BAB I

PENDAHULUAN

A. Latar Belakang

Setiap orang berhak atas lingkungan hidup yang baik dan sehat berdasarkan Undang-Undang Dasar 1945. Berdasarkan Penjelasan Umum Undang-Undang Nomor 32 Tahun 2009 tentang Perlindungan dan Pengelolaan Lingkungan Hidup (UUPPLH), negara, pemerintah, dan pihak terkait lainnya berkewajiban memelihara lingkungan hidup yang sehat bagi seluruh penduduk. Lingkungan hidup yang berkelanjutan ini tidak dapat tercapai tanpa keterlibatan aktif seluruh lapisan masyarakat, termasuk pelaku usaha atau kegiatan yang dapat berdampak buruk terhadap lingkungan hidup. Berbagai upaya telah diinisiasi untuk melestarikan lingkungan, di antaranya melalui perumusan regulasi dan pengembangan kebijakan nasional perlindungan lingkungan yang terintegrasi, yang wajib diimplementasikan secara konsisten dari tingkat pusat hingga daerah. Diharapkan, pelaksanaan upaya-upaya ini dapat meminimalisir pencemaran dan kerusakan lingkungan akibat aktivitas manusia.

Rencana pengendalian dan pemantauan masalah lingkungan hidup tertuang dalam berkas lingkungan hidup, yang merupakan dokumen resmi. Berkas-berkas ini terbagi dalam tiga kategori utama: Analisis Mengenai Dampak Lingkungan (AMDAL), Upaya Pengelolaan dan Pemantauan Lingkungan (UKL-UPL), dan Surat Persetujuan Pengelolaan dan Pemantauan Lingkungan (SPPL). Sebagai contoh, proyek-proyek berskala besar seperti pembangunan fasilitas manufaktur,

infrastruktur jalan tol, atau pengembangan sektor pariwisata dan perhotelan, umumnya mensyaratkan AMDAL. Sementara itu, kegiatan pertanian skala besar biasanya diwajibkan memiliki UKL-UPL. Adapun SPPL diperlukan untuk jenis usaha seperti industri pengolahan limbah dan fasilitas kesehatan seperti klinik..

Seperti kita ketahui semakin berkembangnya zaman apalagi dilihat dari teknologi dan informasi baik yang dilihat dari segi media online dan televisi, yang menjadi terbarukan yakni dengan membuat aplikasi yang tadinya berupa dokumen yang diketik harus dirumah dan dikantor secara *offline* menjadi bisa dilakukan dimana pun dan kapan pun secara online. Dan datanya bisa disimpan di *cloud* (diawan) maksudnya provider menyediakan tempat penyimpanan gratis dimana data yang sudah kita olah dapat disimpan di *cloud* tersebut, contohnya di google drive atau di terabox. Dan adapula penyimpanan yang berupa database yang memang secara khusus dibayar untuk menyimpan data kita supaya terhindar dari *hacker* (orang yang iseng menggunakan keahliannya untuk merusak data perusahaan atau data pribadi), sebagai contoh data dokumen lingkungan hidup yang dapat kita kelola dengan secara online.

Secara tidak langsung dokumen yang ada di dinas sebelumnya disimpan di kantor saja dan ada kemungkinan dokumen tersebut tercecer atau tidak sengaja bercampur dengan dokumen yang lain. Masalah lainnya juga yakni penggunaan kertas yang banyak, saat menyerahkan dokumen lalu direvisi tentunya menggunakan kertas yang baru untuk mencetaknya. Dan kertas sebelumnya tentu tidak bisa digunakan lagi, yang menyebabkan penumpukan sampah kertas. Selain

penggunaan kertas yang banyak, masalah lain mungkin dari segi biaya, baik itu biaya pencetakan atau mungkin biaya transportasi.

Sehingga penulis berinisiatif melakukan penelitian dengan membuat aplikasi pengelolaan dokumen pada dinas lingkungan hidup. Dengan menggunakan analisis data terlebih dahulu terus nantinya akan mendesign, lanjut ke implementasi dan testing sistem yang digunakan nantinya apakah lebih efektif dan bermanfaat dalam mengelola dokumen secara online. Dalam aplikasi nantinya pemohon akan registrasi sebelum login, setelah login pemohon akan melakukan pengisian dokumen sesuai yang dibutuhkan dan mengikuti panduan pengisian yang terdapat pada aplikasi. Kemudian setelah pengisian dokumen secara online tadi, pemohon menunggu verifikasi dari admin.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka perumusan masalah yang dapat diambil adalah sebagai berikut:

Bagaimana membuat rancangan dan mengimplementasikan sistem pengelolaan dokumen lingkungan kedalam sistem web yang efektif dan sesuai keperluan operasional DLH Kota Parepare?

C. Tujuan Penelitian

Adapun tujuan yang penulis dapatkan untuk tugas akhir ini yaitu:

Merancang dan mengimplementasikan sistem pengelolaan dokumen di DLH Kota Parepare menjadi sistem web yang efektif.

