.3**, proses dimulai dengan ESP32 yang terlebih dahulu terhubung ke jaringan *Wi-Fi*, kemudian dilanjutkan dengan inisialisasi koneksi ke Firebase dan LCD. Setelah sistem siap, kondisi *standby* diaktifkan dan *RFID reader* menunggu kendaraan yang akan dipindai.

Ketika *RFID tag* terbaca, sistem akan mengirimkan UID ke Firebase. *Database* kemudian akan membaca data pengguna berdasarkan UID tersebut dan melakukan validasi apakah pengguna terdaftar atau tidak. Jika pengguna tidak terdaftar, sistem akan menampilkan pesan “RFID tidak terdaftar” pada LCD,

menyimpan data sebagai *log* transaksi gagal di Firebase, serta mengirimkan status akses ditolak ke ESP32.

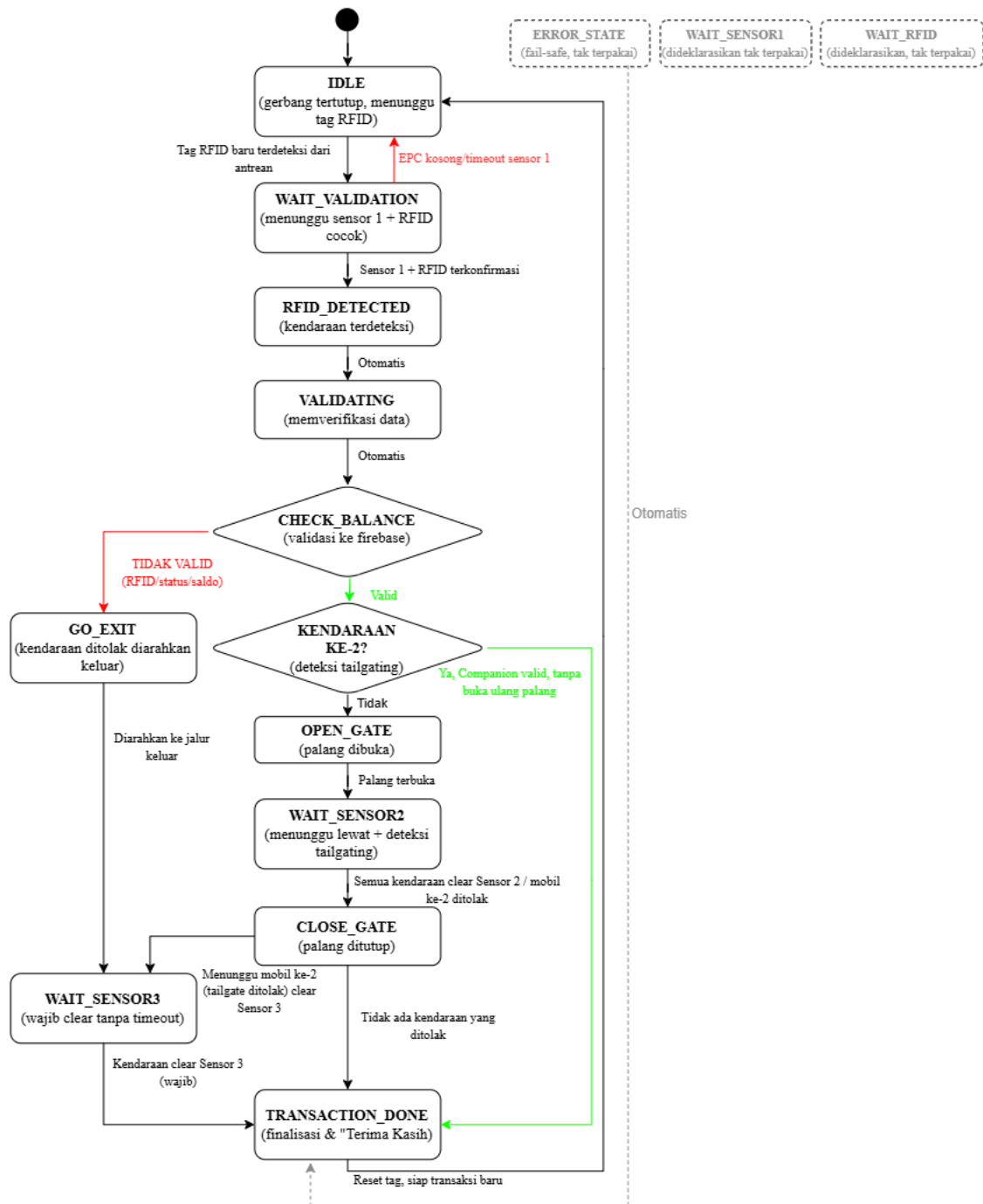
Jika pengguna terdaftar, sistem akan melanjutkan pengecekan saldo di *database*. Apabila saldo tidak mencukupi, sistem akan menampilkan pesan “saldo tidak cukup”, serta menyimpan data transaksi gagal sebagai *log* di Firebase. Namun, jika saldo mencukupi, sistem akan melanjutkan proses transaksi.

