BAB I

PENDAHULUAN

A. Latar Belakang

Kelurahan Jaya merupakan salah satu dari 8 kelurahan yang ada di Kecamatan Watang Sawitto, yang juga merupakan bagian dari 39 kelurahan di Kabupaten Pinrang. Terdapat 8 desa dan kelurahan di Kecamatan Watang Sawitto. Pada tahun 2020, jumlah penduduk Kecamatan Watang Sawitto sebanyak 56.570 jiwa. Kecamatan Watang Sawitto memiliki luas wilayah sebesar 58,97 km² dan terbagi menjadi 8 desa dan kelurahan.

Jumlah penduduk Kelurahan Jaya sebanyak 9.358 jiwa, Luas wilayah kelurahan jaya adalah 1 km^2 dan terbagi atas wilayah untuk bangunan, lapangan, tempat ibadah, pemukiman, perkantoran, perdangan, dan sebagainya. Kelurahan Jaya terkenal merupakan kawasan perdagangan kota Pinrang yang letaknya sangat strategis karena terletak di tengah kota Pinrang.

Pelayanan administrasi penduduk merupakan bagian integral dari tata kelola pemerintahan yang efektif dan efisien. Kantor Kelurahan Jaya, yang berada di Kecamatan Watang Sawitto, memiliki peran strategis dalam memberikan pelayanan administrasi kepada penduduknya. Namun, dalam praktiknya, masih terdapat beberapa kendala yang perlu diatasi untuk meningkatkan kualitas layanan tersebut.

Data menunjukkan bahwa sebagian penduduk di Kelurahan Jaya mengalami kesulitan dalam mengakses layanan administrasi penduduk yang disediakan oleh

kantor kelurahan. Hal ini disebabkan oleh faktor jarak, waktu operasional yang terbatas, serta prosedur yang kompleks. Selain itu, proses administrasi yang masih manual juga menyebabkan keterlambatan dalam penyelesaian dokumen administrasi, seperti pembuatan Kartu Tanda Penduduk (KTP), Kartu Keluarga (KK), dan surat keterangan lainnya.

Kurangnya transparansi dan akuntabilitas dalam proses pelayanan administrasi penduduk juga menjadi perhatian. Penduduk seringkali tidak mendapatkan informasi yang jelas mengenai proses administrasi yang mereka jalani, sehingga menimbulkan ketidakpastian dan ketidakpuasan.

Dengan adanya tantangan tersebut, dirancanglah aplikasi pelayanan administrasi penduduk di Kantor Kelurahan Jaya, Kecamatan Watang Sawitto menjadi sangat relevan. Aplikasi ini diharapkan dapat meningkatkan aksesibilitas, efisiensi, transparansi, dan akuntabilitas dalam layanan administrasi penduduk. Dengan demikian, penggunaan teknologi informasi dalam pelayanan administrasi dapat memberikan solusi yang efektif untuk memperbaiki masalah-masalah yang teridentifikasi dan meningkatkan kualitas layanan publik di tingkat kelurahan.

B. Rumusan Masalah

Dengan latar belakang yang telah dijelaskan, maka yang menjadi rumusan masalah pada penelitian ini adalah bagaimana membuat aplikasi pelayanan *admin*istrasi penduduk di kantor Kelurahan Jaya Kecamatan Watang Sawitto.

C. Batasan Masalah

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

1. Aplikasi ini mencakup pelayanan administrasi penduduk di kantor Kelurahan Jaya Kecamatan Watang Sawitto.
2. Aplikasi ini bebasis *desktop* dengan menggunakan *framework Electron.js*.

D. Tujuan Penelitian

Penelitian ini bertujuan merancang dan membangun Aplikasi Pelayanan *Admin*istrasi Penduduk di Kantor Kelurahan Jaya Kecamatan Watang Sawito Berbasis *Web* untuk meningkatkan efisiensi layanan *admin*istrasi, mempercepat proses pengurusan dokumen penduduk, serta meningkatkan transparansi dan akuntabilitas layanan *admin*istrasi dalam pemerintahan setempat. Jika aplikasi ini digunakan pada kantor kelurahan maka staff dapat dengan mudah mengolah data penduduk dan membuat surat dengan lebih efisien dan meminimalisir adanya kesalahan.

E. Manfaat Penelitian

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

1. Sebagai bahan referensi bagi penulis lain untuk mengembangkan kemampuan di bidang yang sama.
2. Kemudahan akses layanan *admin*istrasi bagi masyarakat.
3. Monitoring dan evaluasi yang lebih mudah untuk perbaikan layanan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Sistem

Definisi sistem telah banyak dikemukakan oleh para ahli dari berbagai disiplin ilmu. Walaupun definisi sistem begitu banyak namun system dapat diartikan dengan mengacu pada pendapat para ahli.

(Susanti, 2020) sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Suatu sistem dibangun bukan dinilai dari segi tampilan semata, namun dinilai dari pola aliran informasi dan implementasi. Faktor faktor yang harus dipertimbangkan dalam pembangunan sistem, yaitu:

1) Efisiensi dan efektifitas.

   Pola sistem yang dibangun harus sederhana, lengkap dan akurat.

2) Prosedur yang singkat. Sistem yang dirancang tidak menjemukan pada saat melakukan input data.

3) Sistem yang optimal. Sistem dapat bekerja secara optimal dalam pemanfaatan sumber daya yang dimiliki oleh perusahaan.

4) Tren di masa depan. Sistem yang dibangun mampu beradaptasi dengan perkembangan jaman.

5) Efisiensi dalam biaya. Sistem dibangun berdasarkan pada perencanaan dan perancangan secara matang guna menghemat biaya.

6) Integritas dan keamanan data. Sistem yang baru dibangun harus memenuhi standar integritas dan keamanan data.
7) Interaktif. Sistem yang dibangun harus mudah dipahami dengan cara mengidentifikasi, menyeleksi dan menyusun komponen-komponen yang sesuai dengan latar belakang pemakaiannya.

2. Informasi

Dalam hal ini penulis akan mendeskripsikan definisi dari Informasi menurut pendapat ahli yang memiliki gagasan yang sama yaitu :

Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat (Dewi Maharani, 2019).

Informasi adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan, sebagaimana perannya, pengguna membuat keputusan yang lebih baik sebagai kuantitas dan kualitas dari peninggkatan informasi (Retno sari, 2020).

Dari beberapa pengertian informasi menurut pendapat ahli diatas dapat penulis simpulkan, bahwa informasi adalah data mentah yang telah diolah sedemikian rupa sehingga menghasilkan sesuatu yang bermakna bagi penggunanya dalam mengambil sebuah keputusan.

3. Pelayanan Administrasi Kelurahan

Pelayanan administrasi kelurahan mencakup berbagai layanan yang diberikan oleh pemerintah kelurahan kepada masyarakat di wilayah tersebut. Ini bisa mencakup pembuatan KTP, Surat Keterangan Domisili, Surat Izin

Usaha, pendaftaran kelahiran dan kematian, serta bantuan dalam proses administrasi lainnya. Pelayanan administrasi kelurahan bertujuan untuk memudahkan warga dalam mengurus dokumen resmi dan keperluan administratif lainnya. Pemerintah membentuk sistem administrasi dan birokrasi mulai dari pemerintahan yang paling tinggi sampai ke tingkat pemerintahan yang terendah. Hal tersebut dimaksudkan agar dapat memberikan pelayanan yang lebih optimal bagi seluruh masyarakat dan fungsinya sebagai pelayanan masyarakat.

Pelayanan mempunyai dua istilah penting yang perlu diketahui yaitu melayani dan pelayanan. Pengertian melayani adalah membantu menyiapkan apa yang diperlukan seseorang. Sedangkan pelayanan adalah usaha melayani kebutuhan orang lain (Poerwadarminta, 2020).

Berikut adalah beberapa jenis pelayanan yang umumnya terdapat dalam sistem administrasi kelurahan:

1. Pelayanan Administrasi Kependudukan
    - Pencatatan dan pemeliharaan data kependudukan (kelahiran, kematian, perkawinan)
    - Penerbitan Kartu Keluarga dan Kartu Tanda Penduduk
    - Pengurusan dokumen kependudukan lainnya seperti surat pindah dan pengantar nikah.
2. Pelayanan Administrasi Pemerintahan
    - Pengelolaan surat menyurat dan arsip dokumen pemerintahan

- Pelayanan informasi terkait regulasi dan kebijakan pemerintah daerah

3. Pelayanan Sosial
    - Pendaftaran dan penyaluran bantuan sosial kepada masyarakat yang membutuhkan
    - Penyediaan informasi terkait program-program sosial dari pemerintah

4. Administrasi

Secara etimologis, administrasi berasal dari kata *ad* dan *ministrate* yang berarti sebagai berikut: melayani, membantu, memenuhi, melaksanakan. menerapkan, mengendalikan, menyelenggarakan, mengarahkan, mengahasilgunakan, mengelola, melaksanakan mengemudikan, mengatur, mengurus mengusahakan dan mendayagunakan. (Inu Kencana Syafiie. 2021).

Sedangkan administrasi secara luas mempunyai arti sebagai suatu kerja sama yang dilakukan oleh sekelompok orang dan/organisasi berdasarkan pembagian kerja sebagaimana ditentukan dengan struktur dengan mendayagunakan sumber daya untuk mencapai tujuan secara efektif dan efisien. Bertitik tolak dari fitrah manusia sebagai makhluk sosial yang memiliki kemampuan terbatas, maka untuk mengendalikan dan memenuhi aneka ragam kebutuhannya, suka atau tidak suka kembali dia membutuhkan uluran tangan dari lingkungan sekitarnya termasuk manusia lain. Dapat dipahami pameo klasik bahwa kehidupan manusia tidak berarti tanpa kehadiran orang lain disisinya. Administrasi juga dapat diartikan sebagai salah satu cabang studi

ilmu sosial, administrasi dalam arti luas mencakup keseluruhan proses aktivitas kerjasama sejumlah manusia di dalam organisasi untuk mencapai satu atau sejumlah tujuan yang telah disepakati sebelumnya.

5. *WEB*

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik antara *page* yang disimpan dalam *server* yang sama maupun *server* di seluruh dunia. Halaman dapat diakses dan dibaca melalui browser seperti Google Chrome, Mozilla Firefox, dan lainnya.

Web berasal dari kata Bahasa Inggris yang bila diterjemahkan dalam Bahasa Indonesia berarti "Jaring Laba-Laba". Hampir sama dengan arti dari kata *web* itu sendiri, *web* telah membentang ke seluruh penjuru dunia. Tidak hanya terbatas pada lembaga-lembaga penelitian yang ingin memublikasikan hasil riset, tetapi juga telah banyak digunakan oleh perusahaan bisnis yang ingin mengiklankan produk atau untuk melakukan transaksi bisnisnya (Chandra Kesuma, 2020)

6. *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 (Duckett & Wiley, 2011). *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 (Hartl & Donahoe, 2016).

Dokumen yang dibuat dengan menggunakan bahasa bahasa *markup* seperti *HTML* memiliki bagian yang direnderpada 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*" (Nematrian, 2020).

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 *CCS*, 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*.

7. *CCS*

CCS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten *web* di layar, printer, sintesis ucapan, pembaca layar, dan

jendela obrolan. *CCS* 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) *CCS* adalah bahasa pemrograman *web* yang digunakan untuk mengendalikan tampilan halaman *web*. (Castro, 2022) menjelaskan bahwa *CCS* memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. *CCS* 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. *CCS* juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. Beberapa properti *CCS* 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.

8. *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.

9. *JavaScript*



Gambar 2. 3 *Javascript*

JavaScript adalah bahasa pemrograman *open-source* untuk teknologi *web* yang dapat digunakan untuk memodifikasi *HTML* dan *CCS*. 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 penggunanya. Karena kemampuannya untuk menginstruksikan *browser* untuk melakukan tugas, sekarang *JavaScript* menjadi *client-side scripting Language* yang paling populer (Shute, 2019).

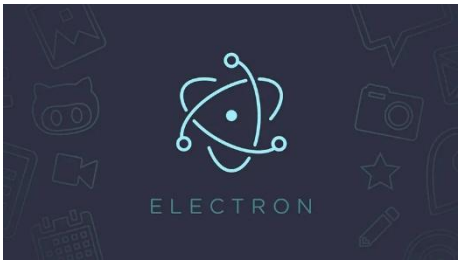
JavaScript dapat meningkatkan kegunaan halaman *web* dengan memberikan umpan balik yang cepat. Dengan bantuan bahasa pemrograman *JavaScript*, Anda dapat meningkatkan *HTML* Anda dengan menambahkan animasi, interaksi, dan efek visual yang dinamis untuk pengguna *browser*. (McFarland, 2012). Selain itu, *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).

10. Electron.js



Gambar 2. 4 *Electron.js*

Electron.js adalah *framework* o*pen-source* yang dikembangkan oleh *GitHub* untuk membangun aplikasi *desktop* lintas platform menggunakan teknologi *web* seperti *HTML*, *CCS*, dan *JavaScript*. Dengan *Electron*,

pengembang dapat membuat aplikasi yang berjalan di Windows, macOS, dan Linux dengan basis kode yang sama.

Beberapa fitur utama *Electron.js* meliputi:

a) Lintas Platform: Aplikasi yang dibangun dengan *Electron* dapat dijalankan di berbagai sistem operasi tanpa perubahan kode yang signifikan.
b) Integrasi *Native*: *Electron* menyediakan *API* untuk mengakses fitur sistem operasi seperti *File* system, notifikasi, dan menu *native*.
c) Pengembangan Cepat: Menggunakan teknologi *web* yang sudah familiar, memungkinkan pengembangan aplikasi *desktop* yang cepat dan efisien.
d) Komunitas Besar: *Electron* memiliki komunitas yang aktif dan banyak sumber daya yang tersedia, termasuk *plugin* dan alat bantu.

11. 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*

12. *Unified modelling Language*

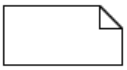
UML adalah serangkaian alat yang digunakan untuk menyederhanakan suatu sistem atau perangkat lunak berbasis objek. Singkatan dari *Unified Modelling Language*, *UML* juga berperan dalam mempermudah pengembangan aplikasi secara berkelanjutan. Sistem atau aplikasi yang tidak

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.

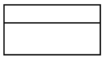
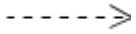
UML (*Unified Modelling Language*), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. *UML* menyediakan format standar untuk merancang model sistem. Namun, karena *UML* memanfaatkan kelas dan operasi sebagai bagian dari konsep intinya, 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:

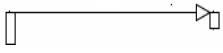
Tabel 2. 1 *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
4		*Generalization*	Menspesifikasikan bahwa *Use Case* sumber secara eksplisit.
5		*Include*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Extend*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*Association*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

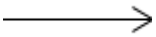
Tabel 2. 2 *Symbol Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *Entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi-in*form*asi tentang aktivitas yang terjadi

Tabel 2. 4 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai *Link* pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah *State* objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 5 *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	*State* dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	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.

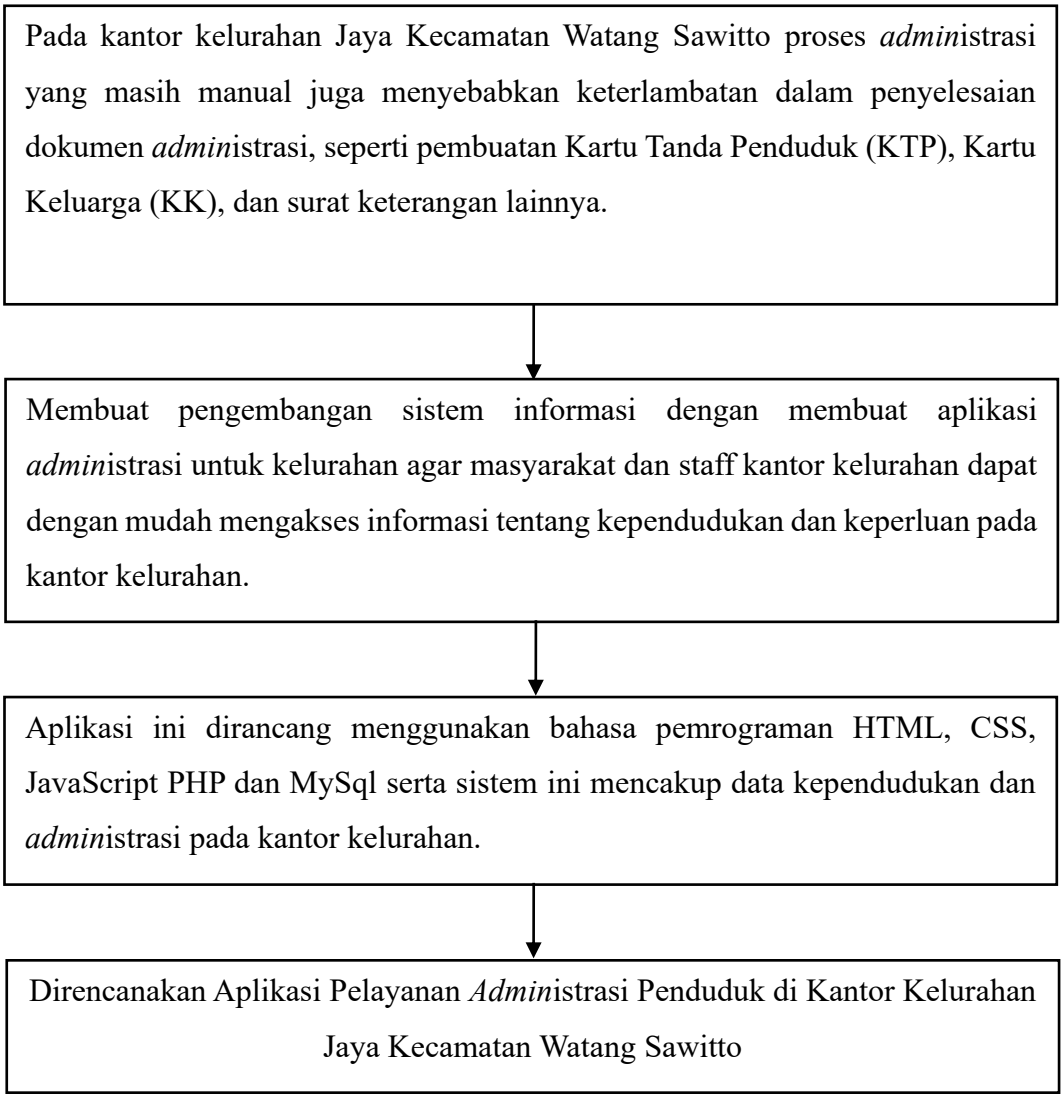
(Lulu et al.,2019) dengan judul "Sistem Informasi *Admin*istrasi kependudukan Kelurahan Rembige Berbasis *web*). menghasilkan sebuah aplikasi yang dapat digunakan untuk keperluan *admin*istrasi kelurahan Rembige berupa keperluan surat menyurat, laporan kependudukan, dan data penduduk.

