

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam era digital saat ini, teknologi informasi dan komunikasi telah menjadi bagian integral dari berbagai aspek kehidupan masyarakat. Pemerintah, sebagai penyelenggara pelayanan publik, dituntut untuk beradaptasi dengan perkembangan teknologi guna meningkatkan efisiensi dan kualitas layanan. Kantor kelurahan, sebagai unit pemerintahan yang terdekat dengan masyarakat, memiliki peran penting dalam memberikan layanan administrasi dan informasi yang cepat, tepat, dan transparan. Meskipun demikian, masih banyak kantor kelurahan yang menghadapi tantangan dalam mengelola administrasi dan pelayanan publik dengan efektif.

Namun, pada kenyataannya, banyak kantor kelurahan yang masih menggunakan sistem manual dalam pengelolaan administrasi dan pelayanan publik. Hal ini menyebabkan beberapa permasalahan, yaitu: Proses pelayanan yang lambat dimana penggunaan sistem manual seringkali menyebabkan antrean panjang dan waktu tunggu yang lama bagi masyarakat; Pengelolaan data secara manual menyebabkan kesulitan dalam pencarian dan pengarsipan data, serta meningkatkan risiko kehilangan atau kerusakan data; Proses manual mengharuskan pegawai kelurahan melakukan pekerjaan administratif berulang yang memakan waktu dan tenaga.

Untuk mengatasi permasalahan tersebut, diperlukan sebuah solusi berbasis teknologi informasi yang mampu mengotomatisasi proses pelayanan publik di kantor kelurahan. Solusi ini harus dapat menangani berbagai aspek administrasi dan operasional secara efisien, mulai dari pendaftaran, pengolahan data, hingga pengarsipan dan pelaporan. Salah satu solusi yang dapat diterapkan adalah pengembangan aplikasi berbasis web yang dirancang khusus untuk mendukung administrasi dan pelayanan publik di kantor kelurahan. Aplikasi ini tidak hanya akan mempercepat proses layanan, tetapi juga meningkatkan akurasi data, meminimalkan kesalahan manusia, dan menyediakan akses yang lebih mudah bagi masyarakat untuk mendapatkan layanan yang mereka butuhkan. Dengan demikian, aplikasi berbasis web ini dapat menjadi alat yang efektif untuk modernisasi sistem pelayanan di kantor kelurahan, sekaligus meningkatkan kepuasan masyarakat terhadap layanan publik.

Oleh karena itu, dengan "**Aplikasi Layanan Publik di Kantor Kelurahan Menggunakan Teknologi Web**" menggambarkan solusi konkret yang diusulkan untuk mengatasi permasalahan yang ada di kantor kelurahan. Judul ini mencerminkan penerapan teknologi modern dalam mengelola dan menyederhanakan proses administrasi serta layanan publik. Pengembangan aplikasi berbasis web sebagai solusi teknologi informasi merupakan langkah strategis untuk meningkatkan kualitas dan efisiensi pelayanan publik. Diharapkan aplikasi ini dapat memberikan manfaat yang signifikan bagi masyarakat dengan menyediakan layanan yang lebih cepat, akurat, dan transparan, serta mendukung upaya pemerintah dalam modernisasi sistem administrasi di tingkat kelurahan.

B. Rumusan Masalah

Dengan latar belakang yang telah dijelaskan, maka yang menjadi rumusan masalah pada penelitian ini adalah bagaimana membuat aplikasi layanan publik di kantor kelurahan menggunakan teknologi web.

C. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini mencakup layanan administrasi dasar yang disediakan oleh kantor kelurahan, seperti pembuatan KTP, surat keterangan domisili, surat pengantar, dan layanan administrasi lainnya. Layanan di luar lingkup kelurahan tidak termasuk dalam aplikasi ini.
2. Aplikasi ini akan diujicobakan di satu kantor kelurahan sebagai studi kasus. Generalisasi ke seluruh kelurahan di wilayah yang lebih luas atau nasional tidak termasuk dalam penelitian ini.
3. Pengguna aplikasi terdiri dari dua kelompok utama yaitu staf kantor kelurahan dan masyarakat yang membutuhkan layanan administrasi. Pengguna dari instansi lain atau pihak ketiga tidak termasuk dalam penelitian ini.

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan aplikasi berbasis web yang mampu mempercepat dan menyederhanakan proses administrasi di kantor kelurahan, sehingga masyarakat dapat menerima layanan dengan lebih cepat.

E. Manfaat Penelitian

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Bagi akademik

Penelitian ini dapat memberikan kontribusi baru dalam bidang teknologi informasi terutama terkait pengembangan aplikasi web untuk pelayanan publik di tingkat kelurahan.

2. Bagi penulis

Memungkinkan penulis untuk mendapatkan pengalaman langsung dalam merancang, mengembangkan, dan mengimplementasikan solusi teknologi dalam lingkungan pelayanan publik.

3. Bagi masyarakat

Masyarakat dapat mengakses layanan administrasi kelurahan secara online, mengurangi waktu tunggu dan memudahkan proses pengajuan dokumen.

4. Bagi kantor kelurahan

Mengurangi beban kerja administratif dengan otomatisasi proses pelayanan, sehingga pegawai dapat fokus pada pelayanan yang lebih kompleks dan interaktif.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Pelayanan Publik

Pelayanan publik mengacu pada berbagai layanan yang disediakan oleh pemerintah atau lembaga publik untuk memenuhi kebutuhan dan kepentingan masyarakat. Secara umum, pelayanan publik mencakup segala hal yang dilakukan oleh pemerintah untuk kesejahteraan masyarakat, baik dalam bentuk layanan langsung seperti kesehatan, pendidikan, infrastruktur, keamanan, hingga layanan administratif seperti perizinan dan registrasi. Undang-Undang Nomor 25 Tahun 2009 tentang pelayanan publik menjelaskan bahwa kegiatan atau rangkaian kegiatan dalam rangka pemenuhan kebutuhan layanan sesuai dengan peraturan undang undang bagi setiap warga negara dan penduduk atas barang, jasa, dan/atau pelayanan yang disediakan oleh penyelenggara pelayanan publik (Fakhriyah et al., 2022). Kualitas pelayanan publik merupakan kemampuan organisasi pelayanan publik untuk memberikan pelayanan yang dapat memuaskan para pengguna jasa baik melalui pelayanan teknis maupun administrasi (Along, 2020). Pelayanan publik yang baik memberikan kepuasan terhadap masyarakat atas pelayanan tersebut. Dalam pemberian layanan, menjadi tugas dan tanggung jawab pemerintah untuk menyediakannya secara profesional, akuntabel dan optimal. Pelayanan yang optimal adalah harapan semua masyarakat agar tercipta kualitas pelayanan yang lebih baik.

Berikut karakteristik pelayanan publik

- a. Aksesibilitas: Pelayanan publik harus dapat diakses oleh semua warga negara tanpa diskriminasi. Ini berarti layanan tersebut harus tersedia bagi setiap individu atau kelompok yang memenuhi syarat-syarat yang ditentukan, tanpa adanya hambatan yang tidak diperlukan.
- b. Universalitas: Prinsip ini mengindikasikan bahwa layanan publik harus tersedia untuk semua individu atau kelompok dalam masyarakat tanpa memandang latar belakang sosial, ekonomi, atau budaya mereka. Universalitas menjamin bahwa setiap warga negara memiliki hak yang sama untuk mendapatkan layanan yang dibutuhkan.
- c. Kesetaraan: Layanan publik harus disediakan secara adil dan setara bagi semua penerima. Ini berarti bahwa setiap individu harus diperlakukan dengan cara yang sama tanpa adanya preferensi atau diskriminasi yang tidak adil.
- d. Tanggung jawab publik: Pelayanan publik dijalankan atas nama masyarakat dan bertanggung jawab untuk memberikan manfaat yang maksimal bagi kesejahteraan umum.

Pelayanan publik adalah serangkaian kegiatan yang dilakukan oleh organisasi publik atau instansi pemerintah yang bertujuan untuk memenuhi kebutuhan masyarakat akan barang dan jasa yang dilakukan sesuai standar dan peraturan yang telah ditetapkan, pemerintah melalui lembaga dan segenap aparturnya bertugas menyediakan dan menyelenggarakan pelayanan kepada masyarakat (Laia et al., 2022).

2. Sistem Informasi

Sistem Informasi (SI) dapat didefinisikan sebagai kombinasi dari orang, proses, dan teknologi yang berfungsi bersama-sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang dibutuhkan untuk mendukung operasi bisnis dan pengambilan keputusan dalam sebuah organisasi. Tujuan utama dari sistem informasi adalah untuk menyediakan informasi yang akurat, tepat waktu, dan relevan kepada pengguna yang berwenang, sehingga dapat mendukung pengambilan keputusan yang baik dan efisiensi dalam operasi organisasi.

Teknologi berbasis web dan internet saat ini telah menjadi salah satu kebutuhan yang penting dalam aktifitas kehidupan. Setiap hari terus berkembang, perkembangan yang ramai dibicarakan dan dibahas sekarang ini adalah teknologi yang mengarah pada sistem informasi dan aplikasi berbasis web. Dengan adanya penerapan teknologi berbasis web, komunikasi antara pengguna, admin dan pengembang lebih luas karena waktu siaga sistem informasi dan aplikasi berbasis web sangat tinggi, bahkan ada yang sampai 24 jam perhari dalam satu minggu yaitu web yang menggunakan server yang sudah maju (Faqih & Wahyudi, 2022). Sistem informasi adalah sebuah sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Dukungan sistem informasi di abad ke-21 ini sangat penting. Sebab munculnya banyak arus perubahan yang bersifat luas (Primawanti & Ali, 2022).

Berbagai jenis sistem informasi termasuk:

- a. Sistem Informasi Manajemen (SIM): Digunakan oleh manajemen tingkat menengah untuk mengambil keputusan yang berhubungan dengan perencanaan, pengendalian, dan pengorganisasian.
- b. Sistem Informasi Eksekutif (SIE): Memberikan informasi yang diperlukan oleh manajemen tingkat atas untuk pengambilan keputusan strategis.
- c. Sistem Pendukung Keputusan (SPK): Sistem yang membantu pengambilan keputusan berdasarkan analisis data dan model matematis.
- d. Sistem Informasi Akuntansi (SIA): Digunakan untuk memproses transaksi keuangan dan menghasilkan laporan keuangan.

Dalam konteks pelayanan publik, sistem informasi memiliki peran penting dalam mempercepat proses pengolahan data dan informasi, mengurangi biaya administrasi, dan meningkatkan responsifitas terhadap kebutuhan masyarakat, dengan menyediakan akses mudah terhadap informasi yang relevan, sistem informasi membantu meningkatkan transparansi dalam proses pengambilan keputusan dan akuntabilitas dalam penyediaan layanan publik, Memberikan informasi yang akurat dan tepat waktu kepada pengguna layanan publik, sehingga memungkinkan pemerintah untuk memberikan layanan yang lebih baik kepada masyarakat.

3. Sistem Informasi Pelayanan Publik

Sistem Informasi Pelayanan Publik (SIPP) adalah infrastruktur teknologi informasi yang digunakan oleh pemerintah atau lembaga publik untuk mengelola dan menyediakan informasi yang diperlukan dalam penyediaan layanan publik kepada masyarakat. SIPP mencakup aplikasi-aplikasi berbasis teknologi informasi yang mendukung berbagai fungsi administratif, operasional, dan layanan kepada publik

4. HTML



Gambar 2.1 *Hypertext Markup Language*

Hypertext Markup Language atau *HTML*, adalah bahasa *markup* standar industri untuk membangun halaman *web*. Ini menawarkan kumpulan *tag* dan properti yang menentukan pengorganisasian halaman *web*, konten, dan format halaman *web*. Untuk merepresentasikan elemen pada halaman *web*, *HTML* menggunakan struktur hierarkis yang dikenal sebagai *Document Object Model (DOM)*. Teks, foto, tautan, multimedia, dan elemen lainnya semuanya dapat dimuat dalam setiap elemen, yang dibatasi oleh *tag* pembuka dan penutup.

Istilah "*HyperText*" mengacu pada kemampuan *HTML* untuk membuat tautan yang memungkinkan pengguna dengan mudah dan cepat berpindah dari satu situs *web* ke situs *web* lainnya. *HyperText Markup Language*, juga dikenal sebagai

HTML, adalah bahasa global *World Wide Web*. Seperti yang tersirat dari namanya *HyperText Markup Language*, *HTML* adalah bahasa *markup* bukan bahasa pemrograman

Dokumen yang dibuat dengan menggunakan bahasa bahasa markup seperti *HTML* memiliki bagian yang dirender pada hasil akhir serta bagian yang menginstruksikan program rendering cara menginterpretasikan teks yang tersisa. Proses mengubah dokumen teks yang berisi teks *HTML* menjadi, misalnya, representasi visualnya di layar disebut sebagai "*rendering*".

Perkembangan *HTML* sangat pesat. Dari permulaan *HTML* 1.0 yang sederhana, yang mencakup elemen dasar seperti judul dan paragraf, hingga penambahan tabel dan gambar pada *HTML* 2.0, dan akhirnya ke *frame* dan dukungan *form* yang disempurnakan pada *HTML* 3.2. Dengan *CSS*, konten dan presentasi sekarang dapat dipisahkan dalam *HTML* 4.01, yang juga meningkatkan komponennya. Dengan elemen semantiknya, dukungan multimedia, bentuk yang lebih baik, fitur aksesibilitas, dan kemampuan *mobile-friendly*, *HTML5* menandai periode transformasi.

Versi *HTML* terbaru, yang dikenal sebagai *HTML5*, digunakan untuk membuat dan mengatur dokumen di internet. Ini mencakup elemen dan fitur baru yang memungkinkan pembuatan halaman *web* yang lebih dinamis dan menarik, termasuk dukungan untuk audio dan video, kanvas untuk grafik dan animasi, dan peningkatan kompatibilitas dengan perangkat seluler (McLaughlin, 2011).

Fungsionalitas baru yang diperkenalkan oleh *HTML5* memungkinkan pembuatan halaman *web* yang lebih dinamis dan interaktif, yaitu:

- a. Dukungan audio dan video.
- b. Kanvas visual dan animasi.
- c. Dukungan yang ditingkatkan untuk mobile..
- d. Semantic Element baru termasuk nav, article, sections dan headers.
- e. Kontrol untuk form, yaitu tanggal, waktu, email, pencarian, dll.
- f. Penyimpanan web untuk menyimpan informasi secara lokal.
- g. Fitur drag and drop.

5. CSS

CSS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten *web* di layar, printer, sintesis ucapan, pembaca layar, dan jendela obrolan. CSS digunakan oleh semua *browser* pada semua ukuran layar di semua jenis perangkat *IoT*, termasuk ponsel, komputer, video game, televisi, jam tangan, dan konsol otomatis (Meyer & Weyl, 2014). Sedangkan menurut (Duckett, 2022) CSS adalah bahasa pemrograman *web* yang digunakan untuk mengendalikan tampilan halaman *web*. (Castro, 2022) menjelaskan bahwa CSS memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. CSS terdiri dari dua bagian utama, yaitu deklarasi dan selektor. Deklarasi adalah bagian yang menentukan properti dari suatu halaman *web*. Selektor adalah bagian yang menentukan elemen *web* mana yang akan diatur oleh deklarasi. CSS juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. Beberapa properti CSS yang umum digunakan antara lain:

- a. Ukuran: properti ini digunakan untuk mengatur ukuran halaman web, seperti ukuran font, ukuran gambar, dan ukuran jarak antar elemen.
- b. Warna: properti ini digunakan untuk mengatur warna halaman web, seperti warna font, warna latar belakang, dan warna border.
- c. Font: properti ini digunakan untuk mengatur font elemen web, seperti jenis font, ukuran font, dan gaya font.
- d. Tata letak: properti ini digunakan untuk mengatur tata letak elemen web, seperti posisi elemen, ukuran elemen, dan jarak antar elemen.

6. *PHP*

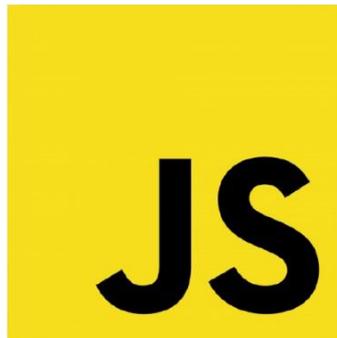


Gambar 2.2 *Hypertext Preprocessor*

PHP adalah bahasa *server-side* yang secara khusus digunakan untuk aplikasi *web*. Bahasa *PHP* berjalan di sisi *server*, artinya ia berjalan pada *server* dan bisa dimasukkan di antara bahasa *HTML* (*HyperText Markup Language*). Oleh karena itu, kode *PHP* tidak akan terlihat lagi dan hasil akhir dalam bentuk *HTML* akan dikirimkan ke *browser* (Sutarman, 2007). Karena *PHP* dijalankan di komputer *server*, itu dikenal sebagai bahasa pemrograman sisi *server*. Ini berbeda dengan bahasa pemrograman sisi klien, seperti JavaScript, yang beroperasi pada klien, yaitu *web browser*.

Pada awalnya, *PHP* adalah singkatan dari *Personal Home Page*. *PHP* digunakan untuk mengembangkan situs *web* pribadi, sesuai dengan namanya. Seiring berjalannya waktu, *PHP* berkembang menjadi bahasa pemrograman *web* yang sangat kuat, tidak hanya digunakan untuk membuat halaman *web* dasar tetapi juga untuk situs-situs populer seperti *Wikipedia*, *WordPress*, *Joomla*, dan lainnya yang dikunjungi oleh jutaan orang.

7. *Javascript*



Gambar 2.3 *Javascript*

Javascript adalah bahasa pemrograman *open-source* untuk teknologi *web* yang dapat digunakan untuk memodifikasi *HTML* dan *CSS*. Biasanya disingkat menjadi *JS* saja. Sebagian besar *browser web*, termasuk *Internet Explorer*, *Google Chrome*, dan *Mozilla Firefox*, menggunakan *JavaScript* untuk operasi yang dilakukan di antarmuka pengguna. Karena kemampuannya untuk menginstruksikan *browser* untuk melakukan tugas, sekarang *JavaScript* menjadi *client-side scripting language* yang paling populer (Shute, 2019).

JavaScript dapat meningkatkan kegunaan halaman *web* dengan memberikan umpan balik yang cepat. Dengan bantuan bahasa pemrograman

JavaScript, Anda dapat meningkatkan *HTML* Anda dengan menambahkan animasi, interaksi, dan efek visual yang dinamis untuk pengguna *browser*. (McFarland, 2012). Selain itu, *JavaScript* dapat digunakan pada *platform* apa pun dan merupakan alat yang berguna untuk mengembangkan *program* dan makro sementara. Anda dapat dengan cepat membuat perangkat lunak yang layak dengan menggunakan *text editor* dan *browser web* berkat ketersediaan *browser* yang tersebar luas (Patrick, 2016).

Saat pertama kali keluar, *JavaScript* hanya memiliki sedikit "bagian yang bagus". Saat ini, sebagian besar komponen sangat baik. Meskipun masih ada beberapa fitur lawas, fitur baru yang ditawarkan oleh standar *ES6/7/8* membantu Anda menulis kode yang lebih aman, mudah, dan singkat (Subramaniam, 2018) Kode *JavaScript* mengalami perubahan yang cukup signifikan saat *ES6* atau *ECMAScript 6* diterbitkan. Sekarang Anda akan dapat dengan cepat membedakan antara kode yang ditulis menggunakan sintaks sebelum *ES6* diterbitkan dan kode yang ditulis menggunakan sintaks *ES6* karena perubahannya sangat signifikan. *Javascript* modern saat ini sangat menyenangkan untuk ditulis. Saya telah lama menggeluti dunia pemrograman, tetapi saya akui bahwa kadang-kadang saya merasa sintaks sebelumnya canggung dan sulit dijelaskan daripada sintaks *Javascript* modern ini yang lebih unggul (Morgan, 2018).

Tidak seperti *ES5*, *ES6* menambahkan lebih dari sekadar sejumlah kecil *API* baru ke dalam bahasa. Ini mencakup banyak struktur sintaksis baru, beberapa di antaranya mungkin memerlukan waktu untuk membiasakan diri. Selain itu, ada banyak *organization form* baru dan dukungan *API* tambahan untuk berbagai jenis

data. *ES6* adalah kemajuan yang signifikan untuk bahasa ini. *ES6* penuh dengan fitur baru yang belum Anda ketahui (Simpson, 2015).

8. MySQL



Gambar 2.4 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang populer digunakan untuk aplikasi web. MySQL menjadi pilihan utama karena keandalannya dalam mengelola data untuk berbagai aplikasi web. Kelebihan MySQL meliputi kemudahan penggunaan dengan sintaks SQL yang mudah dipelajari dan digunakan. Selain itu, MySQL menawarkan skalabilitas yang memungkinkan untuk dijalankan pada berbagai platform, mulai dari komputer pribadi hingga server enterprise. MySQL juga dikenal dengan performanya yang tinggi dalam menangani query dan database besar. Sebagai software open-source, MySQL didistribusikan secara gratis dan bebas digunakan untuk tujuan apa pun. Kelebihan lain dari MySQL adalah adanya komunitas pengguna yang besar dan aktif yang menyediakan dukungan dan sumber daya.

MySQL mendukung berbagai fitur penting dalam pengelolaan basis data, termasuk transaksi, indeks, replikasi, dan keamanan data. "MySQL menyediakan

mekanisme keamanan berlapis, termasuk enkripsi data, autentikasi pengguna, dan manajemen hak akses yang granular, memungkinkan administrator basis data untuk mengontrol akses ke data dengan lebih baik.". Fleksibilitas MySQL dalam mendukung berbagai platform dan bahasa pemrograman juga menjadi salah satu keunggulannya, Kemampuan MySQL untuk berintegrasi dengan berbagai bahasa pemrograman seperti PHP, Java, Python, dan .NET membuatnya menjadi pilihan yang versatil untuk pengembangan aplikasi lintas platform.

9. PhpMyAdmin

Phpmyadmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *phpmyadmin*, anda dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng- *update data* dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara *manual*. Karena berbasis *web*, maka *phpmyadmin* dapat dijalankan di banyak *OS*, selama dapat menjalankan *web server* dan *MySQL*.

10. UML (Unified Modelling Language)

UML adalah serangkaian alat yang digunakan untuk menyederhanakan suatu sistem atau perangkat lunak berbasis objek. Singkatan dari *Unified Modelling Language*, *UML* juga berperan dalam mempermudah pengembangan aplikasi secara berkelanjutan. Sistem atau aplikasi yang tidak didokumentasikan dengan baik seringkali menghambat proses pengembangan karena memaksa para pengembang untuk melakukan penyelidikan dan memahami kode program yang kompleks. *UML* juga berfungsi sebagai alat untuk men-*transfer* pengetahuan tentang suatu sistem atau aplikasi dari satu pengembang ke pengembang lainnya.

Keberadaan *UML* tidak hanya mempermudah komunikasi antara para pengembang, tetapi juga memungkinkan orang di luar lingkaran pengembangan, seperti pihak bisnis atau siapapun, untuk memahami suatu sistem dengan lebih baik.

UML (Unified Modelling Language), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. *UML* menyediakan format standar untuk merancang model sistem. Namun, karena *UML* memanfaatkan kelas dan operasi sebagai bagian dari konsep intinya, lebih sesuai digunakan untuk pengembangan perangkat lunak dengan bahasa berbasis objek seperti C++, Java, C#, atau VB.NET. Meskipun begitu, *UML* masih bisa digunakan untuk memodelkan aplikasi prosedural dalam bahasa seperti VB atau C.

UML mulai dikembangkan oleh *Object Management Group* sejak versi 1.0 dirilis pada bulan Januari 1997. Dalam pengembangan berbasis objek, terdapat beberapa prinsip kunci yang harus dipahami, seperti Objek, Kelas, Abstraksi, Enkapsulasi, Pewarisan, dan Polimorfisme. Adapun daftar simbol *UML* yaitu:

Tabel 2.1 *Symbol Use case Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use case</i> .
2		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>Independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

No	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan di mana objek anak (<i>Descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>Ancestor</i>).
4		<i>Include</i>	Menyepesifikasikan bahwa <i>Use case</i> sumber secara Eksplisit.
5		<i>Extend</i>	Menyepesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menyepesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.2 *Symbol Class Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

No.	GAMBAR	NAMA	KETERANGAN
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2.3 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

Tabel 2.4 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai Link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.

No	GAMBAR	NAMA	KETERANGAN
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.5 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

B. Kajian Hasil Penelitian Terdahulu

Untuk dapat digunakan sebagai data pendukung, diperlukan berbagai penelitian sebelumnya. Studi sebelumnya yang berkaitan dengan masalah yang dibahas dalam penelitian ini merupakan salah satu informasi pendukung yang peneliti butuhkan. Dalam hal ini, referensi yang berkaitan dengan tantangan teknologi informasi dari studi sebelumnya digunakan. Oleh karena itu penelitian dilakukan secara *online* dengan menggunakan berbagai temuan penelitian yang disajikan dalam makalah akhir, tesis, atau publikasi.

(Irwan et al., 2023) dengan judul “Sistem Informasi Pelayanan Kantor Urusan Agama Pitu Riase”. Penelitian ini menghasilkan sistem informasi pelayanan berbasis web untuk Kantor Urusan Agama Pitu Riase. Sistem ini mempermudah pegawai dalam mengelola data dan layanan terkait pernikahan, surat masuk dan keluar, serta informasi masjid dan kurban. Pengujian dilakukan menggunakan metode *Black Box* dan *White Box*, dan hasilnya menunjukkan bahwa sistem bebas dari kesalahan.

(Hidayat et al., 2023). Penelitian berjudul “Perancangan Aplikasi Pelayanan Masyarakat di Wilayah Hukum Polres Pinrang”. Penelitian ini menghasilkan aplikasi yang memudahkan masyarakat melaporkan kejadian kepada petugas kepolisian melalui platform Android. Aplikasi ini dilengkapi dengan fitur-fitur seperti laporan ke polisi, pelaporan kejadian, dan peta untuk melihat titik koordinat pengguna. Tujuan dari penelitian ini adalah untuk membangun sebuah aplikasi yang membantu masyarakat melaporkan kejadian dengan cepat dan efisien kepada petugas kepolisian. Metodologi penelitian menggunakan metode survei dan dilaksanakan di Polres Pinrang selama 4 bulan.

(Hasan et al., 2023) melakukan penelitian dengan judul Sistem Informasi Manajemen PT. Calidad Coco Indonesia. Penelitian ini dilatarbelakangi oleh kebutuhan PT Calidad Coco Indonesia untuk meningkatkan efisiensi pengelolaan data perusahaan dan mengurangi ketidakakuratan data. Metodologi penelitian mengadopsi pendekatan waterfall, mencakup tahapan kebutuhan sistem, analisis, desain, coding, pengujian/verifikasi, dan pemeliharaan, dengan pengumpulan data dilakukan melalui metode kualitatif. Hasil penelitian menunjukkan bahwa sistem informasi yang dikembangkan memungkinkan pimpinan perusahaan untuk mengakses laporan secara fleksibel, mendukung pengambilan keputusan yang lebih baik. Selain itu, masyarakat Desa Tana Cellae merespons positif terhadap perusahaan ini karena sumbangan terhadap peningkatan lapangan kerja. Secara kesimpulan, sistem berbasis Android ini telah membawa kemudahan dalam pengelolaan data dan penyajian laporan yang akurat, berperan penting dalam proses pengambilan keputusan perusahaan.