Pada tahap transaksi, Firebase akan mengurangi saldo sesuai tarif yang ditentukan, lalu menyimpan data transaksi yang meliputi ID kendaraan, waktu transaksi, tarif, dan sisa saldo. Setelah itu, informasi transaksi ditampilkan pada LCD, yaitu saldo awal, tarif, dan saldo akhir.

Selanjutnya, Firebase mengirimkan status transaksi berhasil ke ESP32, sehingga sistem merespons dengan membuka palang gerbang dan menampilkan pesan “silakan jalan”. Selain itu, sistem juga mengirimkan status gerbang ke Firebase sebagai bagian dari *monitoring* sistem. Setelah seluruh proses selesai, sistem kembali ke kondisi *standby* dan siap memproses kendaraan berikutnya.

3.2.3 Diagram State *Finite State Machine* (FSM)

Selain *flowchart* proses yang telah dijelaskan sebelumnya, sistem transaksi pada tugas akhir ini juga dirancang menggunakan pendekatan *Finite State Machine* (FSM) untuk menstrukturkan alur transisi antarkondisi secara eksplisit. Berbeda dengan *flowchart* yang menggambarkan urutan proses secara umum, diagram *state* pada **Gambar 3.4** menunjukkan kondisi (*state*) spesifik yang dapat ditempati sistem beserta kejadian yang memicu perpindahan antar-*state* tersebut.



Gambar 3.4 Diagram State *Finite State Machine*

Sumber: Hasil Perancangan Penulis

Dari **Gambar 3.4**, sistem dimulai dari *state* IDLE, yaitu kondisi gerbang tertutup sambil menunggu *tag* RFID terdeteksi. Ketika *tag* RFID baru terdeteksi dari antrian, sistem berpindah ke *state* WAIT_VALIDATION untuk menunggu kecocokan antara Sensor 1 dan pembacaan RFID. Apabila EPC kosong atau Sensor

1 mengalami *timeout*, sistem akan kembali ke *state* IDLE. Jika Sensor 1 dan RFID terkonfirmasi cocok, sistem berpindah ke *state* RFID_DETECTED, kemudian secara otomatis menuju ke *state* VALIDATING untuk memverifikasi data kendaraan.

Setelah proses verifikasi, sistem melakukan pengecekan saldo pada *state* CHECK_BALANCE melalui validasi ke Firebase. Apabila data tidak valid (RFID atau saldo tidak sesuai), sistem berpindah ke *state* GO_EXIT dan mengarahkan kendaraan menuju jalur keluar tanpa membuka palang. Sebaliknya, apabila data valid, sistem berpindah ke *state* OPEN_GATE untuk membuka palang, dilanjutkan ke *state* WAIT_SENSOR2 yang menunggu kendaraan melintas sekaligus mendeteksi kemungkinan *tailgating* (kendaraan kedua yang mengikuti tanpa validasi terpisah). Setelah kendaraan melintasi Sensor 2 atau kendaraan kedua ditolak, sistem berpindah ke *state* CLOSE_GATE untuk menutup palang, kemudian menuju *state* TRANSACTION_DONE sebagai tahap finalisasi transaksi.

Pada skenario *tailgating*, apabila kendaraan kedua (*companion*) dinyatakan valid, sistem dapat langsung menuju *state* TRANSACTION_DONE tanpa membuka ulang palang kembali. Sementara itu, pada jalur penolakan (GO_EXIT), sistem tetap mewajibkan kendaraan untuk melewati *state* WAIT_SENSOR3 sebelum kembali menuju TRANSACTION_DONE, guna memastikan area gerbang benar-benar kosong sebelum sistem mereset *tag* dan siap menerima transaksi berikutnya. Perancangan ini turut mengantisipasi kondisi kegagalan melalui *state* cadangan seperti ERROR_STATE, WAIT_SENSOR1, dan WAIT_RFID yang dideklarasikan dalam program sebagai mekanisme *fail-safe*, meskipun pada pengujian normal ketiga *state* tersebut belum digunakan karena seluruh skenario pengujian berjalan sesuai dengan alur utama yang dirancang.

