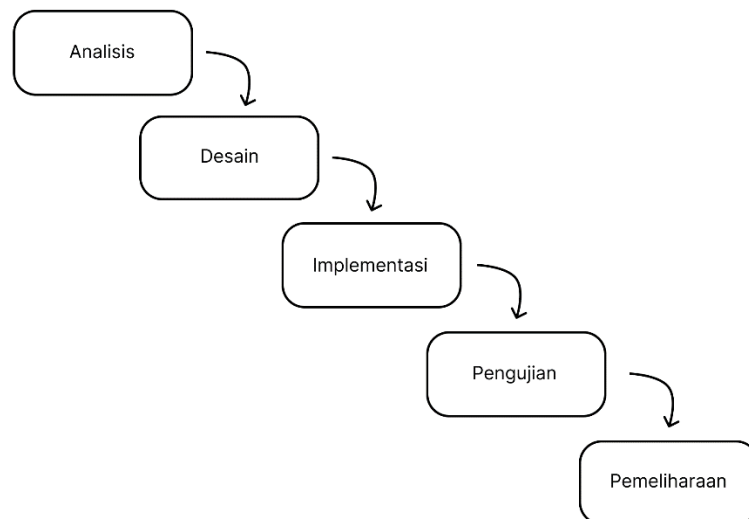


BAB III METODE PENELITIAN

3.1 Metode Penelitian

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering (SE)*. saat ini model waterfall merupakan model pengembangan perangkat lunak yang sering digunakan. Model pengembangan ini melakukan pendekatan secara sistematis dan berurutan. Disebut waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya [14].



Gambar 3. 1 *Waterfall*

3.1.1 Analisis

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi, atau survei langsung. Informasi yang telah dikumpulkan kemudian dianalisis secara mendalam untuk mendapatkan data yang benar-benar dibutuhkan oleh pengguna. Hasil analisis ini akan menjadi dasar dalam merumuskan kebutuhan sistem (requirement), baik kebutuhan fungsional maupun nonfungsional, sehingga perangkat lunak yang dikembangkan dapat berjalan sesuai dengan tujuan, kebutuhan pengguna, serta kondisi lingkungan tempat sistem tersebut akan diterapkan. Selain itu, tahap ini juga membantu meminimalkan kesalahan dalam pengembangan sistem di tahap selanjutnya karena kebutuhan pengguna telah didefinisikan dengan jelas sejak awal.

3.1.2 Desain

Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (hardware) dan sistem persyaratan serta membantu dalam mendefinisikan arsitektur sistem secara keseluruhan. Desain ini mencakup perancangan struktur data, antarmuka pengguna (user interface), alur proses sistem, serta hubungan antar komponen yang akan saling berinteraksi. Dengan adanya desain yang matang, pengembang dapat memiliki gambaran yang jelas mengenai cara kerja sistem sebelum proses pengkodean dimulai, sehingga dapat mengurangi risiko kesalahan implementasi. Selain itu, tahap perancangan ini juga berfungsi sebagai pedoman bagi seluruh tim pengembang agar proses pembangunan sistem dapat berjalan secara terarah, konsisten, dan sesuai dengan kebutuhan yang telah dianalisis pada tahap sebelumnya.

3.1.3 Implementasi

Pada tahap ini, sistem pertama kali dikembangkan dalam program-program kecil yang disebut unit, yang kemudian akan diintegrasikan pada tahap selanjutnya. Setiap unit dikembangkan secara terpisah sesuai dengan fungsi yang telah dirancang sebelumnya dan diuji untuk memastikan fungsionalitasnya berjalan dengan baik, yang disebut sebagai *unit testing*. Pengujian ini bertujuan untuk menemukan dan memperbaiki kesalahan sejak dini, sehingga setiap unit dapat bekerja secara optimal sebelum digabungkan dengan unit lainnya. Dengan adanya pengujian pada tingkat unit, kualitas kode dapat lebih terjaga dan risiko terjadinya kesalahan yang lebih besar pada tahap integrasi dan pengujian sistem secara keseluruhan dapat diminimalkan.

3.1.4 Pengujian

Pada tahap ini, sistem dilakukan proses verifikasi dan pengujian untuk memastikan apakah sistem, baik secara keseluruhan maupun sebagian, telah memenuhi persyaratan yang telah ditetapkan. Pengujian dapat dikategorikan ke dalam beberapa jenis, yaitu *unit testing* yang dilakukan pada modul atau bagian kode tertentu untuk memastikan setiap fungsi berjalan dengan benar, *system testing* yang bertujuan untuk melihat bagaimana sistem bereaksi ketika seluruh modul telah terintegrasi dan dijalankan secara bersamaan, serta *acceptance testing* yang dilakukan bersama atau atas nama pelanggan untuk memastikan bahwa sistem telah memenuhi seluruh kebutuhan dan harapan pengguna. Melalui tahap pengujian ini, pengembang dapat mengidentifikasi dan memperbaiki kesalahan atau kekurangan sebelum sistem diimplementasikan, sehingga kualitas dan keandalan sistem yang dihasilkan dapat terjamin.

3.1.5 Pemeliharaan

Ini adalah tahap akhir dari metode *waterfall*. Perangkat lunak yang telah selesai dikembangkan kemudian dijalankan dan digunakan oleh pengguna, serta dilakukan proses pemeliharaan secara berkelanjutan. Pemeliharaan ini mencakup kegiatan memperbaiki kesalahan atau bug yang tidak ditemukan pada tahap sebelumnya, melakukan penyesuaian terhadap perubahan kebutuhan pengguna, serta meningkatkan kinerja dan keamanan sistem. Selain itu, tahap pemeliharaan juga bertujuan untuk memastikan perangkat lunak tetap dapat beroperasi secara optimal dalam jangka waktu yang panjang dan tetap relevan dengan perkembangan teknologi maupun lingkungan pengguna.