(Wida et al.,2021) dengan judul "Sistem Informasi *Admin*istrasi Kependudukan (SIASIK) Pada Kelurahan Berbasis *Web*". Penelitian ini menghasilkan Sistem Informasi *Admin*istrasi Kependudukan (SIASIK) sangat bermanfaat bagi masyarakat dalam pembuatan surat keterangan yang dapat diakses dari rumah, membantu *admin*istrator atau petugas dalam penyimpanan data masyarakat dan dalam pengisian data pemohon. SIASIK membantu pengolahan data secara lebih tepat, cepat, praktis dan efisien. Data kependudukan dapat tersimpan dengan aman dan mudah dicari oleh *admin*istrator atau petugas, sehingga pelayanan kepada masyarakatpun akan lebih cepat.

(Vicky Al et al., 2020) dengan judul "Sistem Informasi Pelayanan *Admin*istrasi Kependudukan Berbasis *Website* Kelurahan Banaran". Sistem Informasi Pelayanan *admin*istrasi kependudukan berdasarkan hasil pengujian dengan menggunakan *BlackBox* secara keseluruhan fitur bekerja sesuai dan berjalan dengan baik, hal itu dibuktikan selama dilakukan pengujian tidak terdapat error yang ditemukan. Berdasarkan pengujian SUS didapatkan nilai rata-rata 74.9 yang dapat disimpulkan sistem yang dibangun termasuk kedalam kategori baik

yang dapat memberikan kemudahan atas kendala dalam pelayanan *admin*istrasi secara konvensional atau datang langsung ke kantor.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian *R&D* (*Research and Development*). Jenis penelitian yang digunakan adalah penelitian deskriptif dimana memberikan gambaran mengenai fenomena yang sesungguhnya terjadi dan menggunakaan pendekatan kualitatif. Dalam pendekatan kualitatif perlu menekankan pada pentingnya kedekatan dengan orang-orang dan situasi penelitian, termasuk dengan menjelaskan tingkah laku, persepsi, motivasi, tindakan, dan lain-lain secara keseluruhan, dan dengan cara deskripsi dalam bentuk kata-kata dan bahasa, pada suatu konteks khusus yang alamiah dan dengan memanfaatkan berbagai metode alamiah, agar peneliti memperoleh pemahaman jelas tentang realitas dan kondisi kehidupan nyata.

B. Lokasi dan Waktu Penelitian

Penelitian dilakukan di Kabupaten Pinrang, Kecamatan Watang Sawitto, Kelurahan Jaya, Sulawesi Selatan. Penelitian ini akan berlangsung selama ± 2 (dua) bulan di tahun 2024.

C. Alat dan Bahan Penelitian

1. Perangkat keras

a. Laptop Acer Aspire 3 dengan spesifikasi hardware:

- *Processor* : AMD A4-9120E Dual-Core
- *Installed RAM* : 4GB

- *HDD* : 1000GB

2. Perangkat lunak

a. Windows 11 Home

b. Visual Studio Code

c. Xampp

d. *PHP*

e. *JavaScript*

D. Teknik Pengumpulan Data

1. Wawancara

Penulis melakukan tanya jawab dan wawancara dengan pihak yang dibutuhkan dalam penelitian ini seperti perwakilan bagian kurikulum, bagian tata usaha, guru, siswa, sehingga dari pertanyaan tersebut diharapkan dapat memberikan sebuah informasi yang dapat digunakan dalam penelitian (Sugiono, 2019).

2. Observasi

Observasi merupakan teknik pengumpulan data dengan cara mengamati secara langsung maupun tidak langsung. Observais tidak langsung dilakukan dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

3. Dokumentasi

Pada tahap ini dilakukan mengumpulkan data yang berkaitan dalam bentuk dokumen. Dokumen-dokumen tersebut dapat berupa dokumen tertulis dan dokumen elektronik.

E. Tahapan Penelitian

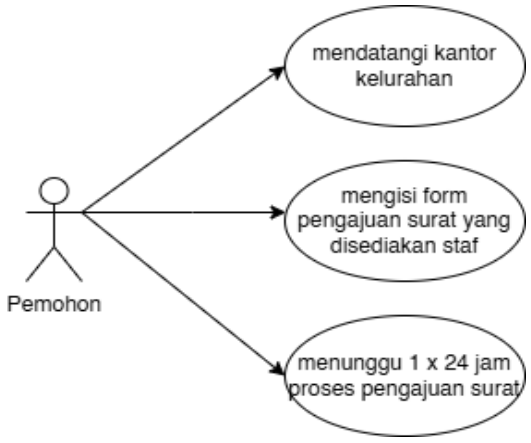
Ada banyak langkah ke tahap penelitian yang disebutkan dalam penelitian ini, termasuk persiapan penelitian, pengumpulan data, analisis, perencanaan, pengujian, dan implementasi. Hasil dari tahap ini adalah sebagai berikut:

1. Persiapan penelitian Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topic penelitian serta software yang digunakan selama penelitian.
2. Pengumpulan data Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya adalah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berpikir ilmiah.
3. Analisis Pada tahap analisis, peneliti memeriksa sistem yang sudah digunakan dan kemudian merumuskan topik penelitian untuk menemukan strategi pemecahan masalah alternatif.
4. Perancangan Berdasarkan pendekatan yang berbeda, para peneliti kemudian membangun aplikasi yang ingin mereka kembangkan.
5. Pengujian Temuan desain diuji oleh para peneliti setelah mereka menyelesaikan desain. Mulai ulang tahap analisis jika ada masalah atau kelemahan.
6. Implementasi Program ini siap untuk digunakan oleh pengguna setelah tidak ada cacat.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Sistem Yang Berjalan



Gambar 4.1 *UseCase* pemohon sistem yang berjalan

Diagram di atas adalah ***Use Case Diagram*** yang menggambarkan proses pengajuan surat di kantor kelurahan oleh seorang pemohon. Berikut adalah penjelasan dari *Diagram* tersebut:

1. **Mendatangi kantor kelurahan**

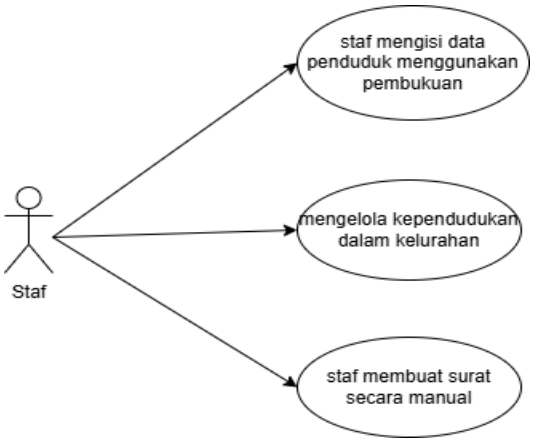
   Pemohon harus datang langsung ke kantor kelurahan untuk mengajukan surat yang dibutuhkan.

2. **Mengisi form pengajuan surat yang disediakan staf**

   Setelah sampai di kantor kelurahan, pemohon harus mengisi formulir yang telah disediakan oleh staf sebagai bagian dari prosedur pengajuan surat.

3. **Menunggu 1 x 24 jam proses pengajuan surat**

Setelah formulir pengajuan diisi dan diserahkan, pemohon harus menunggu proses *admin*istrasi yang memakan waktu sekitar 1 hari kerja sebelum surat selesai dan bisa diambil.



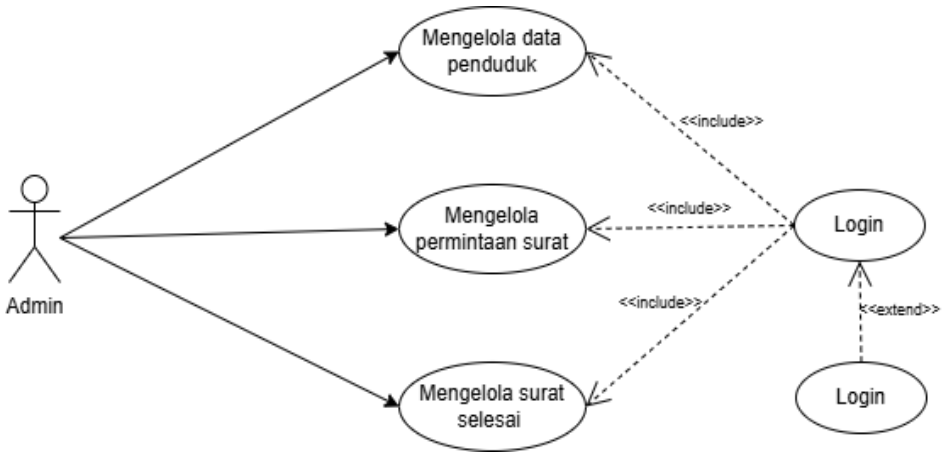
Gambar 4.2 *UseCase* staf sistem yang berjalan

Gambar di atas merupakan *Diagram Use Case* yang menggambarkan sistem yang sedang berjalan dalam pengelolaan kependudukan di kelurahan. Dalam sistem ini, **Staf** memiliki tiga tugas utama, yaitu:

1. **Mengisi data penduduk menggunakan pembukuan** – Pencatatan data penduduk masih dilakukan secara manual menggunakan buku atau dokumen fisik.
2. **Mengelola kependudukan dalam kelurahan** – Staf bertanggung jawab dalam pengelolaan data kependudukan, termasuk pencatatan, perubahan, dan pemrosesan informasi terkait penduduk.
3. **Membuat surat secara manual** – Proses pembuatan surat untuk keperluan *admin*istratif masih dilakukan tanpa bantuan sistem digital, sehingga memerlukan waktu lebih lama dan rentan terhadap kesalahan.

B. Desain Sistem Yang Diusulkan

a. UserCase Diagram



Gambar 4.3 *UseCase* sistem yang diusulkan

Gambar di atas adalah ***Diagram Use Case*** yang menggambarkan **sistem yang diusulkan** untuk pengelolaan kependudukan dan surat-menyurat oleh *admin*. Berikut adalah penjelasan lebih spesifik tentang setiap elemen dalam *Diagram*:

1. **Aktor**

   *Admin***:** Sebagai pengguna sistem yang memiliki akses untuk mengelola data penduduk dan surat-menyurat.

2. ***UseCase* dan Hubungannya**

   1) Mengelola Data Penduduk

   - *Admin* dapat menambahkan, mengedit, atau menghapus data penduduk.

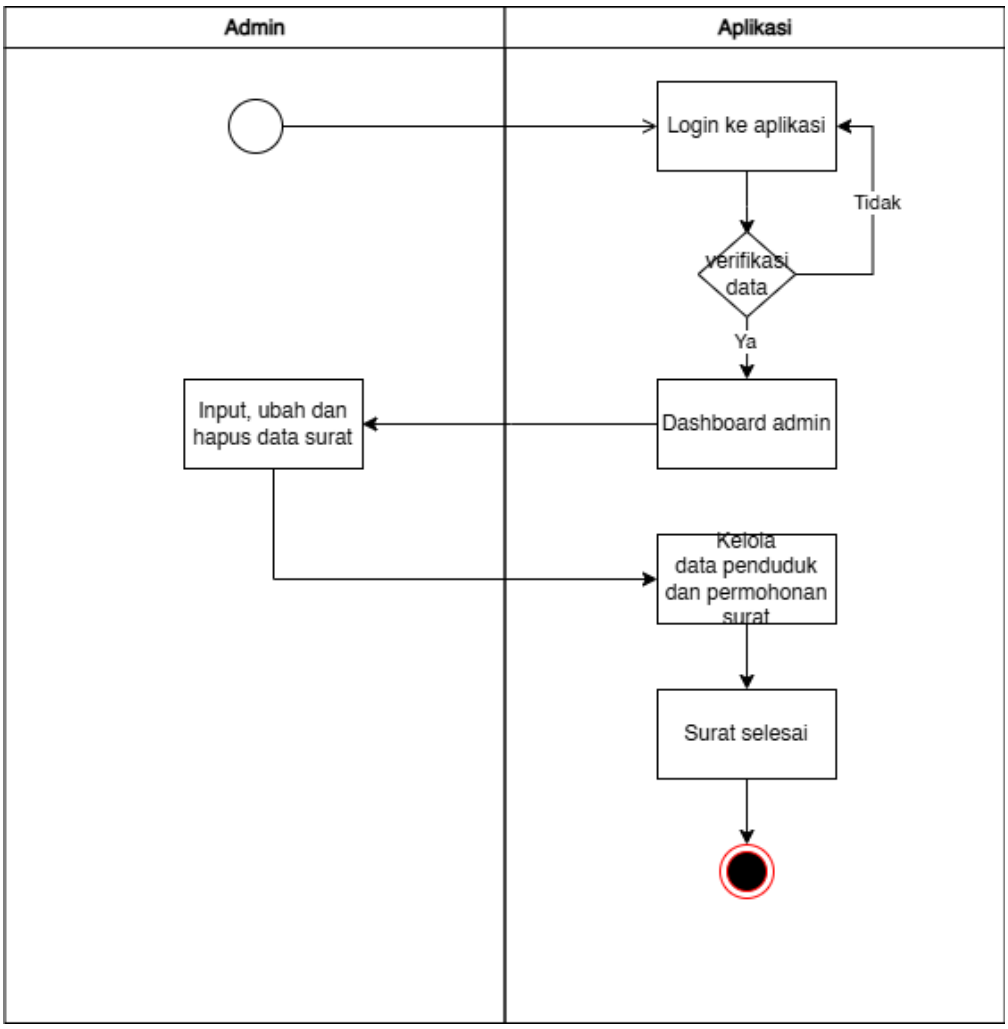
- *Use Case* ini memiliki hubungan **«include»** dengan ***Login***, artinya sebelum *admin* dapat mengelola data penduduk, mereka harus *login* terlebih dahulu.

2) **Mengelola Permintaan Surat**
   - *Admin* bertanggung jawab untuk menerima dan memproses permintaan pembuatan surat dari warga.
   - Seperti *Use Case* sebelumnya, ***Login*** di-*include* dalam *Use Case* ini, yang berarti *admin* harus *login* sebelum mengakses fitur ini.

3) **Mengelola Surat Selesai**
   - Setelah surat diproses, *admin* dapat menyelesaikan dan menyimpannya dalam sistem.
   - *Use Case* ini juga memiliki hubungan **«include»** dengan ***Login***, menunjukkan bahwa akses hanya diberikan setelah *admin* masuk ke dalam sistem.

4) ***Login***
   - Sebagai langkah autentikasi, *admin* harus memasukkan kredensialnya sebelum dapat mengakses fitur manajemen dalam sistem.
   - Terdapat hubungan **«extend»** antara dua *Use Case* ***Login***, yang mungkin menandakan adanya fitur tambahan seperti *logout* atau autentikasi multi-level setelah *login*.

b. Activity Diagram



Gambar 4.4 *Activity* sistem yang diusulkan

a. ***Admin* memulai proses dengan *login***

a) *Admin* memasukkan ***User*name dan password**.

b) Sistem memverifikasi kredensial yang diberikan.

c) Jika valid, *admin* diarahkan ke halaman utama. Jika gagal, sistem menampilkan pesan error.

b. ***Admin* memilih fitur yang ingin digunakan**

a) Mengelola **data penduduk** (menambah, mengedit, menghapus data).

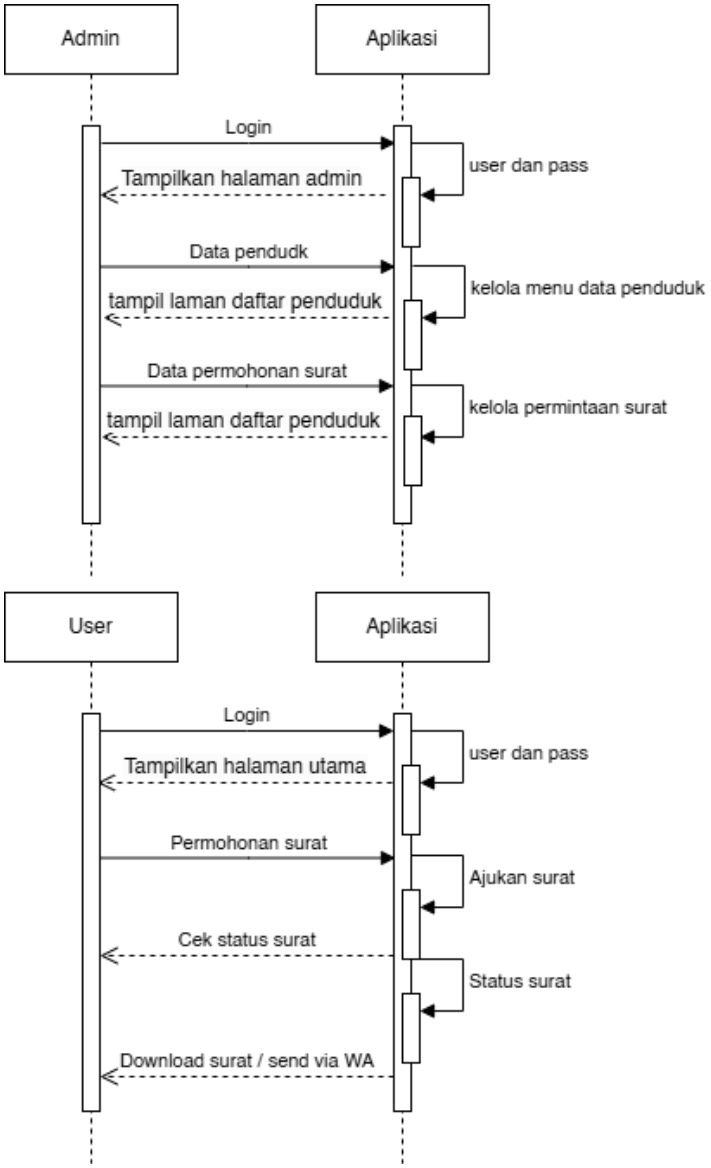
b) Mengelola **permintaan surat** (menerima dan memproses).

c) Mengelola **surat selesai** (menyimpan sebagai arsip).

c. **Setelah selesai, *admin* dapat *logout* dari sistem**

a) Sistem mengakhiri sesi *admin* dan menampilkan halaman *login* kembali.

c. Sequence Diagram



Gambar 4.5 *Sequence* sistem yang diusulkan

- *Admin*

    a. ***Login*:**

        a) *Admin* memasukkan *User*name dan password ke dalam aplikasi.

        b) Aplikasi memverifikasi kredensial dan menampilkan halaman *admin* jika berhasil.

    b. **Mengelola Data Penduduk:**

        a) *Admin* memilih menu "Data Penduduk".

        b) Aplikasi mengambil data penduduk dari *database* dan menampilkan daftar penduduk kepada *admin*.

        c) *Admin* dapat mengelola data tersebut (menambah, mengedit, atau menghapus data).

    c. **Mengelola Permohonan Surat:**

        a) *Admin* memilih menu "Data Permohonan Surat".

        b) Aplikasi mengambil daftar permohonan surat dari *database* dan menampilkannya.

        c) *Admin* dapat mengelola permintaan surat sesuai kebutuhan.