Perancangan FSM pada **Gambar 3.4** diimplementasikan langsung ke dalam *firmware* ESP32. Cuplikan kode fungsi pengendali (*controller*) yang menjalankan logika perpindahan *state* tersebut ditunjukkan pada **Gambar 3.5**.

```

switch(currentState)
{
    case IDLE:                Serial.println("IDLE"); break;
    case WAIT_SENSOR1:        Serial.println("WAIT_SENSOR1"); break;
    case WAIT_RFID:           Serial.println("WAIT_RFID"); break;
    case WAIT_VALIDATION:     Serial.println("WAIT_VALIDATION"); break;
    case RFID_DETECTED:       Serial.println("RFID_DETECTED"); break;
    case VALIDATING:          Serial.println("VALIDATING"); break;
    case CHECK_BALANCE:       Serial.println("CHECK_BALANCE"); break;
    case OPEN_GATE:           Serial.println("OPEN_GATE"); break;
    case WAIT_SENSOR2:        Serial.println("WAIT_SENSOR2"); break;
    case GO_EXIT:             Serial.println("GO_EXIT"); break;
    case WAIT_SENSOR3:        Serial.println("WAIT_SENSOR3"); break;
    case CLOSE_GATE:          Serial.println("CLOSE_GATE"); break;
    case TRANSACTION_DONE:    Serial.println("TRANSACTION_DONE"); break;
}

```

Gambar 3.5 Kode Program Cuplikan Fungsi Dispatcher *runFSM()*

Sumber: Hasil Perancangan Penulis

Fungsi pengendali ini dipanggil pada setiap iterasi *loop* program, memeriksa kondisi yang sedang aktif, lalu menjalankan rutinitas penanganan yang sesuai dengan kondisi tersebut. Struktur ini juga dilengkapi dengan mekanisme pencatatan (*logging*) otomatis yang mendeteksi setiap kali terjadi perpindahan *state*, kemudian mencetaknya ke Serial Monitor, sehingga transisi antar-*state* dapat ditelusuri secara *real-time* selama pengujian.

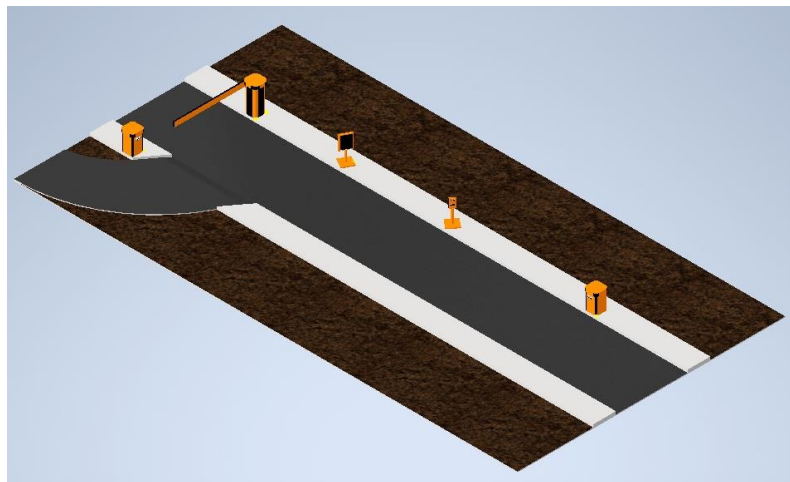
Perlu dicatat bahwa *state* WAIT_SENSOR1 dan WAIT_RFID pada **Gambar 3.4** merupakan *state* cadangan yang dirancang sebagai mekanisme *fail-safe*. Namun, pada alur transaksi normal kedua *state* tersebut tidak pernah dimasuki, karena perpindahan dari *state* IDLE menuju WAIT_VALIDATION terjadi secara langsung begitu kendaraan baru terdeteksi dari antrean RFID. Kondisi ini sejalan dengan penandaan pada **Gambar 3.4** yang menunjukkan kedua *state* tersebut sebagai *state* cadangan yang belum terpakai dalam skenario pengujian yang dilakukan.