D. Batasan Masalah

Pembatasan masalah berikut diterapkan pada pelaksanaan studi ini:

1. Objek penelitian pada dokumen AMDAL,UKL-UPL, dan SPPL yang terdapat di Dinas Lingkungan Hidup Kota Parepare.
2. Bahasa komputer yang disebut PHP digunakan untuk membuat aplikasi berbasis web.

E. Manfaat Penelitian

Adapun manfaat penelitian yang dilakukan penulis pada tugas akhir ini yaitu:

1. Manfaat untuk Mahasiswa

   Temuan studi ini dapat membantu Dinas Lingkungan Hidup memperoleh lebih banyak pemahaman dan wawasan tentang proses penerapan sistem manajemen dokumen berbasis web.

2. Manfaat untuk Universitas

   Hasil penelitian ini juga diharapakan dapat berguna bagi universitas yakni peningkatan pengetahuan, solusi praktis, dukungan kebijakan yang berkualitas, perubahan praktik kerja yang berkelanjutan, dan pengembangan kapasitas yang lebih baik dalam menghadapi tantangan lingkungan hidup.

3. Manfaat untuk Instansi

   Dinas Lingkungan Hidup dapat meningkatkan produktivitas, kepatuhan, dan transparansi dalam proses pengelolaan dokumen. Hal ini dapat memberikan manfaat jangka panjang dalam pengelolaan informasi dan operasional secara keseluruhan.

F. Sistematika Penulisan

Berikut ini adalah pemaparan umum sistematika penulisan:

1. BAB I PENDAHULUAN

Bab I terdiri dari data latar belakang, konseptualisasi masalah, tujuan, dan proses penulisan.

2. BAB II TINJAUAN PUSTAKA

   Bab II memeriksa ide-ide dan literatur yang relevan.

3. BAB III METODE PENELITIAN

   Bab III menjelaskan teknik penelitian yang digunakan.

4. BAB IV HASIL DAN PEMBAHASAN

   Bab IV menampilkan hasil desain dan penerapan sistem.

5. BAB V PENUTUP

   Bab V termasuk kesimpulan dan saran untuk penelitian tambahan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Hasil Peneliti Terdahulu

Untuk memperkuat penelitian dan memastikan keabsahan hasil, penelitian terdahulu akan diuraikan sebagai berikut:

1. Berdasarkan penelitian yang dipublikasikan pada tahun 2023 oleh Dinas Lingkungan Hidup Kota Cirebon dengan judul Perancangan dan Pembangunan Sistem Informasi Arsip Surat Menggunakan Metode *Waterfall* oleh Cep Lukman Rohmat, Dea Eryanti Putri, Martanto, dan Willy Prihartono.Baik di lembaga pemerintahan maupun komersial, penggunaan komputer dan teknologi informasi telah menjadi hal yang esensial bagi operasional sehari-hari. Khususnya dalam layanan publik, penggunaan komputer dan teknologi informasi tidak diragukan lagi sangat mendukung operasional saat ini. Untuk mengatasi tingginya volume surat masuk dan keluar di Badan Pusat Statistik, yang menyulitkan pengarsipan, sebuah sistem pengarsipan surat masuk dan keluar berbasis situs web telah diciptakan. Sistem manajemen arsip surat masuk dan keluar yang dapat menyimpan surat secara digital merupakan salah satu hasil penelitian ini. Sistem ini akan membantu karyawan mengelola surat mereka dan memudahkan setiap divisi untuk mengambil surat. *PHP* dan *MySQL* adalah bahasa pemrograman yang digunakan dalam sistem ini, bersama dengan kerangka kerja *CodeIgniter*. Dengan masuk ke aplikasi pengarsipan ini menggunakan akun pengguna yang tepat, staf administrasi

dapat menambah, mengubah, menghapus, dan menyimpan surat masuk dan keluar, mencari berdasarkan nomor surat, subjek, pengirim, tujuan, dan tanggal, serta mengunduh surat digital berdasarkan hasil pencarian. Berkat sistem pengarsipan surat yang dirancang dengan baik ini, staf dapat memproses data arsip surat masuk dan keluar secara lebih efektif dan efisien.

2. Menurut Kartomiharjo, I. D., Hayuhardhika, W., Putra, N., & Purnomo, W. (2019), jurnal penelitian. Studi kasus Dinas Lingkungan Hidup dan Kebersihan Kabupaten Sidoarjo digunakan untuk menganalisis dan merancang sistem informasi pelaporan sampah. 3(11), 10756–10764, Jurnal Pengembangan Ilmu Komputer dan Teknologi Informasi. Memastikan tidak adanya sampah di lingkungan Kabupaten Sidoarjo merupakan salah satu tanggung jawab Dinas Lingkungan Hidup dan Kebersihan Kabupaten Sidoarjo. Sampah menjadi permasalahan utama yang dihadapi Dinas Lingkungan Hidup dan Kebersihan Kabupaten Sidoarjo.Agar Dinas Lingkungan Hidup dan Kehutanan (DLHK) dapat segera menangani penumpukan sampah, pelaporan oleh masyarakat sangatlah penting. Saat ini, warga harus datang ke DLHK untuk melaporkan sampah karena masih dilakukan secara manual. Oleh karena itu, pemantauan menjadi tantangan dan memakan waktu. Sistem informasi diperlukan untuk mengatasi masalah ini.Perancangan yang mempertimbangkan kebutuhan Dinas Lingkungan Hidup dan Kehutanan (DLHK) dan masyarakat diperlukan untuk memastikan sistem informasi memenuhi kriteria. Oleh karena itu, metode yang dikenal sebagai Analisis dan Perancangan Berorientasi Objek (OOAD) digunakan. Analisis kebutuhan yang dilakukan untuk studi ini