C. Kerangka Pikir

Pemerintah, sebagai penyelenggara pelayanan publik, dituntut untuk beradaptasi dengan perkembangan teknologi guna meningkatkan efisiensi dan kualitas layanan. Kantor kelurahan, sebagai unit pemerintahan yang terdekat dengan masyarakat, memiliki peran penting dalam memberikan layanan administrasi dan informasi yang cepat, tepat, dan transparan.



Banyak kantor kelurahan yang masih menggunakan sistem manual dalam pengelolaan administrasi dan pelayanan publik. Hal ini menyebabkan beberapa permasalahan seperti proses pelayanan yang lambat, proses manual mengharuskan pegawai kelurahan melakukan pekerjaan administratif berulang



Teknologi informasi yang mampu mengotomatisasi proses pelayanan publik di kantor kelurahan. Teknologi informasi ini harus dapat menangani berbagai aspek administrasi dan operasional secara efisien



Direncanakan Aplikasi Layanan Publik Di Kantor Kelurahan

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini merupakan penelitian pengembangan (*development research*) yang bertujuan untuk merancang, mengembangkan, dan mengimplementasikan aplikasi layanan publik di kantor kelurahan menggunakan teknologi web.

B. Lokasi dan Waktu Penelitian

Penelitian dilakukan di Kantor Lurah Kelurahan Temmassarangnge, Kabupaten Pinrang, Sulawesi Selatan. Penelitian ini akan berlangsung selama ± 2 (dua) bulan di tahun 2024.

C. Alat dan Bahan Penelitian

1. Perangkat keras

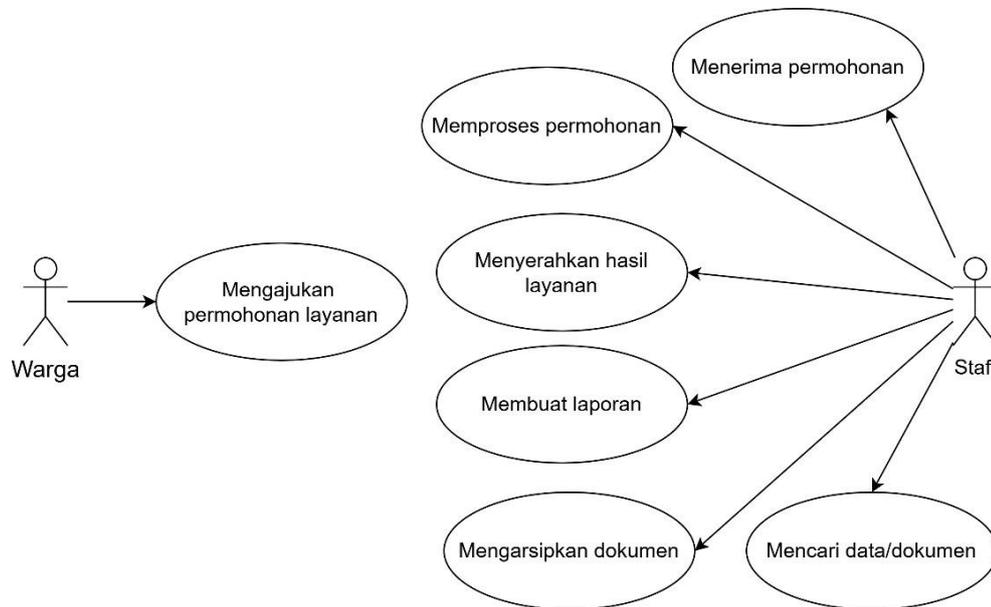
- a. Laptop ASUS A416EA dengan spesifikasi hardware:
 - *Processor* : Intel Core i3-8145U
 - *Installed RAM* : 4GB DDR4
 - *Penyimpanan* : 512GB

2. Perangkat lunak

- a. Windows 11 Home Single Language
- b. Visual Studio Code
- c. XAMPP
- d. PHP
- e. Javascript

D. Rancangan Sistem

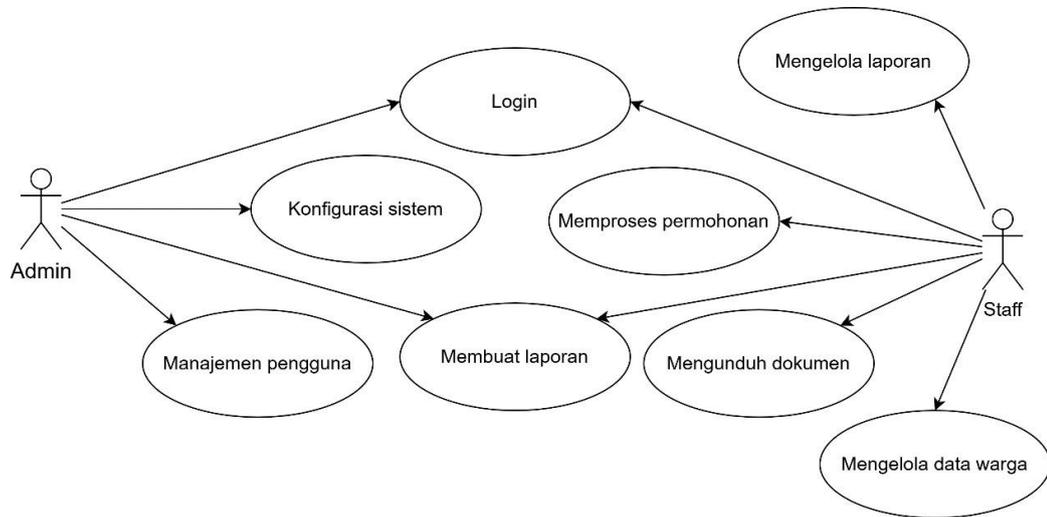
1. Sistem yang berjalan



Gambar 3.1 Sistem yang berjalan

Pada gambar 3.1 menunjukkan Diagram *use case* yang menggambarkan proses pengajuan permohonan layanan di sebuah kantor pelayanan publik, yang melibatkan dua aktor utama, yaitu warga dan *staff*. Warga sebagai pemohon memulai proses dengan mengajukan permohonan layanan kepada *staff*. Setelah menerima permohonan tersebut, *staff* bertanggung jawab untuk memprosesnya. Proses ini melibatkan beberapa langkah, termasuk pencarian data atau dokumen yang diperlukan untuk mendukung pemrosesan. Setelah permohonan diproses, *staff* akan menyerahkan hasil layanan kembali kepada warga. Selain itu, *staff* juga memiliki tanggung jawab tambahan, seperti membuat laporan terkait permohonan yang telah diproses dan mengarsipkan dokumen-dokumen yang berhubungan dengan permohonan tersebut.

2. Sistem yang diusulkan



Gambar 3.2 Sistem yang diusulkan

Pada gambar 3.2 menunjukkan Diagram *use case* yang menggambarkan proses kerja dalam aplikasi pelayanan kelurahan dengan dua peran utama: *Admin* dan *Staff*. *Admin* bertanggung jawab untuk melakukan *Login*, Konfigurasi sistem, Manajemen pengguna, dan Membuat laporan. *Admin* memastikan sistem berjalan sesuai kebutuhan kelurahan dan mengelola akun-akun pengguna serta membuat laporan untuk evaluasi kinerja. *Staff* juga melakukan *Login* untuk mengakses sistem dan memiliki tugas utama seperti Memproses permohonan, Mengunduh dokumen, Mengelola data warga, dan Mengelola laporan. *Staff* menangani permohonan layanan dari warga, memastikan data warga selalu diperbarui, serta membuat dan mengelola laporan terkait layanan yang diberikan.

E. Teknik Pengumpulan Data

1. Wawancara

Melakukan wawancara dengan pegawai kelurahan dan warga yang menggunakan layanan publik untuk mendapatkan informasi mengenai kebutuhan dan masalah yang dihadapi dalam pelayanan publik saat ini.

2. Observasi

Melakukan observasi langsung terhadap proses pelayanan publik di kantor kelurahan untuk memahami alur kerja dan mengidentifikasi area yang memerlukan perbaikan.

3. Dokumentasi

Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis, dokumen elektronik, atau dokumen *audiovisual*.

BAB IV

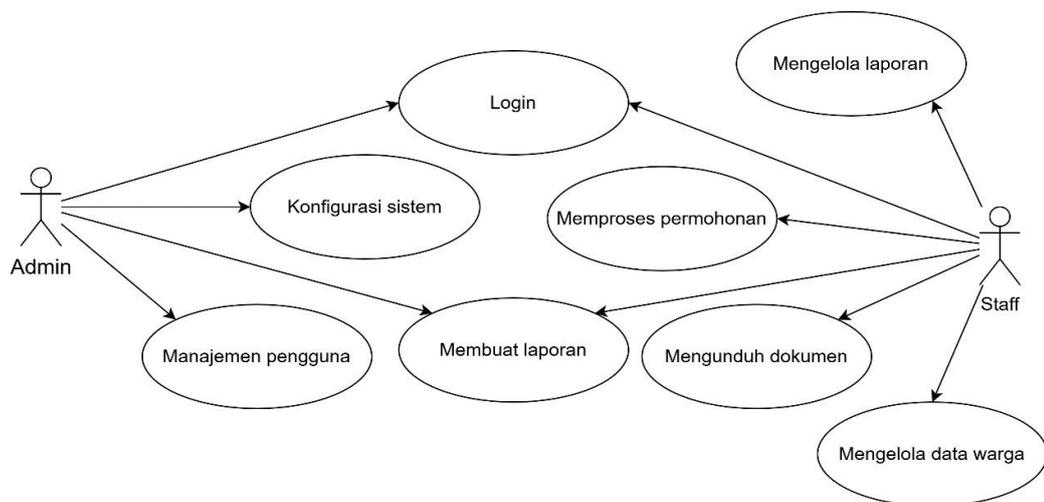
HASIL PEMBAHASAN

A. Analisis Aliran Data

Analisis aliran data menggunakan *Unified Modeling Language* (UML) adalah langkah penting dalam merancang sistem informasi yang kompleks. UML merupakan bahasa pemodelan yang digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Salah satu aspek penting dalam UML adalah kemampuan untuk memetakan aliran data di dalam sistem, yang memungkinkan pengembang untuk memahami bagaimana informasi bergerak dan diolah di berbagai bagian sistem.

1. Use Case Diagram

Use case diagram merupakan gambaran dari interaksi antara aktor dengan sistem pada suatu rancangan sistem.



Gambar 4.1 Use Case

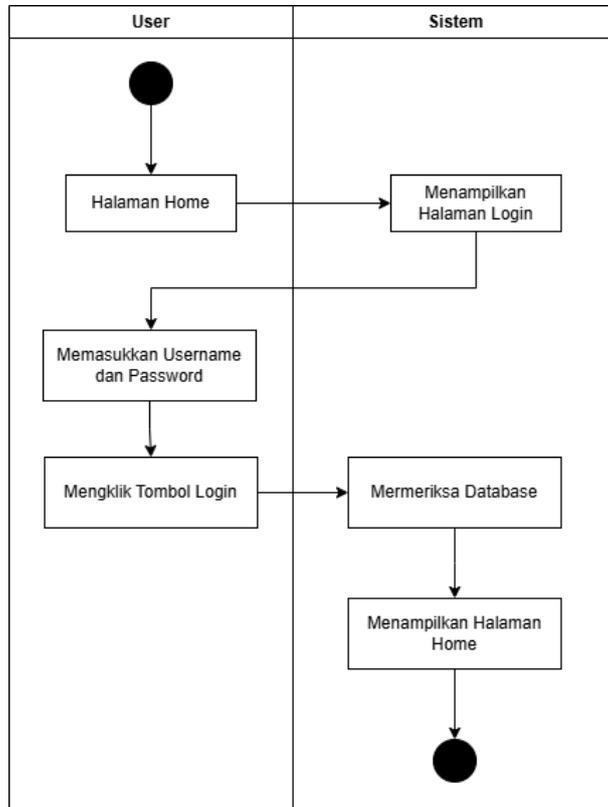
Tabel 4.1 *Use Case Scenario*

Nama <i>Use case</i>	Deskripsi
Login	<i>Use case</i> ini berfungsi sebagai tempat menginput <i>username</i> dan <i>password</i> untuk login ke halaman utama.
Konfigurasi Sistem	<i>Use case</i> ini berfungsi untuk mengatur segala sistem yang ada pada web
Membuat Laporan	<i>Use case</i> ini berfungsi untuk membuat laporan yang di butuhkan.
Manajemen Pengguna	<i>Use case</i> ini berfungsi untuk mengelola data, akses, dan hak pengguna dalam suatu sistem atau situs web.
Mengelola Laporan	<i>Use case</i> ini berfungsi untuk mengelolah laporan data.
Memproses Permohonan	<i>Use case</i> ini berfungsi untuk memproses data yang di ajukan.
Mengunduh Dokumen	<i>Use case</i> ini berfungsi untuk mengunduh dokumen yang telah di buat.
Mengelolah Data Warga	<i>Use case</i> ini berfungsi untuk menambah, mengubah dan menghapus data warga

2. Activity Diagram

Activity diagram digunakan untuk memodelkan aliran kerja atau aliran kontrol dari sebuah proses bisnis atau *use case*. Diagram ini menunjukkan aktivitasaktivitas yang terjadi dalam sistem dan urutan eksekusinya.

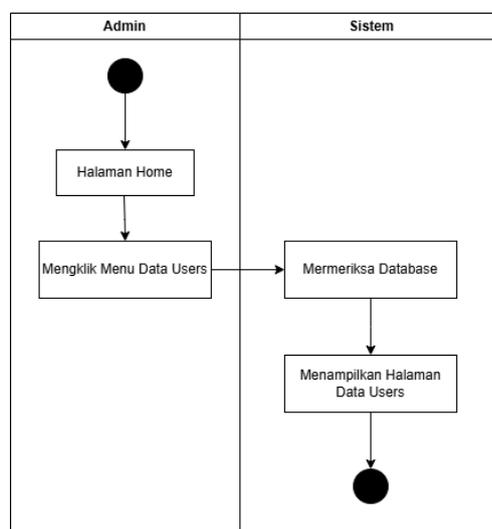
a. *Activity Diagram Login*



Gambar 4.2 *Activity Diagram Login*

Pada Gambar 4.2 diatas menjelaskan tentang menu login, dimana *user* ingin login maka sistem akan menampilkan halaman login, setelah *user* login akan menampilkan halaman home dan pada aplikasi ini terdapat dua *users* yaitu admin dan staff

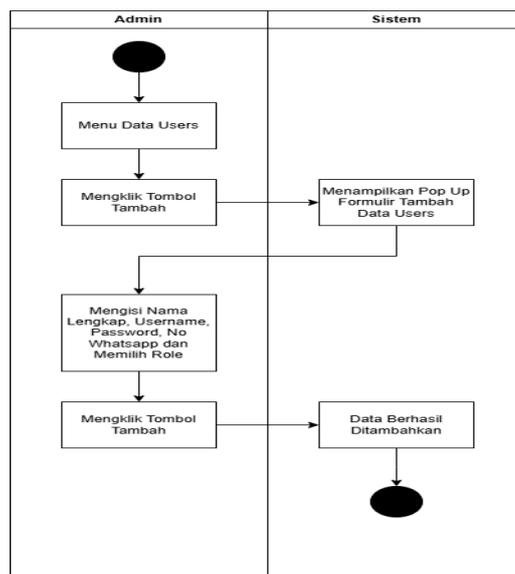
b. *Activity Diagram Menu Data Users*



Gambar 4.3 *Activity Diagram* Halaman Data Users

Pada Gambar 4.3 diagram tersebut menunjukkan alur kerja antara admin dan sistem dalam pengelolaan data *users*. Admin memulai dari halaman home, lalu mengklik menu data *users*. Sistem merespons dengan memeriksa database untuk mengambil data pengguna, kemudian menampilkan halaman data *users* yang berisi informasi pengguna. Proses berakhir setelah data berhasil ditampilkan.

c. *Activity Diagram* Tambah Data Users

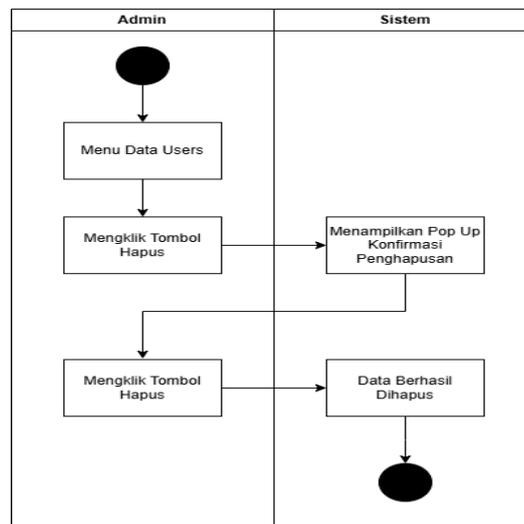


Gambar 4.4 *Activity Diagram* Tambah Data Users

Pada Gambar 4.4 diagram tersebut menunjukkan proses penambahan pengguna dimulai ketika admin membuka menu data *users* dan mengklik tombol tambah, yang kemudian memicu sistem untuk menampilkan popup formulir tambah pengguna. admin lalu mengisi formulir dengan data seperti nama lengkap, *username*, password, nomor whatsapp, serta memilih *role* pengguna. Setelah data

terisi, admin menekan tombol tambah, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

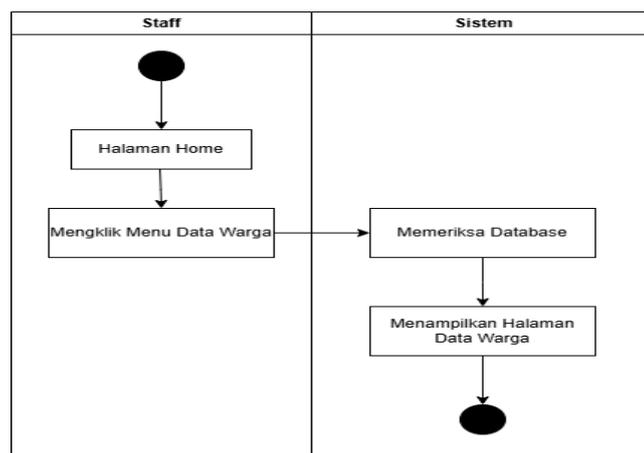
d. *Activity Diagram Hapus Data Users*



Gambar 4.5 *Activity Diagram Hapus Data Users*

Pada Gambar 4.5 diagram tersebut menunjukkan proses penghapusan pengguna dimulai ketika admin membuka menu data *users* dan mengklik tombol hapus pada data pengguna yang ingin dihapus. Setelah itu, sistem akan menampilkan popup konfirmasi penghapusan untuk memastikan bahwa admin benar-benar ingin menghapus data tersebut. Jika admin mengonfirmasi dengan mengklik tombol hapus, sistem akan memproses permintaan dan menghapus data pengguna dari database.

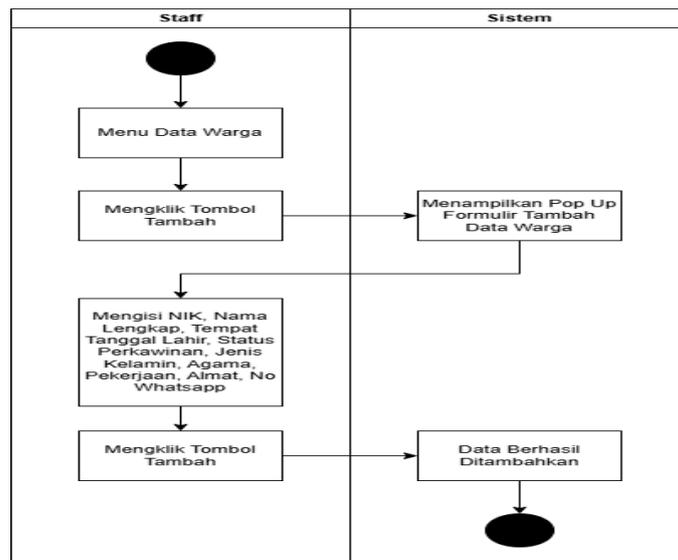
e. *Activity Diagram Menu Data Warga*



Gambar 4.6 Activity Diagram Menu Data Warga

Pada Gambar 4.6 diagram tersebut menunjukkan alur kerja antara staff dan sistem dalam pengelolaan data warga. Staff memulai dari halaman home, lalu mengklik menu data warga. Sistem merespons dengan memeriksa database untuk mengambil data pengguna, kemudian menampilkan halaman data warga yang berisi informasi pengguna. Proses berakhir setelah data berhasil ditampilkan.

f. Activity Diagram Tambah Data Warga

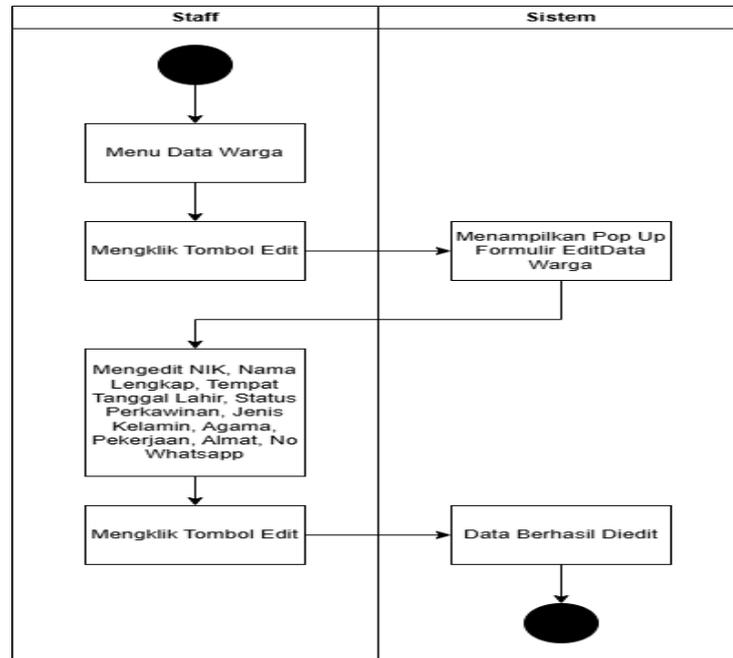


Gambar 4.7 Activity Diagram Tambah Data Warga

Pada Gambar 4.7 diagram tersebut menunjukkan proses penambahan warga dimulai ketika staff membuka menu data warga dan mengklik tombol tambah, yang kemudian memicu sistem untuk menampilkan popup formulir tambah warga. Staff lalu mengisi formulir dengan data seperti nik, nama lengkap, tempat tanggal lahir, status perkawinan, jenis kelamin, agama, pekerjaan, alamat, no whatsapp. Setelah

data terisi, staff menekan tombol tambah, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

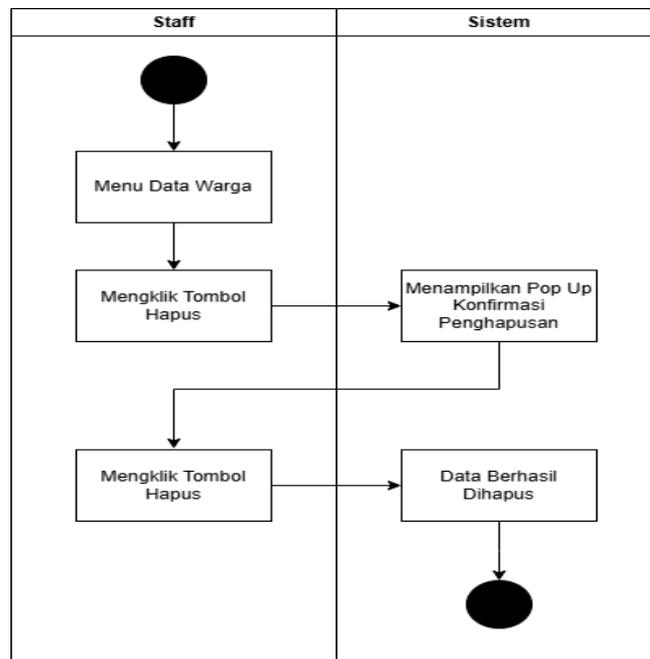
g. *Activity Diagram* Edit Data Warga



Gambar 4.8 *Activity Diagram* Edit Data Warga

Pada Gambar 4.8 diagram tersebut menunjukkan proses pengeditan warga dimulai ketika staff membuka menu data warga dan mengklik tombol edit pada data warga yang ingin di edit, yang kemudian memicu sistem untuk menampilkan popup formulir edit warga. Staff lalu mengedit formulir dengan data seperti nik, nama lengkap, tempat tanggal lahir, status perkawinan, jenis kelamin, agama, pekerjaan, alamat, no whatsapp. Setelah data diedit, staff menekan tombol edit, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

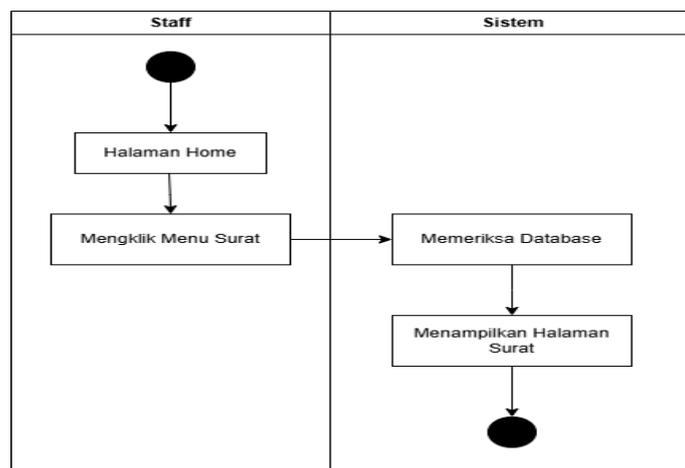
h. *Activity Diagram* Hapus Data Warga



Gambar 4.9 *Activity Diagram* Hapus Data Warga

Pada Gambar 4.9 diagram tersebut menunjukkan proses penghapusan pengguna dimulai ketika staff membuka menu data warga dan mengklik tombol hapus pada data warga yang ingin dihapus. Setelah itu, sistem akan menampilkan popup konfirmasi penghapusan untuk memastikan bahwa staff benar-benar ingin menghapus data tersebut. Jika staff mengonfirmasi dengan mengklik tombol hapus, sistem akan memproses permintaan dan menghapus data warga dari database.