Sebagai catatan tambahan, proses validasi saldo pada *state* CHECK_BALANCE berkomunikasi dengan Firebase secara sinkron melalui mekanisme *task notification* pada *task* jaringan yang berjalan terpisah di salah satu core ESP32, sementara *task* FSM tetap berjalan di core lain untuk terus memantau sensor secara *non-blocking*. Pendekatan *multitasking* dua core ini turut mendukung

sinkronisasi antara deteksi kendaraan, validasi data, dan eksekusi aktuator agar tetap berjalan stabil, termasuk pada skenario kendaraan beriringan (*tailgating*) yang memerlukan proses validasi kendaraan kedua tanpa mengganggu pemantauan sensor kendaraan pertama.

3.3 Gambar 3D

Dalam perancangan tugas akhir ini, dibuatlah gambar 3D dari prototipe yang bertujuan sebagai gambaran penulis dalam mengembangkan alat atau prototipe yang dibuat. Berikut merupakan bentuk 3d dari prototipe sistem transaksi gerbang tol *free flow* yang tertera pada **Gambar 3.6**.

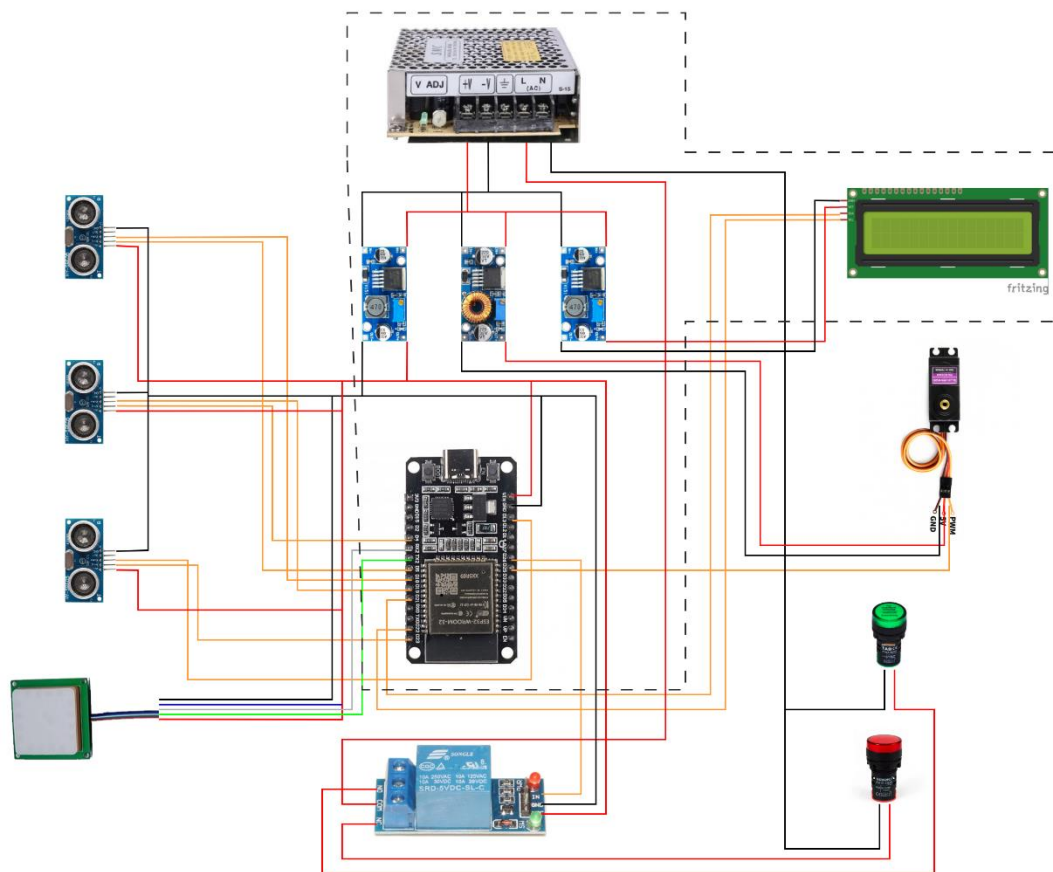


Gambar 3. 6 3D Design Gerbang Tol Free Flow
Sumber: Hasil Perancangan Penulis

Pada **Gambar 3.6**, terdapat sensor ultrasonik pertama untuk mendeteksi kendaraan yang melintas. Kemudian terdapat UHF RFID Reader dan Pilot Lamp Merah dan Hijau yang akan membaca UHF RFID Tag yang melintas serta indikator pada lampu apakah tag tersebut terdaftar atau tidak. Selanjutnya ada layar LCD I2C 20×4 untuk menampilkan informasi terkait tarif, saldo awal, dan saldo akhir. Palang di sistem ini juga terdapat sensor ultrasonik kedua yang berfungsi untuk mendeteksi kendaraan yang akan melewati palang, sehingga memberikan informasi agar palang kembali menutup. Terakhir, terdapat sensor ultrasonik ketiga yang berfungsi untuk mendeteksi kendaraan yang tag-nya tidak terdaftar maupun yang saldonya tidak mencukupi tarif yang telah ditentukan.

3.4 Skema Pengkabelan (Wiring Diagram)

Skema pengkabelan sistem gerbang tol *free flow* menggambarkan bagaimana seluruh komponen elektronik, mulai dari sumber daya, mikrokontroler, sensor, hingga aktuator, saling terhubung untuk membentuk satu kesatuan sistem yang dikendalikan oleh ESP32. Secara umum, arsitektur pengkabelan dibagi menjadi tiga blok utama: blok catu daya, blok akuisisi data (sensor & identifikasi), dan blok aktuator & indikator (*output*), dengan ESP32 sebagai pusat pemrosesan yang menjembatani ketiganya. Rancangan pengkabelan keseluruhan sistem ditunjukkan pada **Gambar 3.7**.