- *User*

    a. *Login*:

        a) *User* memasukkan *User*name dan password.

        b) Aplikasi memverifikasi data dan menampilkan halaman utama *User* jika berhasil *login*.

b. Mengajukan Permohonan Surat:

a) *User* mengajukan permohonan surat melalui aplikasi.

b) Aplikasi memproses dan menyimpan permintaan ke *database*.

c. Mengecek Status Surat:

a) *User* mengecek status permohonan surat yang telah diajukan.

b) Aplikasi mengambil status surat dari *database* dan menampilkannya kepada *User*.

d. Download atau Kirim Surat via WhatsApp:

a) Jika permohonan surat telah selesai diproses, *User* dapat mengunduh surat tersebut atau mengirimnya melalui WhatsApp.

b) Aplikasi menyediakan opsi untuk mengunduh atau mengirim *File* surat kepada *User*.

A. Perancangan *Input* Dan *Output*

1. *Database* Aplikasi

Gambar di bawah menunjukkan antarmuka pengguna dari aplikasi manajemen basis data yang sedang terhubung ke *server* dengan alamat IP 127.0.0.1. Aplikasi ini digunakan untuk mengelola *database* bernama "db_kel_jaya". Pada bagian atas antarmuka, terdapat berbagai menu seperti Struktur, SQL, Cari, *Query*, Ekspor, Impor, Operasi, Hak Akses, Routine, Event, Trigger, Pelacakan, dan Lainnya. Di bagian tengah antarmuka, terdapat tabel yang menampilkan daftar tabel

dan relasinya yang ada di dalam *database* "db_kel_jaya". Tabel-tabel tersebut adalah:

Tabel tb_*admin*

- **Primary Key:** *User*name
- Menyimpan data akun *admin* dan lurah.
- Relasi ke tb_ajukan

Tabel tb_penduduk

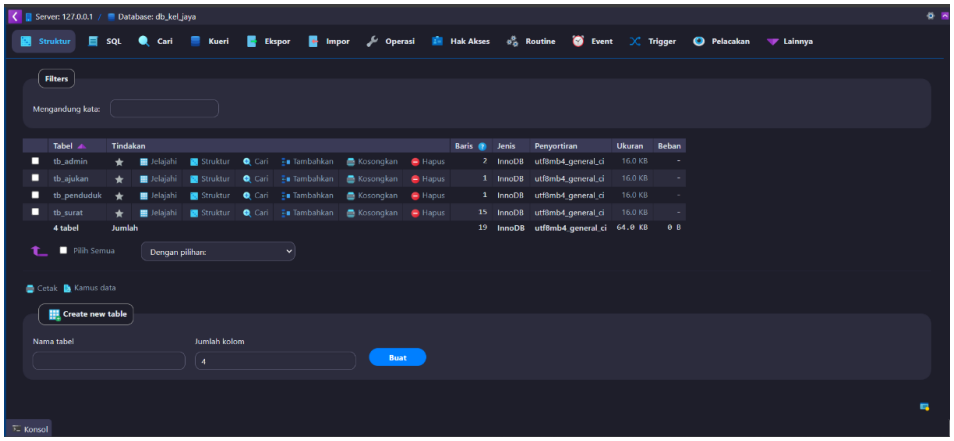
- **Primary Key:** NIK
- Menyimpan data penduduk yang dapat mengajukan surat.
- Relasi **One-to-Many** ke tb_ajukan melalui tb_ajukan.NIK.

Tabel tb_surat

- **Primary Key:** KdSurat
- Menyimpan data master jenis surat beserta template *File*.
- Relasi **One-to-Many** ke tb_ajukan melalui tb_ajukan.KdSurat.

Tabel tb_ajukan

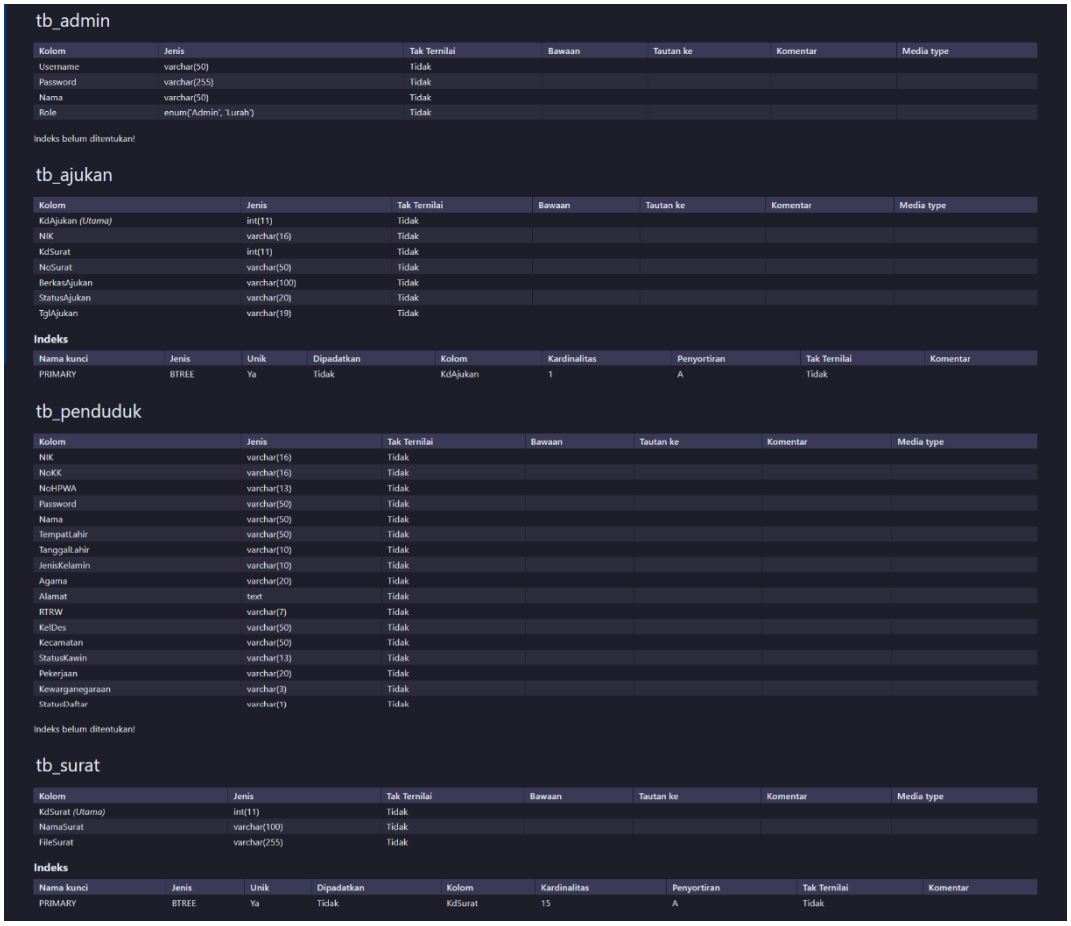
- **Primary Key:** KdAjukan
- **Foreign Keys:**
    - NIK → tb_penduduk.NIK
    - KdSurat → tb_surat.KdSurat
- Menyimpan data pengajuan surat oleh penduduk, termasuk berkas, status, dan tanggal pengajuan.



Gambar 4.6 *Database* Aplikasi

2. **Struktur Tabel Aplikasi**

Gambar di bawah ini menunjukkan tampilan antarmuka dari sebuah aplikasi manajemen basis data yang menampilkan struktur tabel-tabel dalam *database* bernama "db_kel_jaya". Tabel-tabel yang ada dalam *database* tersebut meliputi "tb_*admin*", "tb_ajukan", "tb_penduduk", dan "tb_surat". Setiap tabel memiliki kolom-kolom dengan atribut seperti jenis data, nilai null, default, tautan, komentar, dan tipe media. Selain itu, informasi mengenai indeks yang digunakan pada tabel-tabel tersebut juga ditampilkan, seperti indeks "PRIMARY" pada tabel "tb_ajukan" dan "tb_surat".



tb_admin

Kolom	Jenis	Tak Ternilai	Bawaan	Tautan ke	Komentar	Media type
Username	varchar(50)	Tidak				
Password	varchar(255)	Tidak				
Nama	varchar(50)	Tidak				
Role	enum('Admin', 'Lurah')	Tidak				

Indeks belum ditentukan!

tb_ajukan

Kolom	Jenis	Tak Ternilai	Bawaan	Tautan ke	Komentar	Media type
KdAjukan *(Utama)*	int(11)	Tidak				
NIK	varchar(16)	Tidak				
KdSurat	int(11)	Tidak				
NoSurat	varchar(50)	Tidak				
BerkasAjukan	varchar(100)	Tidak				
StatusAjukan	varchar(20)	Tidak				
TglAjukan	varchar(19)	Tidak				

Indeks

Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
PRIMARY	BTREE	Ya	Tidak	KdAjukan	1	A	Tidak	

tb_penduduk

Kolom	Jenis	Tak Ternilai	Bawaan	Tautan ke	Komentar	Media type
NIK	varchar(16)	Tidak				
NoKK	varchar(16)	Tidak				
NoHPWA	varchar(13)	Tidak				
Password	varchar(50)	Tidak				
Nama	varchar(50)	Tidak				
TempatLahir	varchar(50)	Tidak				
TanggalLahir	varchar(10)	Tidak				
JenisKelamin	varchar(10)	Tidak				
Agama	varchar(20)	Tidak				
Alamat	text	Tidak				
RTRW	varchar(7)	Tidak				
KelDes	varchar(50)	Tidak				
Kecamatan	varchar(50)	Tidak				
StatusKawin	varchar(13)	Tidak				
Pekerjaan	varchar(20)	Tidak				
Kewarganegaraan	varchar(3)	Tidak				
StatusDaftar	varchar(1)	Tidak				

Indeks belum ditentukan!

tb_surat

Kolom	Jenis	Tak Ternilai	Bawaan	Tautan ke	Komentar	Media type
KdSurat *(Utama)*	int(11)	Tidak				
NamaSurat	varchar(100)	Tidak				
FileSurat	varchar(255)	Tidak				

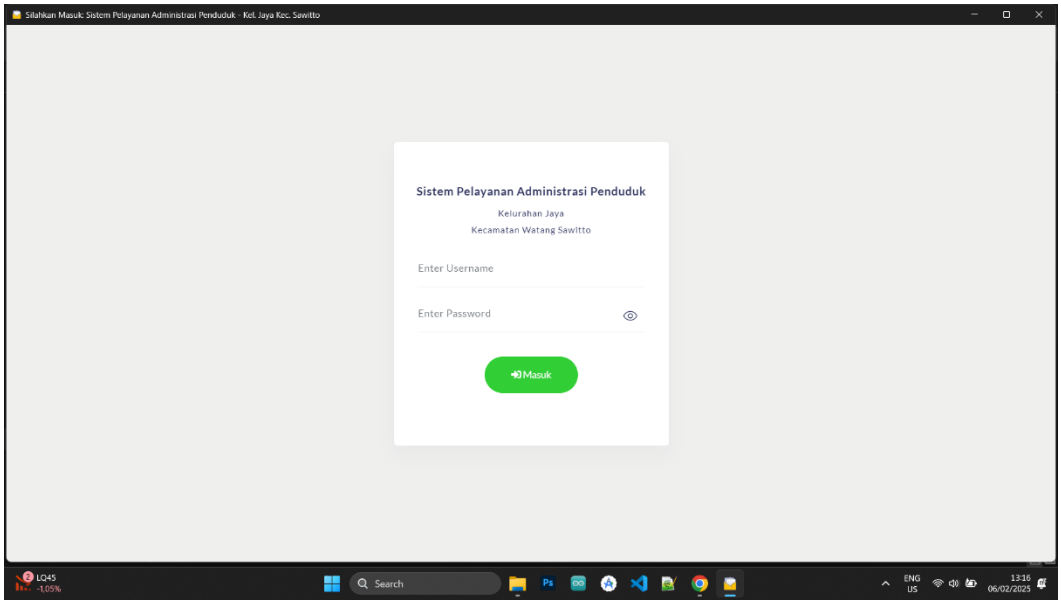
Indeks

Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Ternilai	Komentar
PRIMARY	BTREE	Ya	Tidak	KdSurat	15	A	Tidak	

Gambar 4.7 Struktur Tabel Aplikasi

3. Halaman *Login* Aplikasi

Halaman *Login* adalah tampilan pertama yang muncul saat pengguna membuka aplikasi. Tampilan ini dirancang untuk memberikan pengenalan awal tentang aplikasi dan memudahkan pengguna dalam menavigasi fitur-fitur yang tersedia.



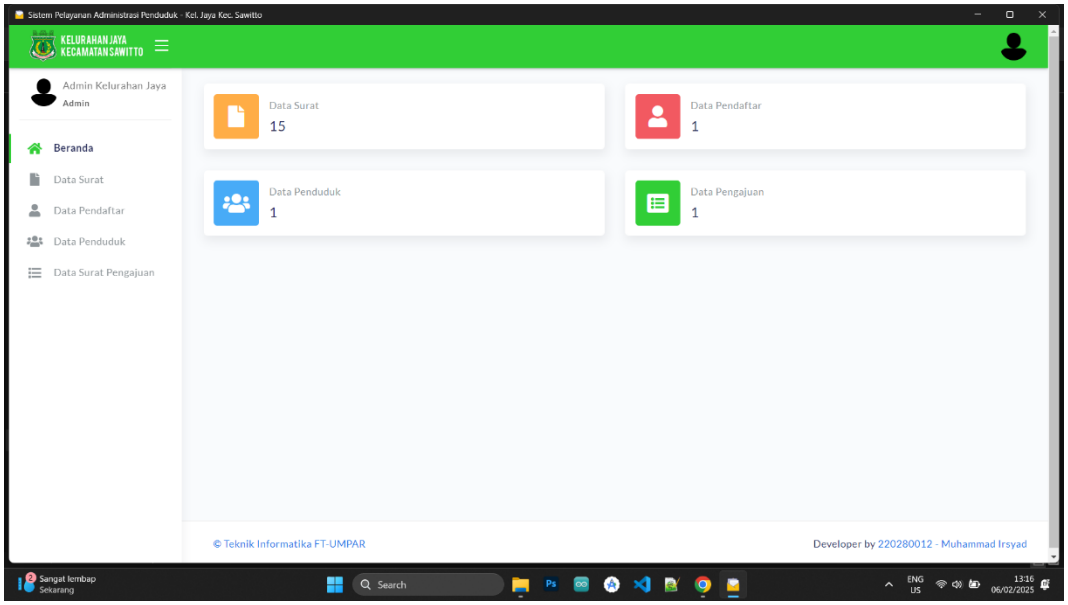
Gambar 4.8 Halaman *Login* Apilkasi

4. Halaman *Dashboard* Aplikasi

Bagian *dashboard* pada gambar menunjukkan tampilan utama dari sistem *administrasi* pelayanan penduduk untuk Kelurahan Jaya, Kecamatan Sawitto. Di dalam *dashboard*, terdapat beberapa kotak informasi yang menampilkan jumlah data yang ada dalam sistem. Diantaranya adalah:

1. Data Surat
2. Data Pendaftar
3. Data Penduduk
4. Data Penjualan

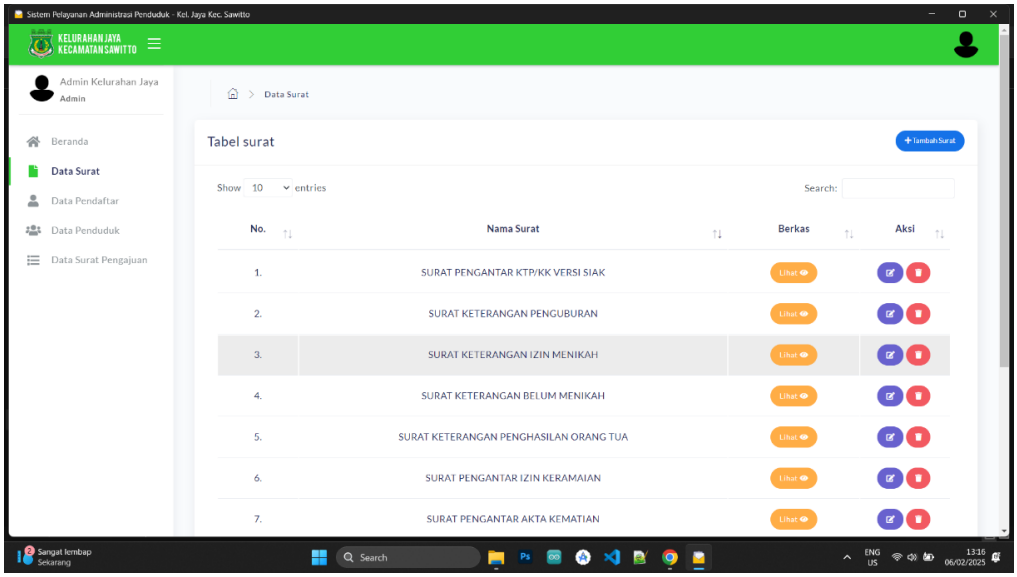
Dashboard ini dirancang untuk memberikan gambaran umum secara cepat dan mudah mengenai jumlah data yang ada dalam sistem kepada pengguna.