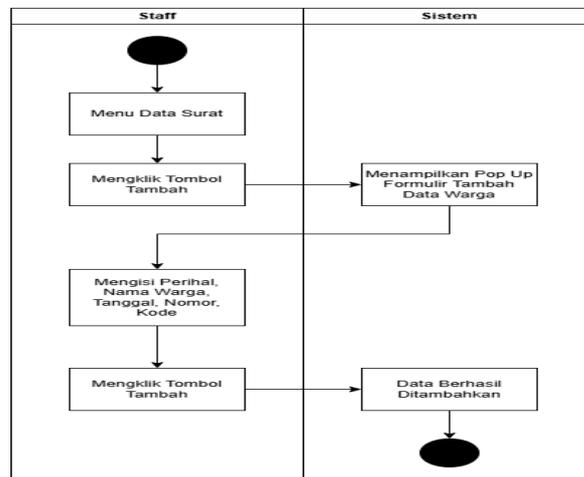
i. *Activity Diagram* Menu Data Surat



Gambar 4.10 *Activity Diagram* Menu Data Surat

Pada Gambar 4.10 diagram tersebut menunjukkan alur kerja antara staff dan sistem dalam pengelolaan data surat. Staff memulai dari halaman home, lalu mengklik menu data surat. Sistem merespons dengan memeriksa database untuk mengambil data pengguna, kemudian menampilkan halaman data surat yang berisi informasi pengguna. Proses berakhir setelah data berhasil ditampilkan.

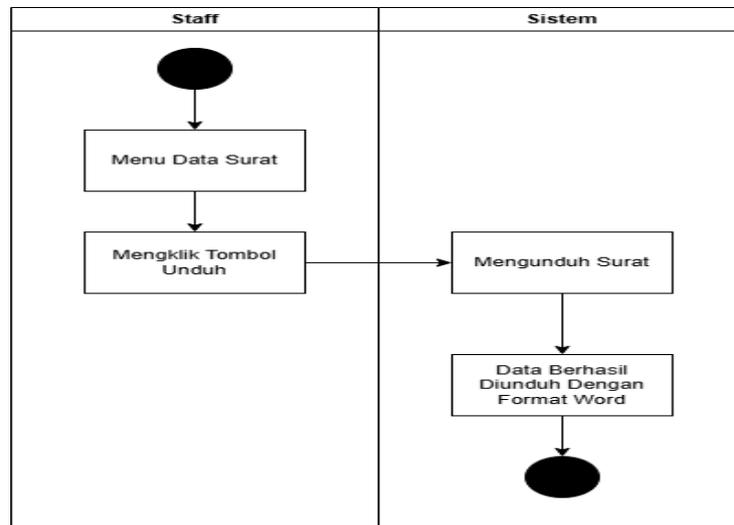
j. *Activity Diagram* Tambah Data Surat



Gambar 4.11 *Activity Diagram* Tambah Data Surat

Pada Gambar 4.11 diagram tersebut menunjukkan proses penambahan surat dimulai ketika staff membuka menu data surat dan mengklik tombol tambah, yang kemudian memicu sistem untuk menampilkan popup formulir tambah surat. Staff lalu mengisi formulir dengan data seperti perihal, nama warga, tanggal, nomor, kode. Setelah data terisi, staff menekan tombol tambah, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

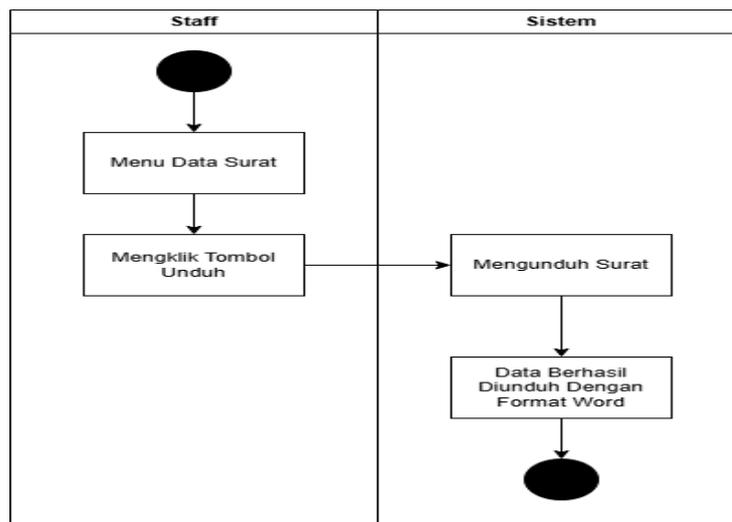
k. *Activity Diagram Unduh Data Surat*



Gambar 4.12 *Activity Diagram Unduh Data Surat*

Pada Gambar 4.12 diagram tersebut menunjukkan proses pengunduhan surat dimulai ketika staff membuka menu data surat dan mengklik tombol unduh pada data surat yang ingin di unduh , yang kemudian memicu sistem untuk mengunduh surat dengan format word.

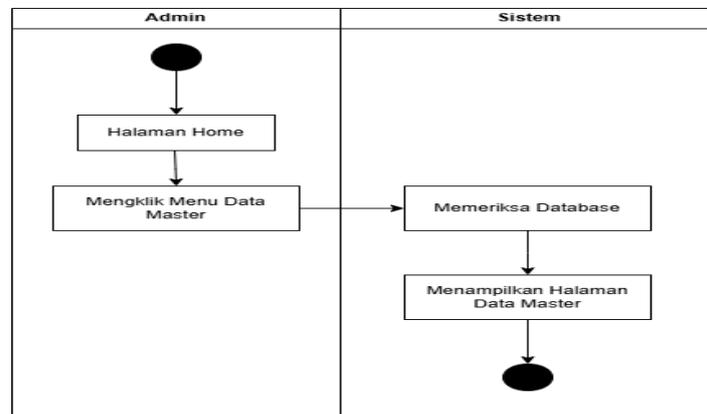
l. *Activity Diagram Hapus Data Surat*



Gambar 4 13 *Activity Diagram Hapus Data Surat*

Pada Gambar 4.13 diagram tersebut menunjukkan proses penghapusan pengguna dimulai ketika staff membuka menu data surat dan mengklik tombol hapus pada data surat yang ingin dihapus. Setelah itu, sistem akan menampilkan popup konfirmasi penghapusan untuk memastikan bahwa staff benar-benar ingin menghapus data tersebut. Jika staff mengonfirmasi dengan mengklik tombol hapus, sistem akan memproses permintaan dan menghapus data warga dari database.

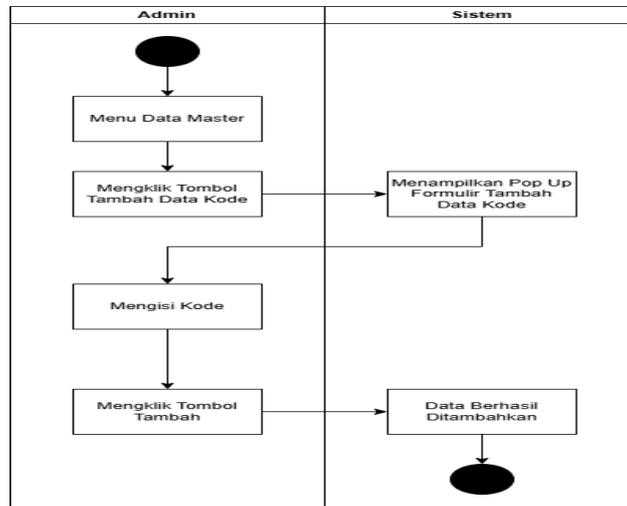
m. *Activity Diagram Menu Data Master*



Gambar 4.14 *Activity Diagram Menu Data Master*

Pada Gambar 4.14 diagram tersebut menunjukkan alur kerja antara admin dan sistem dalam pengelolaan data surat. Admin memulai dari halaman home, lalu mengklik menu data master. Sistem merespons dengan memeriksa database untuk mengambil data pengguna, kemudian menampilkan halaman data master yang berisi informasi pengguna. Proses berakhir setelah data berhasil ditampilkan.

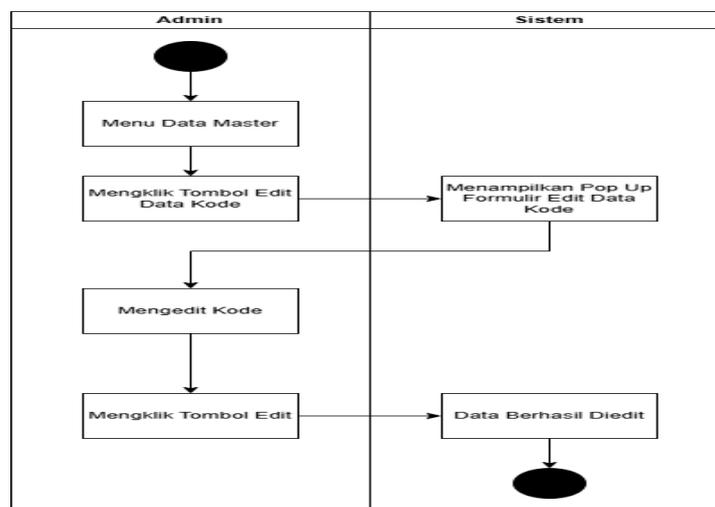
n. *Activity Diagram* Tambah Data Kode



Gambar 4.15 *Activity Diagram* Tambah Data Kode

Pada Gambar 4.15 diagram tersebut menunjukkan proses penambahan kode dimulai ketika admin membuka menu data master dan mengklik tombol tambah, yang kemudian memicu sistem untuk menampilkan popup formulir tambah kode. Admin lalu mengisi formulir dengan data kode. Setelah data terisi, admin menekan tombol tambah, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

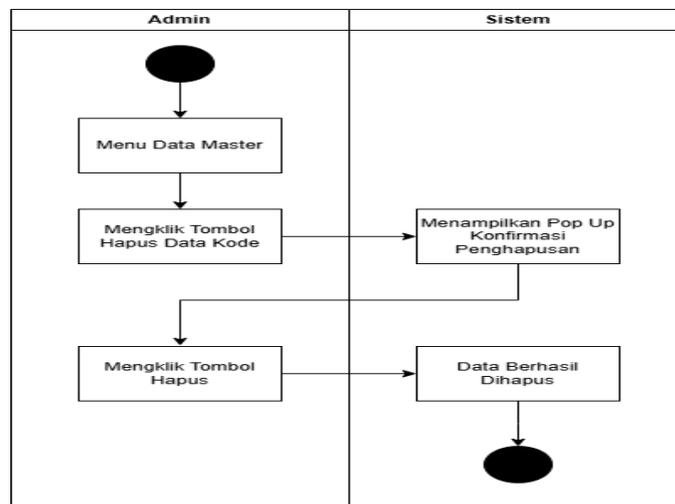
o. *Activity Diagram* Edit Data Kode



Gambar 4.16 *Activity Diagram* Edit Data Kode

Pada Gambar 4.16 diagram tersebut menunjukkan proses pengeditan kode dimulai ketika admin membuka menu data master dan mengklik tombol edit pada data kode yang ingin di edit, yang kemudian memicu sistem untuk menampilkan popup formulir edit kode. Admin lalu mengedit formulir dengan data kode. Setelah data diedit, admin menekan tombol edit, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

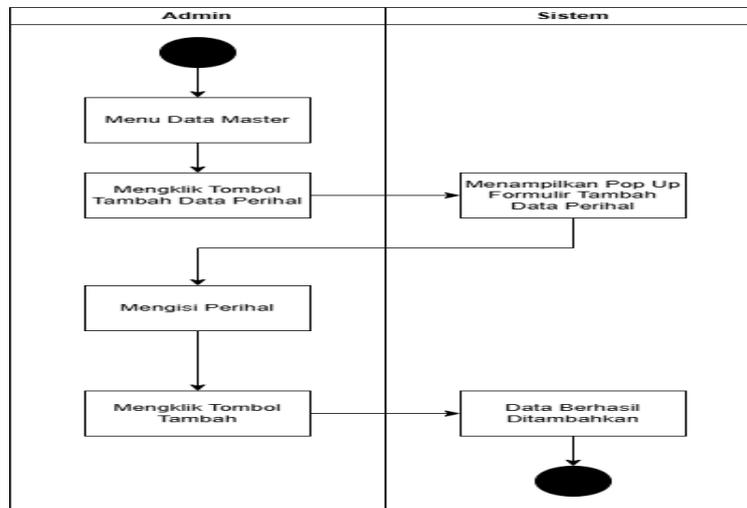
p. *Activity Diagram Hapus Data Kode*



Gambar 4 17 *Activity Diagram Hapus Data Kode*

Pada Gambar 4.17 diagram tersebut menunjukkan proses penghapusan pengguna dimulai ketika admin membuka menu data master dan mengklik tombol hapus pada data kode yang ingin dihapus. Setelah itu, sistem akan menampilkan popup konfirmasi penghapusan untuk memastikan bahwa admin benar-benar ingin menghapus data tersebut. Jika admin mengonfirmasi dengan mengklik tombol hapus, sistem akan memproses permintaan dan menghapus data kode dari database.

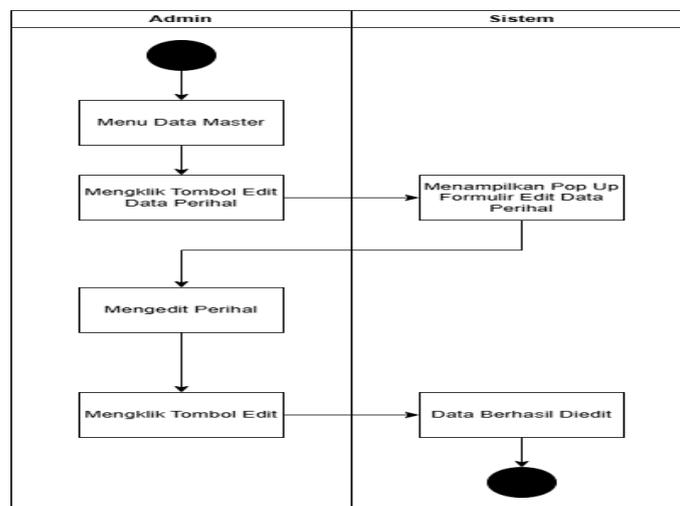
q. *Activity Diagram* Tambah Data Perihal



Gambar 4.18 *Activity Diagram* Tambah Data Perihal

Pada Gambar 4.18 diagram tersebut menunjukkan proses penambahan perihal dimulai ketika admin membuka menu data master dan mengklik tombol tambah, yang kemudian memicu sistem untuk menampilkan popup formulir tambah perihal. Admin lalu mengisi formulir dengan data perihal. Setelah data terisi, admin menekan tombol tambah, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

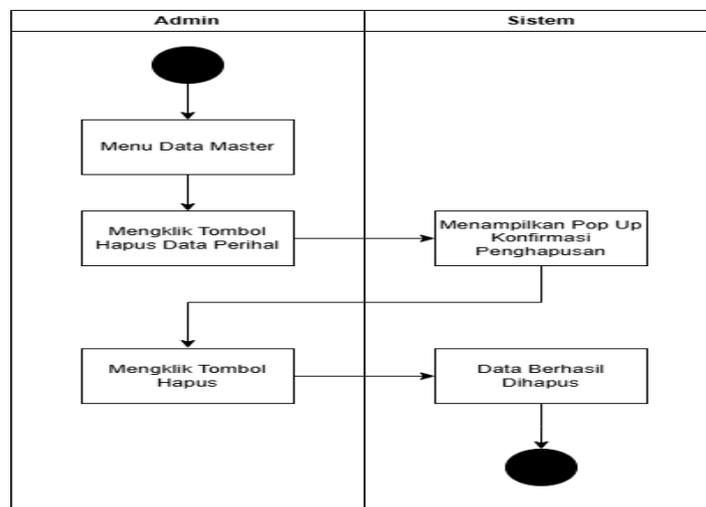
r. *Activity Diagram* Edit Data Perihal



Gambar 4.19 *Activity Diagram* Edit Data Perihal

Pada Gambar 4.19 diagram tersebut menunjukkan proses pengeditan kode dimulai ketika admin membuka menu data master dan mengklik tombol edit pada data perihal yang ingin di edit, yang kemudian memicu sistem untuk menampilkan popup formulir edit perihal. Admin lalu mengisi formulir dengan data perihal. Setelah data diedit, admin menekan tombol edit, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

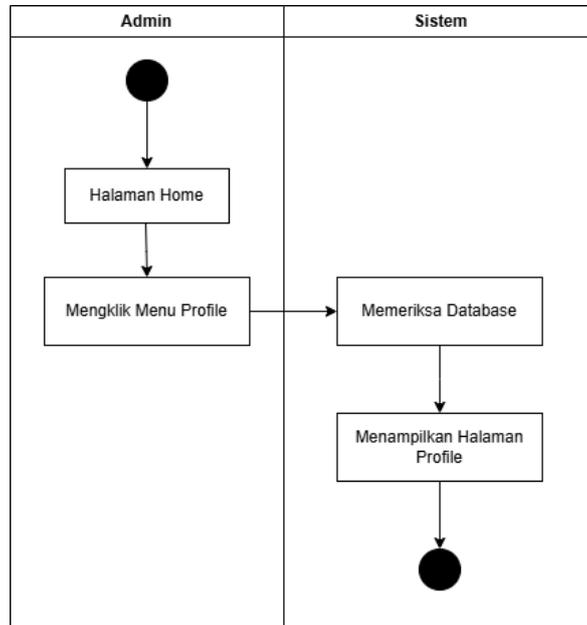
s. *Activity Diagram Hapus Data Perihal*



Gambar 4.20 *Activity Diagram Hapus Data Perihal*

Pada Gambar 4.17 diagram tersebut menunjukkan proses penghapusan pengguna dimulai ketika admin membuka menu data master dan mengklik tombol hapus pada data perihal yang ingin dihapus. Setelah itu, sistem akan menampilkan popup konfirmasi penghapusan untuk memastikan bahwa admin benar-benar ingin menghapus data tersebut. Jika admin mengonfirmasi dengan mengklik tombol hapus, sistem akan memproses permintaan dan menghapus data kode dari database.

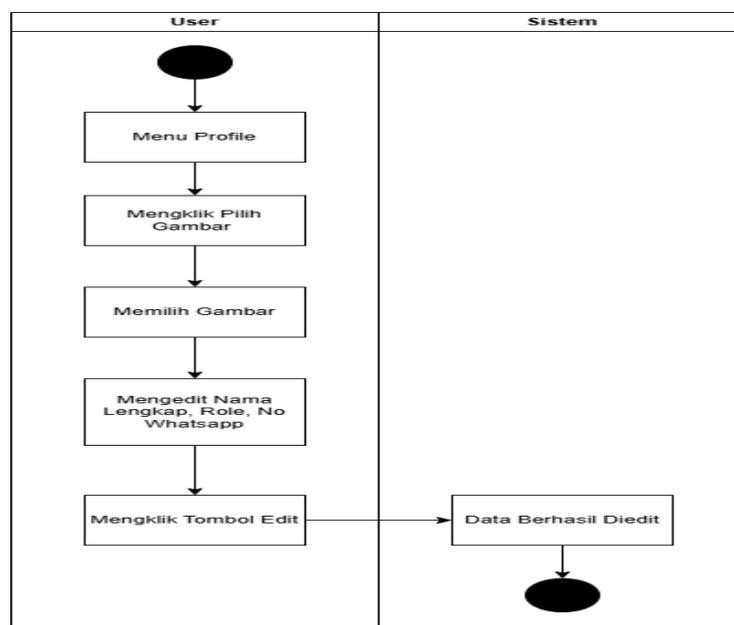
t. *Activity Diagram Profile*



Gambar 4.21 *Activity Diagram Profile*

Pada Gambar 4.7 diagram tersebut menunjukkan alur kerja antara *user* dan sistem dalam pengelolaan data surat. *user* memulai dari halaman home, lalu mengklik menu profile. Sistem merespon dengan memeriksa database untuk mengambil data pengguna, kemudian menampilkan halaman profile yang berisi informasi pengguna. Proses berakhir setelah data berhasil ditampilkan.

u. *Activity Diagram Edit Profile*



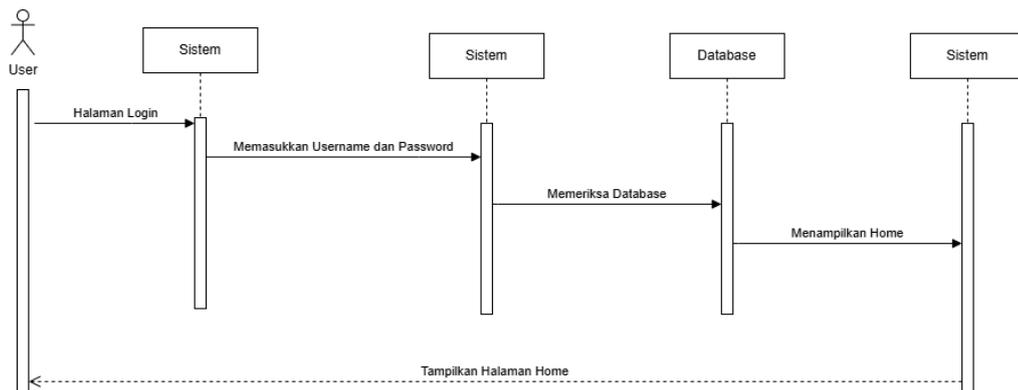
Gambar 4 22 *Activity Diagram Edit Profile*

Pada Gambar 4.8 diagram tersebut menunjukkan proses pengeditan warga dimulai ketika *user* membuka profile . Staff lalu mengedit dengan data seperti foto, nama lengkap, *role*, no whatsapp. Setelah data diedit, *user* menekan tombol edit, sehingga sistem memproses dan menyimpan informasi tersebut ke dalam database.

3. Sequence Diagram

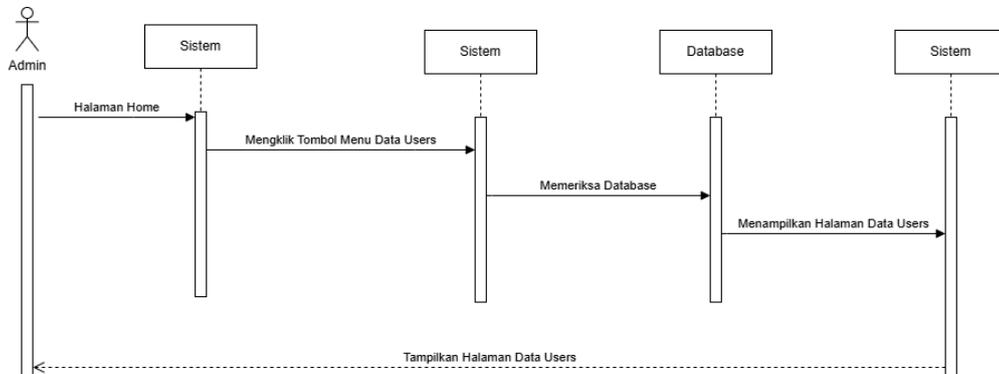
Sequence diagram merupakan salah satu diagram interaction yang menjelaskan bagaimana suatu operasi itu dilakukan message (pesan) apa yang dikirim dan kapan pelaksanaannya.

a. Sequence Diagram Login



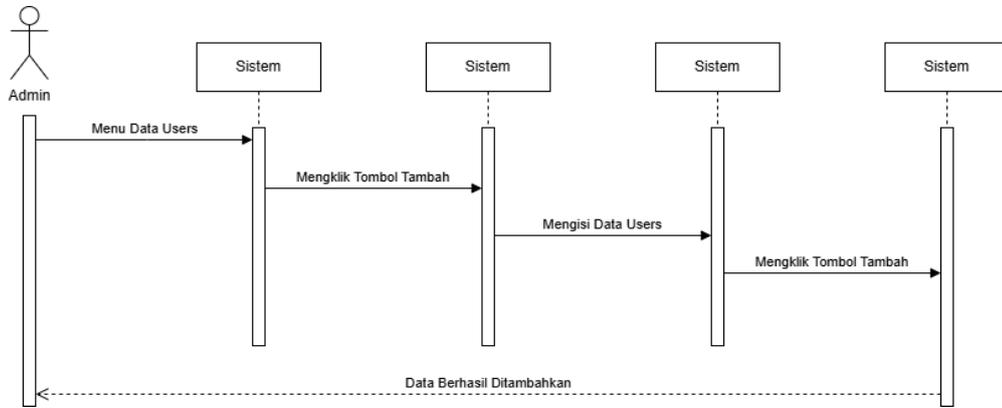
Gambar 4.23 Sequence Diagram Login

b. Sequence Diagram Menu Data Users



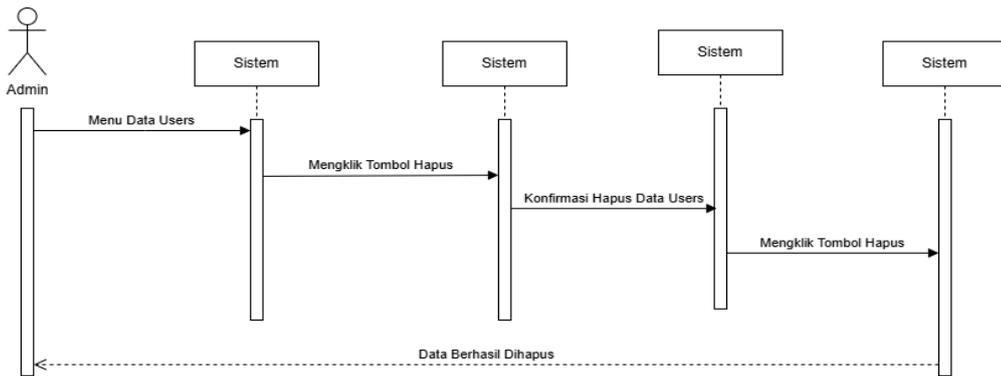
Gambar 4.24 Sequence Diagram Menu Data Users

c. *Sequence Diagram Tambah Data Users*



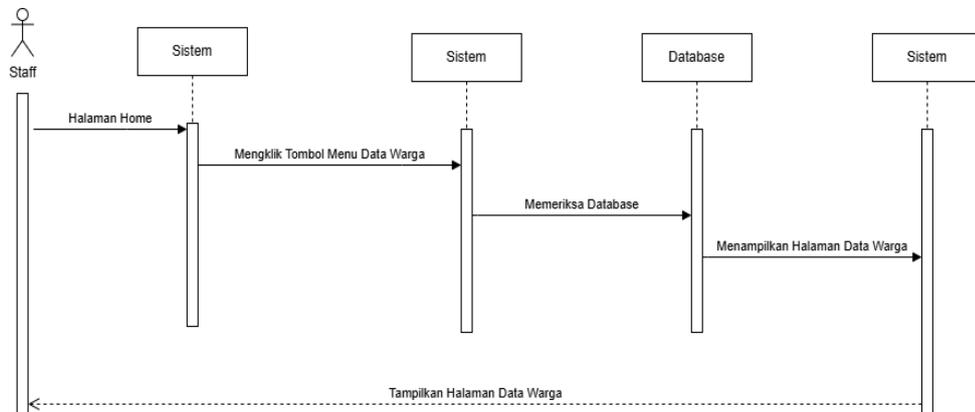
Gambar 4.25 *Sequence Diagram Tambah Data Users*