3.2 Tempat Penelitian

Untuk tempat penelitian, peneliti melakukan penelitian di Kantor Kepala Desa Lingga Tiga yang terdapat di Lingga Tiga I, Lingga Tiga, Bilah Hulu, Labuhanbatu, Sumatera Utara 21451, N Empat Aek Nabara, Kec. Bilah Hulu, Kab. Labuhanbatu, Sumatera Utara 21422.

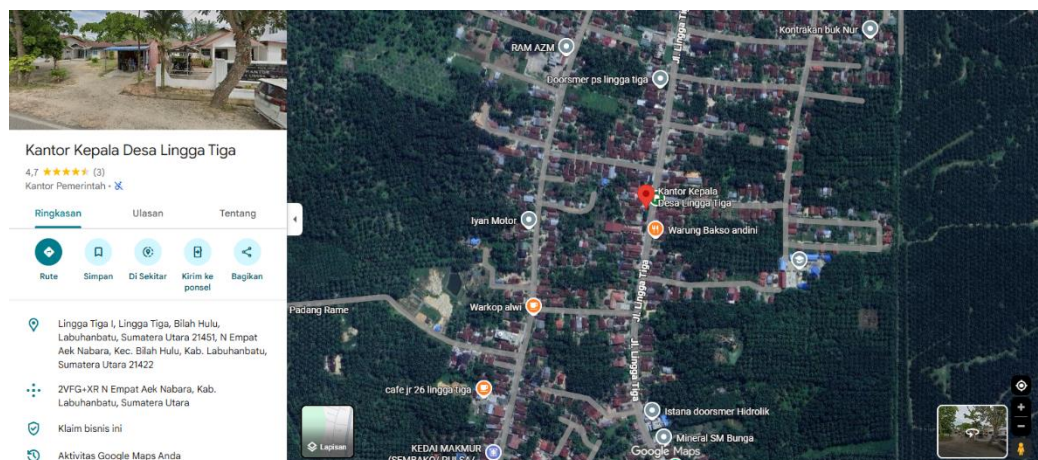
Tabel 3. 1 Waktu Penelitian

No	Kegiatan	Tahun 2025 s.d 2026				
		November	Desember	Januari	Februari	Maret
1.	Analisis					
2.	Desain					
3.	Implementasi					
4.	Pengujian					
5.	Pemeliharaan					

Tabel 3.1 menyajikan jadwal pelaksanaan penelitian yang dilakukan pada rentang waktu November 2025 sampai dengan Maret 2026. Tahapan yang direncanakan meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Penyusunan jadwal tersebut menunjukkan bahwa penelitian dilaksanakan secara bertahap dan sistematis, sehingga setiap tahapan memiliki alokasi waktu yang jelas serta saling berkaitan satu sama lain. Dengan demikian, pelaksanaan penelitian dapat berlangsung secara terarah sesuai dengan metode pengembangan yang digunakan dalam penelitian ini.

3.3 Waktu Penelitian

Dalam melaksanakan penyusunan skripsi penelitian dilakukan selama empat bulan, dimulai pada November 2025 sampai Maret 2026, dengan fokus pada pengambilan sampel dan riset pemantauan waktu pelayanan di Instansi tersebut. Adapun waktu perencanaan penelitian di Kantor Kepala Desa Lingga Tiga.

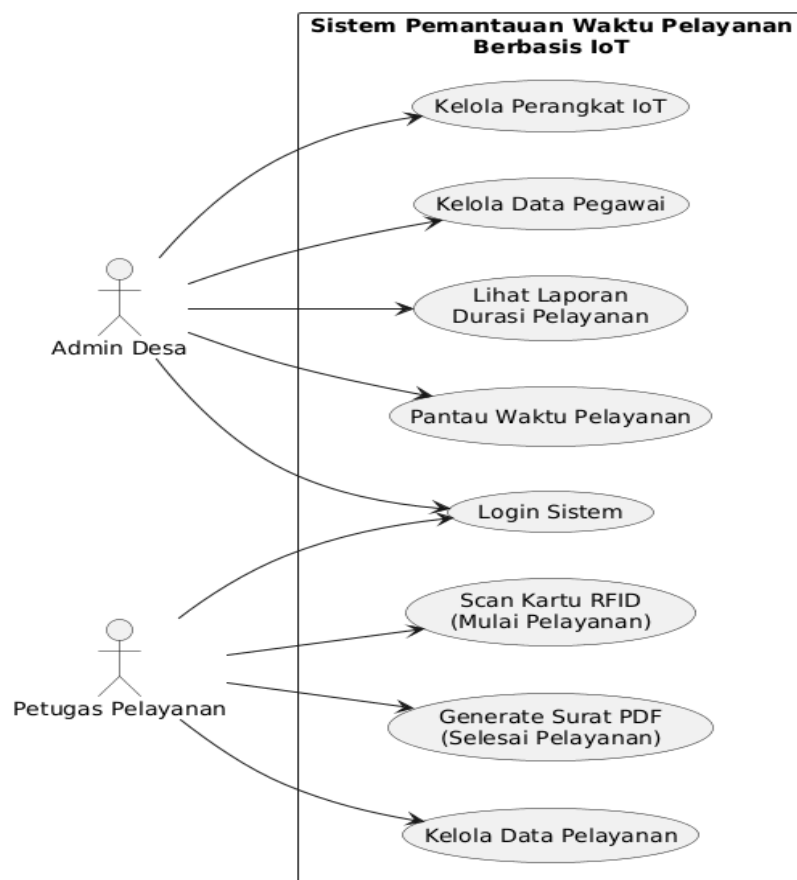


Gambar 3. 2 Alamat Penelitian

3.4 Perancangan Sistem

Perancangan sistem informasi yang terstruktur dan sistematis merupakan langkah penting dalam pengembangan yang berkualitas. *UML (Unified Modeling Language)* adalah salah satu bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [15].