Gambar 3. 7 Skema Pengkabelan Sistem Gerbang Tol *Free Flow*
Sumber: Hasil Perancangan Penulis

Berdasarkan **Gambar 3.7**, sumber daya utama berasal dari modul *power supply* AC-DC yang menghasilkan tegangan DC dengan level yang dapat diatur (*adjustable*) melalui trimpot pada modul tersebut. Tegangan keluaran ini kemudian didistribusikan ke tiga modul *step-down converter* (*buck converter*) yang

menurunkan tegangan hingga level yang sesuai dengan kebutuhan tiap komponen (ESP32, sensor ultrasonik, servo, dan LCD), sehingga setiap perangkat mendapat suplai tegangan yang stabil tanpa membebani satu jalur regulator secara langsung.

Sebagaimana terlihat pada **Gambar 3.7**, blok akuisisi data terdiri dari tiga buah sensor ultrasonik yang ditempatkan pada titik-titik berbeda di sepanjang jalur gerbang untuk mendeteksi keberadaan dan posisi kendaraan secara berurutan, sensor 1 sebagai penanda kendaraan memasuki jalur, sensor 2 sebagai penanda kendaraan telah melewati palang, dan sensor 3 sebagai penanda kendaraan telah melewati jalur keluar. Di samping itu, modul pembaca RFID terhubung ke ESP32 melalui jalur UART dan berfungsi memvalidasi identitas kendaraan/pengguna berdasarkan tag RFID yang terbaca.

Pada bagian keluaran, seperti tampak pada **Gambar 3.7**, ESP32 mengendalikan sebuah motor servo yang berfungsi sebagai penggerak palang pintu, serta sebuah modul *relay* satu kanal yang berfungsi sebagai saklar pengalih (*changeover switch*) untuk dua lampu indikator (hijau dan merah), sehingga hanya satu pin GPIO yang dibutuhkan untuk mengendalikan status "boleh lewat" dan "dilarang lewat" secara bergantian. Informasi status sistem juga ditampilkan secara *real-time* kepada pengguna melalui LCD karakter 20×4 yang berkomunikasi dengan ESP32 melalui protokol I2C, sehingga hanya membutuhkan 2 jalur data (SDA dan SCL).

Seluruh pin GPIO yang digunakan didefinisikan secara terpusat pada bagian pengaturan awal program, sehingga memudahkan penyesuaian apabila terjadi perubahan wiring di kemudian hari. Rincian koneksi setiap komponen ke pin ESP32, yang menjadi acuan detail pada **Gambar 3.7**, ditampilkan pada **Tabel 3.1** berikut.