d. *Sequence Diagram Hapus Data Users*



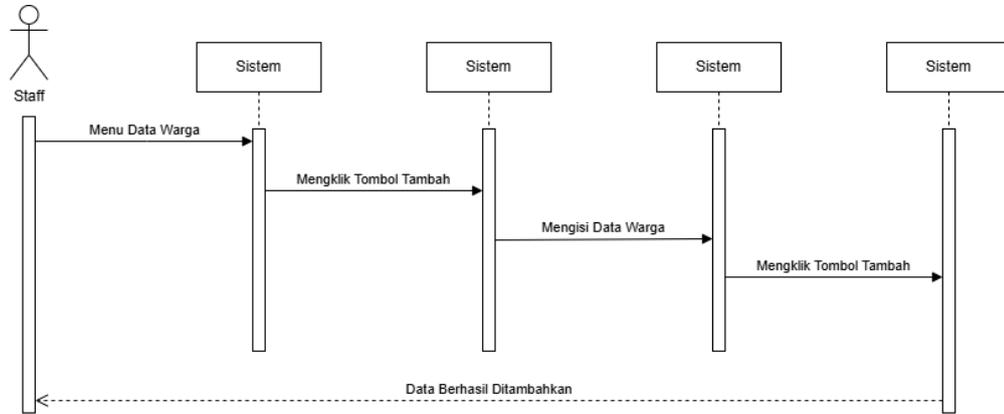
Gambar 4.26 *Sequence Diagram Hapus Data Users*

e. *Sequence Diagram Data Warga*



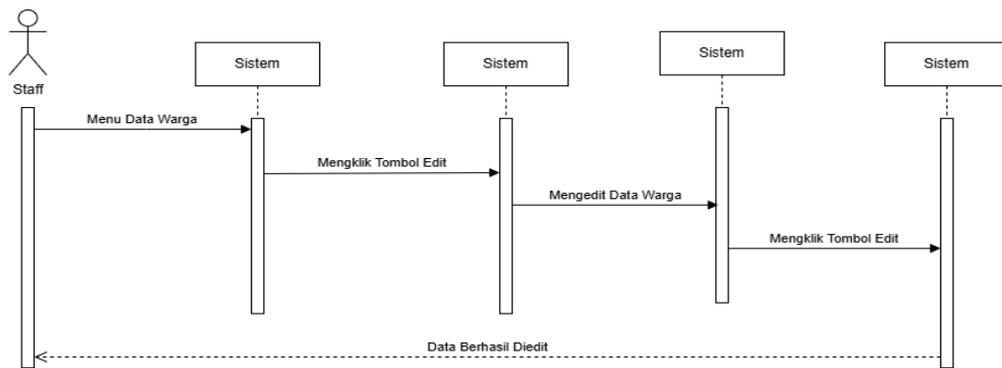
Gambar 4.27 *Sequence Diagram Data Warga*

f. *Sequence Diagram* Tambah Data Warga



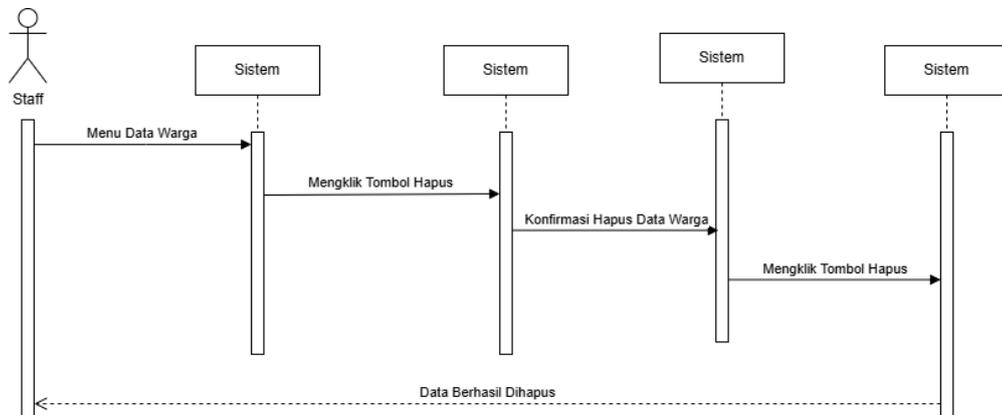
Gambar 4.28 *Sequence Diagram* Tambah Data Warga

g. *Sequence Diagram* Edit Data Warga



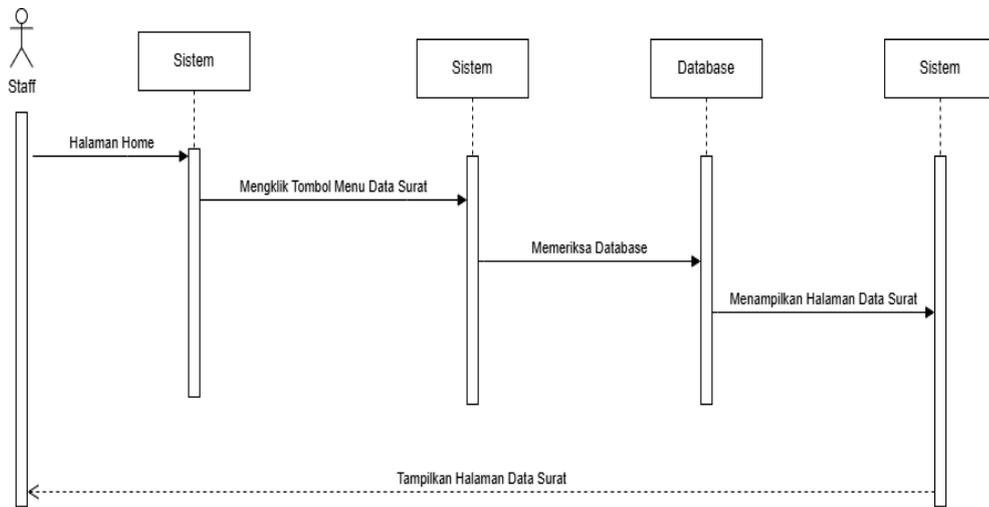
Gambar 4.29 *Sequence Diagram* Edit Data Warga

h. *Sequence Diagram* Hapus Data Warga



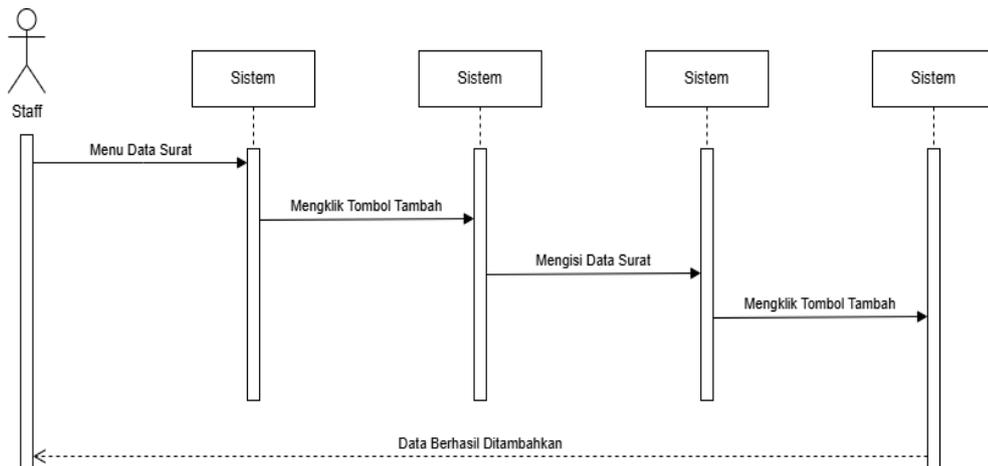
Gambar 4.30 *Sequence Diagram* Hapus Data Warga

i. *Sequence Diagram Data Surat*



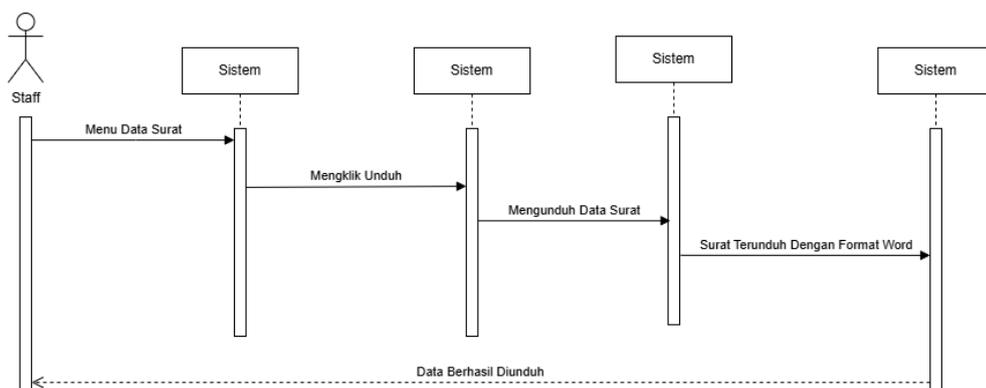
Gambar 4.31 *Sequence Diagram Data Surat*

j. *Sequence Diagram Tambah Data Surat*



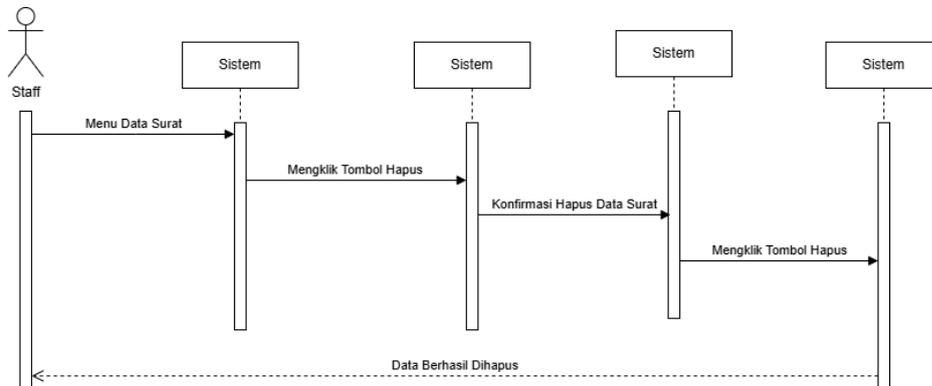
Gambar 4.32 *Sequence Diagram Tambah Data Surat*

k. *Sequence Diagram Unduh Data Surat*



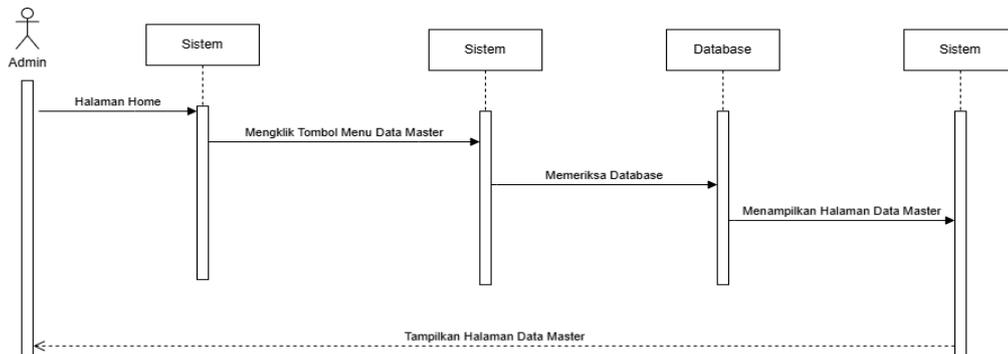
Gambar 4.33 *Sequence Diagram Unduh Data Surat*

1. *Sequence Diagram Hapus Data Surat*



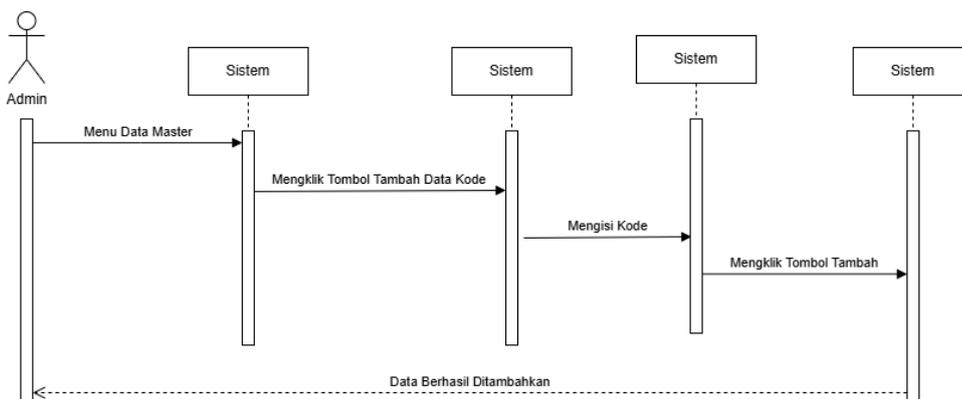
Gambar 4.34 *Sequence Diagram Hapus Data Surat*

m. *Sequence Diagram Data Master*



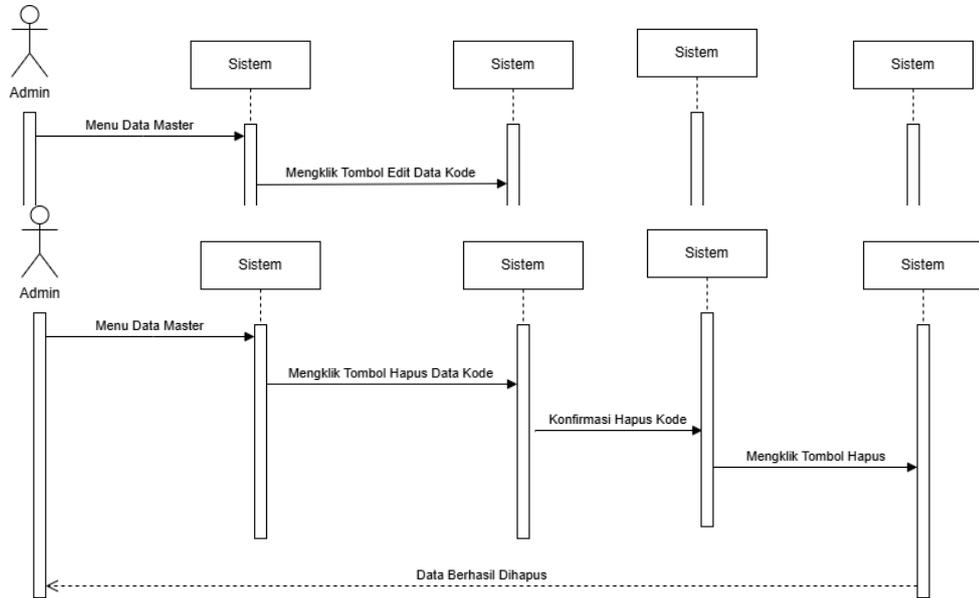
Gambar 4.35 *Sequence Diagram Data Master*

n. *Sequence Diagram Tambah Data Kode*



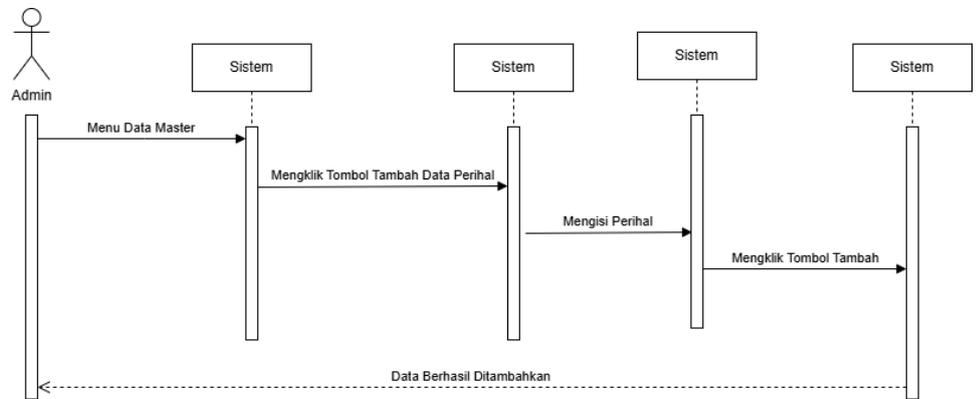
Gambar 4.36 *Sequence Diagram Tambah Data Kode*

o. *Sequence Diagram Edit Data Kode*



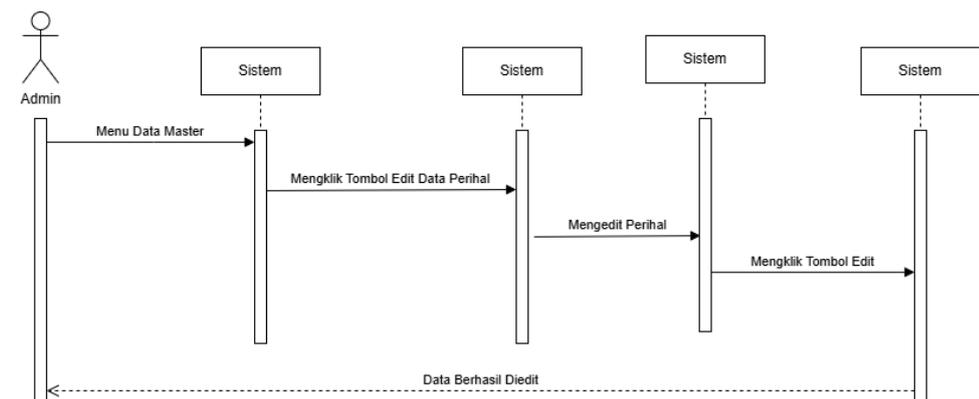
Gambar 4.38 *Sequence Diagram Hapus Data Kode*

q. *Sequence Diagram Tambah Data Perihal*



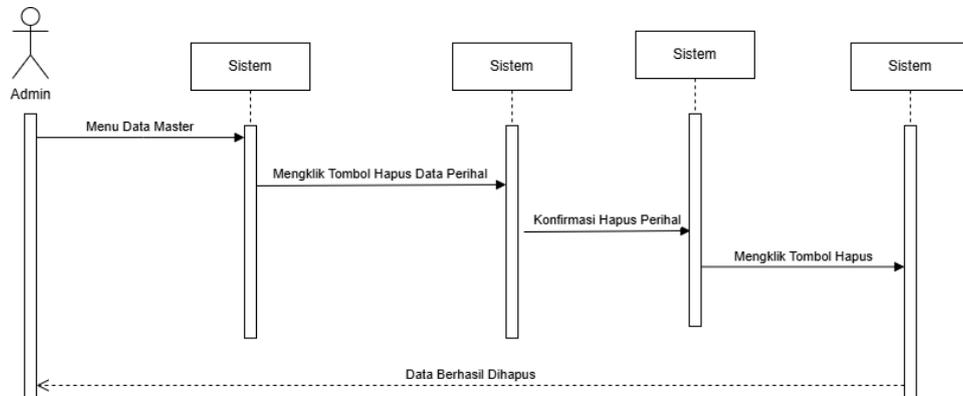
Gambar 4.39 *Sequence Diagram Tambah Data Perihal*

r. *Sequence Diagram Edit Data Perihal*



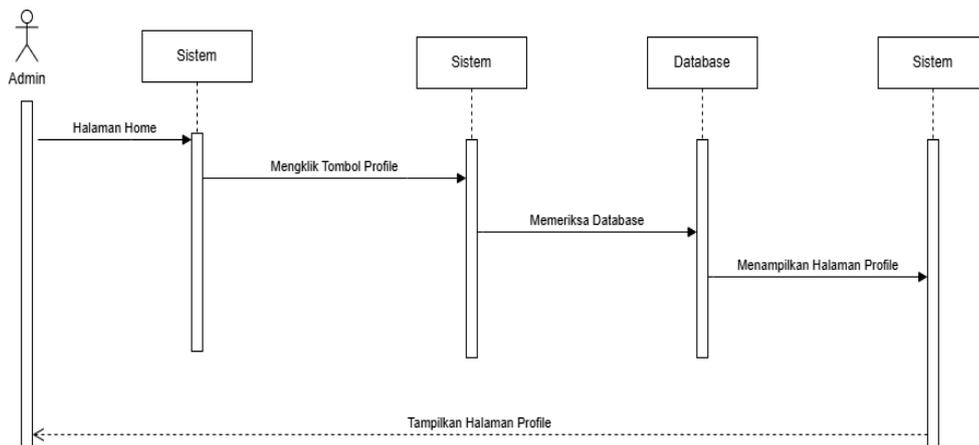
Gambar 4.40 *Sequence Diagram Edit Data Perihal*

s. *Sequence Diagram Hapus Data Perihal*



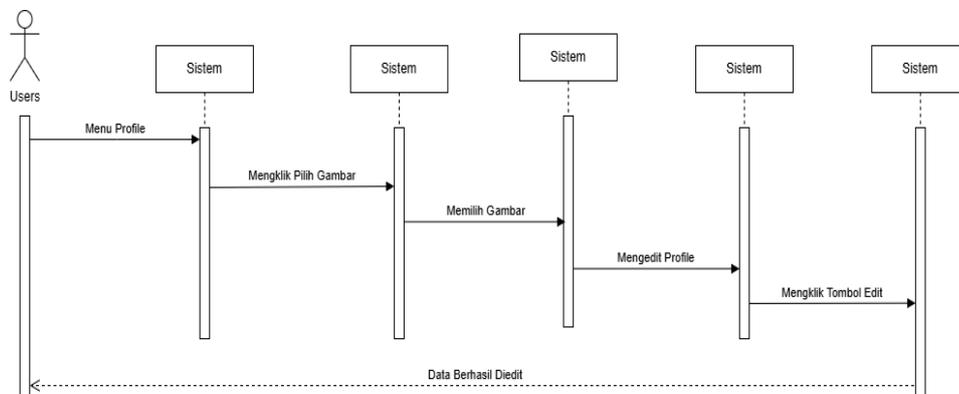
Gambar 4.41 *Sequence Diagram Hapus Data Perihal*

t. *Sequence Diagram Profile*



Gambar 4.42 *Sequence Diagram Profile*

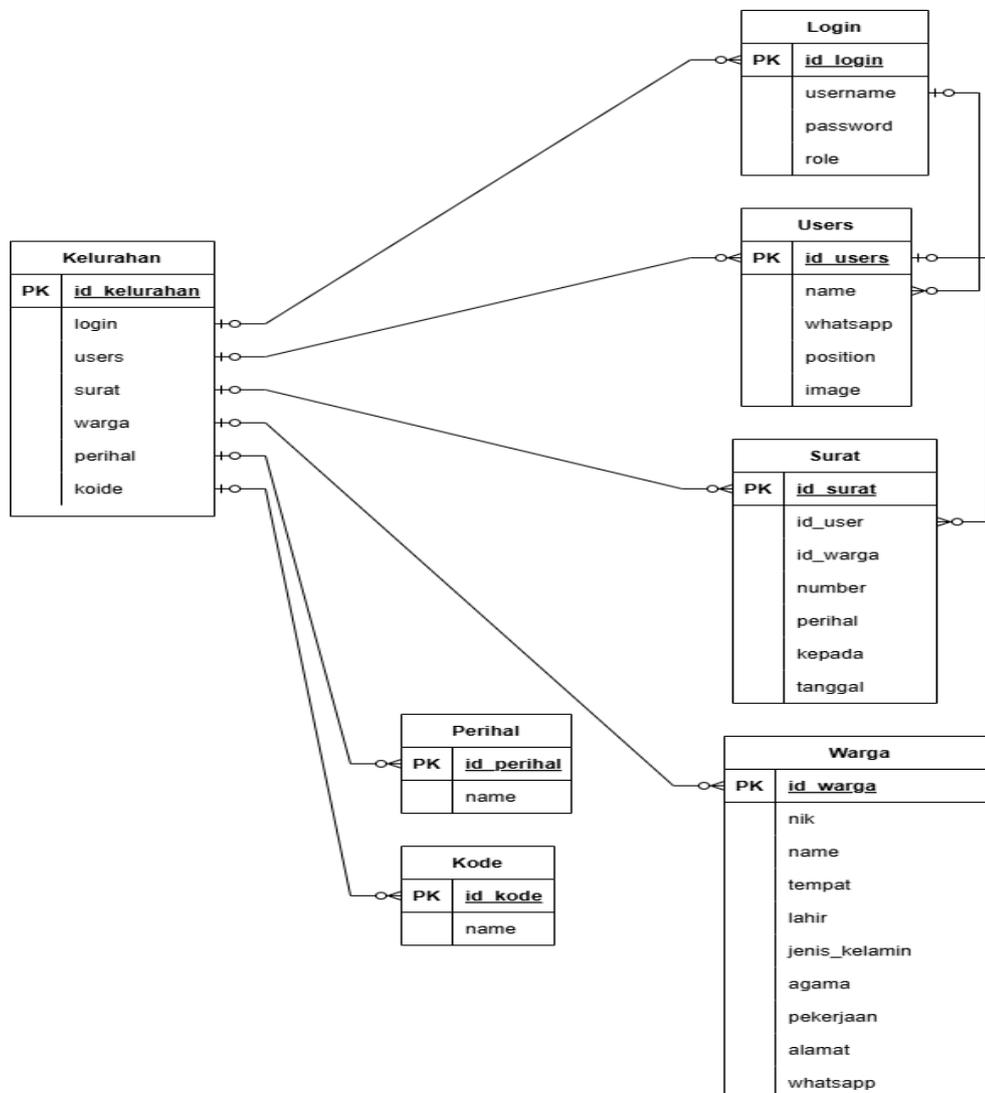
u. *Sequence Diagram Edit Profile*



Gambar 4.43 *Sequence Diagram Edit Profile*

4. Entity-Relationship Diagram

ERD (*Entity-Relationship Diagram*) merupakan diagram yang digunakan untuk memodelkan struktur logis sebuah basis data. ERD menggambarkan bagaimana entitas (objek atau tabel dalam basis data) saling berhubungan, beserta atribut-atributnya. Diagram ini bertujuan untuk memvisualisasikan hubungan antar-data sehingga mempermudah pemahaman, perancangan, dan implementasi sistem basis data.



Gambar 4.44 Entity-Relationship Diagram Kelurahan

ERD (Entity-Relationship Diagram) tersebut menggambarkan struktur data untuk sistem pengelolaan administrasi kelurahan. Entitas utama adalah Kelurahan, yang menghubungkan berbagai entitas lain, seperti Login, *Users*, Surat, Warga, Perihal, dan Kode. Entitas Login berfungsi menyimpan data autentikasi pengguna, termasuk *username*, password, dan peran (*role*) mereka dalam sistem. Entitas *Users* menyimpan informasi tentang pengguna sistem, seperti admin atau staf kelurahan, dengan atribut seperti nama, jabatan, dan kontak WhatsApp.

Entitas Warga menyimpan data penduduk yang mencakup informasi pribadi seperti NIK, nama, tempat lahir, agama, pekerjaan, dan alamat. Data ini dihubungkan dengan entitas **Surat**, yang menyimpan informasi surat-menyurat, seperti nomor surat, kategori surat (perihal), penerima, dan tanggal pembuatan. Kategori surat dikelola melalui entitas Perihal, sementara entitas Kode menyimpan data kode tertentu yang relevan untuk administrasi.