3.3.1 Use Case Diagram



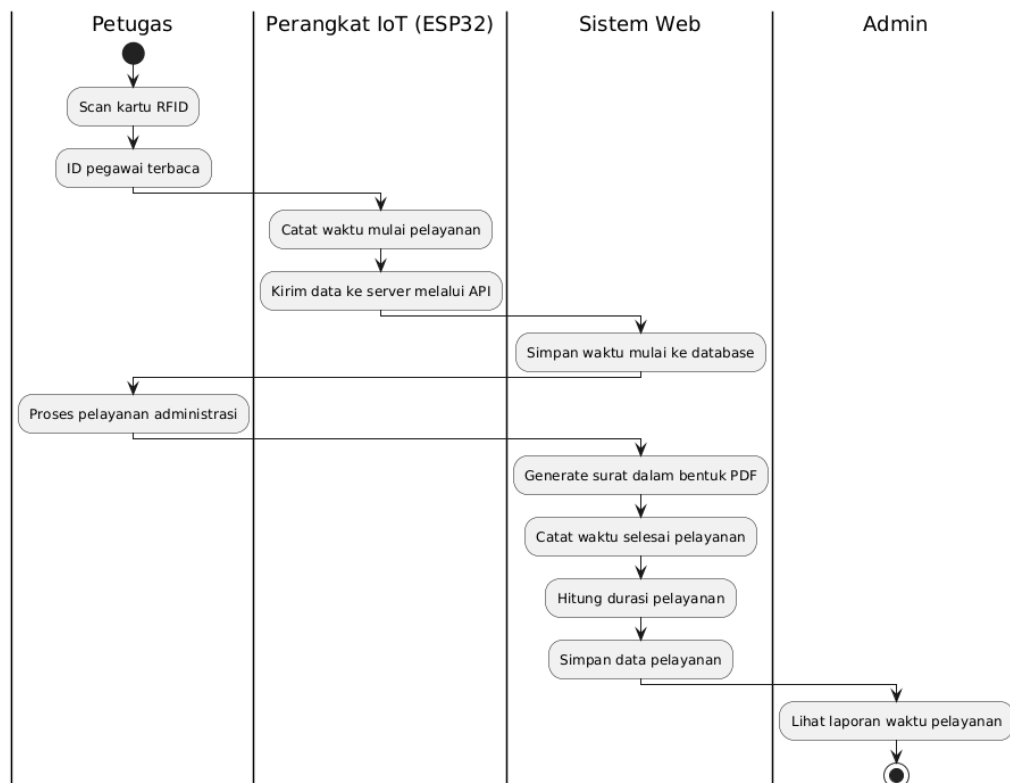
Gambar 3. 3 Use Case Diagram

Use Case Diagram pada sistem pemantauan waktu pelayanan berbasis *IoT* menggambarkan interaksi antara aktor dengan fungsi-fungsi utama yang disediakan oleh sistem. Terdapat dua aktor utama, yaitu Admin Desa dan Petugas Pelayanan.

Di mana seluruh aktivitas pengelolaan dan pemantauan waktu pelayanan dilakukan secara terintegrasi.

Admin Desa memiliki hak akses untuk melakukan login sistem, mengelola perangkat *IoT*, mengelola data pegawai, memantau waktu pelayanan, serta melihat laporan durasi pelayanan. Hak akses ini menunjukkan bahwa Admin berperan sebagai pengelola sistem dan evaluator kinerja pelayanan. Sementara itu, Petugas Pelayanan memiliki hak akses untuk login sistem, memulai pelayanan melalui pemindaian kartu RFID, mengelola data pelayanan, serta mengakhiri pelayanan dengan generate surat dalam bentuk PDF. Pembagian peran ini bertujuan untuk memisahkan fungsi operasional pelayanan dengan fungsi pengelolaan dan pengawasan sistem, sehingga keamanan dan keakuratan data dapat terjaga.

3.3.2 Activity Diagram



Gambar 3. 4 Activity Diagram

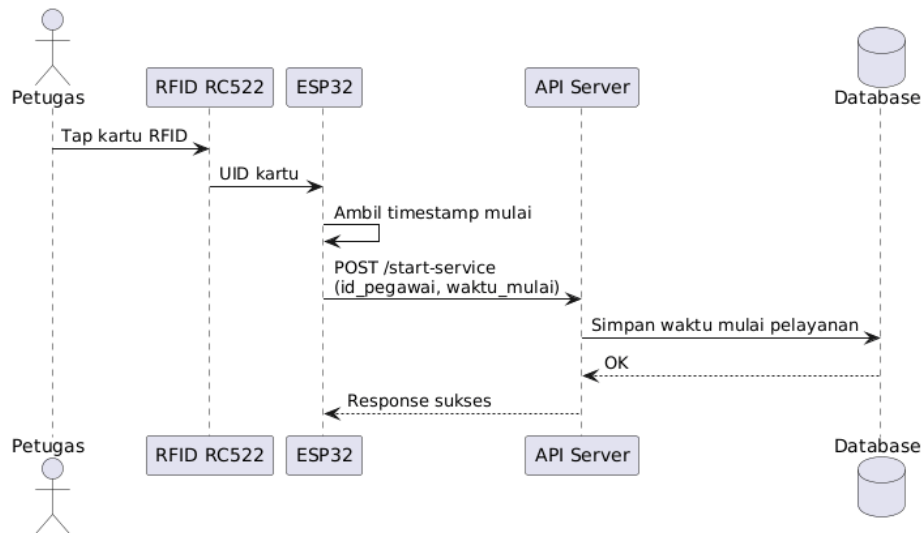
Activity Diagram menggambarkan alur aktivitas pemantauan waktu pelayanan yang terjadi di dalam sistem secara berurutan dan terstruktur. Diagram ini dibagi ke dalam empat swimlane, yaitu Petugas, Perangkat IoT (*ESP32*), Sistem *Web*, dan Admin, yang masing-masing merepresentasikan peran dan tanggung jawab pada setiap tahapan proses.