Tabel 3.1 Konfigurasi Pin

Komponen	Pin Komponen	Pin ESP32 (GPIO)
Sensor Ultrasonik 1	TRIG	GPIO 33
	ECHO	GPIO 32
Sensor Ultrasonik 2	TRIG	GPIO 13
	ECHO	GPIO14

Komponen	Pin Komponen	Pin ESP32 (GPIO)
Sensor Ultrasonik 3	TRIG	GPIO 18
	ECHO	GPIO 5
Modul <i>RFID Reader</i>	TX (modul)	GPIO 16 (RX2)
	RX (modul)	GPIO 17 (TX2)
Motor Servo	Signal (PWM)	GPIO 27
Modul Relay 1 Channel	IN	GPIO 23
LCD I2C	SDA	GPIO 21
	SCL	GPIO 22
<i>Power Supply</i> dan Konverter	VCC/GND	5V, 3.3V, GND

Sumber : Hasil Perancangan Penulis

Berdasarkan **Tabel 3.1**, dapat dilihat bahwa setiap komponen memanfaatkan jalur GPIO yang berbeda sesuai dengan fungsinya masing-masing, baik sebagai antarmuka digital sederhana (TRIG/ECHO, servo, *relay*) maupun antarmuka komunikasi serial (UART untuk RFID dan I2C untuk LCD). Pemetaan pin yang jelas ini memastikan tidak terjadi konflik alamat maupun tumpang tindih fungsi antar-pin selama sistem beroperasi.

Secara keseluruhan, skema pengkabelan yang telah dipaparkan pada **Gambar 3.7** dan **Tabel 3.1** menunjukkan bahwa sistem gerbang tol *free flow* dirancang dengan pembagian jalur yang terstruktur antara blok catu daya, blok akuisisi data, dan blok aktuator/indikator. Struktur pengkabelan ini menjadi dasar bagi ESP32 untuk menjalankan seluruh proses deteksi kendaraan, validasi identitas, hingga pengendalian palang dan lampu indikator secara terkoordinasi.

3.5 Spesifikasi dan Fitur

Berdasarkan kebutuhan konsep gerbang tol *free flow* yang ingin dibuat, spesifikasi ini disusun tidak langsung jadi, tetapi melalui beberapa kali penyesuaian antara komponen yang tersedia, kemampuan sistem, dan tujuan utama, yaitu membuat sistem transaksi tol yang berjalan otomatis dan terhubung ke *database* secara *real-time*.

3.5.1 Spesifikasi

Spesifikasi alat yang dirancang adalah sebagai berikut:

1. Sistem menggunakan mikrokontroler ESP32 sebagai pusat pengolahan data yang dilengkapi dengan koneksi Wi-Fi untuk mendukung konsep IoT.
2. Sistem dilengkapi dengan 3 sensor ultrasonik HY-SRF05 yang digunakan untuk mendeteksi keberadaan dan posisi kendaraan secara bertahap di area gerbang.
3. Sistem menggunakan RFID *reader* UHF (RMT01) yang mampu membaca RFID *tag* pasif (UHF *sticker*) sebagai identitas kendaraan tanpa perlu kontak langsung.
4. Sistem terhubung dengan *database* Firebase yang berfungsi sebagai penyimpanan data transaksi dan pengolahan data secara *real-time*.
5. Sistem memiliki LCD I2C sebagai media tampilan untuk menampilkan informasi seperti status kendaraan, hasil pembacaan RFID, dan kondisi sistem.
6. Sistem menggunakan motor servo sebagai aktuator untuk mensimulasikan mekanisme buka/tutup gerbang pada prototipe.
7. Sistem dilengkapi dengan modul *relay* 5V DC untuk mengontrol lampu indikator (merah dan hijau) sebagai penanda status kendaraan.
8. Sistem menggunakan *power supply* 12V 5A untuk memastikan semua komponen mendapatkan tegangan yang stabil selama sistem bekerja.
9. Sistem dirancang untuk bekerja secara otomatis tanpa *input* manual, sesuai dengan konsep *free flow* pada gerbang tol modern.

3.5.2 Fitur

Fitur yang dikembangkan pada sistem ini disesuaikan dengan kebutuhan sistem transaksi tol berbasis IoT, yaitu:

1. Data transaksi langsung dikirim ke database melalui *Wi-Fi*, sehingga dapat dipantau secara *real-time*.
2. Deteksi kendaraan dilakukan secara otomatis menggunakan tiga sensor ultrasonik yang bekerja secara berurutan sesuai posisi kendaraan.