Gambar 4.9 Halaman Dashboar Apilkasi

5. Halaman Data Surat

Halaman Data Surat ini dirancang untuk memberikan tampilan yang terstruktur dan mudah dipahami, sehingga pengguna dapat dengan cepat mengakses dan mengelola informasi surat yang diperlukan. Fitur pencarian yang disediakan juga meningkatkan efisiensi dalam menemukan data spesifik.



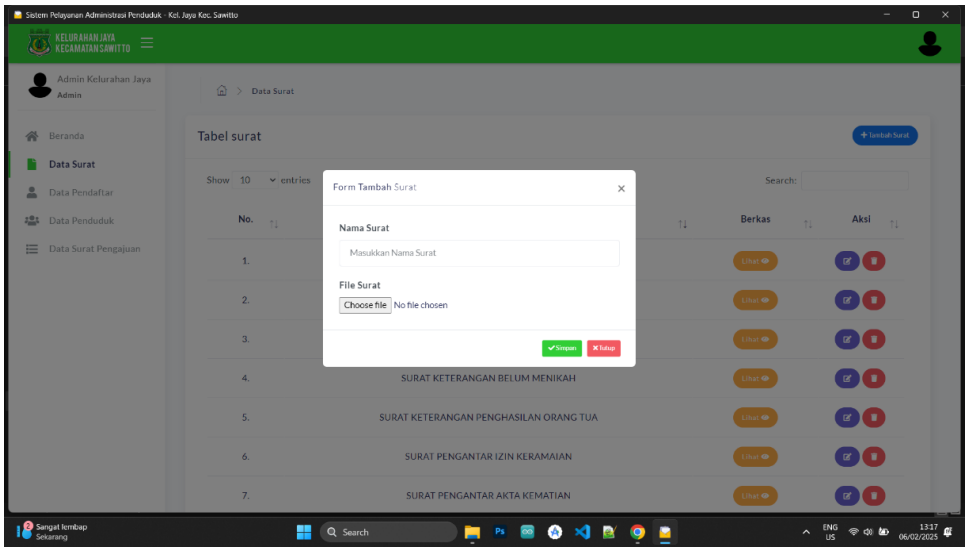
Gambar 4.10 Halaman Data Surat.

6. Form Tambah Data Surat

Gambar tersebut menampilkan Form Tambah Data Surat yang digunakan untuk menambahkan data surat baru ke dalam sistem. Form ini terdiri dari beberapa komponen utama:

1. Nama Surat: Bagian ini memungkinkan pengguna untuk memasukkan nama surat yang akan ditambahkan. Pengguna diharapkan mengisi nama surat sesuai dengan jenis surat yang dimaksud.
2. *File* Surat: Pada bagian ini, pengguna dapat mengunggah *File* surat yang terkait. Terdapat opsi "Choose *File*" yang memungkinkan pengguna memilih *File* dari perangkat mereka. Jika belum ada *File* yang dipilih, sistem akan menampilkan pesan "No *File* chosen".

Form ini dirancang untuk memudahkan pengguna dalam menambahkan data surat baru ke dalam sistem dengan mengisi informasi yang diperlukan dan mengunggah *File* yang relevan. Dengan adanya form ini, proses input data menjadi lebih terstruktur dan efisien.

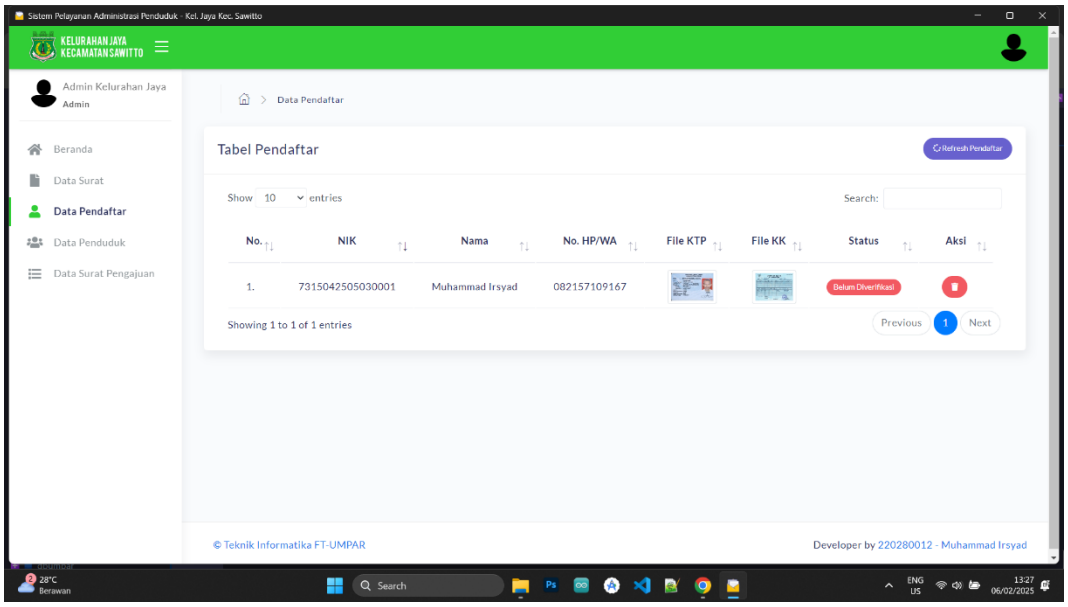


Gambar 4.11 Form Tambah Data Surat

7. Data Pendaftar Belum Diverifikasi

Gambar dibawah ini menampilkan antarmuka dari sistem pelayanan *admin*istrasi, yang memperlihatkan **Data Pendaftar Belum Diverifikasi**. Pada tampilan ini, sistem menampilkan daftar warga yang telah melakukan pendaftaran namun belum mendapatkan verifikasi dari *admin*istrator.

Dengan tampilan ini, sistem membantu memastikan bahwa seluruh proses *admin*istrasi kependudukan berjalan lancar, transparan, dan akuntabel. *Admin*istrator dapat dengan mudah memantau pendaftar baru, melakukan verifikasi, dan memastikan bahwa data penduduk yang ada dalam sistem adalah valid dan terpercaya.

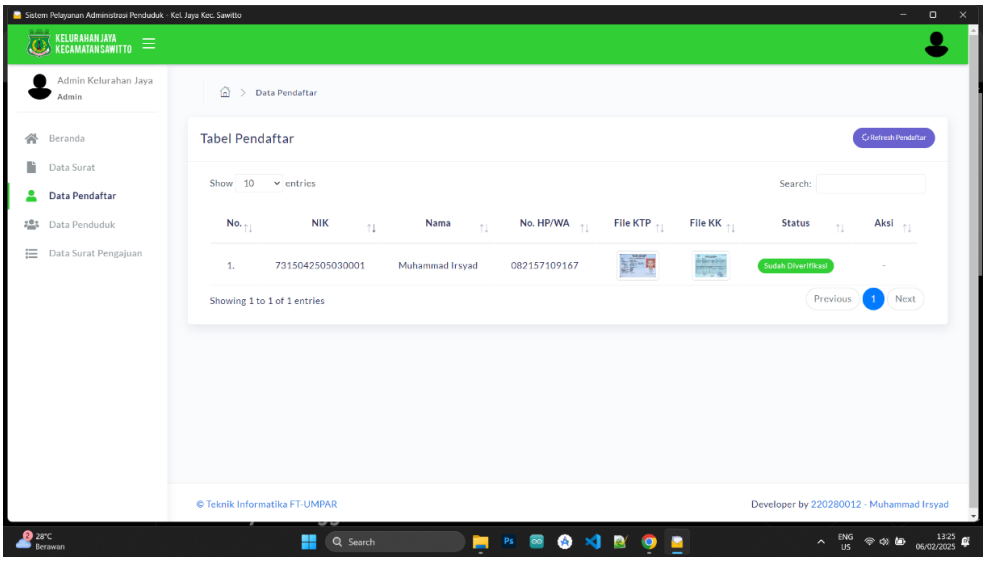


Gambar 4.12 Pendaftar Belum Diverifikasi

8. Data Pendaftar Sudah Diverifikasi

Gambar ini menunjukkan bagaimana sistem menampilkan *Output* data yang sudah diverifikasi, sekaligus mendukung efisiensi dalam pengelolaan informasi kependudukan. Tampilan ini merupakan contoh implementasi dari rancangan

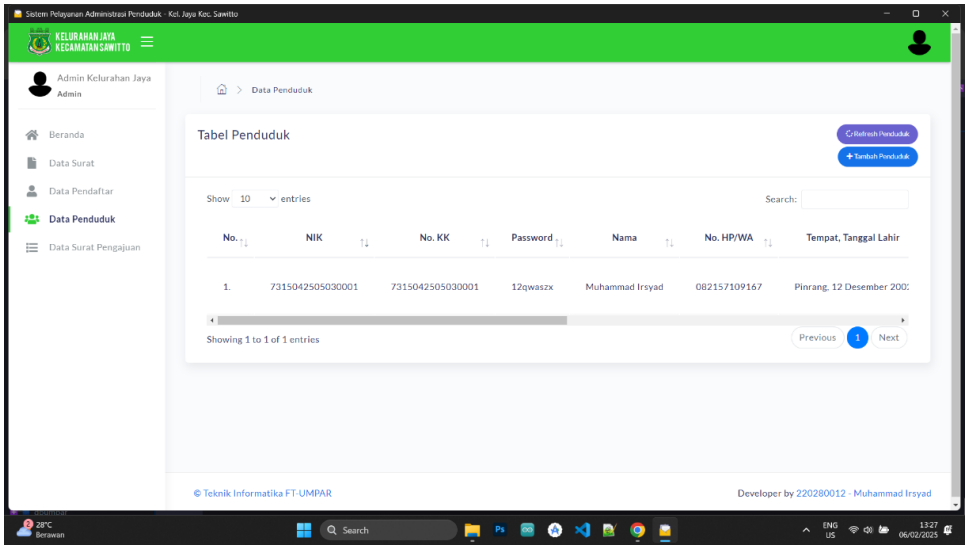
Output yang baik dalam sistem informasi *admin*istrasi, dimana data disajikan secara informatif dan mudah diakses oleh pengguna sistem.



Gambar 4.13 Pendaftar Sudah Diverifikasi.

9. Halaman Data Penduduk Yang Sudah Diverifikasi

Gambar ini menunjukkan halaman Data Penduduk Yang Sudah Diverifikasi dari sebuah sistem pelayanan *admin*istrasi, serta memudahkan *admin*istrator untuk memperbarui data penduduk yang telah diverifikasi atau menambah data baru.



Gambar 4.14 Halaman Data Penduduk Yang Sudah Diverifikasi

10. Halaman Utama Aplikasi Sispaduk

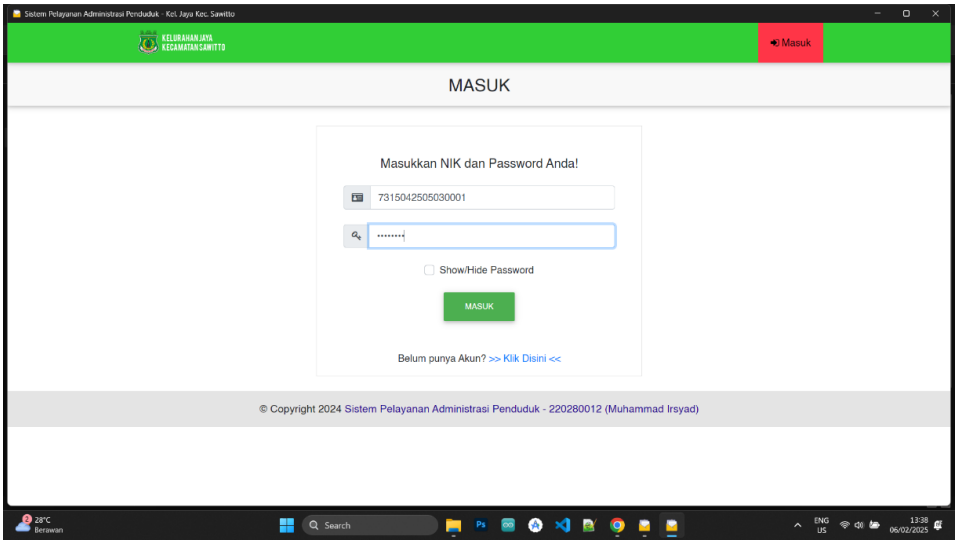
Halaman utama ini dirancang untuk memberikan kemudahan akses kepada pengguna dalam melakukan pengecekan data NIK dan mengunduh berbagai jenis surat yang diperlukan secara langsung dari halaman tersebut. Desain antarmuka yang sederhana dan intuitif memudahkan pengguna untuk menemukan dan menggunakan fitur yang tersedia dalam aplikasi Sispaduk.



Gambar 4.15 Halaman Utama Aplikasi Sispaduk

11. Halaman *Login User*

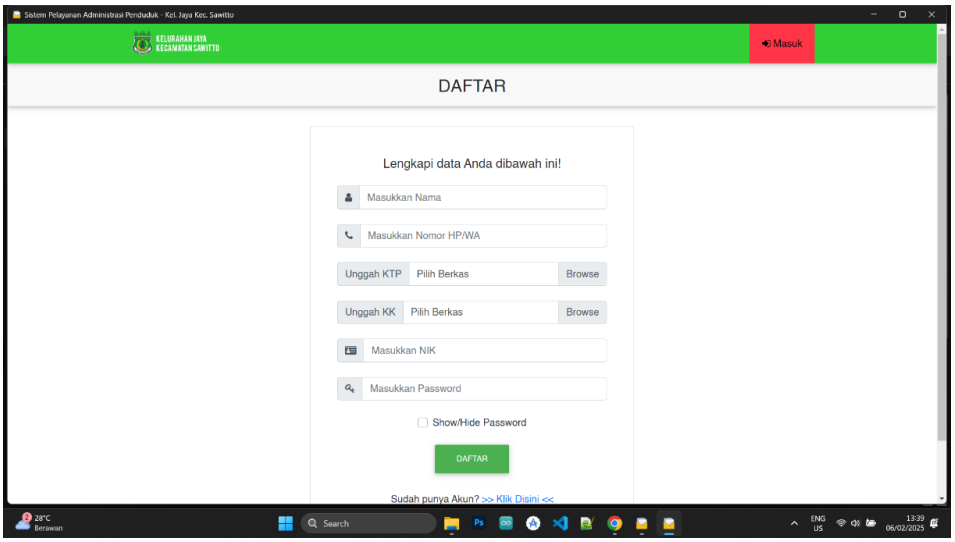
Pada halaman ini, pengguna diminta untuk memasukkan Nomor Induk Kependudukan (NIK) dan kata sandi mereka untuk mengakses sistem. Dan sistem ini dirancang untuk menyediakan akses yang mudah bagi pengguna yang ingin mengakses layanan *admin*istrasi penduduk di Kelurahan Jaya.



Gambar 4.16 Halaman *Login User*

12. Halaman *Register User*

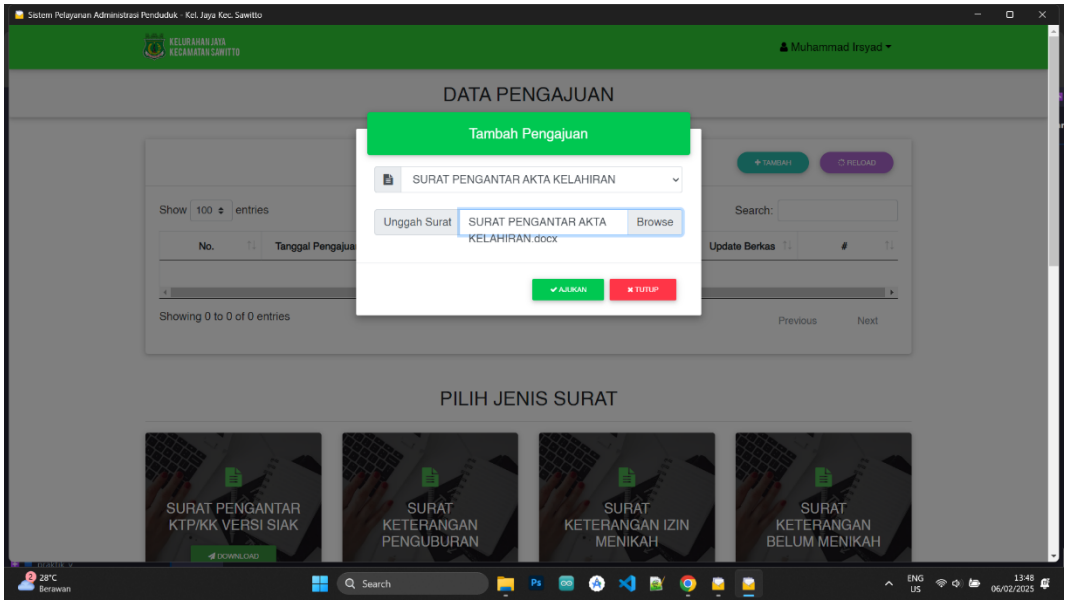
Halaman ini berisi formulir yang harus diisi oleh pengguna untuk mendaftar ke dalam sistem. Dan sistem ini dirancang untuk memudahkan pengguna baru dalam mendaftar ke dalam sistem Sispaduk, memastikan bahwa data yang dimasukkan lengkap dan valid sebelum pengguna dapat mengakses layanan *admin*istrasi penduduk yang disediakan.



Gambar 4.17 Halaman *Register User.*

13. Halaman Pengajuan Surat

Gambar di bawah ini menunjukkan bagian **Form Pengajuan Surat** dari sistem pelayanan administrasi penduduk di Kelurahan Jaya, Kecamatan Sawitto. Formulir ini memungkinkan pengguna untuk mengajukan berbagai jenis surat, seperti **Surat Pengantar Akta Kelahiran**. Pengguna dapat memilih jenis surat yang akan diajukan, mengunggah *File* dokumen yang relevan, dan mengajukan permohonan dengan menekan tombol **"Ajukan"**. Antarmuka ini dirancang untuk memudahkan proses pengajuan surat dengan menyediakan opsi yang jelas dan langkah-langkah yang sederhana.