mengidentifikasi 53 karakteristik fungsional dan 2 karakteristik non-fungsional. Selain itu, desain yang menggunakan diagram UML menghasilkan 24 desain antarmuka pengguna, dua pengontrol diagram kelas, dua model diagram kelas, basis data dalam bentuk PDM, dan 22 diagram sekuens. Kerangka kerja konfigurasi persyaratan kemudian digunakan untuk menilai analisis kebutuhan dan desain yang dibuat, menghasilkan nilai 100%. Pendekatan matriks ketertelusuran juga digunakan dalam evaluasi, dan semua desain pada akhirnya dipastikan sesuai dengan analisis kebutuhan.

3. Berdasarkan jurnal penelitian Adam Afriansyah dan Ade Oktarino. (2020). Pendekatan *prototipe* digunakan dalam pembuatan Sistem Pelaporan Pengelolaan Lingkungan Hidup untuk Dinas Lingkungan Hidup Kota Jambi. Badan usaha diwajibkan untuk menyampaikan kegiatan pelaporan lingkungan hidup mereka kepada Dinas untuk dievaluasi dan ditinjau setiap triwulan oleh personel Dinas Lingkungan Hidup Kota Jambi yang telah disetujui. Berdasarkan wawancara, Dinas Lingkungan Hidup Kota Jambi memiliki sejumlah masalah dalam menangani laporan lingkungan perusahaan, termasuk penyimpanan dokumen yang tidak rapi, keterlambatan pelaporan, dan persyaratan ruang penyimpanan dokumen minimal 100 m^2 per tahun. Untuk mengatasi permasalahan berdasarkan temuan wawancara, diperlukan perancangan sistem pelaporan Manajemen Lingkungan. Agar penelitian ini dapat menggunakan proses perancangan prototipe untuk merancang sistem Informasi Manajemen Pelaporan Lingkungan. Aktor akan meluncurkan aplikasi, sistem akan menyediakan formulir masuk, dan aktor akan

memasukkan nama pengguna dan kata sandinya. Sistem kemudian akan memverifikasi kebenaran informasi tersebut. Menu utama halaman administrator akan ditampilkan oleh sistem jika nama pengguna dan kata sandi dianggap sah. Di menu administrator, aktor dapat menambah, menghapus, dan mengubah data perusahaan dan admin. Sistem akan menunjukkan bahwa data berhasil disimpan jika lengkap. Hasilnya berupa laporan cetak yang membuktikan bahwa Kepala Dinas Lingkungan Hidup (DLH) dapat mencetak laporan limbah B3, laporan limbah non-B3, laporan PPA, laporan PPU, dan laporan UKL/RPL-UKL/UPL.

4. Berdasarkan jurnal penelitian Tri Haryono Patria Mangambe. 2019. Untuk mengelola dan menjaga lingkungan di Kota Yogyakarta, badan usaha dan/atau operasi yang memiliki Surat Pernyataan Komitmen Pengelolaan dan Pemantauan Lingkungan (SPPL) wajib berada di bawah pengawasan Badan Lingkungan Hidup. Pengawasan terhadap badan usaha yang wajib memiliki SPPL dikaji dalam studi ini. Perusahaan yang wajib memiliki SPPL dianggap memiliki dampak yang lebih kecil terhadap lingkungan. Namun pada kenyataannya dari segi jumlah pelaku usaha diwajibkan memiliki SPPL sangat banyak di masyarakat sehingga akumulasi dampak lingkungan juga akan berdampak besar bagi lingkungan. Berdasarkan temuan tersebut, Dinas Lingkungan Hidup Kota Yogyakarta mengalami sejumlah kesulitan dalam mengawasi perusahaan dan/atau kegiatan usaha yang telah mendapatkan SPPL di kota tersebut. Oleh karena itu, untuk mencapai keseimbangan antara

pengelolaan, pelestarian, dan pemanfaatan lingkungan, pemantauan harus segera dimulai.