Proses dimulai ketika Petugas melakukan pemindaian kartu *RFID*. Setelah kartu dipindai, sistem membaca *ID* pegawai yang tersimpan pada kartu *RFID* tersebut. Selanjutnya, Perangkat *IoT* berbasis *ESP32* mencatat waktu mulai pelayanan secara otomatis dan mengirimkan data tersebut ke *server* melalui *API*. Sistem *Web* kemudian menerima data tersebut dan menyimpannya ke dalam database sebagai waktu awal pelayanan.

Setelah proses pelayanan administrasi berlangsung, Petugas menyelesaikan pelayanan melalui Sistem *Web* dengan melakukan generate surat dalam bentuk PDF. Pada saat surat berhasil dibuat, sistem secara otomatis mencatat waktu selesai pelayanan, menghitung durasi pelayanan, dan menyimpan seluruh data pelayanan ke dalam database. Tahap akhir dari proses ini adalah Admin melihat laporan waktu pelayanan yang berisi informasi durasi dan kinerja pelayanan sebagai bahan evaluasi. *Activity Diagram* ini menunjukkan bahwa seluruh pencatatan waktu dilakukan secara otomatis, sehingga meminimalkan kesalahan pencatatan manual.

3.3.3 Sequence Diagram

A. Sequence Diagram Mulai Pelayanan



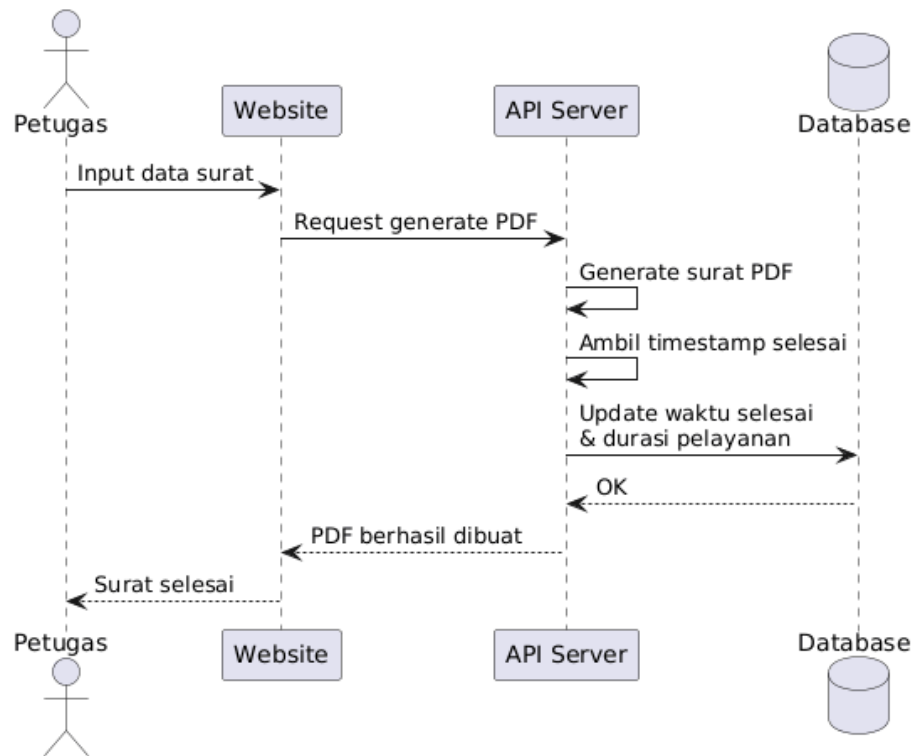
Gambar 3. 5 Sequence Diagram Mulai Pelayanan

Sequence Diagram proses mulai pelayanan menggambarkan urutan interaksi antar objek pada saat pelayanan dimulai menggunakan perangkat *IoT*. Aktor yang terlibat dalam proses ini adalah Petugas, *RFID RC522*, *ESP32*, *API Server*, dan *Database*.

Proses diawali ketika Petugas melakukan tap kartu *RFID* pada *RFID RC522*. *RFID reader* kemudian mengirimkan *UID* kartu ke perangkat *ESP32*. Setelah *UID* diterima, *ESP32* mengambil *timestamp* waktu mulai pelayanan dan mengirimkan data tersebut ke *API Server* melalui *endpoint POST /start-service* yang berisi informasi *ID* pegawai dan waktu mulai pelayanan. *API Server* selanjutnya menyimpan data waktu mulai tersebut ke dalam *Database*. Setelah proses penyimpanan berhasil, *Database* mengirimkan respons sukses ke *API Server*, yang kemudian diteruskan kembali ke *ESP32* sebagai tanda bahwa proses pencatatan waktu mulai pelayanan telah berhasil dilakukan. *Sequence Diagram* ini

menunjukkan integrasi langsung antara perangkat *IoT* dan sistem *backend* melalui *API*.