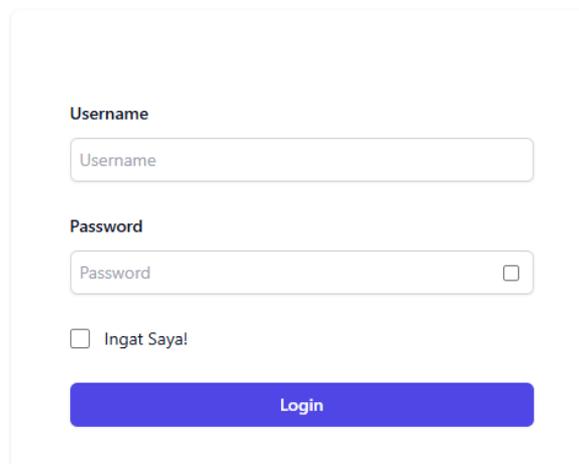
Hubungan antar-entitas ini memastikan integrasi data yang terstruktur, memungkinkan sistem untuk mengelola login pengguna, data warga, pembuatan surat, dan pelacakan informasi lainnya secara efisien. Diagram ini memberikan gambaran jelas tentang bagaimana data diorganisasi untuk mendukung proses administrasi kelurahan.

B. Detail Sistem

1. Login



Login Ke Akun Anda



Username

Password

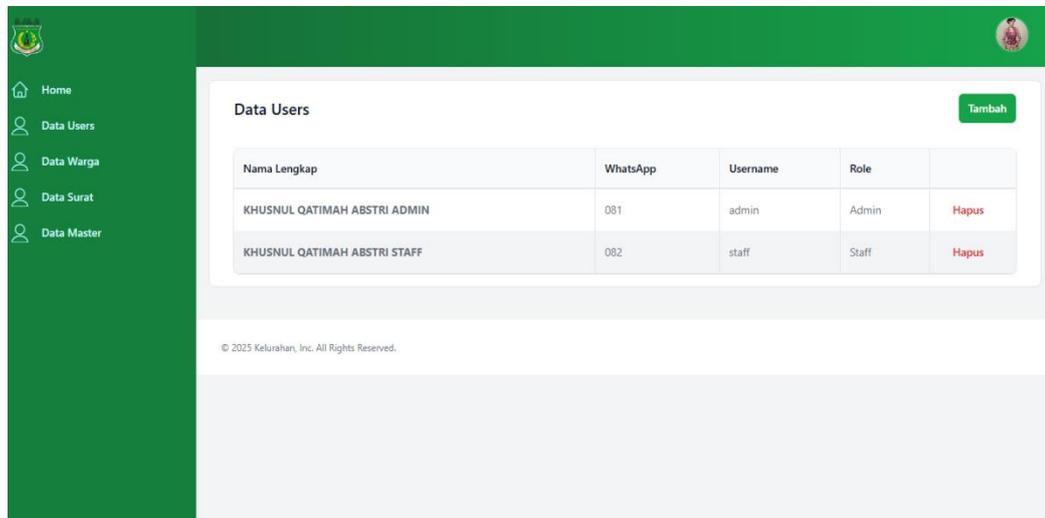
Ingat Saya!

Login

Gambar 4.45 Halaman Login

Pada gambar diatas merupakan Halaman login yang dirancang untuk memungkinkan pengguna mengakses situs web dengan memasukkan *username* serta *password*. Biasanya, halaman ini terdiri dari form input yang sederhana, dengan kolom untuk mengisi *username* dan *password*. Setelah data dimasukkan, pengguna dapat menekan tombol login untuk memvalidasi informasi tersebut. Jika data yang dimasukkan benar, pengguna akan diarahkan ke halaman home atau dashboard. Sebaliknya, jika terjadi kesalahan, sistem akan menampilkan pesan seperti "*Username* atau *password* salah" untuk memberi tahu pengguna.

2. Menu Data Users

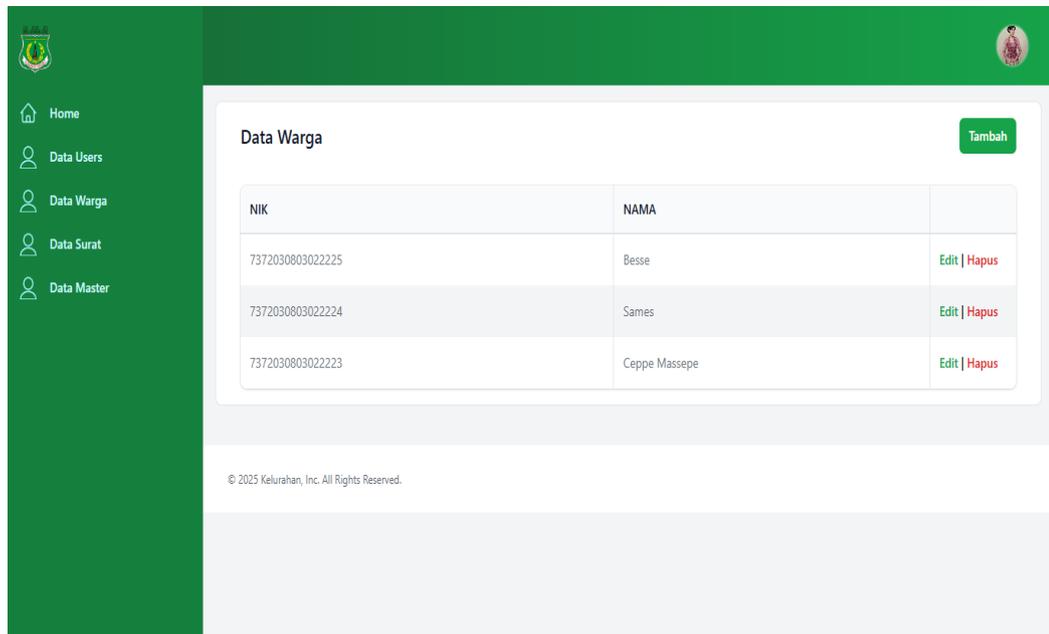


Nama Lengkap	WhatsApp	Username	Role	
KHUSNUL QATIMAH ABSTRI ADMIN	081	admin	Admin	Hapus
KHUSNUL QATIMAH ABSTRI STAFF	082	staff	Staff	Hapus

Gambar 4.46 Menu Data Users

Pada gambar diatas merupakan menu data *user* dalam sistem kelurahan berfungsi untuk mengelola data pengguna, seperti admin dan staff kelurahan, yang memiliki akses ke sistem. Halaman ini memungkinkan admin untuk menampilkan, menambah, mengedit, dan menghapus data pengguna. Data yang dikelola meliputi nama pengguna, kontak *WhatsApp*, dan posisi atau jabatan pengguna, yang penting untuk menentukan peran mereka dalam sistem. Informasi ini disajikan dalam bentuk tabel yang terstruktur, dilengkapi fitur pencarian untuk mempermudah navigasi. Selain itu, terdapat form untuk menambah atau mengedit data pengguna, dengan input seperti nama, nomor *WhatsApp*, dan jabatan. Halaman ini membantu memastikan pengelolaan data pengguna berjalan efisien dan aman, sekaligus mempermudah komunikasi antar pengguna melalui kontak yang disimpan. Halaman ini juga mendukung pengelolaan hak akses berdasarkan jabatan, sehingga hanya pengguna dengan otoritas tertentu yang dapat melakukan tindakan tertentu dalam sistem.

3. Menu Data Warga

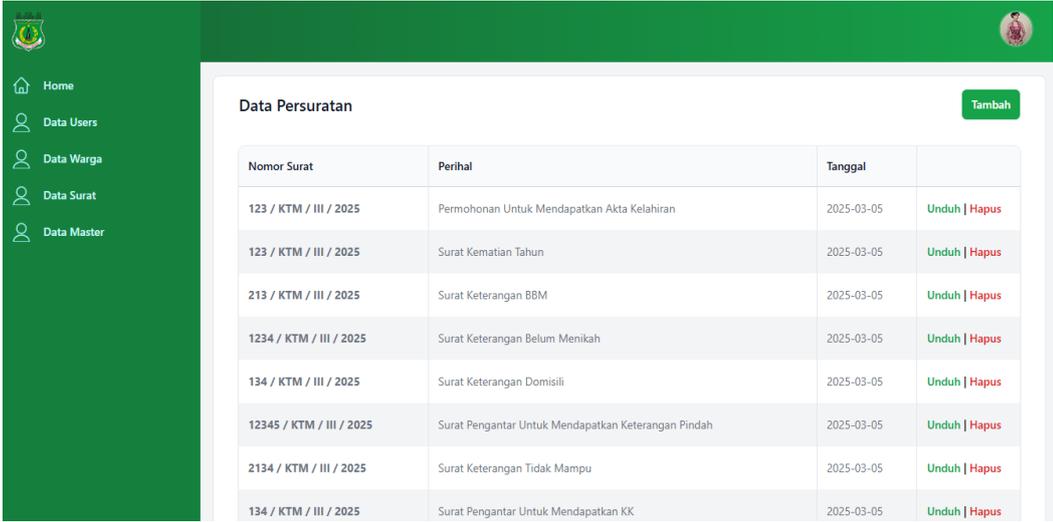


Gambar 4.47 Menu Data Warga

Pada gambar diatas merupakan data warga dalam sistem kelurahan berfungsi sebagai tempat untuk mengelola data penduduk yang tinggal di wilayah kelurahan. Halaman ini menyimpan informasi penting tentang warga, seperti identitas dasar (NIK, nama, tempat lahir, dan tanggal lahir) serta informasi tambahan (jenis kelamin, agama, pekerjaan, alamat, dan kontak *WhatsApp*). Semua data ini ditampilkan dalam bentuk tabel yang terstruktur, sehingga memudahkan admin atau staf kelurahan untuk melihat, mencari, dan mengelola data warga. Halaman ini juga dilengkapi fitur pencarian atau filter yang memungkinkan pengguna menemukan data spesifik berdasarkan atribut seperti nama atau NIK. Selain itu, terdapat form untuk menambah data warga baru atau memperbarui data yang sudah ada.

Fungsi utama halaman ini adalah mendukung berbagai kebutuhan administrasi, seperti pembuatan surat keterangan, pelaporan statistik, atau pemetaan data penduduk. Informasi kontak, seperti nomor *WhatsApp*, juga mempermudah komunikasi antara pihak kelurahan dan warga terkait layanan tertentu. Dengan adanya halaman ini, pengelolaan data warga menjadi lebih terorganisasi, efisien, dan mendukung pelayanan publik yang lebih baik di tingkat kelurahan.

4. Menu Data Surat



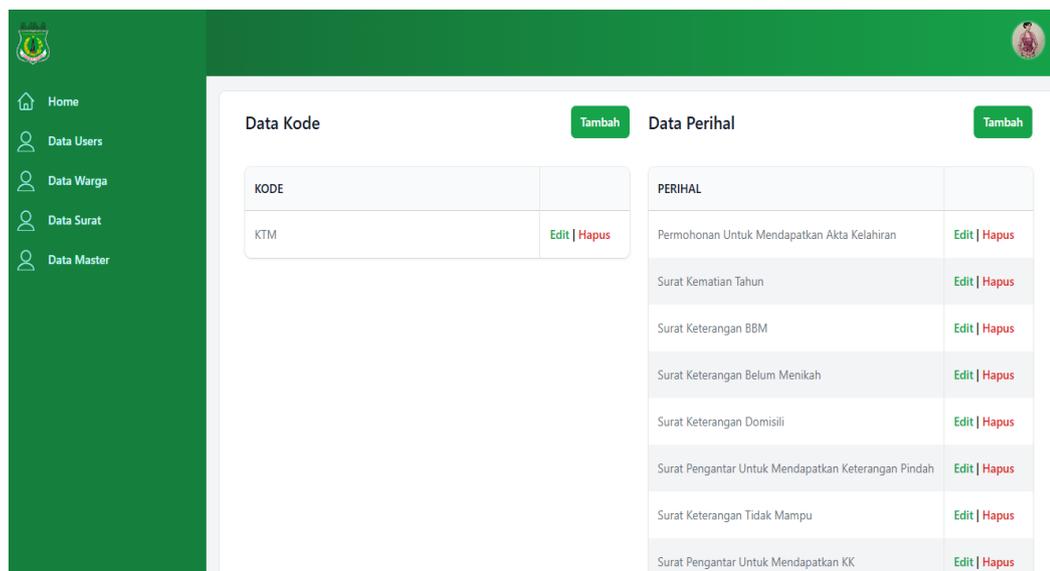
Nomor Surat	Perihal	Tanggal	Tombol Aksi
123 / KTM / III / 2025	Permohonan Untuk Mendapatkan Akta Kelahiran	2025-03-05	Unduh Hapus
123 / KTM / III / 2025	Surat Kematian Tahun	2025-03-05	Unduh Hapus
213 / KTM / III / 2025	Surat Keterangan BBM	2025-03-05	Unduh Hapus
1234 / KTM / III / 2025	Surat Keterangan Belum Menikah	2025-03-05	Unduh Hapus
134 / KTM / III / 2025	Surat Keterangan Domisili	2025-03-05	Unduh Hapus
12345 / KTM / III / 2025	Surat Pengantar Untuk Mendapatkan Keterangan Pindah	2025-03-05	Unduh Hapus
2134 / KTM / III / 2025	Surat Keterangan Tidak Mampu	2025-03-05	Unduh Hapus
134 / KTM / III / 2025	Surat Pengantar Untuk Mendapatkan KK	2025-03-05	Unduh Hapus

Gambar 4.48 Menu Data Surat

Pada gambar diatas merupakan menu surat dalam sistem kelurahan merupakan antarmuka yang dirancang untuk memudahkan admin atau staf kelurahan dalam mengelola surat-surat resmi. Halaman ini biasanya memiliki berbagai fungsi, seperti pembuatan surat baru, dan pengelolaan arsip surat. Pada bagian utama halaman, terdapat tombol aksi, Tambah Surat yang membuka form untuk memasukkan informasi surat baru, termasuk nomor surat, perihal, penerima,

dan tanggal pembuatan, serta memiliki tombol unduh untuk mengunduh surat dan memiliki tombol untuk menghapus surat

5. Menu Data Master



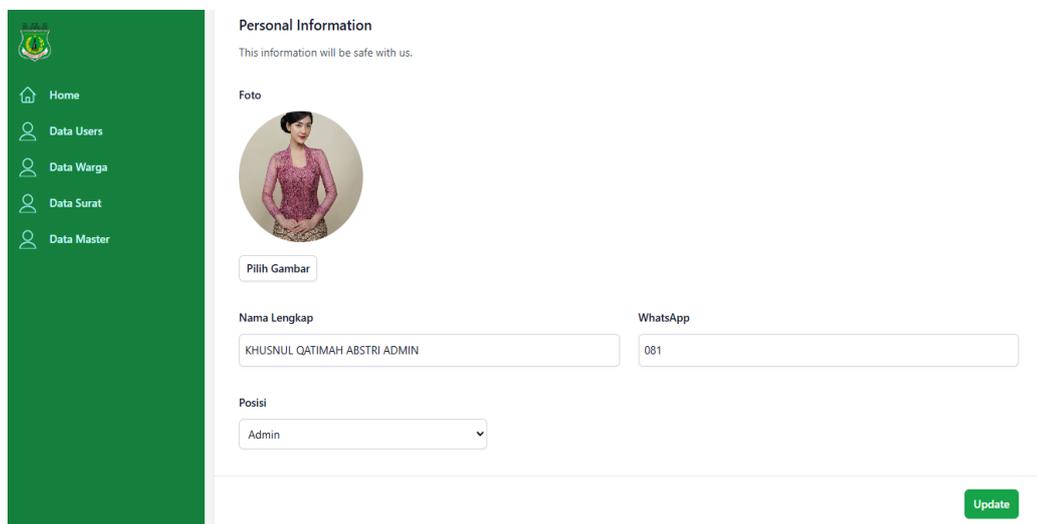
Gambar 4.49 Menu Data Master

Pada gambar diatas merupakan menu data master, terdapat dua bagian utama yang berperan penting dalam pengelolaan administrasi surat, yaitu Kode dan Perihal. Kedua bagian ini bekerja sama untuk memastikan pengelolaan surat dan data kelurahan dapat berjalan dengan lebih efisien dan terstruktur.

Bagian Kode pada halaman Master digunakan untuk menyimpan dan mengelola kode-kode yang berkaitan dengan identifikasi surat atau wilayah kelurahan. Pengguna dapat menambah, memperbarui, atau menghapus kode yang relevan, yang kemudian akan memudahkan dalam pengelompokan surat atau administrasi wilayah berdasarkan kode yang telah ditetapkan.

Sementara itu, bagian Perihal pada halaman Master memungkinkan pengguna untuk mengatur dan mengelola kategori surat atau jenis perihal yang ada. Perihal surat dapat mencakup berbagai kategori, seperti surat masuk, surat keluar, atau jenis surat lainnya yang disesuaikan dengan kebutuhan administrasi. Pengelompokan surat berdasarkan perihal ini membantu pengguna dalam pencarian dan pengelolaan surat yang lebih sistematis.

6. Profile



Personal Information
This information will be safe with us.

Foto

Pilih Gambar

Nama Lengkap: KHUSNUL QATIMAH ABSTRI ADMIN

WhatsApp: 081

Posisi: Admin

Update

Gambar 4.50 Profile

Pada gambar diatas merupakan profile memberikan pengguna kemudahan untuk mengelola informasi pribadi mereka, termasuk mengubah gambar, nama lengkap, posisi dan kontak whatsapp akun. Pada bagian ini, pengguna dapat memperbarui nama lengkap, posisi, kontak *whatsapp* mereka, yang merupakan identitas utama yang digunakan untuk sistem. Selain itu, halaman Profile juga menyediakan opsi untuk mengganti gambar.

C. Pengujian Sistem

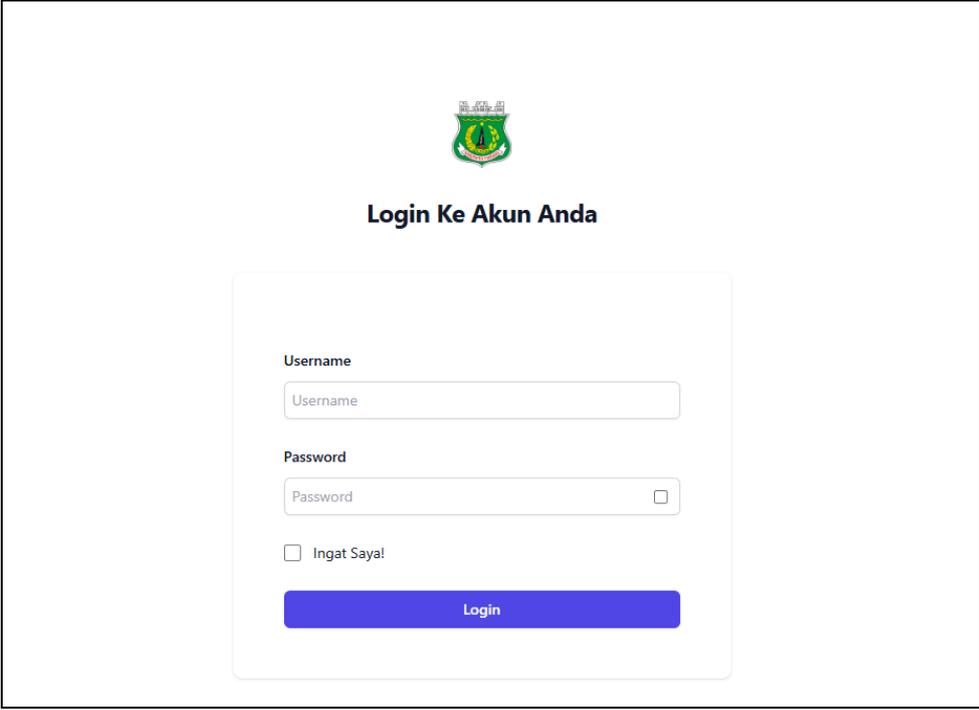
Dalam penelitian ini, tiga pendekatan pengujian sistem digunakan, yaitu pengujian black box, pengujian white box dan pengujian akurasi. Ini adalah hasil pengujian yang dilakukan dengan pendekatan ini.

1. *Black Box Testing*

Black box testing adalah sebuah metode pengujian perangkat lunak di mana *tester* menguji fungsionalitas sistem tanpa mengetahui detail internal dari kode atau struktur implementasinya.

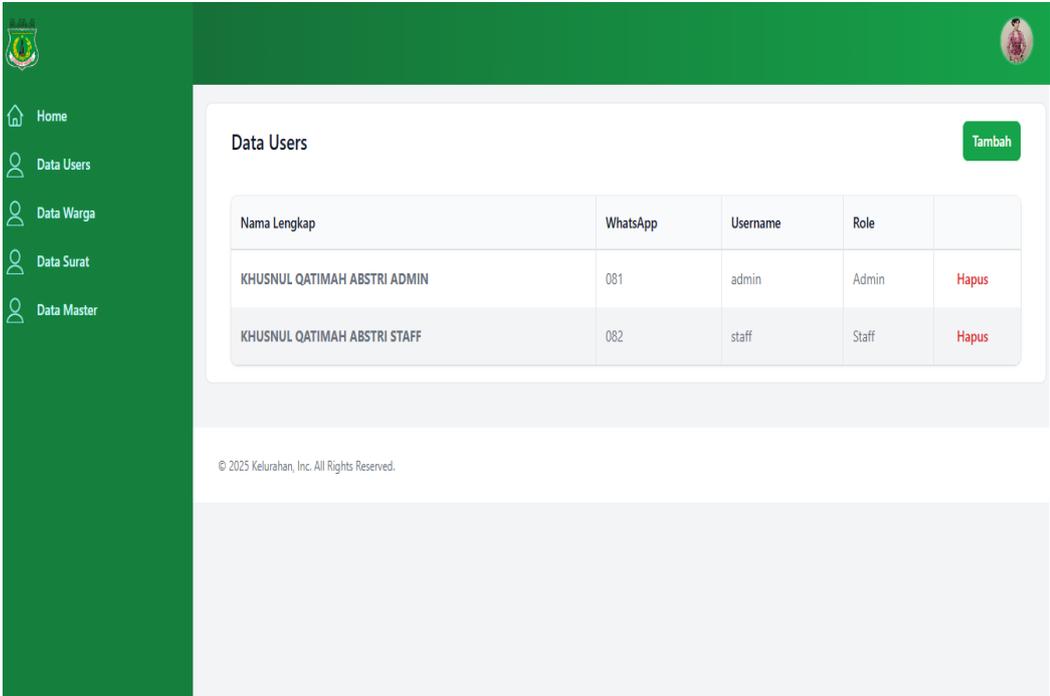
a. Pengujian *Black Box* Login

Tabel 4.2 Pengujian *Black Box* Login

Tes Faktor	Hasil	Keterangan
<i>User</i> login dengan memasukkan <i>username</i> dan password, setelah itu menekan tombol login	✓	Berhasil, login dan menampilkan halaman home
Screenshot		
		

b. Pengujian *Black Box* Menu Data Users

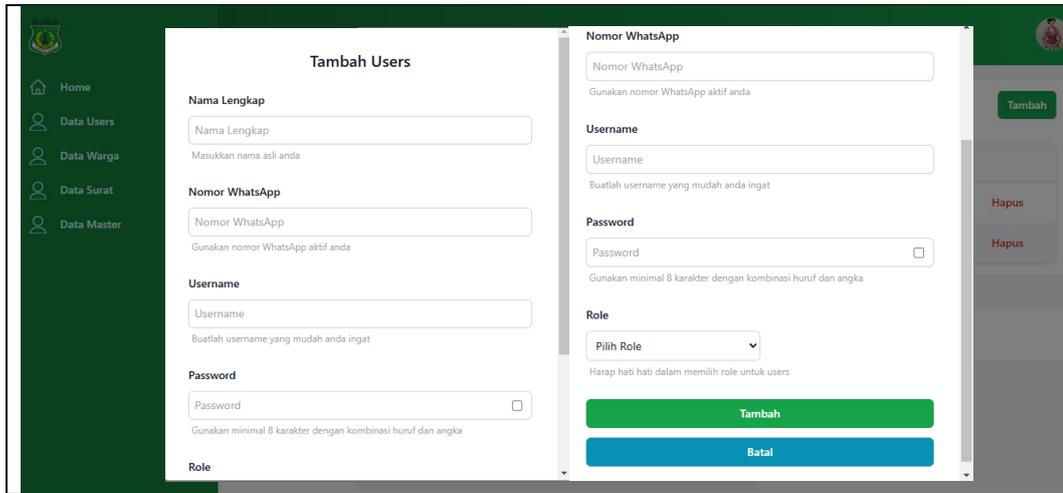
Tabel 4.3 Pengujian *Black Box* Menu Data Users

Tes Faktor	Hasil	Keterangan
User menekan tombol menu data users	✓	Berhasil, menampilkan menu data users
Screenshot		
		

c. Pengujian *Black Box* Tambah Data Users

Tabel 4.4 Pengujian *Black Box* Tambah Data Users

Tes Faktor	Hasil	Keterangan
Admin menekan tombol tambah	✓	Berhasil, menambahkan data users
Screenshot		
		

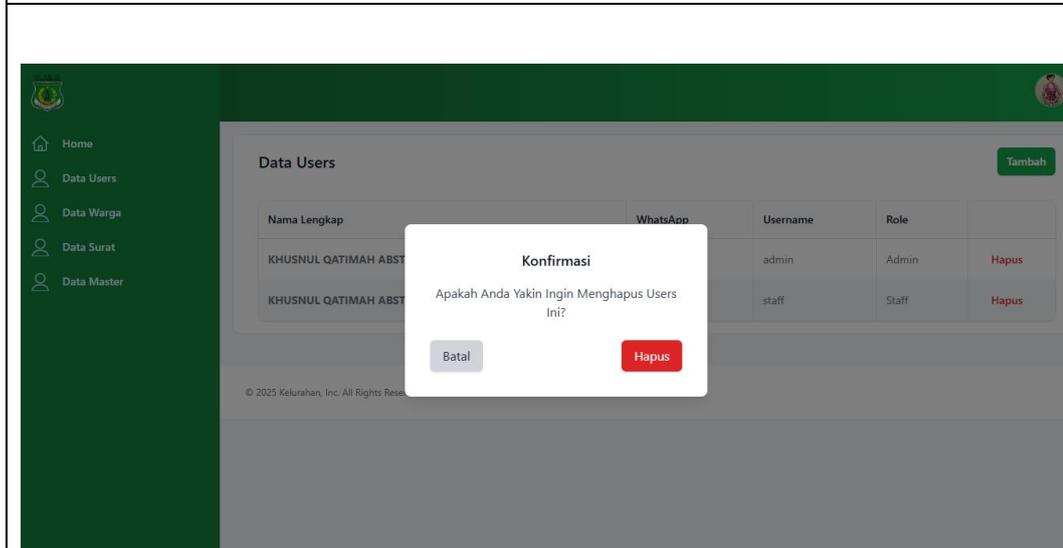


d. Pengujian *Black Box* Hapus Data Users

Tabel 4.5 Pengujian *Black Box* Hapus Data Users

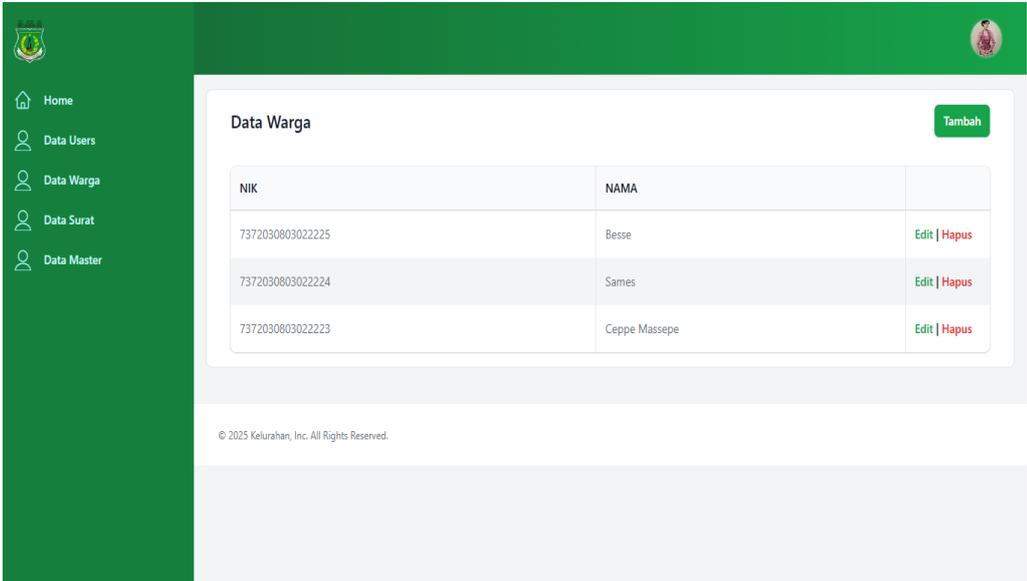
Tes Faktor	Hasil	Keterangan
Admin menekan tombol hapus	✓	Berhasil, menghapus data <i>users</i>