B. Kajian Pustaka

1. Dokumen Perlindungan dan Pengelolaan Lingkungan Hidup

Undang-Undang Nomor 32 Tahun 2009 Republik Indonesia yang menjelaskan tentang perlindungan dan pengelolaan lingkungan hidup, mewajibkan pemerintah daerah untuk membangun sistem informasi lingkungan hidup guna meningkatkan efisiensi penyusunan dan penerapan peraturan pengelolaan dan perlindungan lingkungan hidup. Pemerintah Kota Parepare, sebagai pemegang kewenangan, penting untuk membangun sistem informasi lingkungan yang publik, komprehensif, dan mudah dinavigasi.

Dokumen Informasi Kinerja Pengelolaan Lingkungan Daerah merupakan alat penting untuk menyebarluaskan informasi lingkungan, meningkatkan kesadaran dan pemahaman publik terhadap isu tersebut, serta membantu para pengambil keputusan mengidentifikasi langkah-langkah yang diperlukan untuk meningkatkan pengelolaan lingkungan. Untuk mencapai pembangunan Kota Parepare yang berkelanjutan, maka perlu dikenali isu-isu strategis yang menjadi fokus utama pembangunan, mempertimbangkan pemangku kepentingan sesuai dengan tanggung jawab dan kewenangannya. Isu-isu tersebut kemudian diimplementasikan berdasarkan kondisi permasalahan dan potensi Kota Parepare sesuai dengan dinamika pembangunan daerah dan nasional.

2. Penjelasan Pelaporan dan Pengelolaan Dokumen

a. Dokumen AMDAL. Bagiannya yakni

1) Dokumen Kerangka Acuan Analisis Mengenai Dampak Lingkungan Hidup(KA-ANDAL)
2) Dokumen Analisis Mengenai Dampak Lingkungan Hidup (ANDAL)
3) Dokumen Rencana Pengelolaan Lingkungan Hidup (RKL)
4) Dokumen Rencana Pemanauan Lingkungan Hidup (RPL)

Adapun contoh kegiatan yang membutuhkan AMDAL yakni, Pembangunan pabrik: AMDAL diperlukan untuk menilai potensi dampak lingkungan dari operasi industri saat memulai pabrik baru atau mengembangkan pabrik yang sudah ada.

Pembangunan infrastruktur besar: Proyek pembangunan jalan tol, bendungan, atau bandara yang melibatkan intervensi besar terhadap lingkungan alam juga membutuhkan AMDAL untuk menyelidiki potensi dampak negatif dan menentukan langkah-langkah mitigasi yang diperlukan.

b. Dokumen UKL/UPL. Bagiannya yakni

Aturan Identifikasi Jenis Rencana Usaha dan/atau Operasional yang Memerlukan Upaya Pemantauan dan Pengelolaan Lingkungan (UPL dan UKL)

Adapun contoh usaha yang membutuhkan UKL-UPL yakni, Pertanian skala besar: Usaha pertanian yang melibatkan lahan yang luas atau teknologi pertanian modern (seperti penggunaan pestisida atau pupuk kimia secara intensif) Peraturan lingkungan yang tercantum dalam UKL-UPL (Upaya Pengelolaan Lingkungan dan Upaya Pemantauan Lingkungan) harus dipatuhi.

c. Dokuemen SPPL. Bagiannya yakni

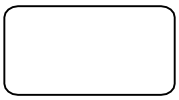
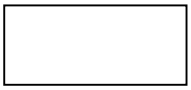
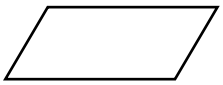
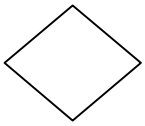
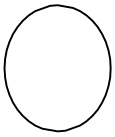
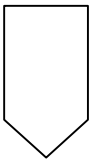
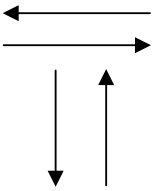
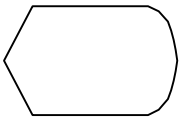
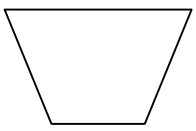
1) Syarat untuk memperoleh Izin Teknis tertentu seperti Izin Usaha Industri (IUI) dan Tanda Daftar Usaha Pariwisata (TDUP).
2) Surat Pernyataan Kesanggupan Pengelolaan dan Pemantauan Lingkungan Hidup.

Adapun contoh kegiatan yang membutuhkan SPPL yakni, Industri pengolahan limbah: Usaha yang bergerak di bidang pengolahan limbah seperti tempat pembuangan akhir sampah atau pengolahan limbah industri harus memenuhi persyaratan lingkungan dan mengajukan SPPL (Surat Pernyataan Penolakan Limbah) yang menunjukkan bahwa mereka memiliki fasilitas yang sesuai dan beroperasi secara aman.