B. *Sequence Diagram* Selesai Pelayanan



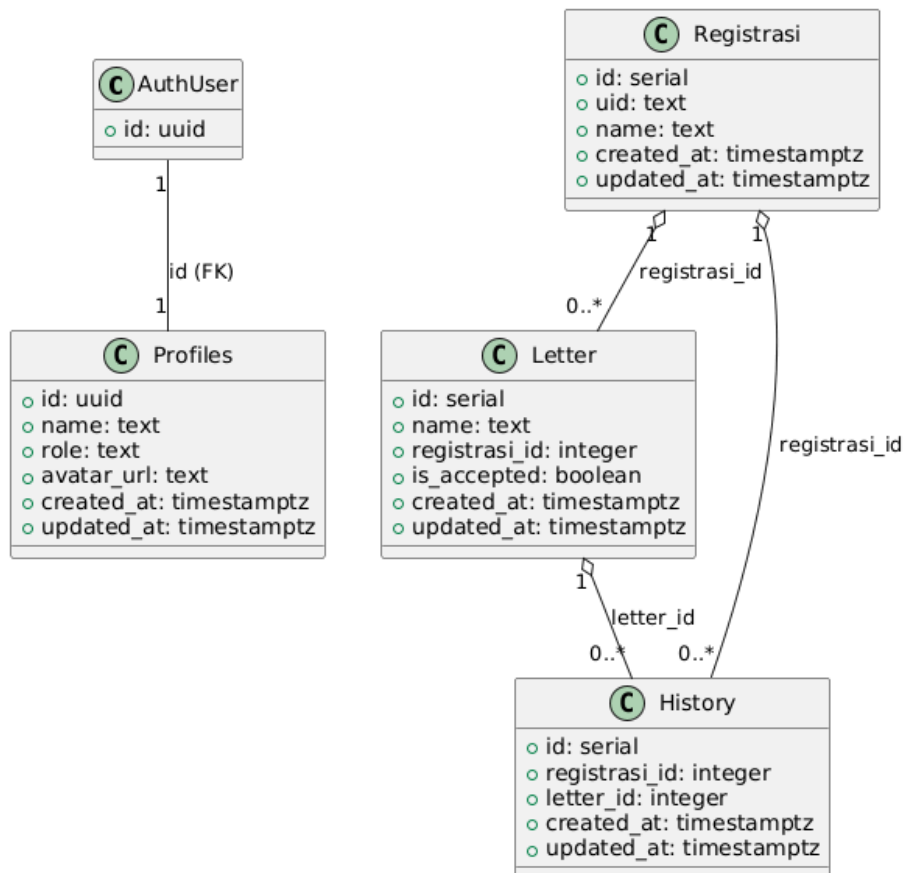
Gambar 3. 6 *Sequence Diagram* Selesai Pelayanan

Sequence Diagram proses selesai pelayanan menjelaskan interaksi antar komponen sistem ketika pelayanan administrasi berakhir. Aktor dan objek yang terlibat dalam proses ini adalah Petugas, *Website*, *API Server*, dan *Database*.

Proses dimulai ketika Petugas menginput data surat melalui *Website*. *Website* kemudian mengirimkan permintaan kepada *API Server* untuk melakukan generate surat dalam bentuk PDF. Setelah surat berhasil dibuat, *API Server* mengambil *timestamp* waktu selesai pelayanan secara otomatis. Selanjutnya, *API Server* melakukan pembaruan data di *Database* dengan menyimpan waktu selesai pelayanan dan menghitung durasi pelayanan berdasarkan selisih antara waktu mulai

dan waktu selesai. Setelah data berhasil diperbarui, *Database* mengirimkan *respons* sukses ke *API Server*. *API Server* kemudian mengembalikan informasi bahwa PDF berhasil dibuat kepada *Website*, dan *Website* menampilkan notifikasi kepada Petugas bahwa proses pelayanan telah selesai. Diagram ini menegaskan bahwa pencatatan waktu selesai pelayanan terintegrasi langsung dengan proses administrasi.

3.3.4 Class Diagram



Gambar 3. 7 Class Diagram

Class Diagram menggambarkan struktur data dan hubungan antar *entitas* yang digunakan dalam sistem pemantauan waktu pelayanan. Diagram ini terdiri dari beberapa kelas utama, yaitu *AuthUser*, *Profiles*, *Registrasi*, *Letter*, dan *History*.

Kelas *AuthUser* merepresentasikan data *autentikasi* pengguna dengan *atribut* utama *id*. Kelas *Profiles* menyimpan informasi profil pengguna seperti nama, peran, avatar, serta waktu pembuatan dan pembaruan data, dengan relasi satu ke satu terhadap *AuthUser* melalui atribut *id* sebagai *foreign key*. Kelas *Registrasi* berfungsi untuk menyimpan data pendaftaran pelayanan, yang memiliki relasi satu ke banyak dengan kelas *Letter*, karena satu data *registrasi* dapat menghasilkan beberapa surat. Kelas *Letter* menyimpan informasi surat yang dibuat selama proses pelayanan, sedangkan kelas *History* berfungsi sebagai pencatat riwayat aktivitas pelayanan yang menghubungkan data registrasi dan surat yang dihasilkan.

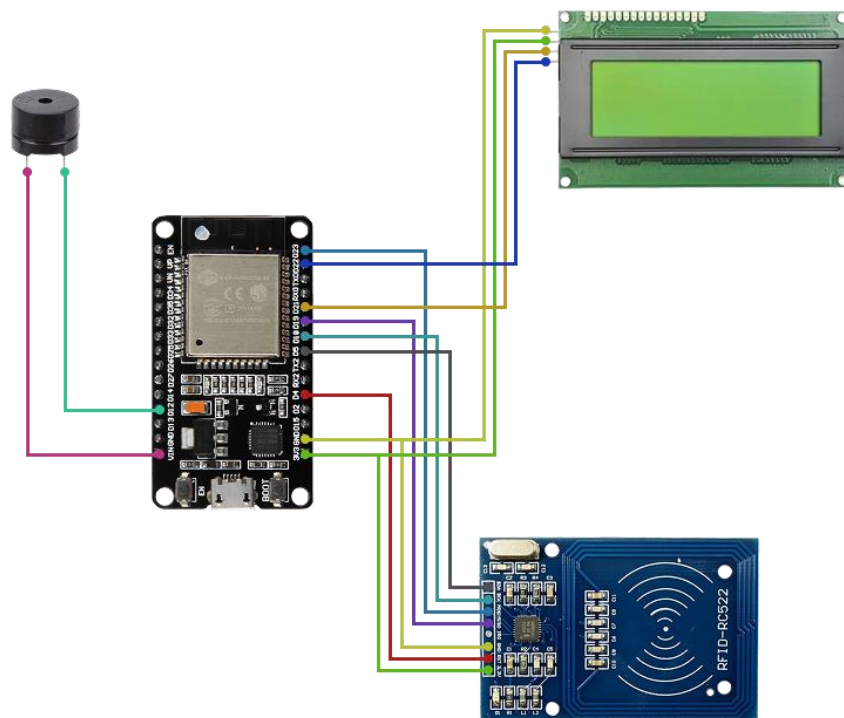
Relasi antar kelas ini menunjukkan bahwa sistem dirancang untuk menyimpan data pelayanan secara terstruktur dan terintegrasi, sehingga memudahkan proses pelacakan, evaluasi, dan pelaporan durasi pelayanan.