3. Identifikasi kendaraan dilakukan secara otomatis melalui pembacaan *RFID tag* UHF tanpa memerlukan kontak langsung.
4. Validasi status RFID (terdaftar/tidak terdaftar) dan saldo pengguna dilakukan secara otomatis melalui *database* Firebase.
5. Saldo pengguna berkurang secara otomatis sesuai tarif ketika transaksi berhasil dilakukan.
6. Palang gerbang terbuka dan tertutup secara otomatis menggunakan motor servo berdasarkan hasil verifikasi RFID dan saldo.
7. Status transaksi ditampilkan melalui indikator visual *pilot lamp* merah dan hijau.
8. Informasi transaksi seperti saldo awal, tarif, dan saldo akhir, ditampilkan secara *real-time* pada LCD I2C.
9. Seluruh data kendaraan dan transaksi tersimpan di Firebase sebagai *log*, sehingga memudahkan proses *monitoring* dan analisis data.

3.6 Teknik Fabrikasi

Tahapan dalam perancangan dan pembuatan sistem transaksi gerbang tol *free flow* terdiri dari beberapa bagian utama, di antaranya:

3.6.1 Alat dan Bahan

Tugas akhir ini membutuhkan berbagai peralatan dan bahan yang mendukung sistem transaksi gerbang tol *free flow* secara *real-time*. Alat dan bahan yang digunakan terdapat pada **Tabel 3.2**.

Tabel 3. 2 Alat dan Bahan

Alat dan Bahan	Fungsi
PSU 12V DC	Sumber tegangan untuk menyuplai daya ke seluruh sistem agar dapat bekerja dengan stabil
ESP32	Mikrokontroler utama untuk mengolah data dari sensor, mengontrol <i>output</i> , serta menghubungkan sistem ke Firebase melalui <i>Wi-Fi</i>

Alat dan Bahan	Fungsi
Sensor Ultrasonik HY-SRF05	Pendeteksi keberadaan dan posisi kendaraan pada area gerbang tol
UHF RFID Reader EL-UHF-RMT01	Pembaca identitas kendaraan melalui RFID tag secara otomatis tanpa kontak
FID Tag	Identitas unik kendaraan yang digunakan dalam proses transaksi tol
Motor Servo MG996R	Aktuator untuk menggerakkan palang gerbang (buka dan tutup) pada prototipe
LCD I2C	Penampilkan informasi sistem seperti status kendaraan dan hasil transaksi
Pilot Lamp Merah dan Hijau	Indikator visual status kendaraan, apakah diizinkan lewat atau tidak

Sumber: Hasil Perancangan Penulis

3.6.2 Perancangan Sistem

Perancangan fabrikasi sistem transaksi gerbang tol *free flow* ini meliputi beberapa tahapan proses, yaitu:

1. Pembuatan Struktur Prototipe

Pembuatan struktur prototipe dilakukan sebagai dasar dari seluruh sistem yang akan dirancang. Proses yang dilakukan meliputi pemotongan bahan sesuai desain jalur kendaraan, pembuatanudukan untuk sensor dan aktuator, serta perakitan menggunakan baut atau lem. Pada tahap ini, penempatan posisi komponen sudah mulai dipikirkan agar alur kendaraan dapat terbaca dengan baik oleh sistem.

2. Pemasangan Sistem Deteksi dan Identifikasi

Tahap ini berfokus pada pemasangan sensor ultrasonik dan RFID reader sebagai bagian utama dalam proses transaksi. Sensor ultrasonik dipasang pada beberapa titik untuk mendeteksi keberadaan kendaraan secara bertahap, sedangkan RFID reader ditempatkan pada posisi yang memungkinkan pembacaan tag dilakukan tanpa harus berhenti. Penentuan posisi ini tidak langsung optimal, sehingga dilakukan beberapa penyesuaian agar pembacaan data menjadi lebih akurat dan stabil.

3. Perakitan Sistem Elektronik dan Kontrol

Pada tahap ini dilakukan perakitan seluruh rangkaian elektronik dengan ESP32 sebagai pusat kontrol. Semua komponen seperti sensor, RFID, LCD, *relay*, dan motor servo dihubungkan sesuai dengan diagram blok yang telah dirancang. Beberapa sambungan dibuat permanen menggunakan solder agar lebih kuat, sementara penataan kabel juga diperhatikan agar rapi dan memudahkan proses pengecekan jika terjadi kesalahan. Tahap ini cukup penting karena kesalahan kecil pada koneksi dapat memengaruhi keseluruhan sistem.