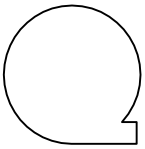
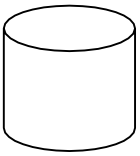
3. *Flowchart*

(Diding Kusnady, S. Pd, MM, 2018:10) *"Flowchart* atau Diagram alir adalah penggambaran visual suatu sistem yang menjelaskan hubungan fisik fundamental antar entitasnya. Aktivitas yang melibatkan pekerjaan manual, pemrosesan komputer, atau keduanya dapat digambarkan menggunakan diagram alir. Unsur-unsur sistem manual, meliputi departemen organisasi yang terlibat dalam proses, catatan akuntansi (kertas, jurnal, buku besar, dan file), serta tugas fisik dan non-fisik yang dilakukan, semuanya direpresentasikan oleh diagram alir dokumen. Simbol-simbol berikut digunakan dalam diagram alir:

Tabel 2. 1 Simbol-Simbol Flowchart

NO	SIMBOL	KETERANGAN
1.		Awal dan akhir diagram alir ditunjukkan oleh simbol Mulai atau Akhir.
2.		Memproses simbol-simbol yang muncul selama suatu proses.
3.		Simbol input/output yang menentukan input dan output proses.
4.		Simbol untuk menentukan apakah proses akan dilanjutkan dalam keadaan tertentu.
5.		Gunakan simbol koneksi untuk menghubungkan proses pada lembar kerja yang sama.
6.		Simbol konektor digunakan untuk menghubungkan operasi di seluruh lembar kerja.
7.		Simbol untuk membuat hubungan antara simbol atau proses.
8.		Simbol untuk perangkat keluaran, seperti printer dan layar monitor, antara lain.
9.		Simbol yang digunakan untuk menggambarkan proses manual.

Lanjutan **Tabel 2.1**

NO	SIMBOL	KETERANGAN
10.		Simbol dapat dimasukkan atau dikeluarkan dari suatu dokumen.
11.		Simbol yang digunakan untuk menunjukkan sub-program, atau bagian dari suatu program.
12.		Simbol yang dimasukkan atau dikeluarkan ke pita magnetik.
13.		Basis data atau simbol basis data

4. *UML* (*Unified Modeling Language*)

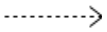
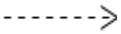
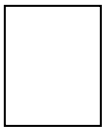
Kumpulan struktur, diagram, dan metode desain untuk sistem dan aplikasi berorientasi objek disebut *UML*. *UML* adalah bahasa standar untuk spesifikasi kebutuhan, analisis, desain, dan karakterisasi arsitektur dalam pemrograman berorientasi objek.

Dalam *UML* sendiri terdapat beberapa diagram yaitu :

a. Use Case Diagram

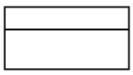
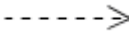
Menurut peneliti (Unhelkar, 2018), Model tingkat tinggi dari kebutuhan sistem disebut *use case*. Tujuan utama diagram *use case* adalah untuk menggambarkan *use case*, industri terkait, dan bagaimana mereka berinteraksi. Diagram *Use Case* menggunakan simbol-simbol berikut:

Tabel 2. 2 Simbol-Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menggambarkan rentang peran yang dapat dimainkan pengguna dalam interaksi kasus penggunaan.
2		*Dependency*	Jika suatu unsur independen berubah, maka akan berdampak pada unsur dependen atau non-independen.
3		*Generalization*	Hubungan di mana objek induk (leluhur) dan objek anak (keturunan) berbagi struktur data dan perilaku.
4		*Include*	Menunjukkan bahwa kasus penggunaan sumbernya jelas.
5		*Extend*	menunjukkan bahwa perilaku kasus penggunaan sumber pada saat tertentu diperluas oleh kasus penggunaan target.
6		*Association*	Ini membuat koneksi antara dua objek.
7		*System*	menemukan paket yang menyediakan tampilan sistem yang terbatas.
8		*Use Case*	Penjelasan mengenai langkah-langkah yang diambil sistem untuk memberikan hasil yang dapat diukur kepada aktor.
9		*Collaboration*	Ketika aturan dan elemen lain bekerja sama, hasilnya adalah perilaku yang lebih kuat daripada jumlah bagian-bagiannya.

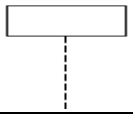
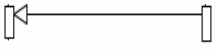
b. Class Diagram

Tabel 2. 3 Simbol-Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan di mana objek induk (leluhur) berbagi struktur data dan perilakunya dengan objek anak (keturunan).
2		*Nary Association*	Hindari mengaitkannya dengan lebih dari dua hal.
3		*Class*	Sekelompok item dengan karakteristik dan fungsi yang serupa.
4		*Collaboration*	Penjelasan mengenai langkah-langkah yang diambil sistem untuk memberikan hasil yang dapat diukur kepada aktor.
5		*Realization*	Tugas yang benar-benar diselesaikan suatu objek.
6		*Dependency*	koneksi di mana perubahan yang dibuat pada satu komponen independen memengaruhi komponen dependen atau non-dependen.

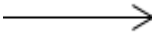
c. *Sequence* Diagram

Tabel 2. 4 *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek entitas merupakan antarmuka dengan kemampuan komunikasi antarpribadi.
2		*Message*	Antarmuka dengan kemampuan berkomunikasi dengan yang lain dikenal sebagai objek entitas.
3		*Message*	persyaratan untuk komunikasi antara objek yang menyimpan informasi tentang kejadian.