3.5 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna (*user interface design*) merupakan komponen yang sangat penting dalam pengembangan sistem informasi, karena berperan sebagai media interaksi antara pengguna dan sistem. Antarmuka pengguna yang dirancang dengan baik dapat meningkatkan kemudahan penggunaan (*usability*), efektivitas, serta efisiensi dalam pengoperasian sistem. Selain itu, desain antarmuka yang tepat juga mampu meminimalkan kesalahan pengguna, mempercepat proses pemahaman terhadap fitur sistem, dan meningkatkan tingkat kepuasan pengguna terhadap sistem informasi yang dikembangkan [16].

3.4.1 Perancangan *Internet of Things (IoT)*

Perancangan *Internet of Things* yang terdiri dari beberapa komponen utama, yaitu mikrokontroler ESP32, modul RFID RC522, modul LCD 20x4, dan buzzer sebagai perangkat output. Seluruh komponen dirangkai dan diintegrasikan untuk membentuk sebuah sistem yang mampu melakukan proses identifikasi, pengolahan data, serta penyampaian informasi secara real time.



Gambar 3. 8 Perancangan Internet of Things

Mikrokontroler ESP32 berperan sebagai pusat kendali (*control unit*) dalam sistem IoT ini. ESP32 dipilih karena memiliki kemampuan pemrosesan yang baik serta telah dilengkapi dengan modul Wi-Fi dan Bluetooth, sehingga sangat mendukung implementasi konsep IoT. Mikrokontroler ini bertugas untuk menerima data dari modul RFID, memproses data tersebut sesuai dengan logika sistem yang telah diprogram, serta mengendalikan perangkat output seperti LCD dan buzzer.

Modul RFID RC522 berfungsi sebagai perangkat input yang digunakan untuk melakukan identifikasi objek atau pengguna melalui kartu RFID. Modul ini bekerja dengan cara membaca UID (*Unique Identifier*) dari kartu RFID yang didekatkan ke sensor. Data UID tersebut kemudian dikirimkan ke mikrokontroler ESP32 melalui komunikasi SPI (*Serial Peripheral Interface*).

LCD 20x4 digunakan sebagai media tampilan informasi dalam sistem. LCD ini berfungsi untuk menampilkan pesan atau status sistem, seperti informasi keberhasilan atau kegagalan proses identifikasi RFID. Dengan adanya LCD, pengguna dapat dengan mudah memahami kondisi sistem secara langsung. LCD dihubungkan ke ESP32 melalui beberapa pin input-output digital yang berfungsi untuk mengirimkan data dan perintah tampilan.

Buzzer berperan sebagai perangkat output tambahan yang berfungsi memberikan notifikasi suara. Buzzer akan aktif sesuai dengan kondisi tertentu, misalnya ketika kartu RFID berhasil dikenali atau ketika akses ditolak. Penggunaan buzzer bertujuan untuk memberikan umpan balik (*feedback*) secara audio kepada pengguna, sehingga interaksi dengan sistem menjadi lebih efektif.

Tabel 3. 2 Perancangan Pin RFID RC522 Ke Pin ESP32

RFID RC522	ESP32
3.3V	3.3V
GND	GND
SDA	D5
SCK	D18
MOSI	D23
MISO	D19
RST	D4

Tabel 3.2 menunjukkan konfigurasi hubungan pin antara modul RFID RC522 dengan mikrokontroler ESP32. Koneksi yang disusun meliputi jalur catu daya, ground, serta jalur komunikasi SPI yang digunakan untuk proses pengiriman data UID kartu RFID ke ESP32. Perancangan ini memiliki peranan penting karena menjadi dasar komunikasi antara perangkat input dan pusat kendali sistem. Dengan konfigurasi pin yang sesuai, proses identifikasi kartu RFID dapat dilakukan secara optimal sehingga mendukung pencatatan waktu mulai pelayanan secara otomatis.

Tabel 3. 3 Perancangan Pin LCD 20x4 Ke Pin ESP32

LCD 20x4	ESP32
VCC	3.3V
GND	GND
SDA	D21
SCL	D22

Tabel 3.3 menyajikan rancangan hubungan pin antara LCD 20x4 dengan ESP32 sebagai media tampilan informasi pada sistem. Koneksi tersebut mencakup jalur daya dan jalur komunikasi I2C yang digunakan untuk menampilkan status perangkat, hasil identifikasi, maupun informasi lain yang diperlukan pengguna. Rancangan ini menunjukkan bahwa LCD berfungsi sebagai antarmuka visual yang mendukung interaksi pengguna dengan sistem. Dengan adanya tampilan informasi secara langsung, pengguna dapat mengetahui kondisi sistem secara lebih cepat, jelas, dan mudah dipahami.