4. Pemasangan Aktuator dan Indikator

Aktuator dan indikator dipasang sebagai *output* dari proses sistem. Motor servo digunakan untuk mensimulasikan gerbang yang dapat membuka dan menutup secara otomatis, sedangkan lampu indikator merah dan hijau digunakan untuk memberikan informasi visual terkait status kendaraan. Selain itu, LCD dipasang untuk menampilkan informasi transaksi secara langsung. Penempatan komponen ini disesuaikan agar mudah dilihat dan merepresentasikan kondisi sistem dengan jelas.

5. Integrasi Sistem IoT dan Pengujian

Tahap terakhir adalah integrasi sistem dengan Firebase sebagai bagian dari implementasi IoT. ESP32 dikonfigurasi agar dapat terhubung ke *Wi-Fi* dan mengirim data ke *database* secara *real-time*. Setelah itu dilakukan pengujian secara keseluruhan, mulai dari deteksi kendaraan, pembacaan RFID, proses transaksi, hingga pengiriman data ke Firebase. Jika ditemukan kendala, dilakukan perbaikan baik dari sisi perangkat keras maupun perangkat lunak, agar sistem dapat bekerja secara optimal.

3.7 Metode Analisis Data

Untuk mengukur kinerja sistem yang telah dirancang, digunakan beberapa metode analisis kuantitatif terhadap data hasil pengujian, yaitu perhitungan *error*, akurasi, dan nilai rata-rata (*mean*). Metode-metode ini digunakan secara konsisten dalam seluruh pengujian pada BAB IV, baik untuk data numerik (jarak, sudut, RSSI) maupun data kategorikal (status identifikasi, status transaksi).

3.7.1 Perhitungan Error

Error dihitung dengan membandingkan nilai yang terukur oleh sistem dengan nilai aktual/acuan, dinyatakan dalam bentuk persentase. Persamaan yang digunakan adalah sebagai berikut:

$$Error (\%) = \frac{|X_{ukur} - X_{aktual}|}{X_{aktual}} \times 100\% \quad (3.1)$$

dengan:

- X_{ukur} = nilai yang terbaca/tercatat oleh sistem (misalnya jarak hasil pembacaan sensor, atau sudut aktual servo).
- X_{aktual} = nilai acuan/sebenarnya (misalnya jarak hasil pengukuran alat ukur fisik, atau sudut perintah yang diberikan)

3.7.2 Perhitungan Akurasi

Untuk data yang bersifat kategorikal (berhasil/gagal, terdeteksi/tidak terdeteksi), akurasi dihitung berdasarkan proporsi jumlah data yang sesuai terhadap keseluruhan jumlah percobaan:

$$Akurasi (\%) = \frac{\sum \text{Data Sesuai}}{\sum \text{Total Percobaan}} \times 100\% \quad (3.2)$$

3.7.3 Perhitungan Rata-Rata (Mean)

Nilai rata-rata digunakan untuk merepresentasikan kecenderungan umum dari serangkaian data hasil pengujian yang dilakukan secara berulang:

$$\bar{X} = \frac{\sum X_i}{n} \quad (3.3)$$

dengan X_i adalah nilai data pada percobaan ke-i, dan n adalah jumlah total percobaan.

3.7.4 Estimasi Total Waktu Proses

Untuk mengetahui total waktu yang dibutuhkan sistem dalam satu siklus transaksi, dilakukan penjumlahan terhadap waktu tiap tahapan komunikasi dan pemrosesan data:

$$t_{total} = t_{RFID \rightarrow FB} + t_{FB \rightarrow ESP32} + t_{proses} + t_{tulis log} \quad (3.4)$$

dengan $t_{RFID \rightarrow FB}$ adalah waktu pengiriman UID dari RFID reader ke Firebase, $t_{FB \rightarrow ESP32}$ adalah waktu pengambilan data pengguna dari Firebase ke ESP32, t_{proses} adalah waktu validasi dan eksekusi transaksi pada ESP32, dan $t_{tulis log}$ adalah waktu penulisan hasil transaksi ke *log* Firebase.