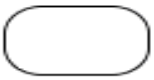
d. *State Chart* Diagram

Tabel 2. 5 *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Atribut dan nilai tautan suatu objek pada titik waktu tertentu.
2		*Initial Pseudo State*	Bagaimana sesuatu ditulis atau dirumuskan
3		*Final State*	Bagaimana sesuatu diciptakan dan dihancurkan
4		*Transition*	Keadaan suatu objek dipicu oleh suatu peristiwa yang mengubah satu atau lebih nilai atributnya.
5		*Association*	yang mengikat dua objek menjadi satu.
6		*Node*	Komponen fisik sumber daya komputasi yang hadir saat aplikasi beroperasi.

e. *Actify* Diagram

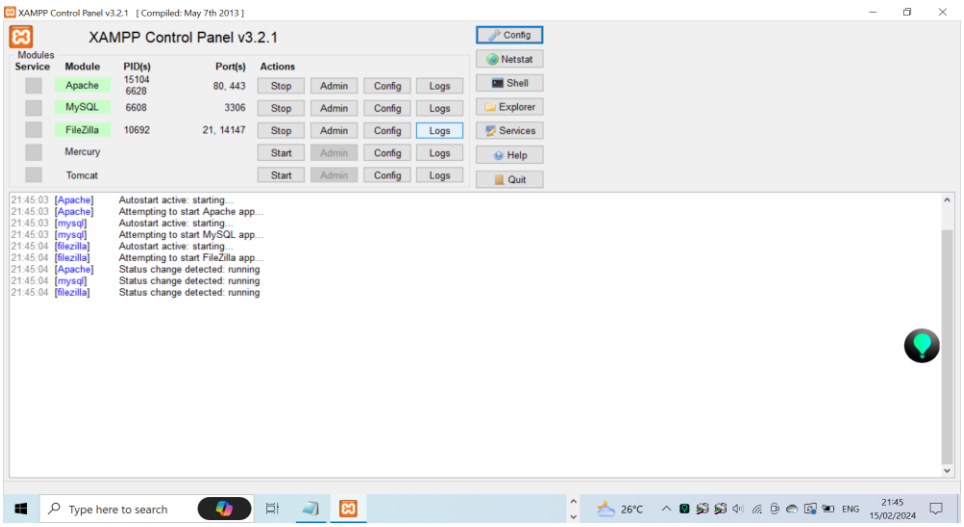
Tabel 2. 6 Simbol-Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	menunjukkan hubungan antara berbagai kelas antarmuka.
2		*Action*	Persyaratan sistem yang menentukan bagaimana suatu tugas dilakukan
3		*Initial Node*	Bagaimana sesuatu dimulai atau dibuat.
4		*Activity Final Node*	Pembentukan dan penghentian objek
5		*Fork Node*	Satu aliran yang akhirnya terbagi menjadi beberapa aliran

5. XAMPP

XAMPP adalah paket perangkat lunak *open-sourceopen-source* yang menyediakan lingkungan *server* local untuk pengembangan web(*Apache, MySQL, PHP, Perl*) . Fungsionalitas server web disediakan oleh *XAMPP*. *XAMPP*, Aplikasi ini memungkinkan Anda melihat pratinjau dan mengedit halaman web tanpa memerlukan koneksi atau akses internet, dan terkadang disebut *server CPanel* virtual.

Penjelasan di atas mengarah pada kesimpulan bahwa *XAMPP* adalah server web yang dapat diinstal *di Windows, Linux, macOS, dan Solaris*, serta membantu dalam melihat pratinjau dan memodifikasi situs web secara offline. Paket perangkat lunak yang ditawarkan *XAMPP*, yang mencakup konfigurasi *Server Web, Apache, dan PHP*, memudahkan kita dalam membangun program web dan aplikasi web terintegrasi.



Gambar 2.1 Contoh Tampilan *XAMPP*

6. PHP

PHP (PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang ditambahkan ke *HTML* untuk mengubah kode sumber program menjadi kode mesin yang dapat dipahami oleh komputer sisi server. *Hypertext Preprocessor (PHP)* adalah bahasa skrip sisi server yang populer untuk pengembangan Java dinamis yang dapat berinteraksi dengan pengguna atau pengunjung.

PHP adalah bahasa skrip yang diproses dan dieksekusi di server. *PHP*, singkatan dari *Hypertext Preprocessor*, adalah bahasa untuk pemrograman web sisi server yang sumbernya bebas. Ini berarti hanya hasilnya yang diberikan ke peramban sintaksis lengkapnya dieksekusi di server.

Contoh *php* yakni:

```
<?php
      $nim = "219280087";
      $nama = 'St.Khaeriani';
      echo  "NIM:  ".  $nim.
      "<br>";
      echo "Nama: $nama";
?>
```

7. *JavaScript*

JavaScript adalah sebagai bahasa script *client-side* untuk membuat halaman web interaktif. Blok kode *JavaScript* diberi nama hampir di mana saja pada halaman untuk mencapai tujuan ini.