Tabel 3. 4 Perancangan Pin Buzzer Ke Pin ESP32

BUZZER	ESP32
PIN POSITIF (+)	D12

PIN NEGATIF (-)	5V
-----------------	----

Tabel 3.4 menunjukkan hubungan pin antara buzzer dan ESP32 yang digunakan sebagai media notifikasi suara pada sistem. Buzzer diintegrasikan untuk memberikan umpan balik audio ketika sistem berada pada kondisi tertentu, seperti keberhasilan pembacaan kartu RFID atau notifikasi terhadap proses yang sedang berlangsung. Perancangan ini memperlihatkan bahwa sistem tidak hanya mengandalkan tampilan visual, tetapi juga memanfaatkan notifikasi suara untuk meningkatkan efektivitas interaksi antara perangkat dan pengguna. Dengan demikian, respons sistem dapat diterima dengan lebih cepat dan lebih informatif.

3.4.2 Perancangan Website

Website atau situs dapat diartikan sebagai kumpulan halaman halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Website berfungsi sebagai media penyampaian informasi yang dapat diakses secara luas melalui jaringan internet dengan menggunakan perangkat komputer maupun perangkat bergerak. Selain itu, website juga dapat dimanfaatkan sebagai sarana komunikasi, publikasi, promosi, serta pengelolaan data dan layanan secara digital. Dengan perkembangan teknologi informasi yang semakin pesat, website tidak hanya berperan sebagai penyedia informasi satu arah, tetapi juga mampu mendukung interaksi dua arah antara pengguna dan sistem melalui fitur-fitur yang bersifat dinamis, seperti formulir, sistem autentikasi, dan basis data terintegrasi [17].

A. Halaman Login



The image shows a login form with the following elements:

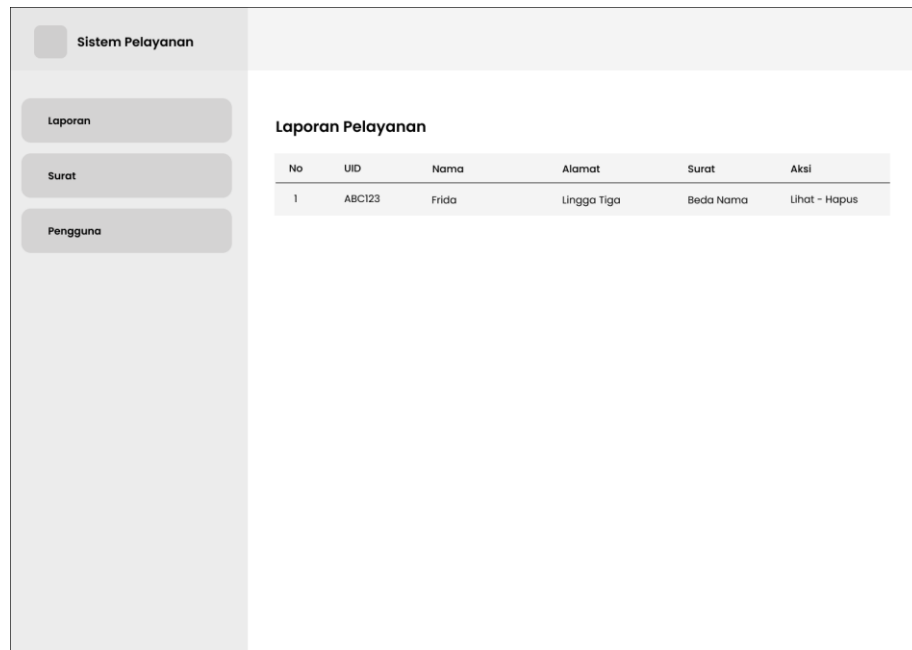
- Title: **Masuk**
- Subtitle: Pemantauan Waktu Pelayanan
- Label: Email
- Input field: A rectangular text box for entering the email address.
- Label: Password
- Input field: A rectangular text box for entering the password.
- Button: A button labeled "Login" positioned below the password field.

Gambar 3. 9 Halaman Login

Halaman login merupakan tampilan awal yang digunakan sebagai gerbang akses ke dalam sistem pelayanan. Pada halaman ini, pengguna diwajibkan untuk memasukkan alamat email dan kata sandi (*password*) yang telah terdaftar sebelumnya. Proses autentikasi ini bertujuan untuk memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat menggunakan sistem sesuai dengan perannya masing-masing.

Desain halaman login dibuat sederhana dan fokus pada fungsi utama, yaitu proses masuk ke sistem. Penempatan komponen input yang terpusat bertujuan untuk meningkatkan kenyamanan dan kemudahan penggunaan (*usability*). Selain itu, keberadaan mekanisme login ini juga berperan penting dalam menjaga keamanan data dan mencegah akses tidak sah terhadap sistem pelayanan.