Screenshot



e. Pengujian *Black Box* Menu Data Warga

Tabel 4.6 Pengujian *Black Box* Menu Data Warga

Tes Faktor	Hasil	Keterangan
Staff menekan tombol menu data warga	✓	Berhasil, menampilkan menu data warga
Screenshot		
		

f. Pengujian *Black Box* Tambah Data Warga

Tabel 4.7 Pengujian *Black Box* Tambah Data Warga

Tes Faktor	Hasil	Keterangan
Staff menekan tombol tambah	✓	Berhasil, menambahkan data warga
Screenshot		

Tambah Warga

NIK:

Nama:

Tempat: Tanggal Lahir:

Status:

Jenis Kelamin:

Agama:

Pekerjaan:

Alamat:

WhatsApp:

g. Pengujian Black Box Edit Data Warga

Tabel 4.8 Pengujian *Black Box* Edit Data Warga

Tes Faktor	Hasil	Keterangan
Staff menekan tombol edit	✓	Berhasil, mengedit data warga

Screenshot

Edit Warga

NIK:

Nama:

Tempat: Tanggal Lahir:

Status:

Jenis Kelamin:

Agama:

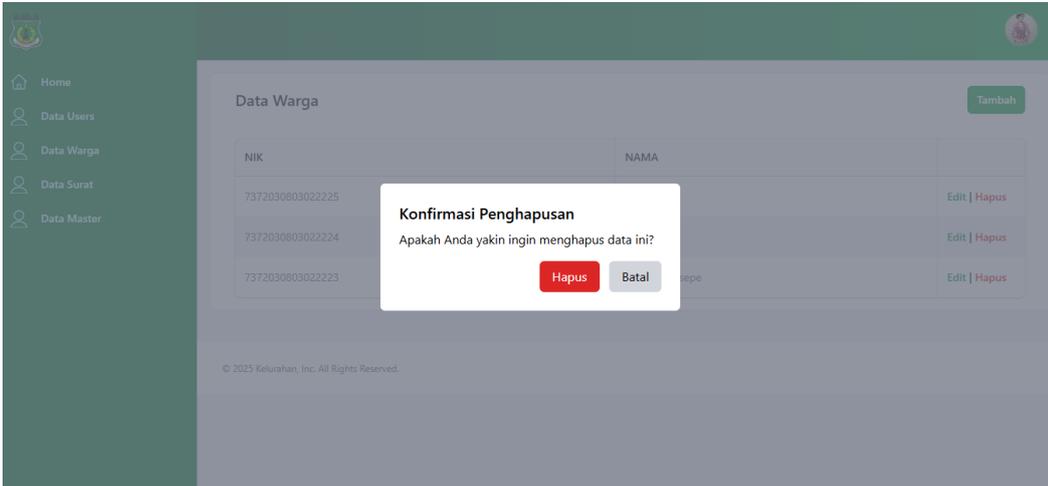
Pekerjaan:

Alamat:

WhatsApp:

h. Pengujian Black Box Hapus Data Warga

Tabel 4.9 Pengujian *Black Box* Hapus Data Warga

Tes Faktor	Hasil	Keterangan
Staff menekan tombol hapus	✓	Berhasil, menghapus data warga
Screenshot		
		

i. Pengujian *Black Box* Menu Data Surat

Tabel 4.10 Pengujian *Black Box* Menu Data Surat

Tes Faktor	Hasil	Keterangan
Staff menekan tombol menu data surat	✓	Berhasil, menampilkan menu data surat
Screenshot		
		

Nomor Surat	Perihal	Tanggal	
123 / KTM / III / 2025	Permohonan Untuk Mendapatkan Akta Kelahiran	2025-03-05	Unduh Hapus
123 / KTM / III / 2025	Surat Kematian Tahun	2025-03-05	Unduh Hapus
213 / KTM / III / 2025	Surat Keterangan BBM	2025-03-05	Unduh Hapus
1234 / KTM / III / 2025	Surat Keterangan Belum Menikah	2025-03-05	Unduh Hapus
134 / KTM / III / 2025	Surat Keterangan Domisili	2025-03-05	Unduh Hapus
12345 / KTM / III / 2025	Surat Pengantar Untuk Mendapatkan Keterangan Pindah	2025-03-05	Unduh Hapus
2134 / KTM / III / 2025	Surat Keterangan Tidak Mampu	2025-03-05	Unduh Hapus
134 / KTM / III / 2025	Surat Pengantar Untuk Mendapatkan KK	2025-03-05	Unduh Hapus

j. Pengujian *Black Box* Tambah Data Surat

Tabel 4.11 Pengujian *Black Box* Tambah Data Surat

Tes Faktor	Hasil	Keterangan
Staff menekan tombol tambah	✓	Berhasil, menambahkan data surat

Screenshot

Tambah Surat

Perihal
Pilih Perihal

Nama Warga
Pilih Nama Warga

Tanggal
dd/mm/yyyy

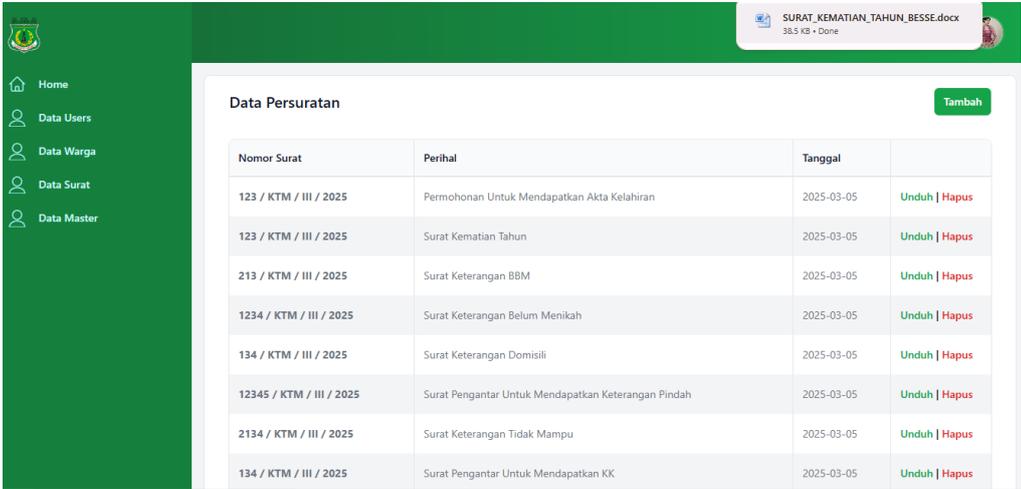
Nomor Kode
Nomor Pilih Kode

Nomor Surat
Nomor Surat

Batal Tambah

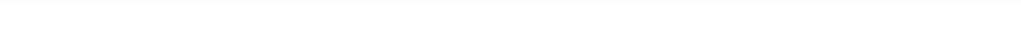
k. Pengujian *Black Box* Unduh Data Surat

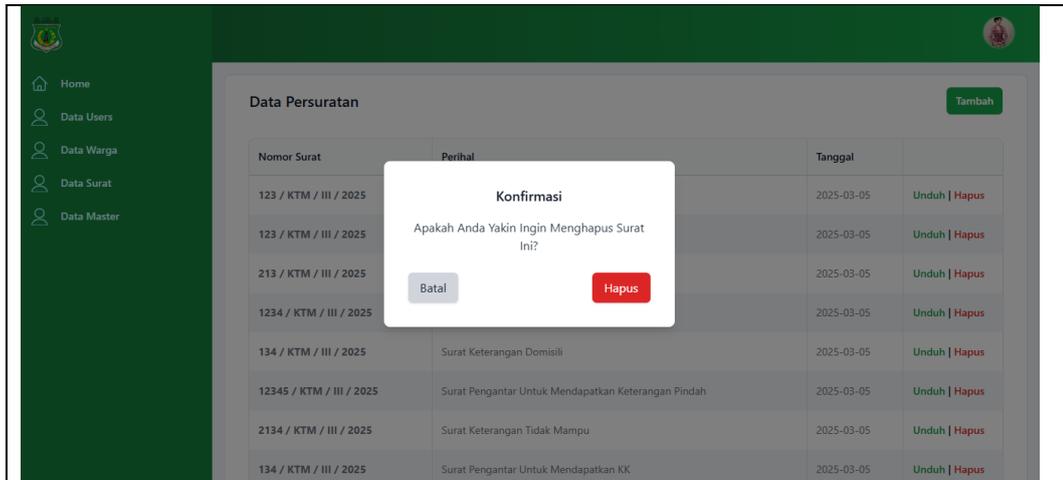
Tabel 4.12 Pengujian *Black Box* Unduh Data Surat

Tes Faktor	Hasil	Keterangan
Staff menekan tombol unduh	✓	Berhasil, mengunduh data surat
Screenshot		
		

l. Pengujian *Black Box* Hapus Surat

Tabel 4.13 Pengujian *Black Box* Hapus Data Surat

Tes Faktor	Hasil	Keterangan
Staff menekan tombol hapus	✓	Berhasil, menghapus data surat
Screenshot		
		

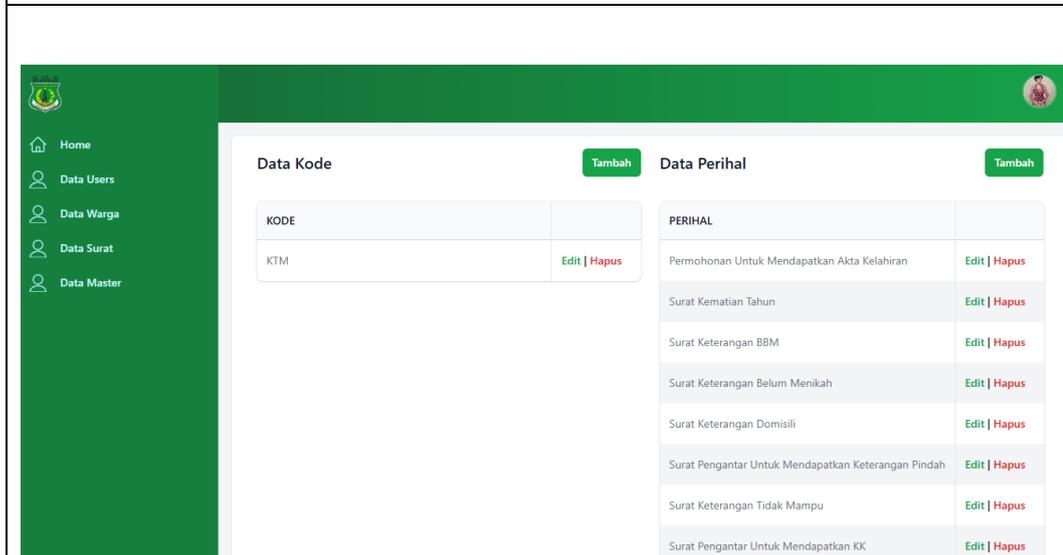


m. Pengujian *Black Box* Menu Data Master

Tabel 4.14 Pengujian *Black Box* Menu Data Master

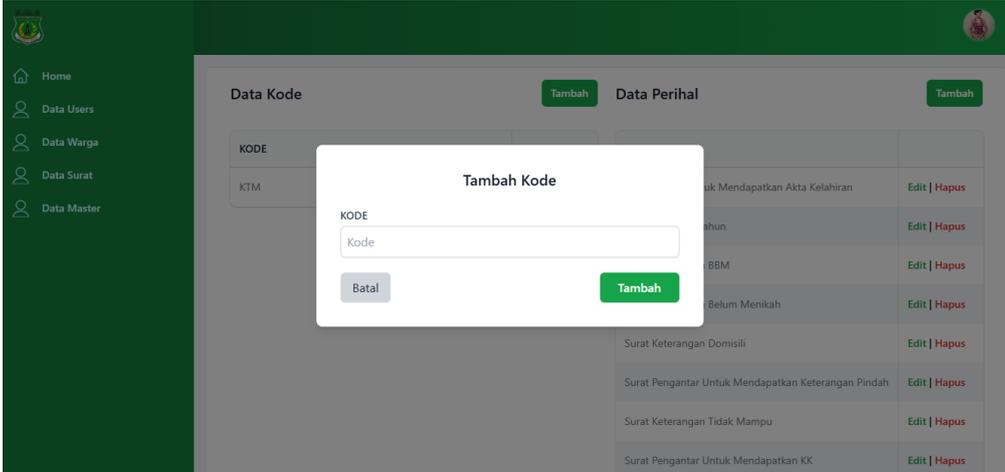
Tes Faktor	Hasil	Keterangan
Admin menekan tombol menu data master	✓	Berhasil, menampilkan menu data master

Screenshot



n. Pengujian *Black Box* Tambah Data Kode

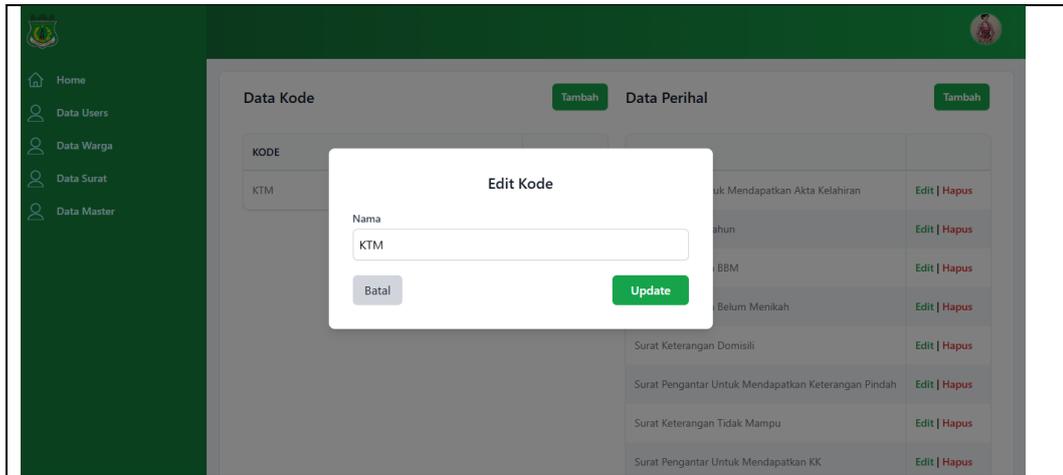
Tabel 4.15 Pengujian *Black Box* Tambah Data Kode

Tes Faktor	Hasil	Keterangan
Admin menekan tombol tambah	✓	Berhasil, menambah data kode
Screenshot		
		

o. Pengujian *Black Box* Edit Data Kode

Tabel 4.16 Pengujian *Black Box* Edit Data Kode

Tes Faktor	Hasil	Keterangan
Admin menekan tombol edit	✓	Berhasil, mengedit data kode
Screenshot		
		

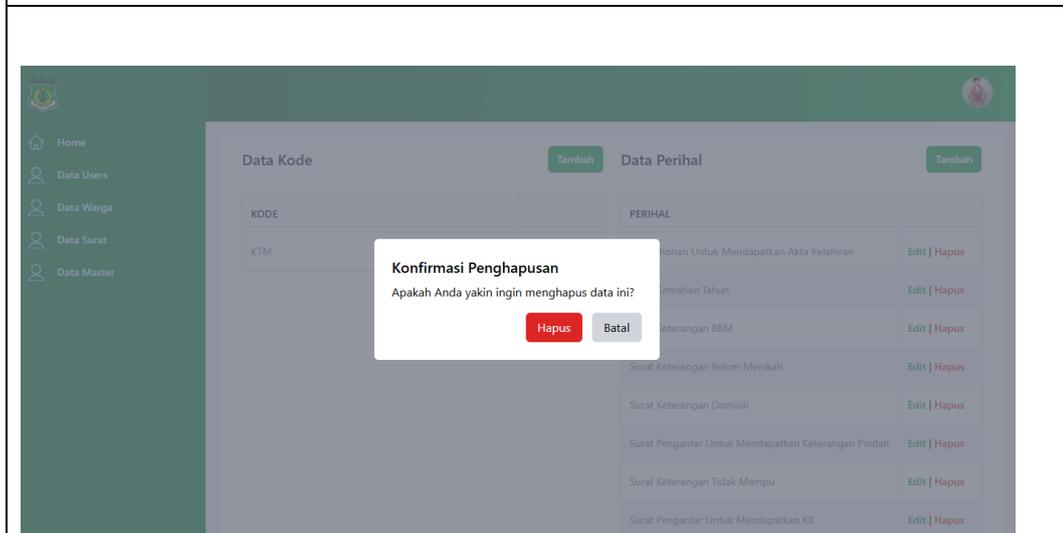


p. Pengujian Black Box Hapus Data Kode

Tabel 4.17 Pengujian *Black Box* Hapus Data Kode

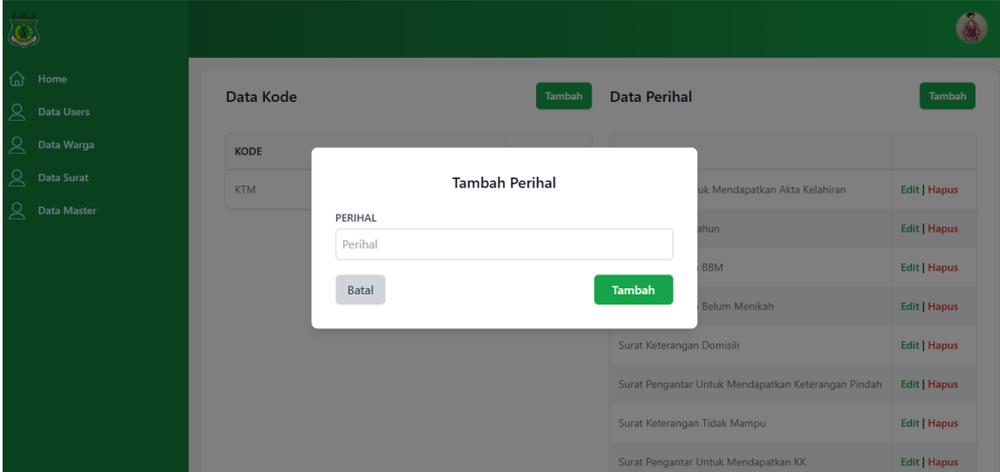
Tes Faktor	Hasil	Keterangan
Admin menekan tombol hapus	✓	Berhasil, menghapus data kode

Screenshot



q. Pengujian *Black Box* Tambah Data Perihal

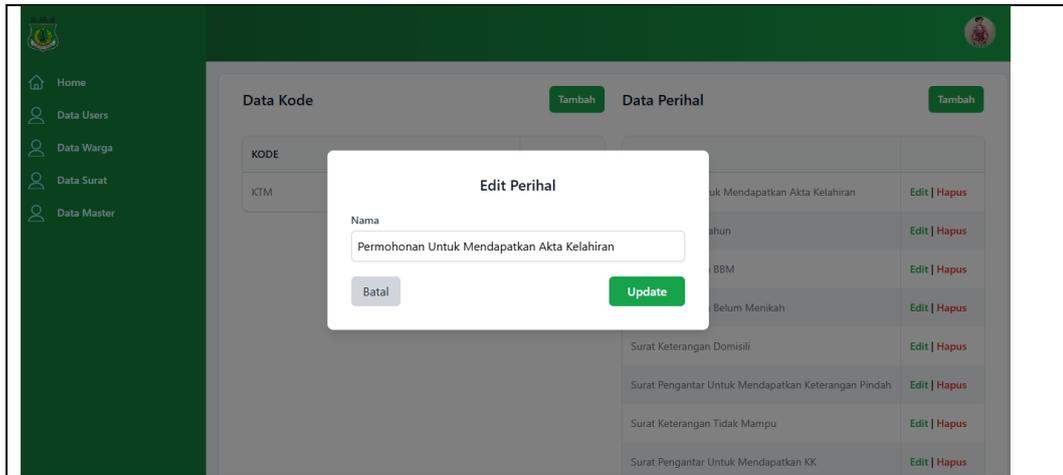
Tabel 4.18 Pengujian *Black Box* Tambah Data Perihal

Tes Faktor	Hasil	Keterangan
Admin menekan tombol tambah	✓	Berhasil, menambah data perihal
Screenshot		
		

r. Pengujian *Black Box* Edit Data Perihal

Tabel 4.19 Pengujian *Black Box* Edit Data Perihal

Tes Faktor	Hasil	Keterangan
Admin menekan tombol edit	✓	Berhasil, mengedit data perihal
Screenshot		

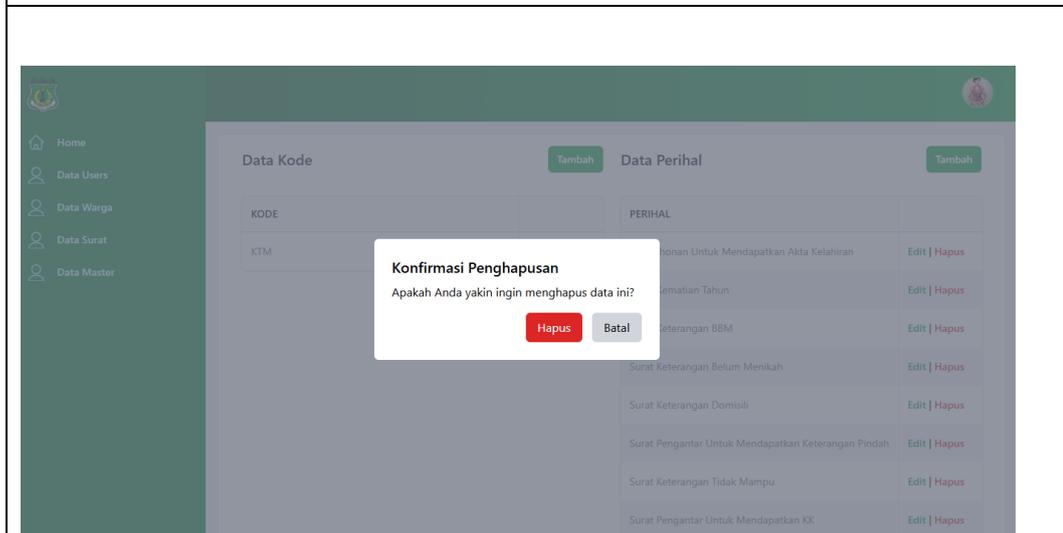


s. Pengujian *Black Box* Hapus Data Perihal

Tabel 4.20 Pengujian *Black Box* Hapus Data Perihal

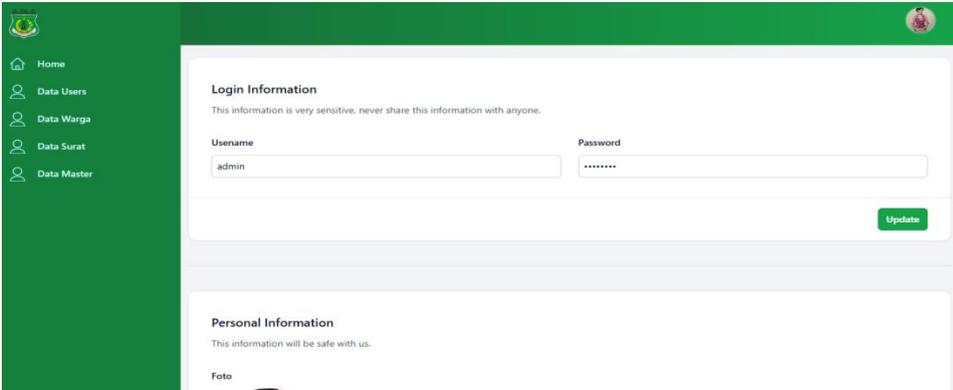
Tes Faktor	Hasil	Keterangan
Admin menekan tombol hapus	✓	Berhasil, menghapus data perihal

Screenshot



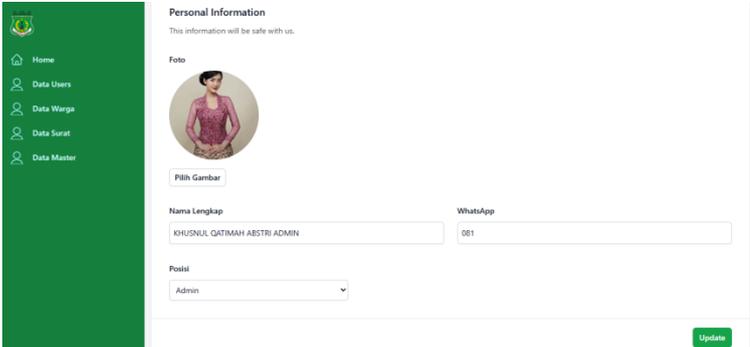
t. Pengujian *Black Box Profile*

Tabel 4.21 Pengujian *Black Box Profile*

Tes Faktor	Hasil	Keterangan
Users menekan tombol profile	✓	Berhasil, menampilkan profile
Screenshot		
 <p>The screenshot shows a web application interface with a green sidebar on the left containing navigation items: Home, Data Users, Data Warga, Data Surat, and Data Master. The main content area is titled 'Login Information' and includes a warning: 'This information is very sensitive, never share this information with anyone.' Below this are input fields for 'Username' (containing 'admin') and 'Password' (masked with dots), and an 'Update' button. A second section titled 'Personal Information' contains the warning 'This information will be safe with us.' and a 'Foto' label above a circular profile picture placeholder.</p>		

u. Pengujian *Black Box Edit Profile*

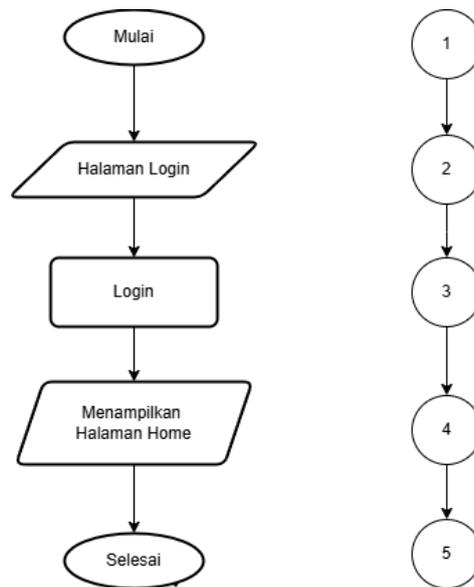
Tabel 4.22 Pengujian *Black Box Edit Profile*