Adapun tag yang digunakan di *javascript* sebagai berikut: Teks ditampilkan ke konsol *JavaScript* oleh fungsi *console.log()*. Kotak dialog ditampilkan oleh *metode alert()*. Objek window berisi fungsi-fungsi yang sebenarnya. Contohnya bisa ditulis seperti ini:

```
window.alert("Hello World!");
```

8. *HTML*

Bahasa markah umum untuk mengelola halaman web adalah *HTML (Hypertext Markup Language)*. Karena *HTML* adalah bahasa komputer bebas, *HTML* dibuat oleh banyak orang di berbagai negara dan tidak dimiliki oleh satu entitas pun. Oleh karena itu, *HTML* dapat dianggap sebagai bahasa yang diproduksi secara kolaboratif dalam skala global.

Dokumen teks apa pun yang dapat diubah dengan editor teks apa pun disebut dokumen *HTML*. Setelah itu, ekstensi berkas *HTML* digunakan untuk menyimpannya. Beberapa item yang diapit oleh tag teks yang diawali dengan simbol "<" dan diakhiri dengan simbol "/>" membentuk dokumen *HTML*.

Tabel 2.7 *Tag HTML*

No	Nama Tag	Fungsi
1	<! DOCTYPE html>	Untuk menentukan dokumen sebagai HTML, gunakan deklarasi ini.
2	<html>	Saat membuat dokumen HTML, tag pembuka
3	<head>	Metadata yang berkaitan dengan dokumen
4	<title>	Buatlah judul untuk halaman yang akan dilihat oleh browser.
5	<body>	Lokasi di mana HTML digunakan untuk menghasilkan semua konten situs web

9. *CSS* (*Cascading Style Sheets*)

Tampilan dan nuansa halaman web ditentukan menggunakan bahasa yang disebut *CSS*. Bahasa pemrograman yang disebut *CSS* digunakan untuk

memperbaiki dan menciptakan tata letak serta tampilan halaman web, sehingga terlihat lebih elegan dan menarik.

Tampilan halaman daring yang dibuat menggunakan *HTML* dan *XHTML* seringkali diformat menggunakan *CSS*. Gambar, teks isi, tabel, batas, hyperlink, kursor tetikus, paragraf, teks, margin atas/bawah, kiri/kanan, dan elemen lainnya dapat diubah warnanya dengan *CSS*. Hal ini memungkinkan CSS untuk menampilkan halaman yang sama dalam beberapa format. Terdapat tiga kategori *CSS*:

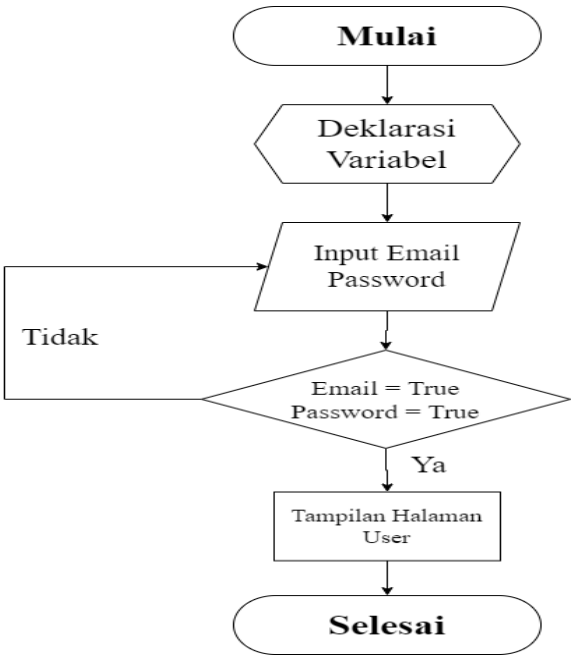
a. Elemen <*head*> dari berkas *HTML* berisi definisi Lembar Gaya *CSS internal.*
b. Di luar berkas *HTML*, lembar gaya *CSS eksternal* didefinisikan dalam berkas yang berbeda.
c. *Inline Style Sheet CSS* didefinisikan di dalam setiap elemen/tag *HTML*