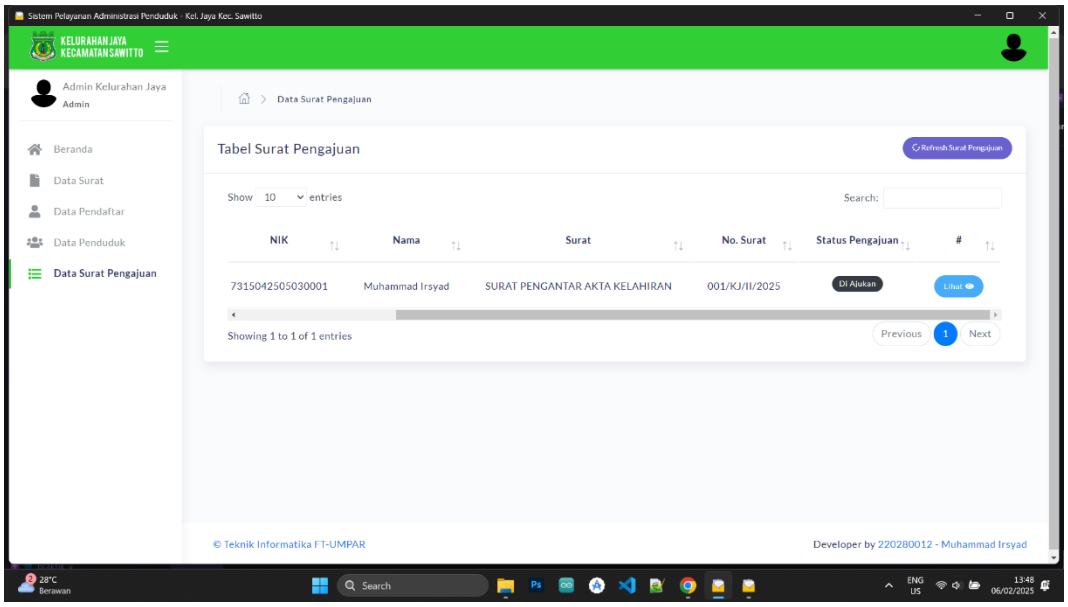


Gambar 4.18 Halaman Pengajuan Surat

14. Halaman Data Surat Pengajuan

Gambar tersebut menunjukkan tabel pada **Halaman Pengajuan Surat** dalam sistem pelayanan administrasi penduduk. Tabel ini menampilkan data pengajuan surat, termasuk **NIK (Nomor Induk Kependudukan)**, **Nama**, **Surat yang Diajukan**, **Nomor Surat**, dan **Status Pengajuan**. Contoh data dalam tabel

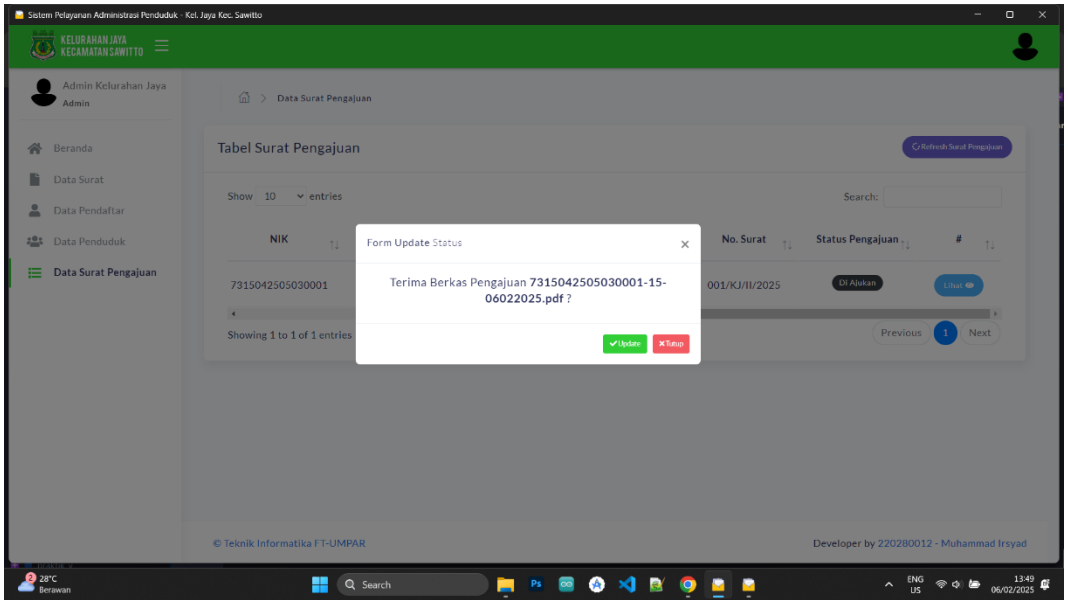
mencakup NIK 7315042505030001, nama Muhammad Irsyad, dan pengajuan "Surat Pengantar Akta Kelahiran" dengan status "Diajukan".



Gambar 4.19 Halaman Data Surat Pengajuan

15. Verifikasi Surat Yang Diajukan

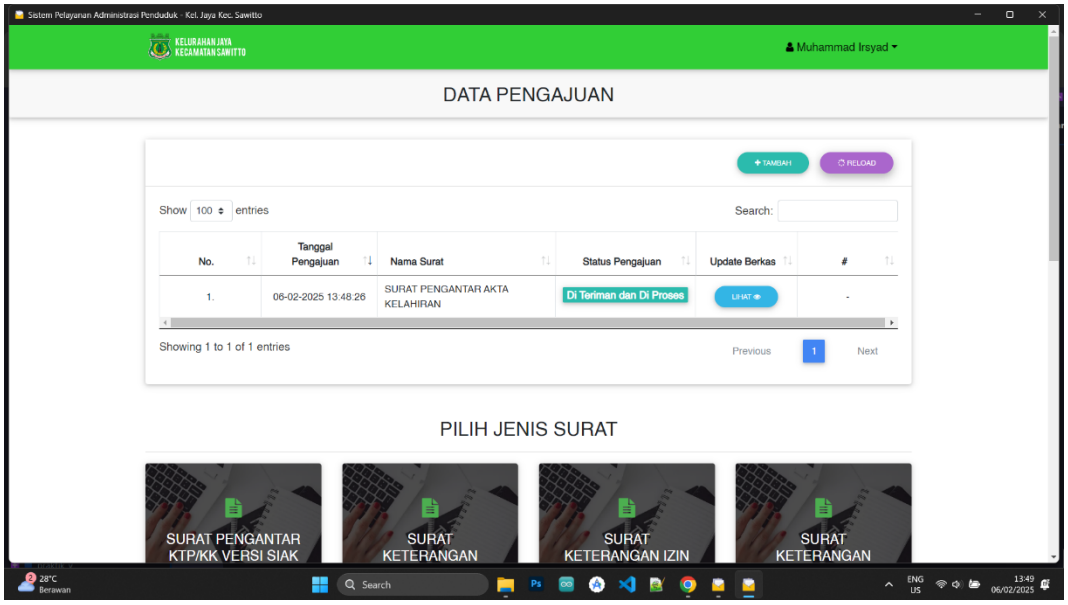
Pada gambar dibawah ini terlihat jendela pop-up dengan judul **"Form Update Status"**, yang menampilkan pertanyaan terkait penerimaan berkas pengajuan dengan nomor NIK 7315042505030001 dan nomor berkas 15-06022025.pdf. Form Verifikasi ini dirancang untuk memudahkan *administrator* dalam memverifikasi dan memperbarui status pengajuan surat secara efisien, memastikan bahwa setiap langkah dalam proses administrasi dapat dilacak dan diatur dengan tepat.



Gambar 4.20 Verifikasi Surat Yang Diajukan

16. Cek Status Surat Pengajuan Yang Diverifikasi

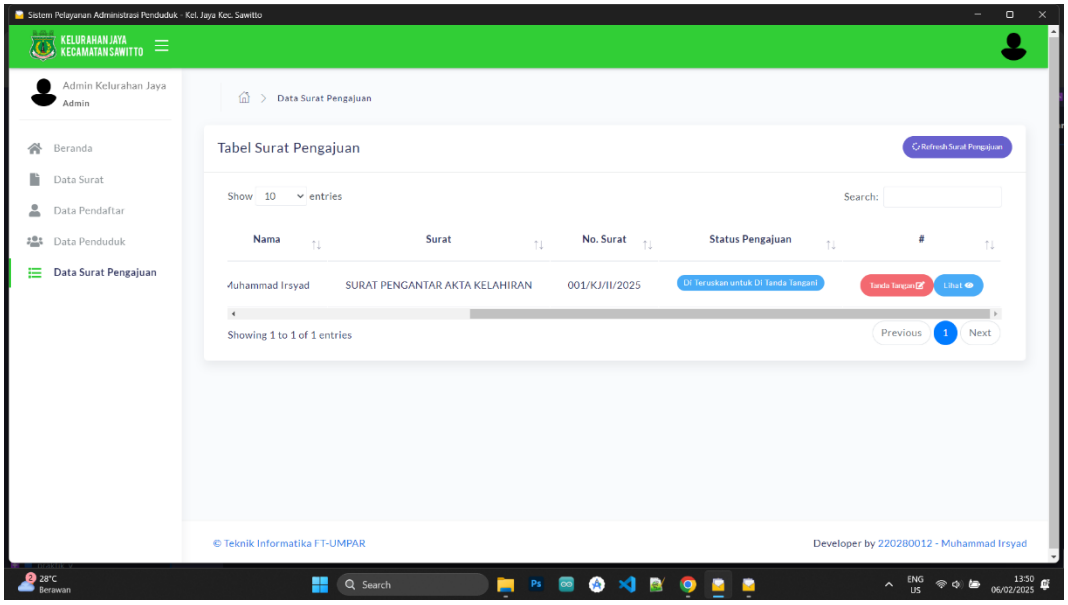
Pada gambar menampilkan data pengajuan surat dengan satu entri yang berisi informasi tanggal pengajuan, nama surat, status pengajuan, dan opsi untuk memperbarui berkas. Surat yang diajukan adalah **"SURAT PENGANTAR AKTA KELAHIRAN"** dengan status **"Di Terima dan Di Proses"**, yang ditampilkan dalam teks berwarna hijau. Status ini menunjukkan bahwa pengajuan telah diverifikasi dan sedang dalam proses lebih lanjut. Selain itu, terdapat tombol "LIHAT" pada kolom "Update Berkas" yang dapat digunakan untuk melihat detail dokumen.



Gambar 4.21 Cek Status Surat Pengajuan Yang Diverifikasi

17. Status Surat Diteruskan Untuk Ditanda Tangani

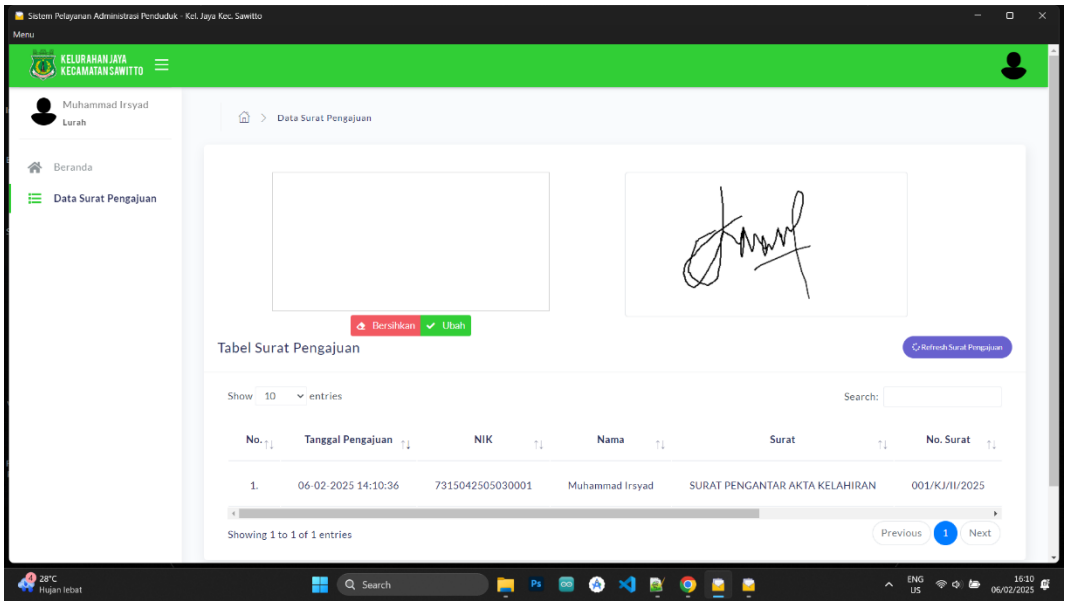
Gambar di atas menampilkan tabel yang berisi data pengajuan surat, khususnya **"SURAT PENGANTAR AKTA KELAHIRAN"** yang diajukan oleh Muhammad Irayad dengan nomor surat O01/KJ/II/2025. Status pengajuan surat ini menunjukkan bahwa surat telah diteruskan untuk ditandatangani oleh Lurah. Ini mengindikasikan bahwa surat tersebut telah diproses oleh admin dan sedang dalam tahap menunggu penandatanganan.



Gambar 4.22 Status Surat Diteruskan Untuk Ditanda Tangani

18. Halaman Tanda Tangan Lurah

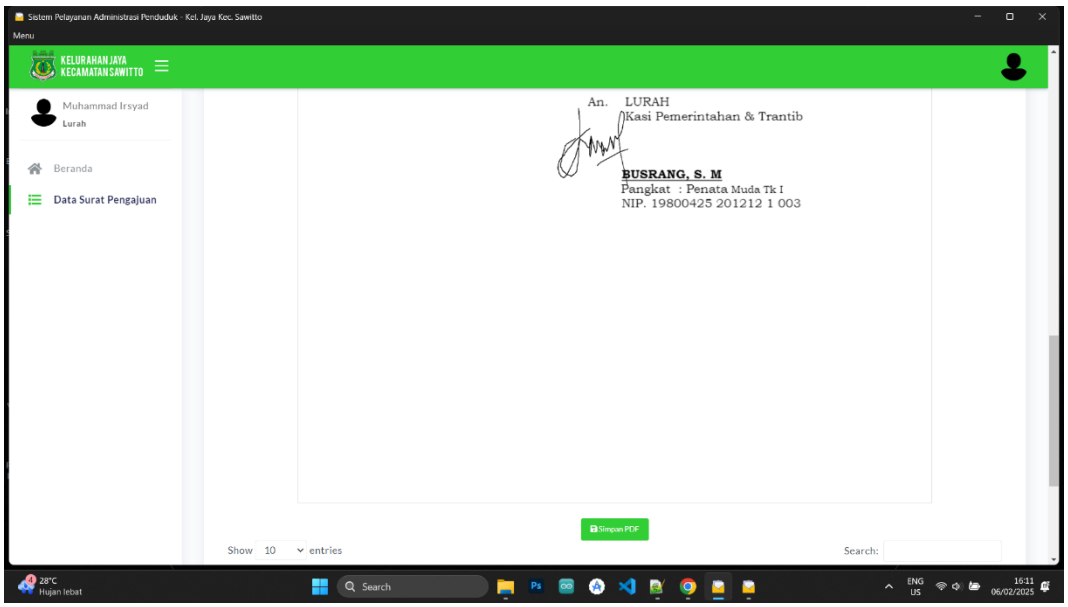
Gambar ini menunjukkan bahwa surat tersebut telah sampai pada tahap "Halaman Tanda Tangan Lurah", yang berarti surat sedang menunggu untuk ditandatangani oleh lurah.



Gambar 4.23 Halaman Tanda Tangan Lurah

19. Form Tanda Tangan Lurah

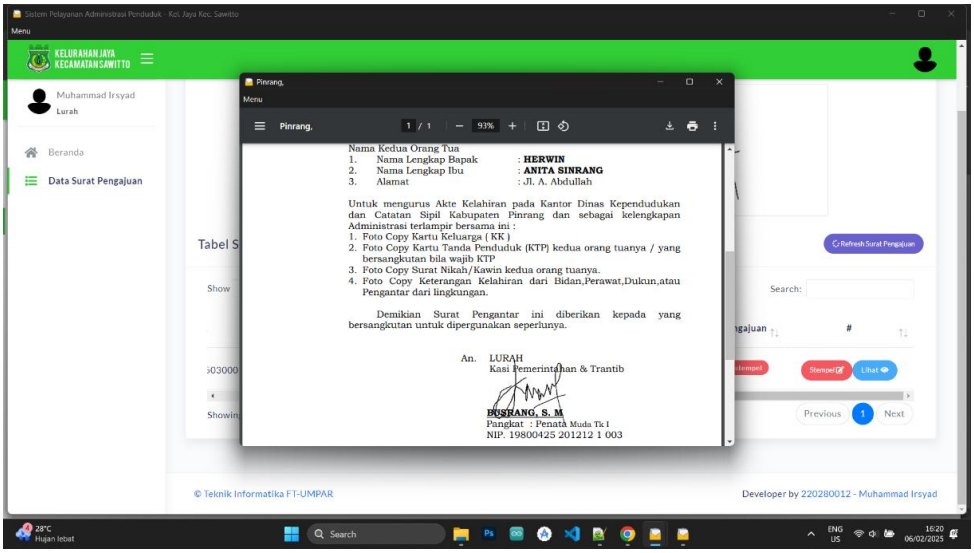
Pada gambar di bawah ini Form Tanda Tangan Lurah pada gambar tersebut mencakup informasi nama lurah, nama kasi pemerintahan & trantib, serta tanda tangan keduanya. Form ini juga menyertakan informasi pangkat dan NIP lurah. Di bagian bawah terdapat opsi untuk menyimpan dokumen dalam format PDF.