Tes Faktor	Hasil	Keterangan
Users menekan tombol edit	✓	Berhasil, mengedit profile
Screenshot		
 <p>The screenshot shows the 'Personal Information' edit page. It features the same green sidebar as the previous page. The main content area has the heading 'Personal Information' and the warning 'This information will be safe with us.' Below this is a 'Foto' section with a circular profile picture of a woman and a 'Pilih Gambar' button. Further down are input fields for 'Nama Lengkap' (containing 'KHUSNUL QATIMAH ABSTRI ADMIN') and 'WhatsApp' (containing '081'). A 'Posisi' dropdown menu is set to 'Admin'. An 'Update' button is located at the bottom right of the form.</p>		

2. White Box Testing

Dalam pengujian ini akan menampilkan flowchart dan flowgraph dari sistem yang telah dibuat. Berikut ini adalah hasil dari pengujian menggunakan white box testing secara keseluruhan.

a. Flowchart dan Flowgraph Login



Gambar 4.51 Flowchart dan Flowgraph Login

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 4$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

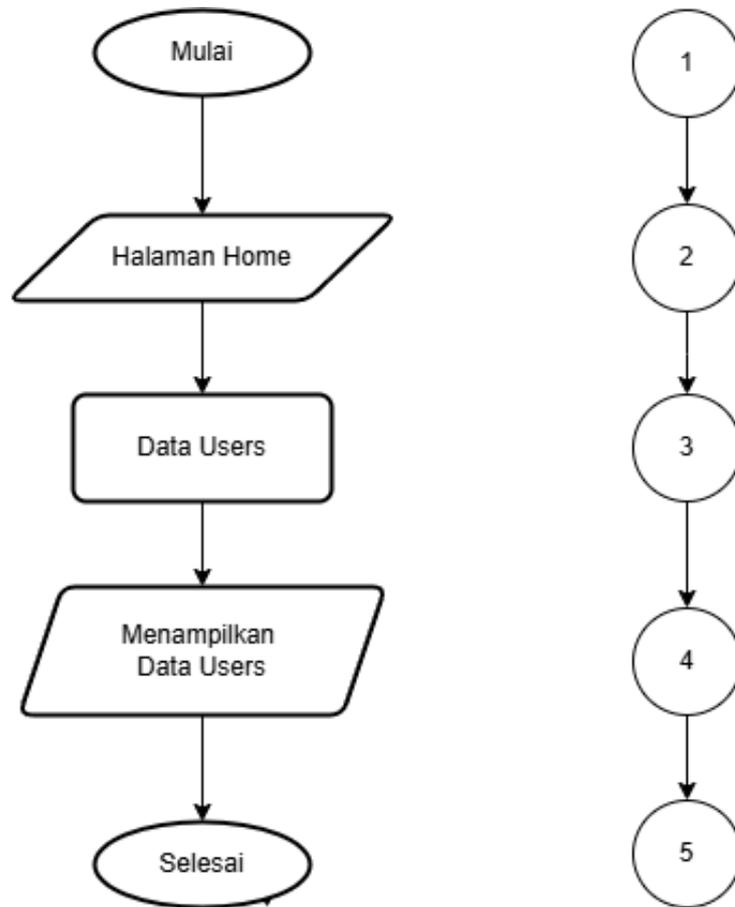
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.23 Grafik Matris Login

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

b. *Flowchart dan Flowgraph Menu Data Users*



Gambar 4.52 *Flowchart dan Flowgraph Menu Data Users*

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

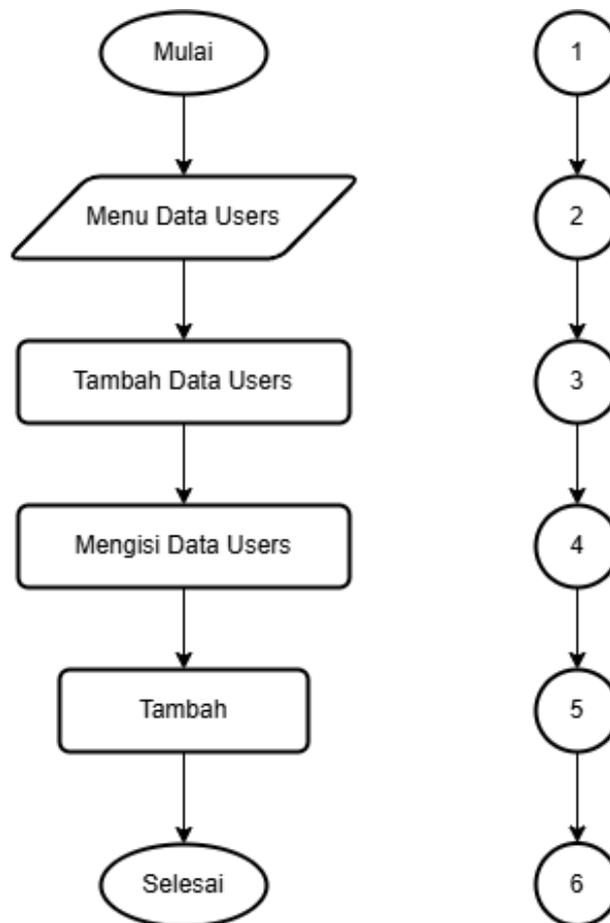
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.24 Grafik Matris Menu Data *Users*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

c. *Flowchart dan Flowgraph* Tambah Data Users



Gambar 4.53 *Flowchart dan Flowgraph* Tambah Data Users

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

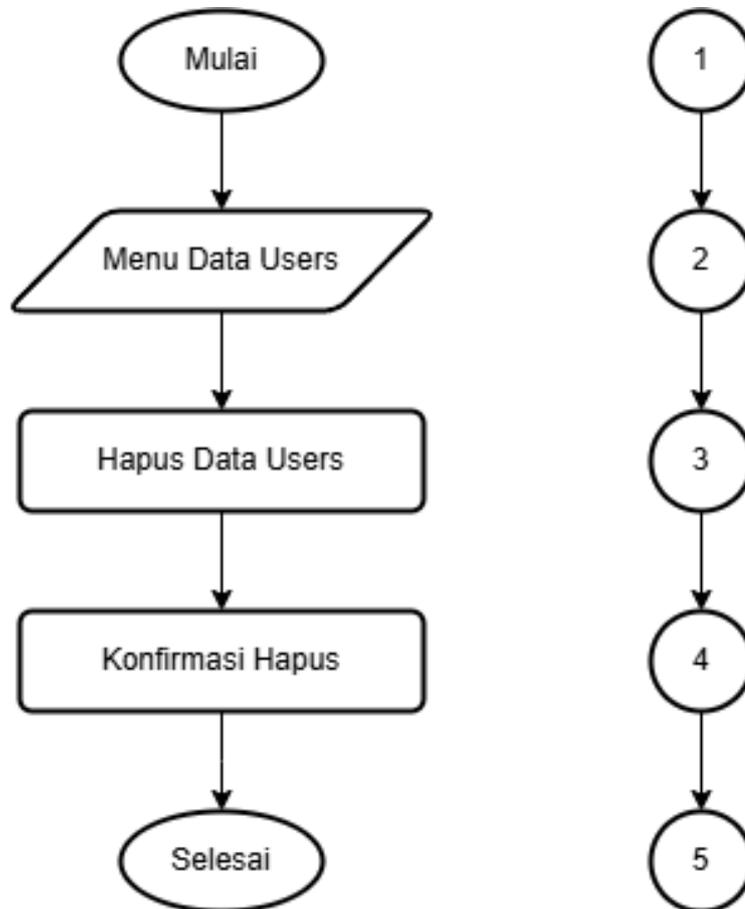
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.25 Grafik Matris Tambah Data *Users*

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

d. *Flowchart dan Flowgraph Hapus Data Users*



Gambar 4.54 *Flowchart dan Flowgraph Hapus Data Users*

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

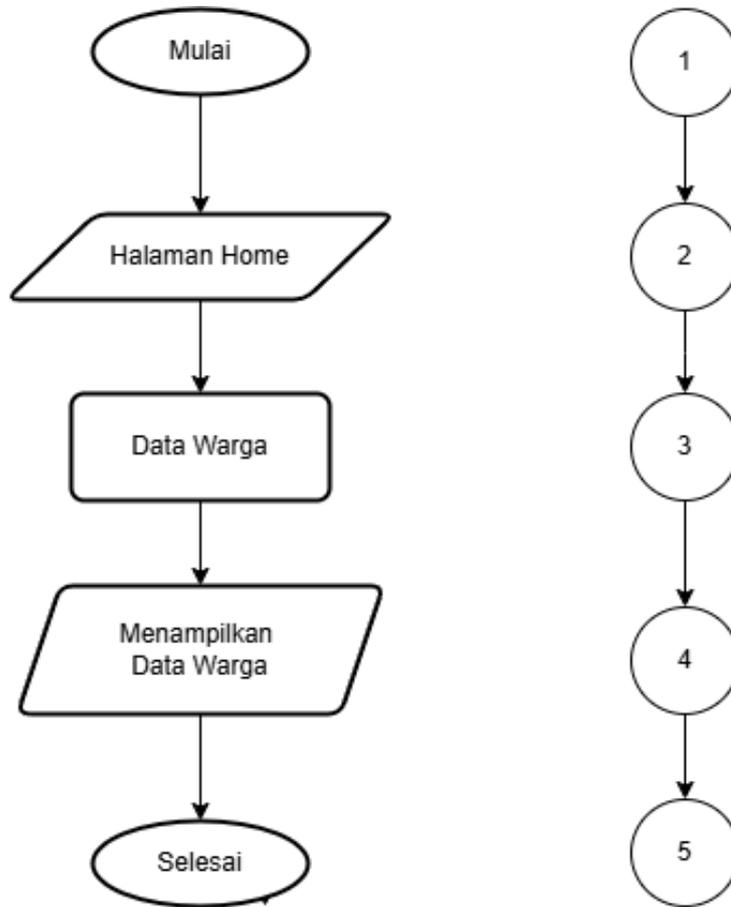
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.26 Grafik Matris Hapus Data *Users*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

e. *Flowchart dan Flowgraph* Menu Data Warga



Gambar 4.55 *Flowchart dan Flowgraph* Menu Data Warga

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

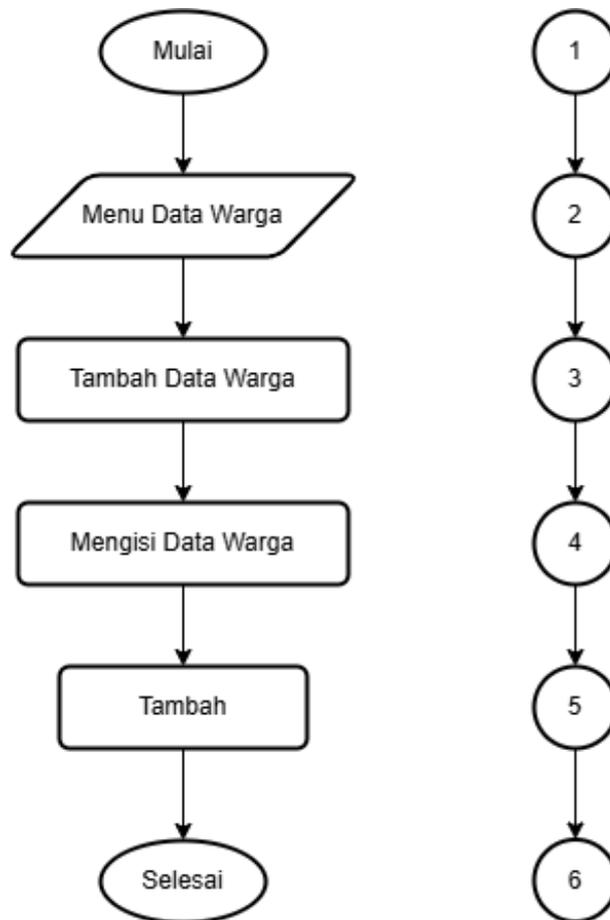
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.27 Grafik Matriks Menu Data Warga

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

f. *Flowchart dan Flowgraph* Tambah Data Warga



Gambar 4.56 *Flowchart dan Flowgraph* Tambah Data Warga

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

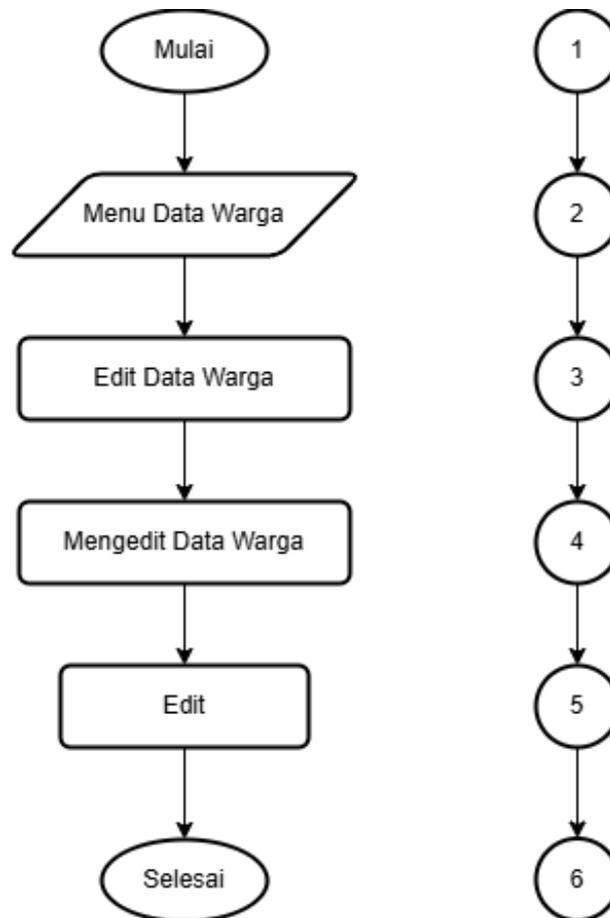
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.28 Grafik Matris Tambah Data Warga

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

g. *Flowchart dan Flowgraph* Edit Data Warga



Gambar 4.57 *Flowchart dan Flowgraph* Edit Data Warga

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

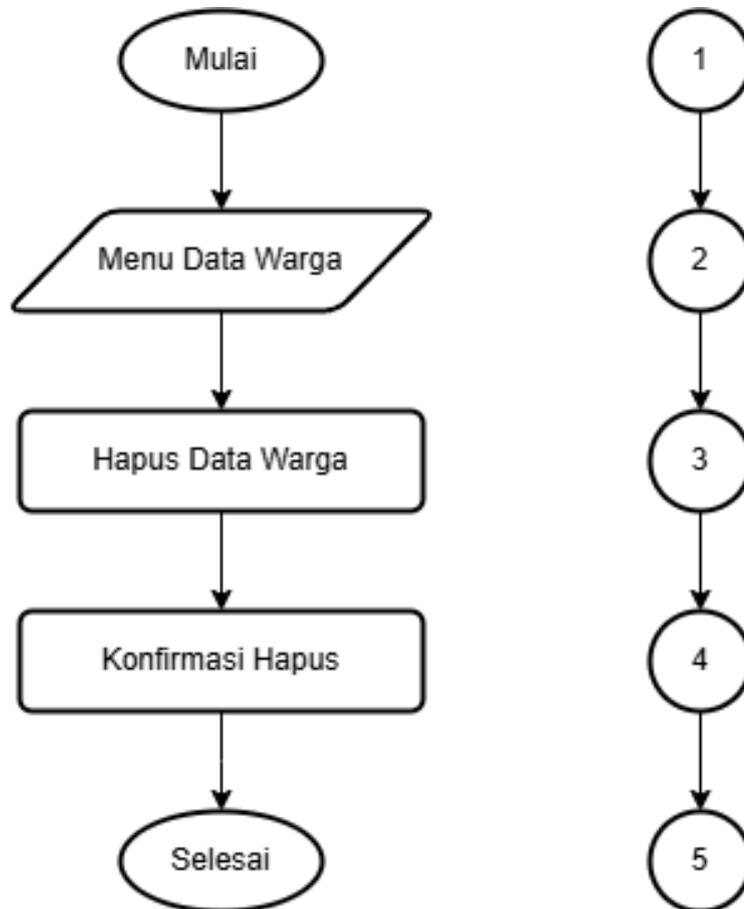
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.29 Grafik Matris Edit Data Warga

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

h. *Flowchart dan Flowgraph* Hapus Data Warga



Gambar 4.58 *Flowchart dan Flowgraph* Hapus Data Warga

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\textit{edge}) = 4$$

$$N (\textit{node}) = 5$$

$$P (\textit{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

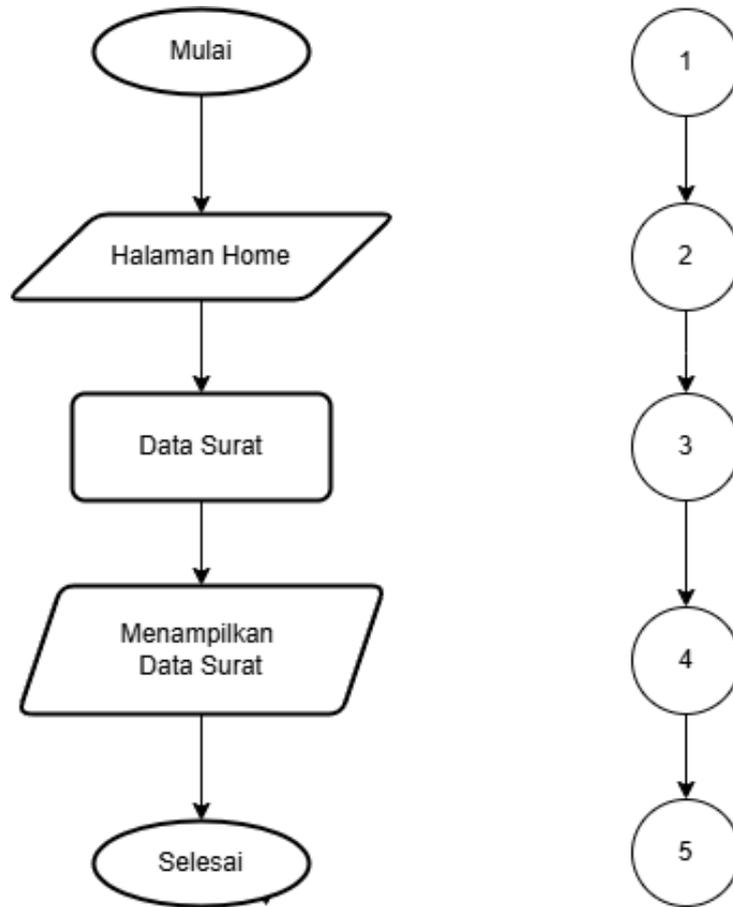
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.30 Grafik Matris Hapus Data Warga

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

i. *Flowchart dan Flowgraph* Menu Data Surat



Gambar 4.59 *Flowchart dan Flowgraph* Menu Data Surat

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

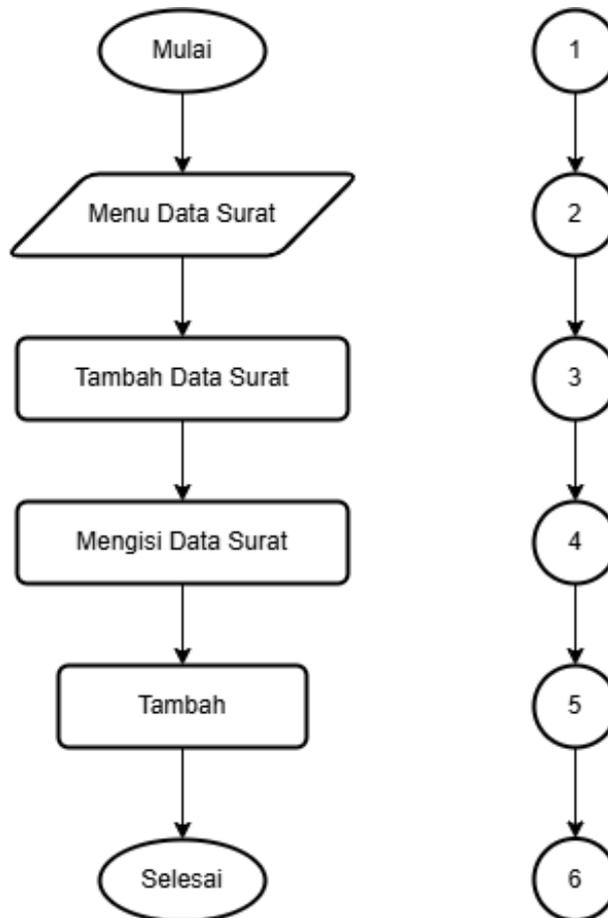
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.31 Grafik Matris Menu Data Surat

	1	2	3	4	5	E-1	
1		1				$1 - 1 = 0$	
2			1			$1 - 1 = 0$	
3				1		$1 - 1 = 0$	
4					1	$1 - 1 = 0$	
5						0	
	SUM (E + 1)						$0 + 1 = 1$

j. *Flowchart dan Flowgraph* Tambah Data Surat



Gambar 4.60 *Flowchart dan Flowgraph* Tambah Data Surat

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\textit{edge}) = 5$$

$$N (\textit{node}) = 6$$

$$P (\textit{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

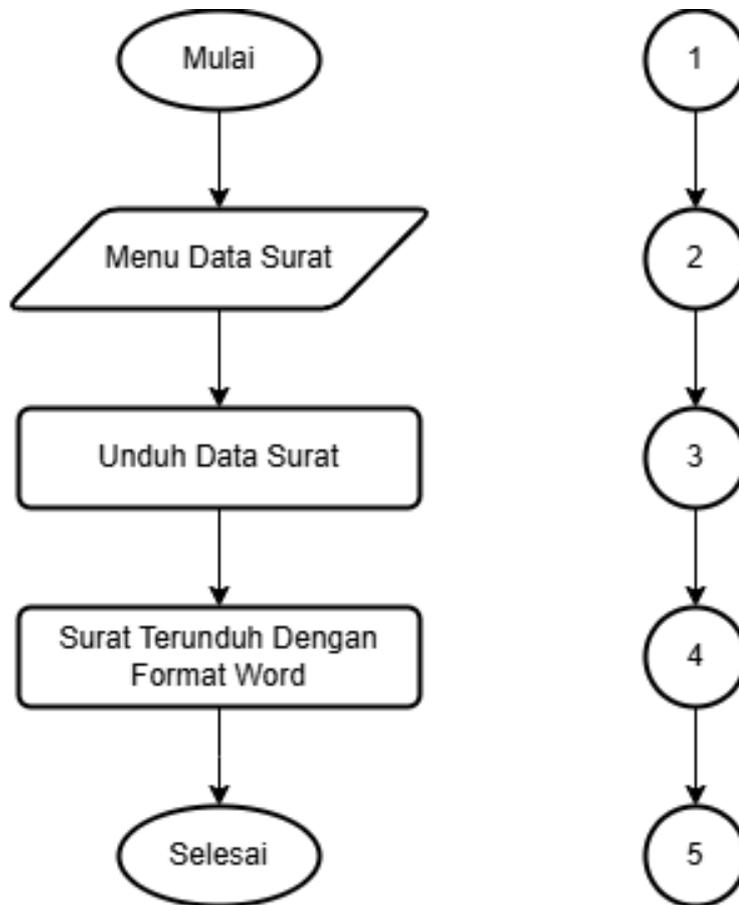
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.32 Grafik Matris Tambah Data Surat

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

k. *Flowchart dan Flowgraph* Unduh Data Surat



Gambar 4.61 *Flowchart dan Flowgraph* Unduh Data Surat

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

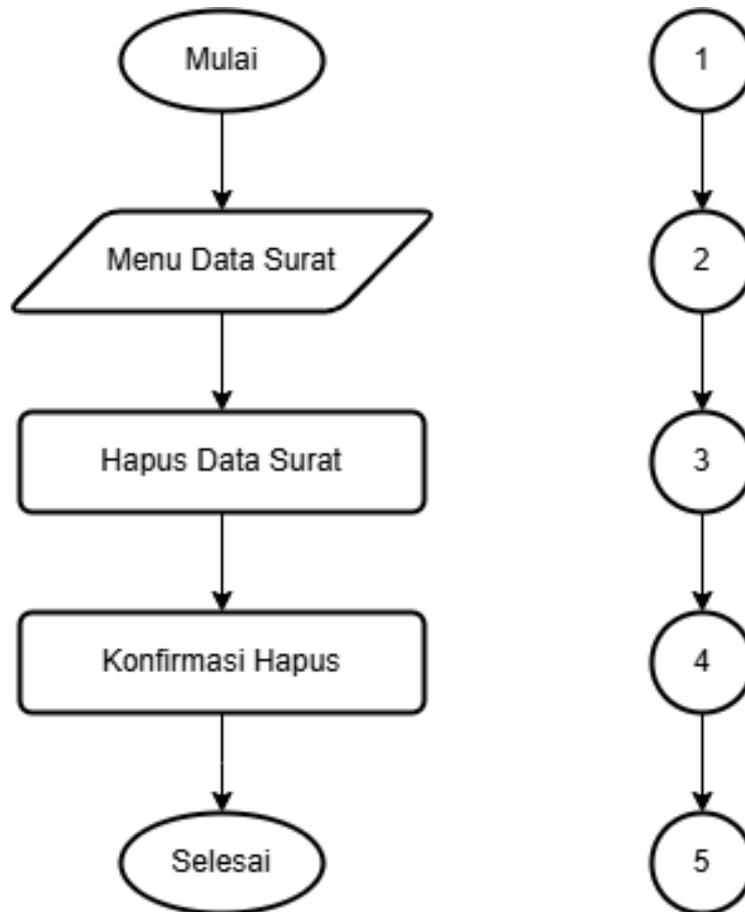
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.33 Grafik Matris Unduh Data Surat