B. Halaman Laporan



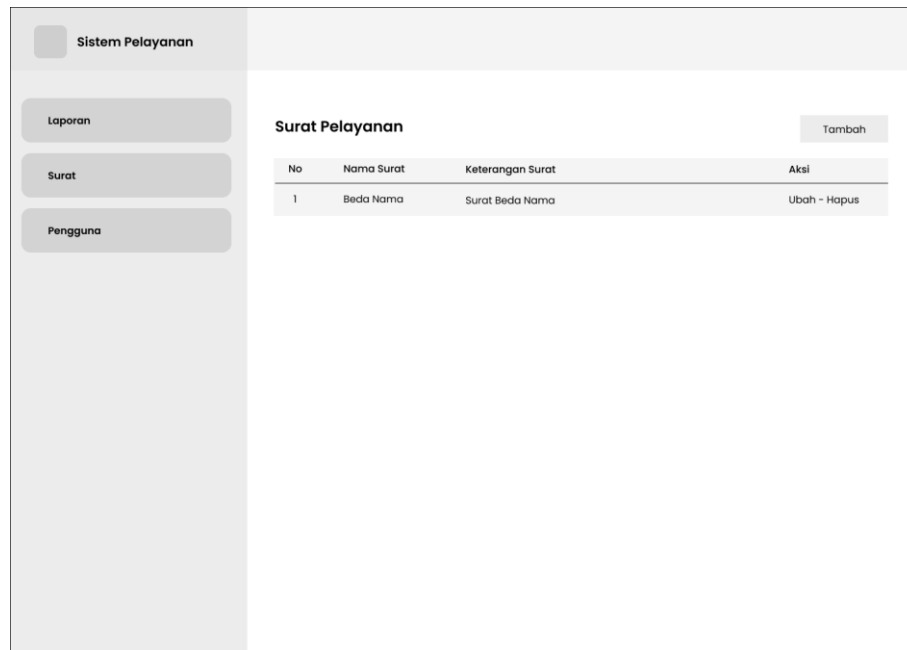
No	UID	Nama	Alamat	Surat	Aksi
1	ABC123	Frida	Lingga Tiga	Beda Nama	Lihat - Hapus

Gambar 3. 10 Halaman Laporan

Halaman laporan pelayanan berfungsi untuk menampilkan data hasil pelayanan yang telah dilakukan oleh sistem. Informasi yang disajikan pada halaman ini meliputi UID, nama pemohon, alamat, jenis surat yang diajukan, serta aksi yang dapat dilakukan seperti melihat detail (*lihat*) dan menghapus (*hapus*) data laporan.

Halaman ini dirancang untuk mendukung proses pemantauan dan evaluasi kinerja pelayanan. Dengan adanya laporan pelayanan yang tersusun dalam bentuk tabel, pengguna dapat dengan mudah mengakses dan menelusuri data pelayanan secara sistematis. Hal ini sangat membantu dalam proses administrasi, pengarsipan, serta pengambilan keputusan berdasarkan data yang tersedia.

C. Halaman Surat

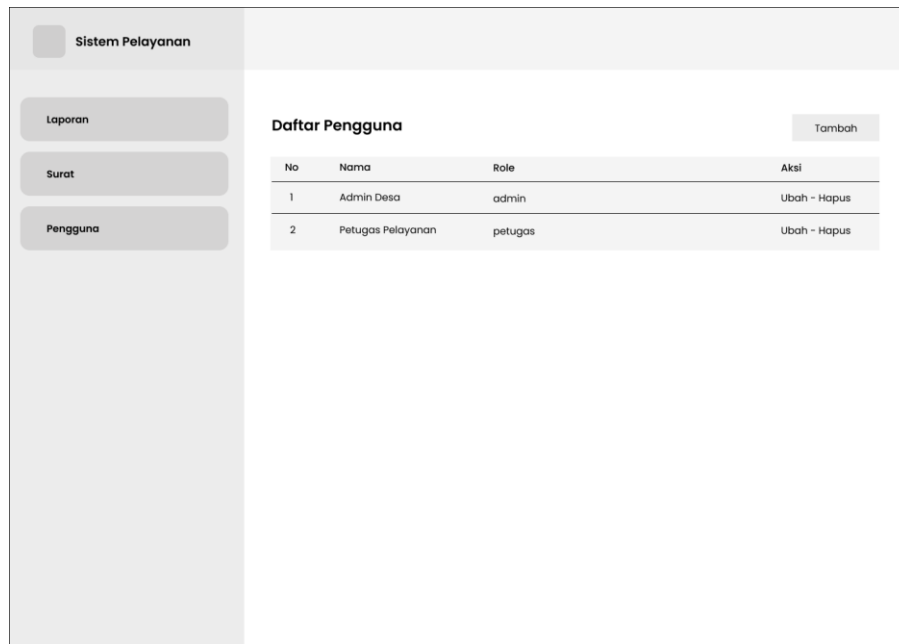


Gambar 3. 11 Halaman Surat

Halaman surat pelayanan digunakan untuk mengelola jenis-jenis surat yang tersedia dalam sistem. Informasi yang ditampilkan mencakup nama surat, keterangan surat, serta opsi aksi untuk mengubah dan menghapus data surat. Selain itu, terdapat tombol tambah yang digunakan untuk menambahkan jenis surat baru ke dalam sistem.

Perancangan halaman ini bertujuan untuk memberikan fleksibilitas kepada administrator dalam mengelola layanan surat sesuai dengan kebutuhan instansi. Dengan sistem pengelolaan surat yang terintegrasi, proses pelayanan administrasi dapat dilakukan secara lebih cepat, terstruktur, dan terdokumentasi dengan baik.

D. Halaman Daftar Pengguna



No	Nama	Role	Aksi
1	Admin Desa	admin	Ubah - Hapus
2	Petugas Pelayanan	petugas	Ubah - Hapus

Gambar 3. 12 Halaman Daftar Pengguna

Halaman daftar pengguna digunakan untuk mengelola data pengguna yang memiliki akses ke dalam sistem. Pada halaman ini ditampilkan informasi berupa nomor urut, nama pengguna, peran (*role*), serta aksi yang dapat dilakukan seperti mengubah (*ubah*) dan menghapus (*hapus*) data pengguna.

Fitur ini umumnya hanya dapat diakses oleh pengguna dengan peran administrator. Tujuan dari halaman ini adalah untuk mempermudah proses manajemen pengguna, termasuk pengaturan hak akses antara admin dan petugas pelayanan. Dengan adanya pembagian peran tersebut, sistem dapat berjalan secara terstruktur dan sesuai dengan tugas masing-masing pengguna, sehingga meningkatkan efektivitas pengelolaan sistem pelayanan.