Gambar 4.24 Form Tanda Tangan Lurah

20. Detail Surat Yang Telah Ditanda Tangani

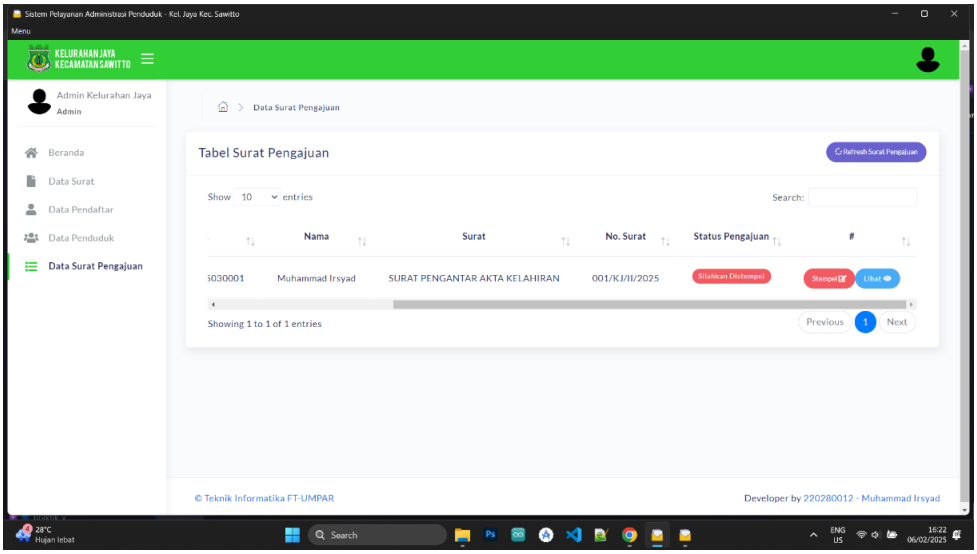
Gambar di bawah ini menampilkan detail surat yang telah ditandatangani, khususnya Surat Pengantar untuk mengurus Akta Kelahiran. Surat ini menjelaskan persyaratan dokumen yang diperlukan. Surat ini juga telah ditandatangani langsung oleh Lurah setempat.



Gambar 4.25 Detail Tanda Tangan Lurah

21. Status Untuk Bubuhi Stempel

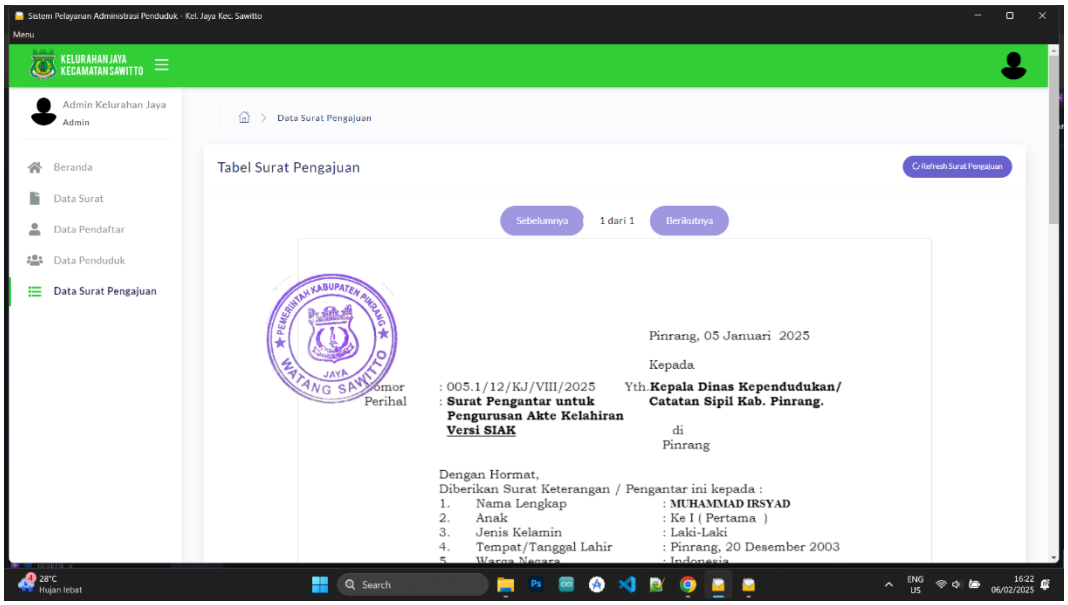
Pada gambar di bawah ini menunjukkan bahwa surat tersebut sedang dalam proses Cek Surat, yang mengindikasikan bahwa surat sedang diperiksa atau diverifikasi sebelum dibubuhi stempel. Status ini menandakan bahwa surat telah melalui beberapa tahap pengajuan dan sedang menunggu untuk diselesaikan dengan pembubuhan stempel resmi.



Gambar 4.26 Bubuhi Stempel Kelurahan

22. Form Bubuhi Stempel Kelurahan

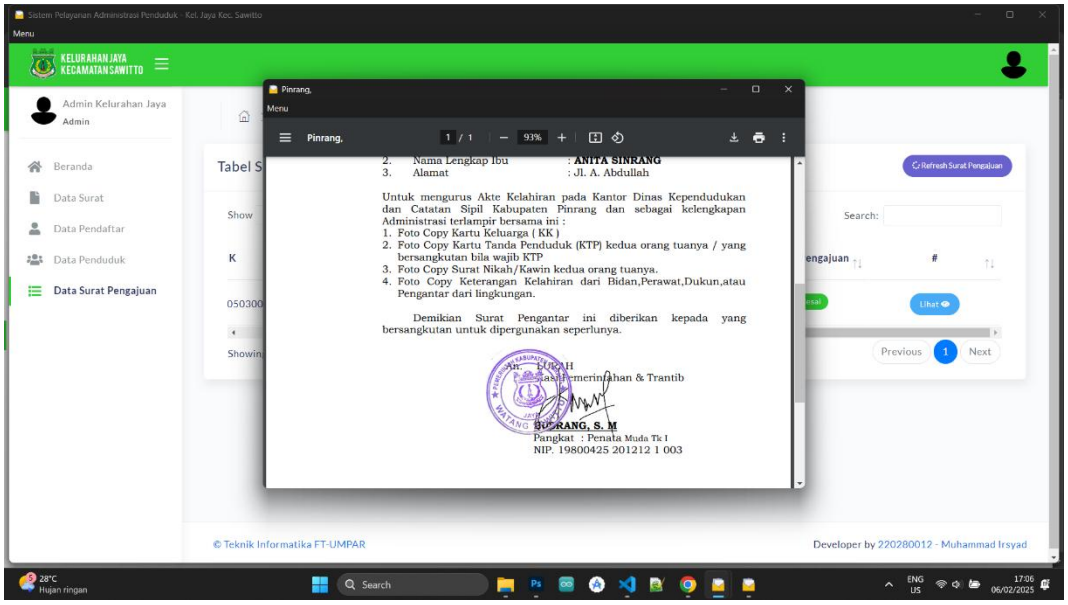
Gambar di bawah ini menunjukkan formulir yang siap untuk dibubuhi stempel resmi sebagai bagian dari proses pengesahan surat pengantar akta kelahiran.



Gambar 4.27 Form Bubuhi Stempel Kelurahan

23. Detail Surat Yang Telah Ditanda Tangani Dan Dibubuhi Stempel

Gambar tersebut menunjukkan Surat Pengantar yang sudah ditandatangani dan distempel. Surat ini digunakan untuk mengurus Akta Kelahiran. Tanda tangan dan stempel resmi menunjukkan bahwa surat ini telah diverifikasi dan disetujui oleh pihak berwenang, siap digunakan untuk keperluan administrasi.



Gambar 4.28 Detail Surat Yang Telah Ditanda Tangani Dan Dibubuhi Stempel

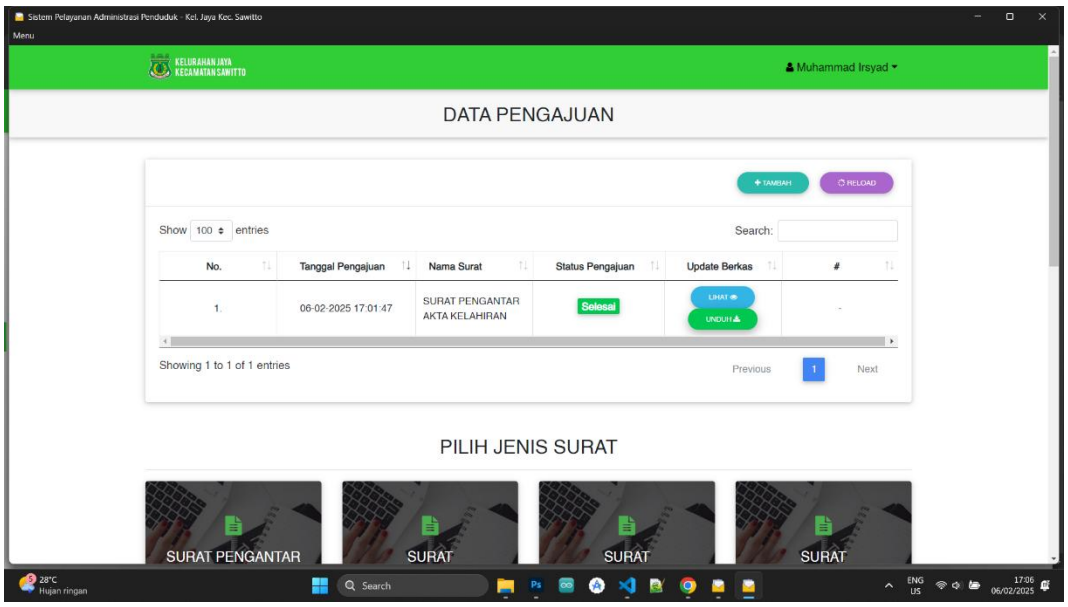
24. Lihat Atau Unduh Surat

Gambar dibawah ini menampilkan antarmuka sistem yang memungkinkan pengguna untuk melihat atau mengunduh surat yang telah selesai diproses. Berikut penjelasan lebih detail:

1. Kolom "Update Berkas": Di kolom ini, terdapat tombol "LIHAT" dan "UNDUH" yang memungkinkan pengguna untuk:
    - LIHAT: Pengguna dapat melihat langsung surat yang telah selesai diproses tanpa perlu meninggalkan sistem.
    - UNDUH: Pengguna dapat mengunduh surat tersebut dalam format *File* PDF untuk disimpan atau dicetak sesuai kebutuhan.
2. Kemudahan Akses: Dengan adanya tombol ini, pengguna tidak perlu melalui proses yang rumit untuk mengakses dokumen. Mereka dapat

langsung melihat atau mengunduh surat yang telah selesai diproses hanya dengan satu klik.

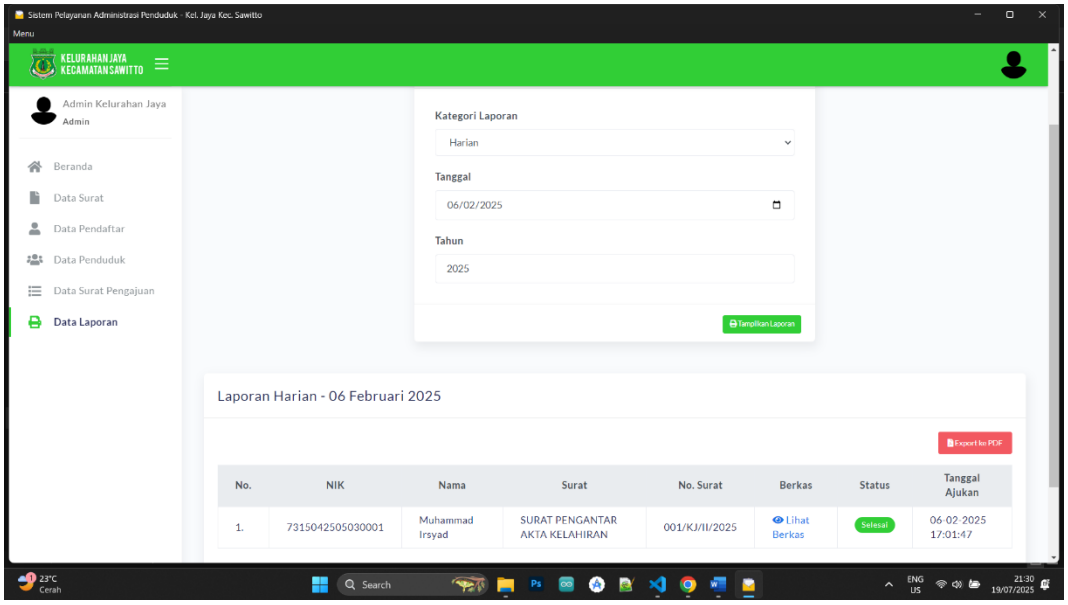
Secara keseluruhan, gambar ini menunjukkan bahwa sistem dirancang untuk memudahkan pengguna dalam mengakses dokumen yang telah selesai diproses, baik untuk dilihat langsung maupun diunduh, melalui tombol yang tersedia di kolom "Update Berkas".



Gambar 4.29 Lihat Atau Unduh Surat

25. Laporan Harian Pengajuan Surat

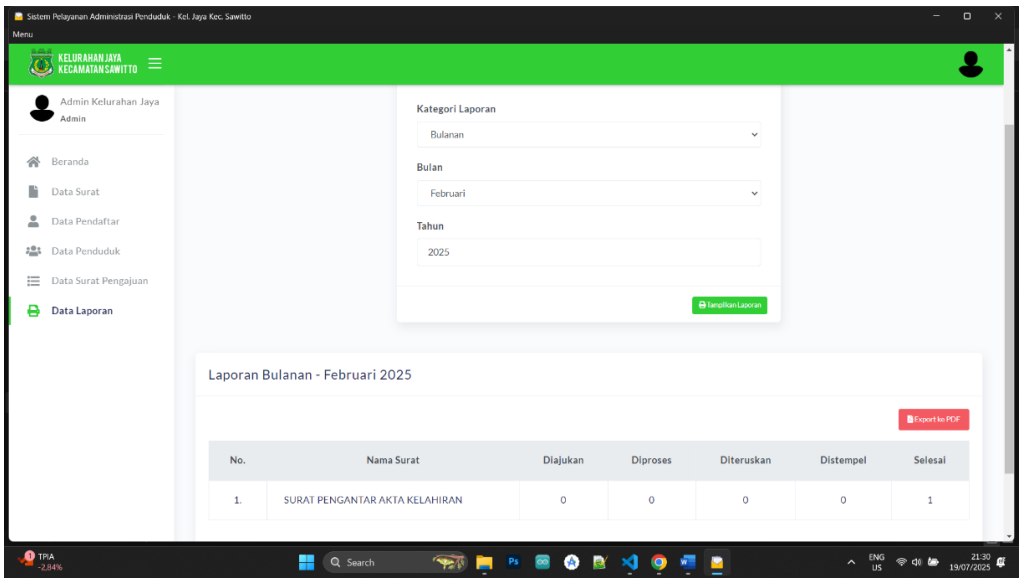
Gambar di bawah ini menunjukkan tampilan laporan harian pengajuan surat yang menampilkan data pemohon, jenis surat, status, serta tanggal pengajuan, dan dilengkapi fitur ekspor ke PDF sebagai dokumentasi resmi.



Gambar 4.30 Laporan Harian Pengajuan Surat

26. Laporan Bulanan Pengajuan Surat

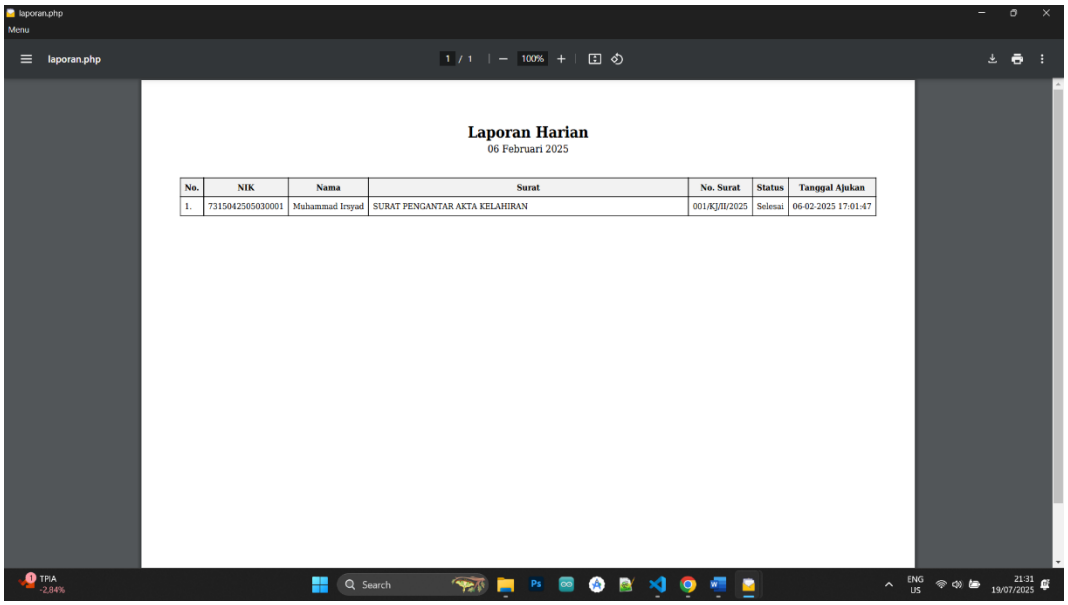
Gambar di bawah ini menunjukkan tampilan laporan bulanan pengajuan surat yang merekap jumlah surat berdasarkan status proses, seperti diajukan, diproses, diteruskan, distempel, dan selesai, serta dilengkapi fitur ekspor ke PDF untuk keperluan dokumentasi bulanan.



Gambar 4.31 Laporan Bulanan Pengajuan Surat

27. Cetak Laporan dalam bentuk PDF

Gambar di bawah ini menunjukkan hasil cetak laporan harian dalam format PDF yang memuat data pengajuan surat, lengkap dengan informasi NIK, nama pemohon, jenis dan nomor surat, status, serta waktu pengajuan.



Gambar 4.32 Cetak Laporan dalam bentuk PDF

B. Pengujian Sistem

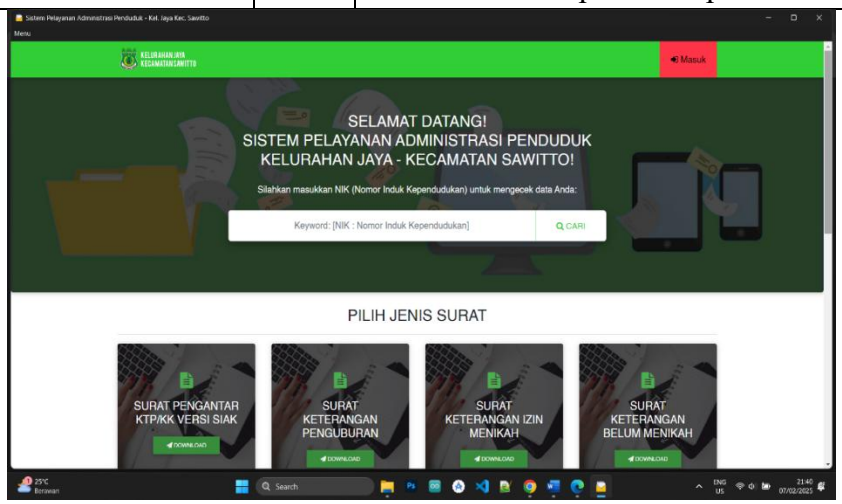
1. Pengujian *Black Box*

Pengujian *Black Box* adalah metode pengujian perangkat lunak yang berfokus pada validasi fungsionalitas aplikasi tanpa memperhatikan struktur internal atau implementasi dari program tersebut. Penguji hanya memeriksa *input* dan *Output* dari sistem untuk memastikan bahwa program berjalan sesuai dengan spesifikasi yang diinginkan.