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

1. *Flowchart dan Flowgraph* Hapus Data Surat



Gambar 4.62 *Flowchart dan Flowgraph* Hapus Data Surat

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

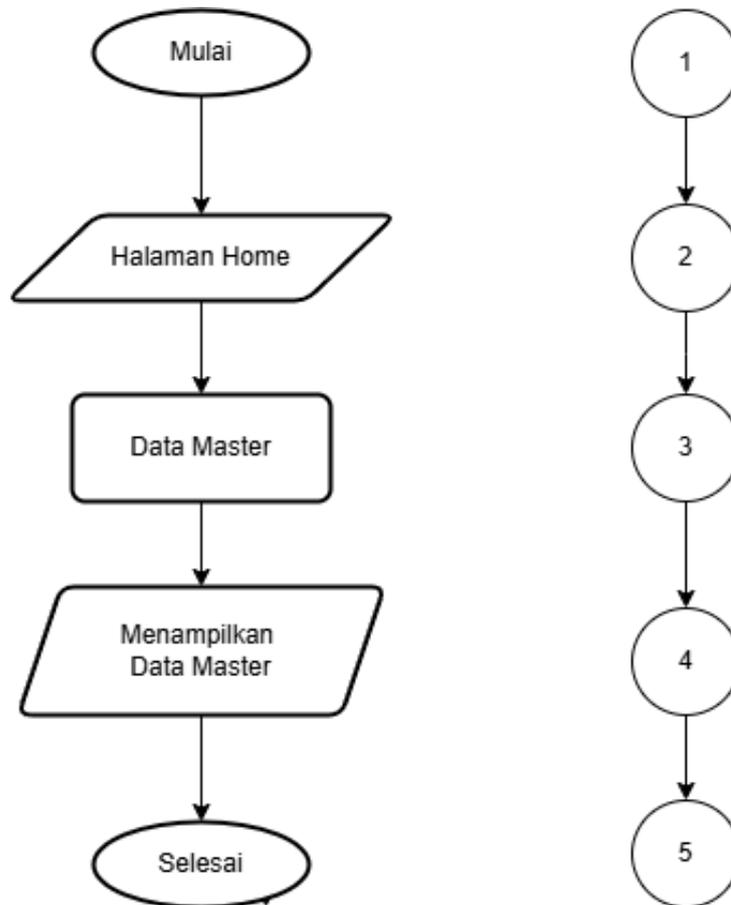
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.34 Grafik Matriks Hapus Data Surat

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

m. *Flowchart dan Flowgraph* Menu Data Master



Gambar 4.63 *Flowchart dan Flowgraph* Menu Data Master

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

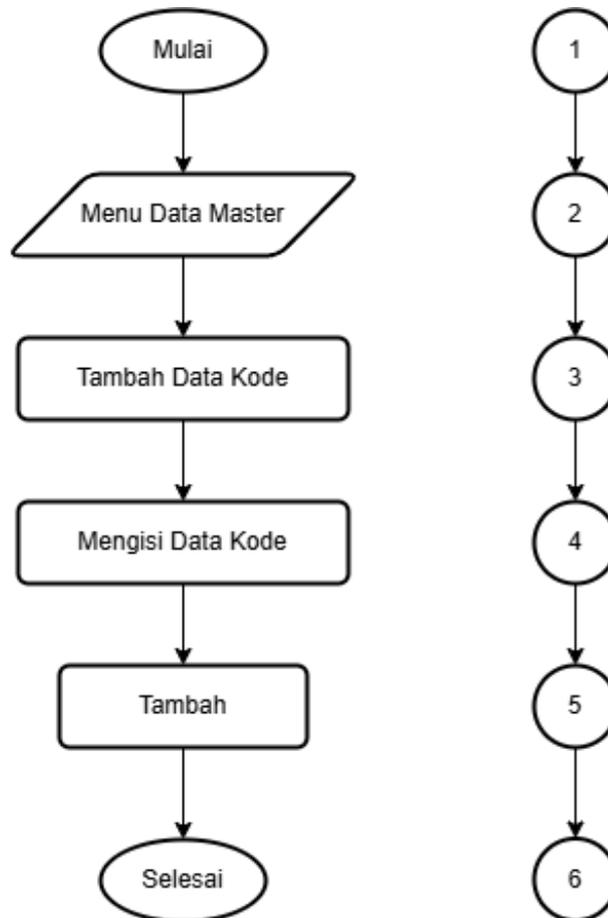
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.35 Grafik Matris Menu Data Master

	1	2	3	4	5	E-1	
1		1				$1 - 1 = 0$	
2			1			$1 - 1 = 0$	
3				1		$1 - 1 = 0$	
4					1	$1 - 1 = 0$	
5						0	
	SUM (E + 1)						$0 + 1 = 1$

n. *Flowchart dan Flowgraph* Tambah Data Kode



Gambar 4.64 *Flowchart dan Flowgraph* Tambah Data Kode

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

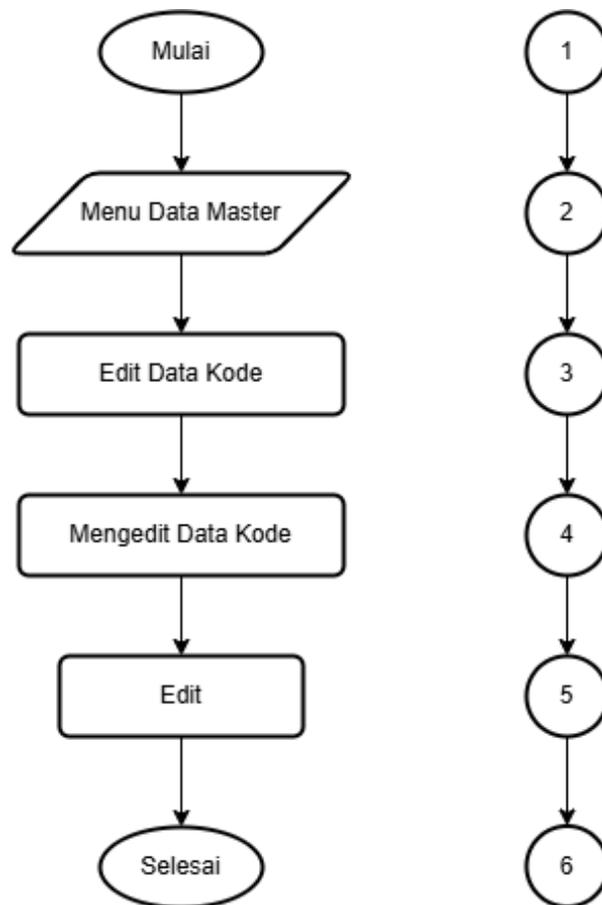
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.36 Grafik Matris Tambah Data Kode

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

o. *Flowchart dan Flowgraph* Edit Data Kode



Gambar 4.65 *Flowchart dan Flowgraph* Edit Data Kode

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

e) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\textit{edge}) = 5$$

$$N (\textit{node}) = 6$$

$$P (\textit{Predikat node}) = 0$$

Penyelesaian :

$$V(G) = E - N + 2$$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

f) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

g) *Independent path* pada *flowgraph* tersebut yakni:

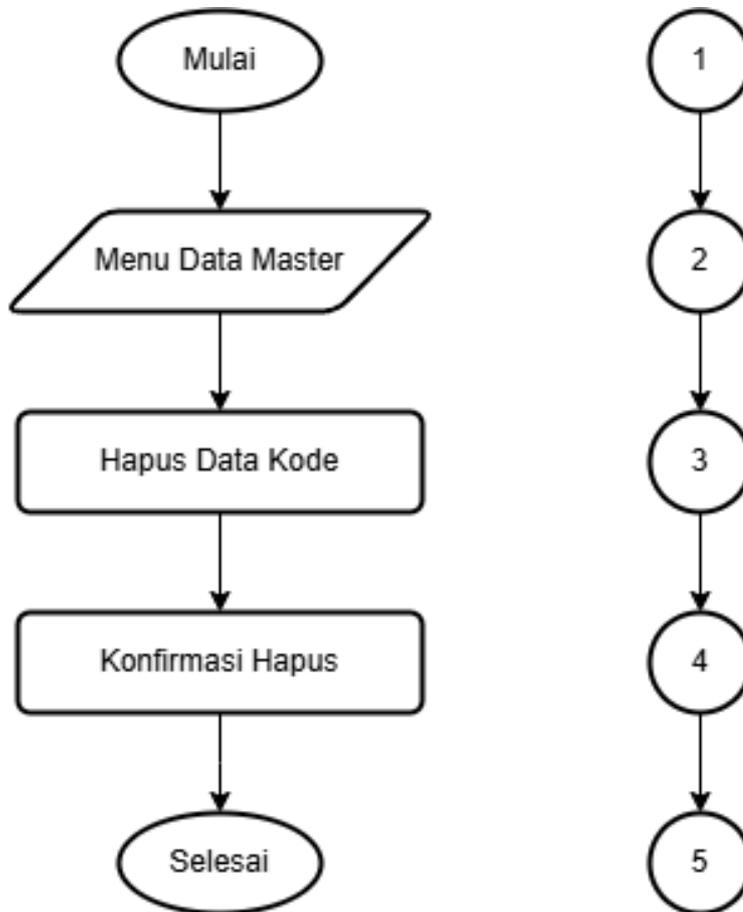
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

h) Grafik Matriks

Tabel 4.37 Grafik Matris Edit Data Kode

	1	2	3	4	5	6	E-1	
1		1					1 - 1 = 0	
2			1				1 - 1 = 0	
3				1			1 - 1 = 0	
4					1		1 - 1 = 0	
5						1	1 - 1 = 0	
6							0	
	SUM (E + 1)							0 + 1 = 1

p. *Flowchart dan Flowgraph* Hapus Data Kode



Gambar 4.66 *Flowchart dan Flowgraph* Hapus Data Kode

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

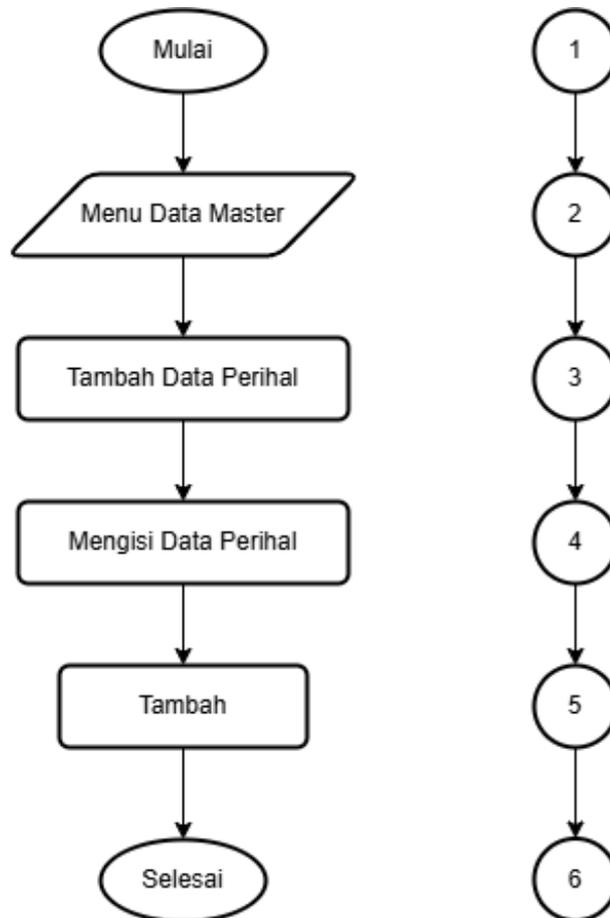
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.38 Grafik Matris Hapus Data Kode

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

q. *Flowchart dan Flowgraph* Tambah Data Perihal



Gambar 4.67 *Flowchart dan Flowgraph* Tambah Data Perihal

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

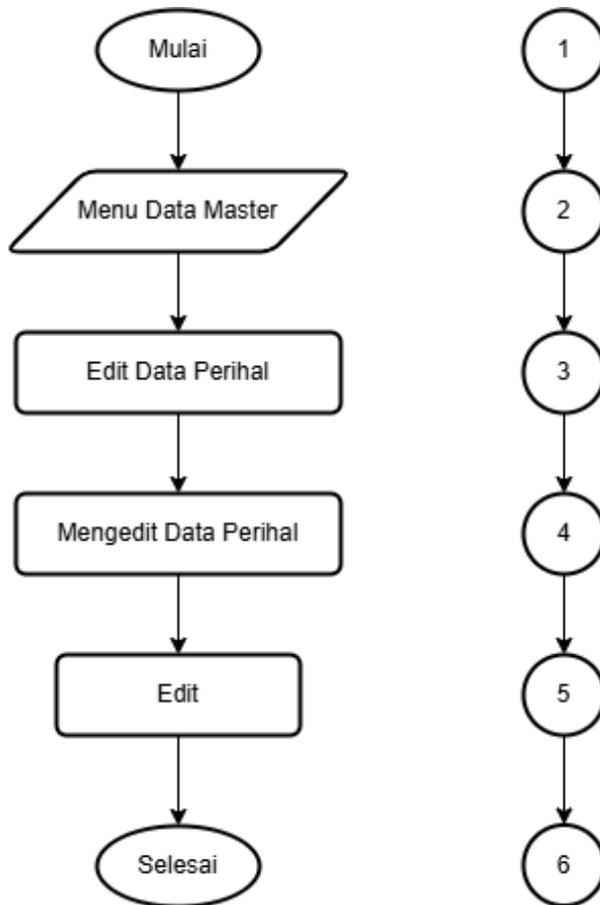
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

d) Grafik Matriks

Tabel 4.39 Grafik Matris Tambah Data Perihal

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

r. *Flowchart dan Flowgraph* Edit Data Perihal



Gambar 4.68 *Flowchart dan Flowgraph* Edit Data Perihal

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

i) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 6

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 5 - 6 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

j) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

k) *Independent path* pada *flowgraph* tersebut yakni:

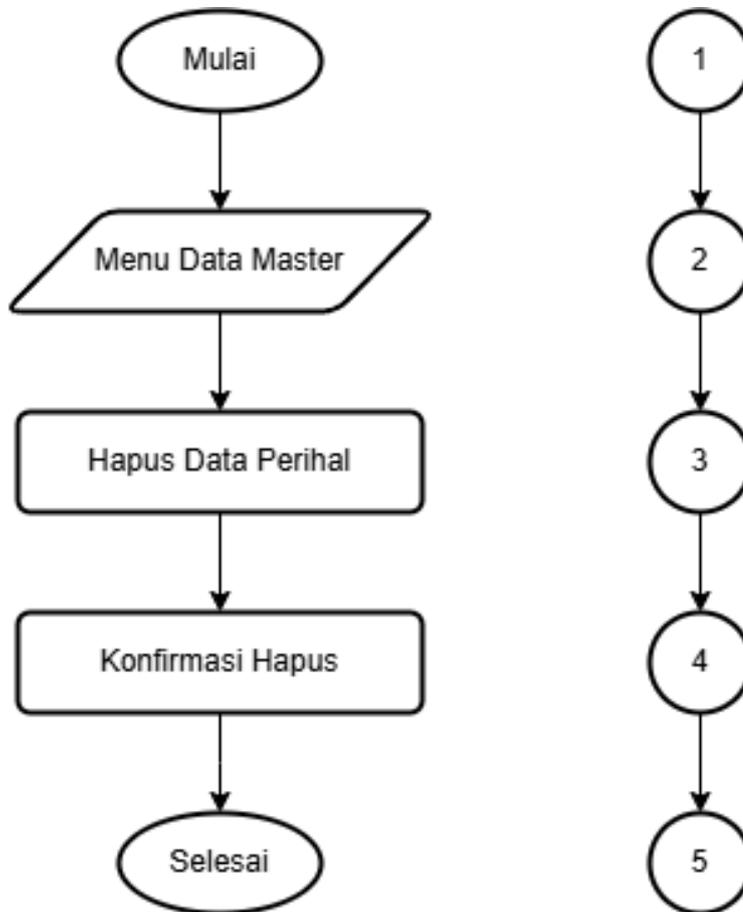
$$\text{Path 1} = 2 - 3 - 4 - 5 - 6$$

l) Grafik Matriks

Tabel 4.40 Grafik Matris Edit Data Perihal

	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4					1		$1 - 1 = 0$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$0 + 1 = 1$

s. *Flowchart dan Flowgraph* Hapus Data Perihal



Gambar 4.69 *Flowchart dan Flowgraph* Hapus Data Perihal

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

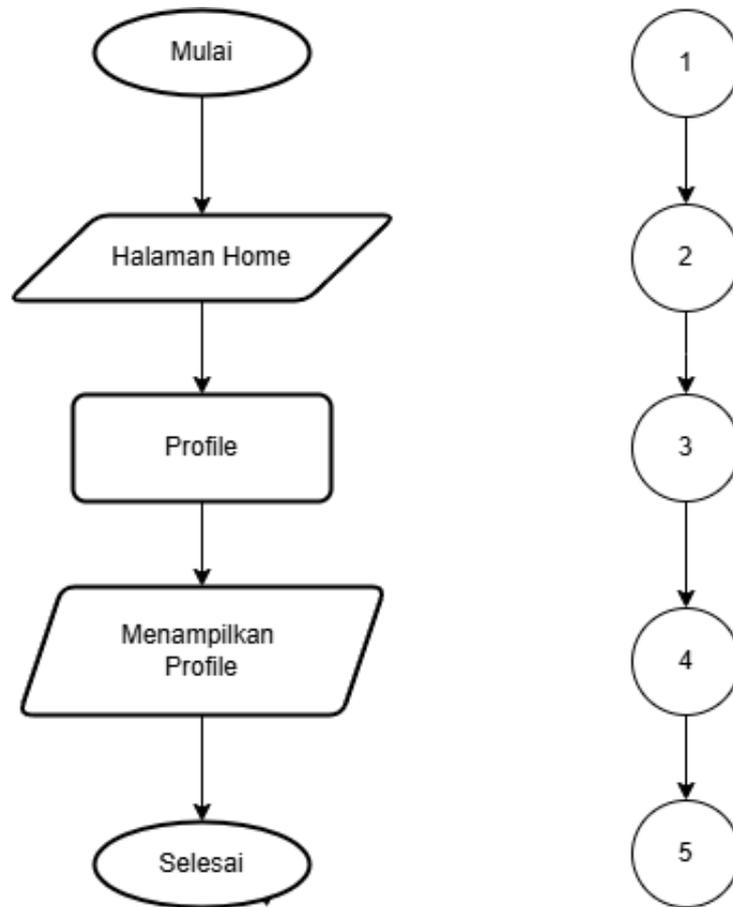
$$\text{Path 1} = 2 - 3 - 4 - 5$$

d) Grafik Matriks

Tabel 4.41 Grafik Matriks Hapus Data Perihal

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

t. *Flowchart dan Flowgraph Profile*



Gambar 4.70 *Flowchart dan Flowgraph Menu Profile*

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 4

N (*node*) = 5

P (*Predikat node*) = 0

Penyelesaian :

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

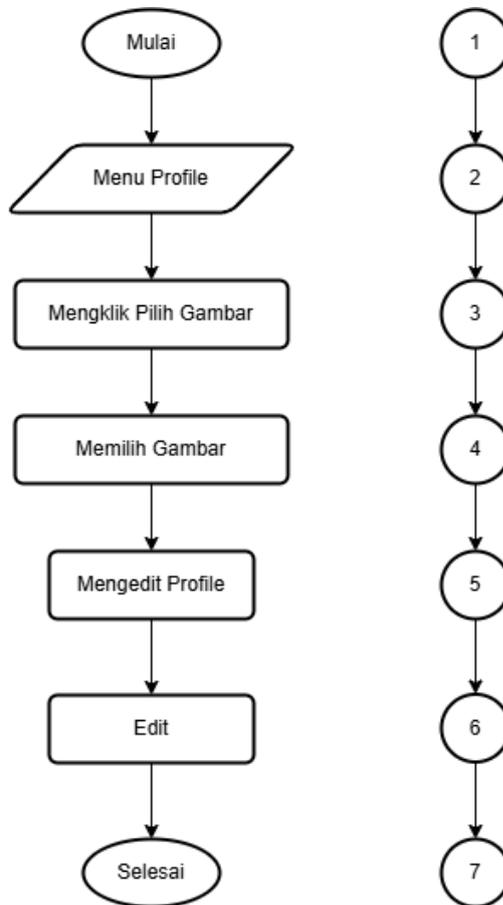
$$\text{Path 1} = 2 - 3 - 4 - 5$$

a) Grafik Matriks

Tabel 4.42 Grafik Matris Profile

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 - 1 = 0
3				1		1 - 1 = 0
4					1	1 - 1 = 0
5						0
	SUM (E + 1)					0 + 1 = 1

u. *Flowchart dan Flowgraph* Edit Profile



Gambar 4.71 *Flowchart dan Flowgraph* Edit Profile

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

a) Menghitung *cyclomatic complexcity* $V(G)$ pada *egde* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 6

N (*node*) = 7

P (*Predikat node*) = 0

Penyelesaian :

$V(G) = E - N + 2$

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *flowgraph* di atas memiliki *Region* = 1

c) *Independent path* pada *flowgraph* tersebut yakni:

$$\text{Path 1} = 2 - 3 - 4 - 5 - 6 - 7$$

d) Grafik Matriks

Tabel 4.43 Grafik Matriks Edit Data Profile

	1	2	3	4	5	6	7	E-1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4					1			1 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
								0
	SUM (E + 1)							0 + 1 = 1

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis web yang dapat mempercepat dan menyederhanakan proses administrasi di kantor kelurahan, sehingga masyarakat dapat menerima layanan dengan lebih cepat dan efisien. Dengan adanya aplikasi ini, diharapkan proses administrasi yang biasanya memakan waktu lama dan bergantung pada sistem manual dapat digantikan dengan sistem yang lebih otomatis dan mudah diakses. Aplikasi berbasis web memungkinkan masyarakat untuk mengakses layanan tanpa terbatas oleh waktu dan lokasi, meningkatkan kualitas pelayanan yang diberikan oleh pihak kelurahan. Selain itu, sistem ini juga dapat mengurangi kemungkinan kesalahan yang sering terjadi pada proses administrasi manual.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat perlu diperbaiki dan pengembangan di penelitian selanjutnya. Oleh karena itu penulis memiliki saran untuk pengembangan selanjutnya :

1. Pelatihan Pegawai yaitu sebelum aplikasi diluncurkan, pegawai kelurahan perlu mendapatkan pelatihan intensif agar dapat mengoperasikan aplikasi dengan baik dan efisien. Hal ini penting agar tidak terjadi kesalahan dalam penggunaan sistem yang bisa mengganggu kelancaran proses administrasi.

2. Pengumpulan Umpan Balik Masyarakat untuk memastikan aplikasi tetap sesuai dengan kebutuhan masyarakat, sebaiknya dilakukan pengumpulan umpan balik dari pengguna secara rutin. Dengan demikian, aplikasi bisa terus diperbaiki dan disesuaikan agar lebih *user-friendly*.
3. Keamanan Data yaitu memperhatikan sistem keamanan data dengan serius, karena aplikasi ini akan menangani data pribadi masyarakat. Penerapan enkripsi, sistem otentikasi yang kuat, dan perlindungan terhadap data akan meningkatkan kepercayaan masyarakat terhadap aplikasi ini.

DAFTAR PUSTAKA

- Along, A. (2020). Kualitas Layanan Administrasi Akademik di Politeknik Negeri Pontianak. *Jurnal Ilmiah Administrasi Publik*, 6(1), 94–99.
- Castro, E. (2022). *CSS: Visual Quickstart Guide* (5th ed.). Peachpit Press.
- Dewi, M. A., & Mulyani, S. (2019). Pengaruh Kualitas Sistem Informasi Terhadap Kepuasan Pengguna Sistem Informasi Pelayanan Publik. *Jurnal Riset Akuntansi Dan Keuangan*, 7(3), 601–614.
- Duckett, J. (2022). *CSS in Action*. Manning Publications.
- Fakhriyah, R., Kencana, N., & Qur'anul Kariem, M. (2022). Efektivitas Penerapan Inovasi Pelayanan Publik Dalam Pengembangan E-Government (Studi Kasus Aplikasi Sp4n Lapor! Di Provinsi Sumatera Selatan). *Jurnal Pemerintahan Dan Politik*, 7(3).
- Faqih, A. S., & Wahyudi, A. D. (2022). Rancang Bangun Sistem Informasi Penjualan Berbasis Web (Studi Kasus : Matchmaker). *Jurnal Teknologi Dan Sistem Informasi*, 3(2), 1–8. <http://jim.teknokrat.ac.id/index.php/JTSSI>
- Hasan, Irmayani, A. P., & Marlina. (2023). Sistem Informasi Manajemen PT. Calidad Coco Indonesia. *Jurnal Sintaks Logika*, X No. X. <https://doi.org/10.31850/jsilog.v1i3>
- Hidayat, T., Basri, M., & Marlina. (2023). Perancangan Aplikasi Pelayanan Masyarakat Di Wilayah Hukum Polres Pinrang. *Jurnal Sintaks Logika*, 1(5). <https://jurnal.umpar.ac.id/index.php/sylog>1
- Irwan, Irmayani, A. P., & Wafiah, A. (2023). Sistem Informasi Pelayanan Kantor Urusan Agama Pitu Riase. *Jurnal Sintaks Logika*, 1(1). <https://jurnal.umpar.ac.id/index.php/sylog>
- Laia, O., Halawa, O., & Lahagu, P. (2022). Pengaruh Sistem Informasi Manajemen Terhadap Pelayanan Publik. *Jurnal Akuntansi, Manajemen Dan Ekonomi*, 1(1), 70–76. <https://doi.org/10.56248/jamane.v1i1.15>
- McFarland, D. (2012). *JavaScript & jQuery: The Missing Manual, Second Edition* (Vol. 2).
- Mclaughlin, B. (2011). *A New Way to Look at the Web O'REILLY*"
- Meyer, E. A., & Weyl, E. (2014). *CSS: The Definitive Guide* (4th ed.). O'Reilly & Associates.
- Morgan, J. (2018). *Simplifying JavaScript: Writing Modern JavaScript with ES5, ES6, and Beyond*.
- Patrick, J. (2016). *JAVASCRIPT A Beginner's Guide to Learning the Basics of JavaScript Programming*.

- Pratama, A. B., Syaifuddin, S., & Rachman, F. F. (2021). Penerapan Sistem Informasi Pelayanan Publik Berbasis Website Untuk Meningkatkan Kualitas Pelayanan Administrasi Kependudukan. *Jurnal Ilmu Administrasi*, 18(1), 105–120.
- Primawanti, E. P., & Ali, H. (2022). *Pengaruh Teknologi Informasi, Sistem Informasi Berbasis Web Dan Knowledge Management Terhadap Kinerja Karyawan (Literature Review Executive Support Sistem (Ess) For Business)*. 3(3). <https://doi.org/10.31933/jemsi.v3i3>
- Shute, Z. (2019). *Advanced JavaScript : Speed up Web Development with the Powerful Features and Benefits of JavaScript*. Packt Publishing Ltd.
- Simpson, K. (2015). *You Don't Know JS: ES6 & Beyond*.
- Subramaniam, V. (2018). *Rediscovering Javascript: Master ES6, ES7, and ES8*.
- Supardi, S., Pratiwi, R. N., & Hanafi, I. (2020). Implementasi Sistem Informasi Manajemen Dalam Pelayanan Administrasi Kependudukan. *Jurnal Ilmiah Ilmu Administrasi Publik*, 10(1), 13–24.
- Sutarman. (2007). *Membangun aplikasi Web dengan PHP & MySQL*. <http://achmatim.net>