1) Halaman Beranda

Tabel 4.1 Pengujian Halaman Beranda

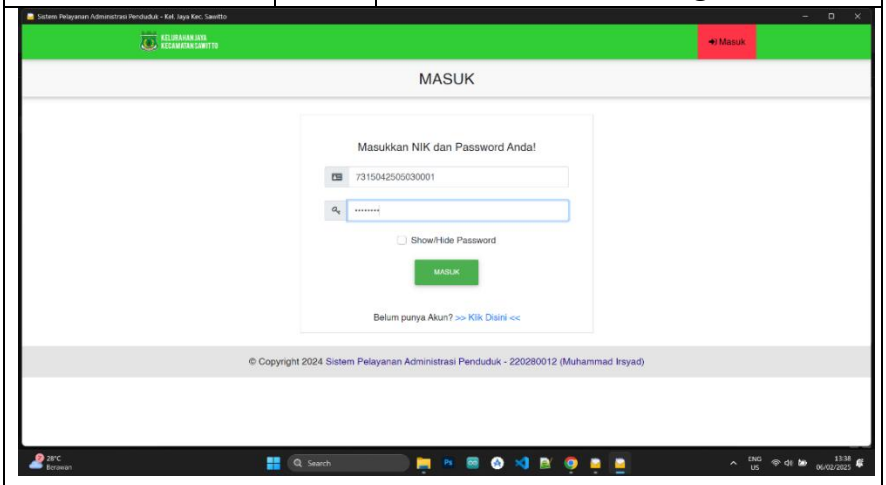
Tes Faktor	Hasil	Keterangan
Halaman Beranda	✓	Sukses pada saat *User* / pengguna membuka aplikasi sispaduk



2) Halaman *Login*

Tabel 4.2 Pengujian Halaman *Login*

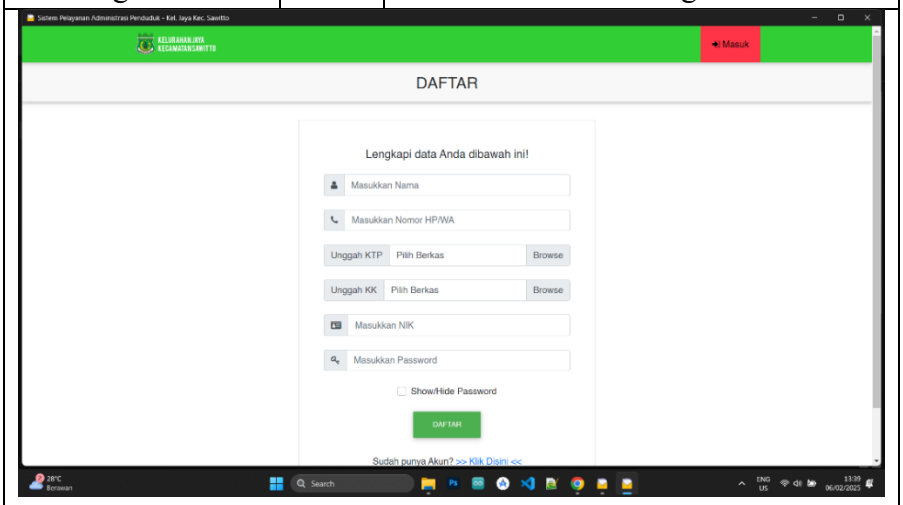
Tes Faktor	Hasil	Keterangan
Halaman *Login*	✓	Sukses pada saat *User* / pengguna melakukan *login*



3) Halaman Registrasi

Tabel 4.3 Pengujian Halaman Registrasi

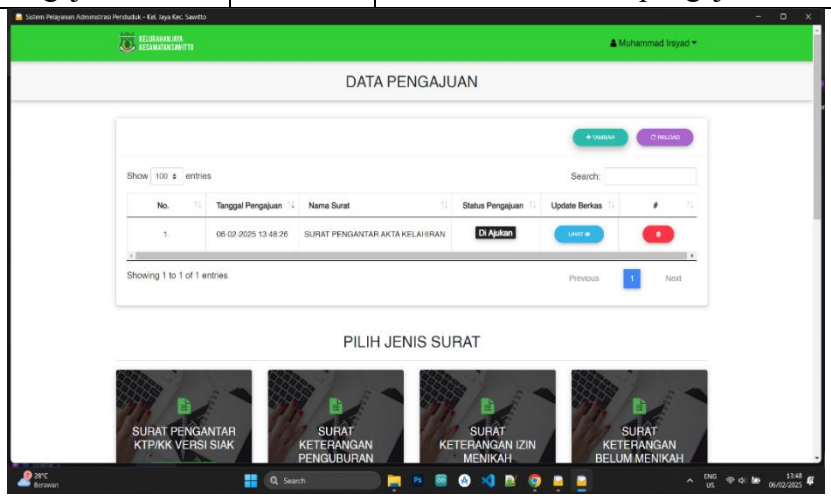
Tes Faktor	Hasil	Keterangan
Halaman Registrasi	✓	Sukses pada saat *User* / pengguna melakukan registrasi



4) Halaman Pengajuan Surat

Tabel 4.4 Pengujian Halaman Pengajuan Surat

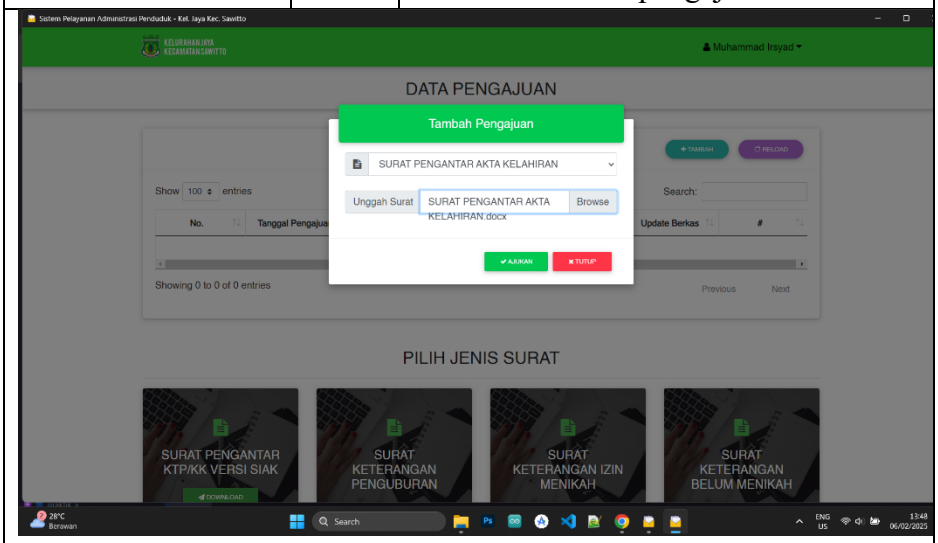
Tes Faktor	Hasil	Keterangan
Halaman Pengajuan Surat	✓	Sukses pada saat *User* / pengguna masuk ke halaman pengajuan surat



5) *Form* Pengajuan Surat

Tabel 4.5 Pengujian *Form* Pengajuan Surat

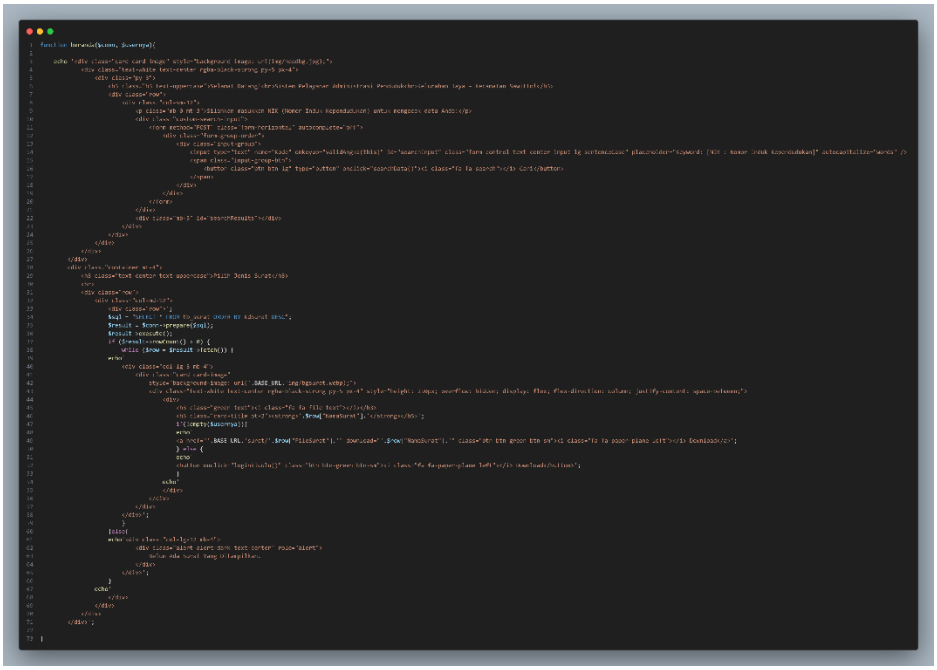
Tes Faktor	Hasil	Keterangan
Halaman Pengajuan Surat	✓	Sukses pada saat *User*/ pengguna melakukan pengajuan surat



2. Pengujian *White Box*

Pengujian *White Box*, juga dikenal sebagai pengujian kode atau pengujian struktur, adalah metode pengujian perangkat lunak yang memeriksa detail internal dan logika program. Penguji memiliki pengetahuan tentang struktur kode dan algoritma yang digunakan dalam aplikasi. Metode ini melibatkan pengujian jalur, cabang, kondisi, dan aliran data dalam program. Tujuan utama dari pengujian *White Box* adalah untuk memastikan bahwa semua jalur kode berfungsi dengan benar dan tidak ada kesalahan logika atau perhitungan dalam program.

1) Halaman Beranda



Gambar 4.33 *Source Code* Halaman Beranda

a) Pengujian Validasi Input NIK:

- Tujuan: Memastikan bahwa input NIK hanya menerima angka dan memiliki panjang 16 digit.
- Langkah Pengujian:
  1. Masukkan NIK dengan karakter non-numerik (misalnya, "73150425abcd0001").
  2. Masukkan NIK dengan panjang kurang dari 16 digit (misalnya, "731504250503000").
  3. Masukkan NIK dengan panjang lebih dari 16 digit (misalnya, "73150425050300011").
  4. Masukkan NIK dengan panjang tepat 16 digit (misalnya, "7315042505030001").

- Hasil yang Diharapkan:
    1. Langkah 1, 2, dan 3: Sistem menampilkan pesan kesalahan dan tidak mengizinkan pengguna untuk melanjutkan.
    2. Langkah 4: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

b) Pengujian Pencarian Data NIK:

- Tujuan: Memastikan bahwa sistem dapat mencari dan menampilkan data pengguna berdasarkan NIK yang valid.
- Langkah Pengujian:
    1. Masukkan NIK yang valid dan klik tombol "Cari".
    2. Masukkan NIK yang tidak valid dan klik tombol "Cari".
    3. Biarkan kolom NIK kosong dan klik tombol "Cari".
- Hasil yang Diharapkan:
    1. Langkah 1: Sistem menampilkan data pengguna yang sesuai dengan NIK yang dimasukkan.
    2. Langkah 2: Sistem menampilkan pesan kesalahan bahwa NIK tidak ditemukan.
    3. Langkah 3: Sistem menampilkan pesan kesalahan bahwa NIK harus diisi.

c) Pengujian Fungsi Tombol MASUK:

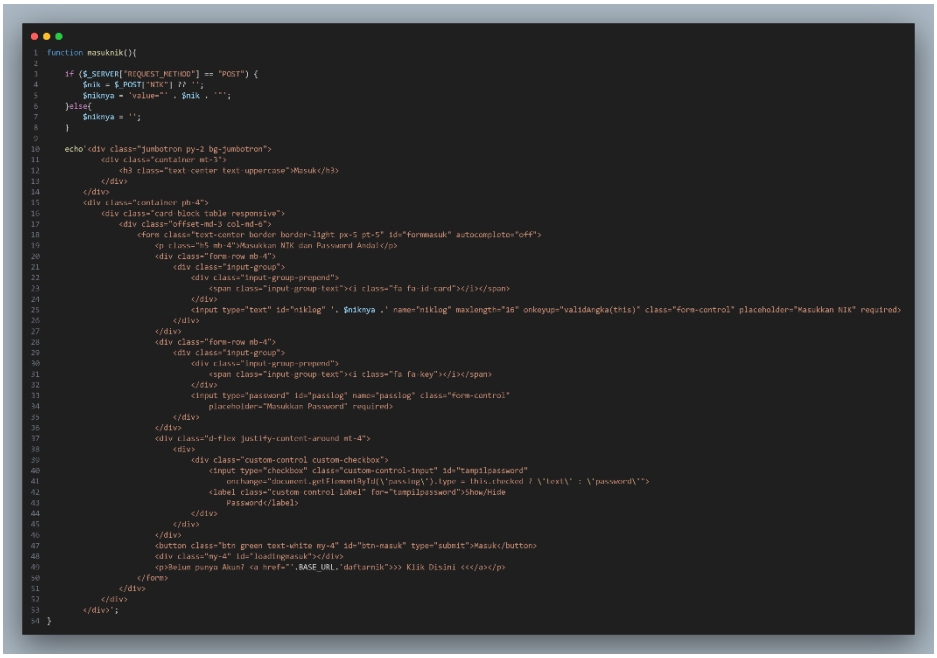
- Tujuan: Memastikan bahwa tombol MASUK berfungsi dengan benar dan mengarahkan pengguna ke halaman *login*.
- Langkah Pengujian:

1. Klik tombol "Masuk" di pojok kanan atas halaman.

- Hasil yang Diharapkan:

1. Sistem mengarahkan pengguna ke halaman *login*.

2) Halaman *Login*



```
function masuknik(){

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $nik = $_POST['NIK'] ?? '';
        $niknya = 'value="' . $nik . '"';
    }else{
        $niknya = '';
    }

    echo'<div class="jumbotron py-2 bg-jumbotron">
            <div class="container mt-3">
                <h3 class="text-center text-uppercase">Masuk</h3>
            </div>
        </div>
        <div class="container ph-4">
            <div class="card-block table responsive">
                <div class="offset-md-3 col-md-6">
                    <form class="text-center border border-light px-5 pt-5" id="formmasuk" autocomplete="off">
                        <p class="h5 mb-4">Masukkan NIK dan Password Anda!</p>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input group text"><i class="fa fa-id-card"></i></span>
                                </div>
                                <input type="text" id="niklog" '. $niknya .' name="niklog" maxlength="16" onkeyup="validangka(this)" class="form-control" placeholder="Masukkan NIK" required>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input-group-prepend">
                                    <span class="input group text"><i class="fa fa-key"></i></span>
                                </div>
                                <input type="password" id="passlog" name="passlog" class="form-control"
                                    placeholder="Masukkan Password" required>
                            </div>
                        </div>
                        <div class="d-flex justify-content-around mt-4">
                            <div>
                                <div class="custom-control custom-checkbox">
                                    <input type="checkbox" class="custom-control-input" id="tampilpassword"
                                        onchange="document.getElementById(\'passlog\').type = this.checked ? \'text\' : \'password\'">
                                    <label class="custom control label" for="tampilpassword">Show/Hide
                                        Password</label>
                                </div>
                            </div>
                        </div>
                        <button class="btn green text-white my-4" id="btn-masuk" type="submit">Masuk</button>
                        <div class="my-4" id="loadingmasuk"></div>
                        <p>Belum punya Akun? <a href="'.BASE_URL.'daftarnik">>> Klik Disini <<</a></p>
                    </form>
                </div>
            </div>
        </div>';
}
```

Gambar 4.34 *Source Code* Halaman *Login*

a) Pengujian Validasi Input NIK:

- Tujuan: Memastikan bahwa input NIK hanya menerima angka dan memiliki panjang 16 digit.
- Langkah Pengujian:

1. Masukkan NIK dengan karakter non-numerik (misalnya, " 731504250503000a").
2. Masukkan NIK dengan panjang kurang dari 16 digit (misalnya, " 731504250503000").

3. Masukkan NIK dengan panjang lebih dari 16 digit (misalnya, "73150425050300010").
4. Masukkan NIK dengan panjang tepat 16 digit (misalnya, "7315042505030001").

- Hasil yang Diharapkan:
  1. Langkah 1, 2, dan 3: Sistem menampilkan pesan kesalahan dan tidak mengizinkan pengguna untuk melanjutkan.
  2. Langkah 4: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

b) Pengujian Validasi Input *Password*:

- Tujuan: Memastikan bahwa input *password* tidak kosong dan memiliki panjang minimal tertentu (misalnya, 8 karakter).
- Langkah Pengujian:
  1. Masukkan *password* kosong.
  2. Masukkan *password* dengan panjang kurang dari 8 karakter (misalnya, "pass").
  3. Masukkan *password* dengan panjang lebih dari atau sama dengan 8 karakter (misalnya, "password123").
- Hasil yang Diharapkan:
  1. Langkah 1 dan 2: Sistem menampilkan pesan kesalahan dan tidak mengizinkan pengguna untuk melanjutkan.
  2. Langkah 3: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

c) Pengujian Fungsi *Show/Hide Password*:

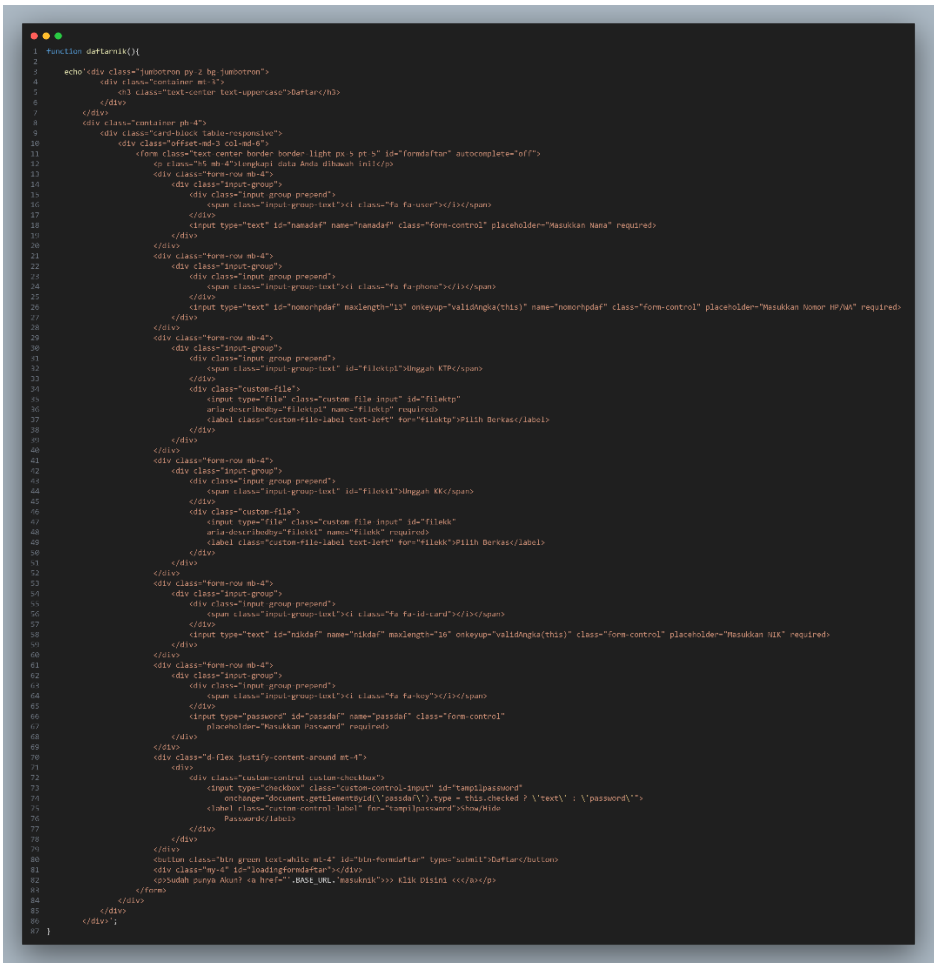
- Tujuan: Memastikan bahwa fungsi *show/hide password* bekerja dengan benar.
- Langkah Pengujian:
  1. Masukkan *password* (misalnya, "password123").
  2. Centang kotak "*Show/Hide Password*" untuk menampilkan *password*.
  3. Hapus centang kotak "*Show/Hide Password*" untuk menyembunyikan *password*.
- Hasil yang Diharapkan:
  1. Langkah 2: *Password* ditampilkan dalam bentuk teks biasa.
  2. Langkah 3: *Password* kembali disembunyikan dengan karakter dot.

d) Pengujian Tombol MASUK:

- Tujuan: Memastikan bahwa tombol MASUK berfungsi dengan benar dan mengarahkan pengguna ke halaman berikutnya jika input valid.
- Langkah Pengujian:
  1. Masukkan NIK dan *password* yang valid.
  2. Klik tombol MASUK.
- Hasil yang Diharapkan:

1. Sistem memverifikasi input dan mengarahkan pengguna ke halaman berikutnya jika input valid. Jika input tidak valid, sistem menampilkan pesan kesalahan.

3) Halaman Registrasi



```
function daftarnik(){

    echo'<div class="jumbotron py-2 bg-jumbotron">
            <div class="container mt-4">
                <h3 class="text-center text-uppercase">Daftar</h3>
            </div>
        </div>
        <div class="container ph-4">
            <div class="card-block table-responsive">
                <div class="offset-md-3 col-md-6">
                    <form class="text center border border light px-5 pt-5" id="formdaftar" autocomplete="off">
                        <p class="h5 mb-4">Lengkapi data Anda dibawah ini!</p>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text"><i class="fa fa-user"></i></span>
                                </div>
                                <input type="text" id="namadaf" name="namadaf" class="form-control" placeholder="Masukkan Nama" required>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text"><i class="fa fa-phone"></i></span>
                                </div>
                                <input type="text" id="nomorhpdaf" maxlength="13" onkeyup="validAngka(this)" name="nomorhpdaf" class="form-control" placeholder="Masukkan Nomor HP/WA" required>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text" id="filektp1">Unggah KTP</span>
                                </div>
                                <div class="custom-file">
                                    <input type="file" class="custom file input" id="filektp"
                                    aria-describedby="filektp1" name="filektp" required>
                                    <label class="custom-filo-label text-left" for="filektp">Pilih Berkas</label>
                                </div>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text" id="filekk1">Unggah KK</span>
                                </div>
                                <div class="custom-file">
                                    <input type="file" class="custom file input" id="filekk"
                                    aria-describedby="filekk1" name="filekk" required>
                                    <label class="custom-filo-label text-left" for="filekk">Pilih Berkas</label>
                                </div>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text"><i class="fa fa-id-card"></i></span>
                                </div>
                                <input type="text" id="nikdaf" name="nikdaf" maxlength="16" onkeyup="validAngka(this)" class="form-control" placeholder="Masukkan NIK" required>
                            </div>
                        </div>
                        <div class="form-row mb-4">
                            <div class="input-group">
                                <div class="input group prepend">
                                    <span class="input-group-text"><i class="fa fa-key"></i></span>
                                </div>
                                <input type="password" id="passdaf" name="passdaf" class="form-control"
                                    placeholder="Masukkan Password" required>
                            </div>
                        </div>
                        <div class="d-flex justify-content-around mt-4">
                            <div>
                                <div class="custom-control custom-checkbox">
                                    <input type="checkbox" class="custom-control-input" id="tampilpassword"
                                        onchange="document.getElementById(\'passdaf\').type = this.checked ? \'text\' : \'password\'">
                                    <label class="custom-control-label" for="tampilpassword">Show/Hide
                                        Password</label>
                                </div>
                            </div>
                        </div>
                        <button class="btn green text-white mt-4" id="btn-formdaftar" type="submit">Daftar</button>
                        <div class="my-4" id="loadingformdaftar"></div>
                        <p>Sudah punya Akun? <a href="'.BASE_URL.'masuknik">>>> Klik Disini <<</b></p>
                    </form>
                </div>
            </div>
        </div>';
}
```

Gambar 4.35 Source Code Halaman Registrasi

d) Pengujian Validasi Input Nama:

- Tujuan: Memastikan bahwa input Nama tidak kosong.
- Langkah Pengujian:

1. Masukkan nama kosong.
2. Masukkan nama yang valid (misalnya, "Muhammad Irsyad").

- Hasil yang Diharapkan:
    1. Langkah 1: Sistem menampilkan pesan kesalahan yang menunjukkan bahwa nama harus diisi.
    2. Langkah 2: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

e) Pengujian Validasi Input Nomor HP/WA:

- Tujuan: Memastikan bahwa input Nomor HP/WA berformat valid.
- Langkah Pengujian:
    1. Masukkan nomor HP/WA dengan format yang tidak valid (misalnya, "12345").
    2. Masukkan nomor HP/WA dengan format yang valid (misalnya, "08123456789").
- Hasil yang Diharapkan:
    1. Langkah 1: Sistem menampilkan pesan kesalahan yang menunjukkan bahwa format nomor HP/WA tidak valid.
    2. Langkah 2: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

f) Pengujian Unggah KTP dan KK:

- Tujuan: Memastikan bahwa *File* KTP dan KK diunggah dengan format yang benar.
- Langkah Pengujian:
    1. Unggah *File* KTP dengan format yang tidak valid (misalnya, *File* .txt).

2. Unggah *File* KK dengan format yang tidak valid (misalnya, *File* .txt).
3. Unggah *File* KTP dan KK dengan format yang valid (misalnya, *File* .pdf atau .jpg).

- Hasil yang Diharapkan:
  1. Langkah 1 dan 2: Sistem menampilkan pesan kesalahan yang menunjukkan bahwa format *File* tidak valid.
  2. Langkah 3: Sistem menerima *File* dan mengizinkan pengguna untuk melanjutkan.

g) Pengujian Validasi Input NIK:

- Tujuan: Memastikan bahwa input NIK hanya menerima angka dan memiliki panjang 16 digit.
- Langkah Pengujian:
  1. Masukkan NIK dengan karakter non-numerik (misalnya, "73150425abcd0001").
  2. Masukkan NIK dengan panjang kurang dari 16 digit (misalnya, "731504250503000").
  3. Masukkan NIK dengan panjang lebih dari 16 digit (misalnya, "73150425050300011").
  4. Masukkan NIK dengan panjang tepat 16 digit (misalnya, "7315042505030001").
- Hasil yang Diharapkan:

1. Langkah 1, 2, dan 3: Sistem menampilkan pesan kesalahan dan tidak mengizinkan pengguna untuk melanjutkan.
2. Langkah 4: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

h) Pengujian Validasi Input *Password*:

- Tujuan: Memastikan bahwa input password tidak kosong dan memiliki panjang minimal tertentu (misalnya, 8 karakter).
- Langkah Pengujian:
  1. Masukkan *password* kosong.
  2. Masukkan *password* dengan panjang kurang dari 8 karakter (misalnya, "pass").
  3. Masukkan *password* dengan panjang lebih dari atau sama dengan 8 karakter (misalnya, "password123").
- Hasil yang Diharapkan:
  1. Langkah 1 dan 2: Sistem menampilkan pesan kesalahan dan tidak mengizinkan pengguna untuk melanjutkan.
  2. Langkah 3: Sistem menerima input dan mengizinkan pengguna untuk melanjutkan.

i) Pengujian Tombol DAFTAR:

- Tujuan: Memastikan bahwa tombol DAFTAR berfungsi dengan benar dan mengarahkan pengguna ke halaman berikutnya jika input valid.

- Langkah Pengujian:
  1. Masukkan semua input yang valid.
  2. Klik tombol DAFTAR.
- Hasil yang Diharapkan:
  1. Sistem memverifikasi input dan mengarahkan pengguna ke halaman berikutnya jika input valid. Jika input tidak valid, sistem menampilkan pesan kesalahan.

C. Proses Pembuatan Aplikasi

Berikut adalah penjelasan proses pembuatan aplikasi **Pelayanan Administrasi Penduduk di Kantor Kelurahan Jaya Kecamatan Watang Sawitto**:

1. **Perancangan Sistem**

   Perancangan aplikasi ini dimulai dengan menganalisis kebutuhan dan kendala yang ada dalam proses administrasi di Kelurahan Jaya. Berdasarkan hasil analisis, aplikasi ini dirancang untuk menangani pengajuan surat administrasi seperti KTP, KK, dan surat keterangan lainnya.

   Teknologi yang digunakan:

   - ***HTML & CCS*:** Untuk membangun tampilan antarmuka (*frontend*).
   - ***JavaScript*:** Digunakan untuk interaktivitas pada antarmuka pengguna.
   - ***PHP & MySQL*:** Digunakan pada bagian *backend* untuk mengolah data penduduk dan pengajuan surat.
   - ***Electron.js*:** Memungkinkan aplikasi berjalan dalam format *desktop* (*cross-platform*).

2. **Pengembangan Antarmuka Pengguna**

   Antarmuka pengguna dirancang menggunakan kombinasi ***HTML*, *CCS*, dan *JavaScript***, dengan fokus pada kemudahan akses dan navigasi. Beberapa halaman utama meliputi:

   - **Halaman *Login* & Registrasi:** Digunakan oleh warga dan *admin* untuk masuk ke sistem.
   - ***Dashboard Admin*:** Menampilkan jumlah data penduduk dan surat yang sedang diproses.
   - **Form Pengajuan Surat:** Memungkinkan warga mengajukan surat secara digital.
   - **Halaman Verifikasi:** Digunakan *admin* untuk memproses pengajuan surat.

3. **Pengolahan Data dengan *PHP* & *MySQL***

   PHP digunakan untuk menangani permintaan pengguna, seperti:

   - Memproses data pendaftaran pengguna.
   - Menyimpan dan menampilkan daftar surat yang diajukan.
   - Mengelola data penduduk.

   MySQL digunakan sebagai *database* utama untuk menyimpan informasi seperti:

   - **Tabel tb_*admin*** → menyimpan data *admin*.
   - **Tabel tb_penduduk** → menyimpan data penduduk.
   - **Tabel tb_ajuan** → menyimpan data pengajuan surat.
   - **Tabel tb_surat** → menyimpan data surat yang telah diverifikasi.

4. Implementasi *Electron.js*

Aplikasi ini dikembangkan sebagai aplikasi *desktop* berbasis ***Electron.js***, yang memungkinkan aplikasi berbasis *web* berjalan dalam format aplikasi mandiri. Dengan *Electron*, sistem dapat dijalankan di ***Windows*, *macOS*, dan *Linux*** tanpa perlu peramban.

Langkah-langkah implementasi *Electron*:

- **Membuat proyek *Electron*** dengan *npm init* dan menginstal *Electron*.
- **Menghubungkan *HTML*, *CCS*, dan *JavaScript*** agar berjalan dalam jendela *Electron.*
- **Menggunakan *PHP* sebagai *server* lokal** untuk menangani data di *backend.*
- **Mengemas aplikasi menjadi format .exe atau .dmg** agar bisa diinstal pada komputer pengguna

BAB V PENUTUP

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan latar belakang yang telah diuraikan, dapat disimpulkan bahwa Kelurahan Jaya, sebagai salah satu kelurahan di Kecamatan Watang Sawitto, Kabupaten Pinrang, memiliki peran strategis dalam memberikan pelayanan administrasi penduduk. Namun, terdapat beberapa kendala yang menghambat efektivitas layanan tersebut, seperti keterbatasan akses, waktu operasional yang terbatas, prosedur yang kompleks, serta proses administrasi yang masih manual. Hal ini menyebabkan keterlambatan dalam penyelesaian dokumen administrasi dan ketidakpuasan masyarakat.

Untuk mengatasi masalah tersebut, dirancanglah aplikasi pelayanan administrasi penduduk yang diharapkan dapat meningkatkan aksesibilitas, efisiensi, transparansi, dan akuntabilitas layanan administrasi. Aplikasi ini bertujuan untuk mempercepat proses pengurusan dokumen, meminimalisir kesalahan, serta memudahkan monitoring dan evaluasi layanan. Dengan demikian, aplikasi ini diharapkan dapat menjadi solusi untuk meningkatkan kualitas layanan publik di tingkat kelurahan.

B. Saran

Setelah melakukan penelitian, maka penulis memberikan saran sebagai berikut:

1. **Pengembangan Aplikasi yang Lebih Komprehensif:**

   Aplikasi pelayanan administrasi penduduk sebaiknya dikembangkan dengan fitur yang lebih lengkap dan *User-friendly* agar dapat memenuhi kebutuhan masyarakat secara menyeluruh. Pengembangan lebih lanjut dapat mencakup integrasi dengan sistem pemerintah daerah lainnya untuk meningkatkan efisiensi.

2. **Sosialisasi dan Pelatihan:**

   Perlu dilakukan sosialisasi dan pelatihan kepada staf Kelurahan Jaya dan masyarakat mengenai penggunaan aplikasi ini. Hal ini penting untuk memastikan bahwa semua pihak dapat memanfaatkan aplikasi dengan optimal.

3. **Peningkatan Infrastruktur Teknologi Informasi:**

   Pemerintah setempat perlu memastikan ketersediaan infrastruktur teknologi informasi yang memadai, seperti jaringan internet yang stabil dan perangkat komputer yang memadai, untuk mendukung operasional aplikasi.

4. **Evaluasi Berkala:**

   Aplikasi ini sebaiknya dievaluasi secara berkala untuk mengidentifikasi kekurangan dan melakukan perbaikan. *Feedback* dari pengguna (staf kelurahan dan masyarakat) juga perlu dipertimbangkan untuk meningkatkan kualitas layanan.

5. **Pengembangan ke Platform Mobile:**

   Untuk meningkatkan aksesibilitas, aplikasi dapat dikembangkan ke *platform mobile* (*smartphone*) agar masyarakat dapat mengakses layanan administrasi kapan saja dan di mana saja.

DAFTAR PUSTAKA

Duckett, J., & Wiley, J. (2011). *HTML & CCS Design and Build Websites*. John Wiley & Sons

Hartl, M., & Donahoe, L. (2016). *Learn Enough HTML to Be Dangerous*. Learn Enough Society

Morgan, J. (2021). *How to code in React.js*. O'Reilly Media

Mustika, W. P., Kumalasari, J. T., Fitriani, Y., & Abdurohim, A. (2021). Sistem informasi administrasi kependudukan (SIASIK) pada kelurahan berbasis *web*. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, *5*(1), 230-240.

Nematrian, N. (2020). *HTML, CCS and JavaScript*. Nematrian

Nurfadillah, N., & Wafiah, A. (2024). Media Pembelajaran Buku Tematik Berbasis *Website*. *Jurnal Sintaks Logika*, *4*(2), 146-156

Patrick, J. (2016). *JavaScript A Beginner's Guide to Learning the Basics of JavaScript Programming*. CreateSpace Independent Publishing Platform

Putri, N., Basri, M., & Wafiah, A. Bantuan Langsung Tunai dan Pangan Non Tunai Pemerintah Berbasis *Web*.

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*. O'Reilly Media

Subramaniam, V. (2018). *Rediscovering JavaScript: Master ES6, ES7, and ES8*. Pragmatic Bookshelf

Sutarman. (2007). *Membangun aplikasi Web dengan PHP & MySQL*. Andi Publisher.

Wafiah, A. (2023). Aplikasi Keperluan Tamu Menggunakan Qr Code Berbasis Cloud Computing Dan Android. *Jurnal Sintaks Logika*, *3*(3), 35-39.

Wahyuddin, W., & Wafiah, A. (2022). Aplikasi Pemesanan Menu Pada Warkop Shearlock Berbasis Abdroid. *Jurnal Sintaks Logika*, *2*(3), 11-16.

Zulkarnaen, M. F., Putri, L. L., & Asyari, H. (2018). Sistem Informasi Administrasi Kependudukan Kelurahan Rembige Berbasis *Web*. *Jurnal Manajemen Informatika Dan Sistem Informasi*, *2*(1), 57-62.