Contoh *CSS* beserta fungsinya yakni:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color:   lightblue;   fungsinya   menampilkan
background berwarna biru
}
h1 {
  color: white;   fungsinya tulisan berwarna putih
  text-align: center;   fungsinya tulisan berada di tengah
lembaran
}
p {
  font-family: verdana;
  font-size: 20px;     fungsinya besar tulisannya ada 20
pixel}
</style>
</head>
<body>
<h1>My First CSS Example</h1>  fungsinya untuk menampilkan
hasil di tengah
<p>This is a paragraph.</p> fungsinya untuk membuat paragraf
</body>
</html>
Hasilnya seperti ini :
                    My First CSS Example
This is a paragraph
```

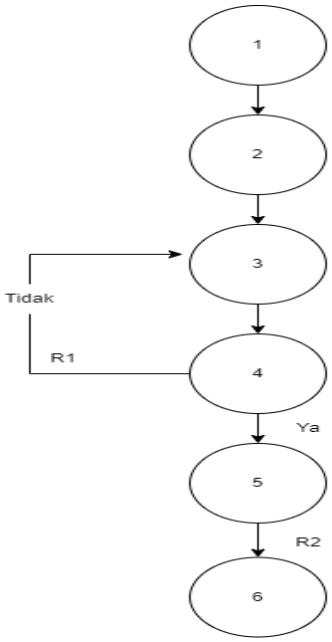
10. *White Box* (Pengujian Kotak Putih)

Pengujian kotak putih adalah jenis pengujian perangkat lunak yang memeriksa kode utama atau arsitektur suatu program. Pengujian kotak putih, yang bertujuan untuk menunjukkan keakuratan metode yang digunakan dan kesesuaiannya dengan proses serta spesifikasi internal lainnya, dapat dilanjutkan dengan pengujian tambahan. Tingkat kelayakan sistem berdasarkan logika algoritma dan fungsi sistem dicantumkan dalam hasil pengujian. Contoh penggunaan *White Box* sebagai berikut:



Gambar 2. 1 *Flowchart* Penggunaan *White Box*

Flowgraph ditentukan berdasarkan Flowchart yang digunakan pada pengujian perangkat lunak berikut:



Gambar 2. 3 *Flowgraph* Penggunaan *White Box*

Dari gambar flowgraph penggunaan *White Box* di atas, proses perhitungan dapat dilakukan dengan cara berikut:

1) Menghitung *Cyclomatic ComplexcityV(G)* dari *Edge* dan *Node*:

   Dengan Rumus: V (G) = E – N + 2

   Dengan E (*edge*) = 6

N (*node*) = 6

Predikat *Node* (P) = 1

Penyelesaian: V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

2) Berdasarkan hitungan dengan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

3) *Independent Path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 3

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6

4) Grafis *Matriks*

Tabel 2.8 *Grafik Matriks Flowgraph* Penggunaan *White Box*

	1	**2**	**3**	**4**	**5**	**6**	**E – 1**
1		1					1 – 1 = 0
2			1				1 – 1 = 0

Lanjutan **Tabel 2. 8**

	1	**2**	**3**	**4**	**5**	**6**	**E - 1**
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
6							0
Sum (E) + 1							1 + 1 = 2

11. *Black Box Testing* (Pengujian Kotak Hitam)

Pengujian "kotak hitam" adalah metode pengujian perangkat lunak, mengabaikan organisasi internal dan hanya berfokus pada fungsionalitas eksternal. Pengamatan ini berfokus pada fungsi perangkat lunak dan data uji. Pengujian kotak hitam tidak melihat detail proses, pengujian ini hanya menilai antarmuka. Hanya proses input dan output yang diperiksa.

Black Box Testing juga memiliki fungsi-fungsi adalah sebagai berikut:

a. Identifikasi fitur perangkat lunak yang hilang atau tidak akurat.
b. Periksa masalah antarmuka yang muncul saat program beroperasi.
c. Untuk menemukan kesalahan dalam akses basis data eksternal atau struktur data suatu aplikasi.
d. Periksa fungsionalitas perangkat lunak.
e. Siapkan dan identifikasi masalah apa pun dengan penghentian perangkat lunak itu sendiri.

Keuntungan dari *Black-box Testing*:

a. Keahlian bahasa pemrograman tidak diwajibkan bagi penguji.

b. Untuk menemukan kontradiksi atau ambiguitas dalam spesifikasi, pengujian dilakukan dari sudut pandang pengguna..
c. Penguji dan pemrogram saling bergantung.

Kekurangan *Black-box Testing*:

a. Tanpa spesifikasi yang tepat, sulit untuk mereplikasi skenario pengujian.jelas
b. Pengujian yang telah dijalankan oleh programmer dapat diulang.
c. Sebagian dari back-end sama sekali belum teruji.

Teknik *Black-box Testing*:

a. *Equivalence Partitioning*

   Data masukan dibagi menjadi beberapa partisi agar teknik ini dapat berfungsi.

b. *Boundary Value Analysis*

   Metode ini lebih berfokus pada batasan, menentukan nilai kesalahan terendah atau maksimum yang diperoleh saat kesalahan terjadi di dalam atau di luar perangkat lunak.

c. *Fuzzing*

   Fuzz adalah teknik yang menggunakan sesi semi-otomatis atau injeksi data yang cacat untuk mendeteksi kesalahan atau gangguan dalam perangkat lunak.

d. *Cause-Effect Graph*

   Grafik digunakan sebagai panduan dalam metode pengujian ini. Grafik ini menunjukkan hubungan antara penyebab dan akibat kesalahan.

e. *Orthogonal Array Testing*

Hal ini dapat diterapkan jika domain masukan cukup berbobot untuk digunakan dalam skala luas meskipun ukurannya relatif kecil.

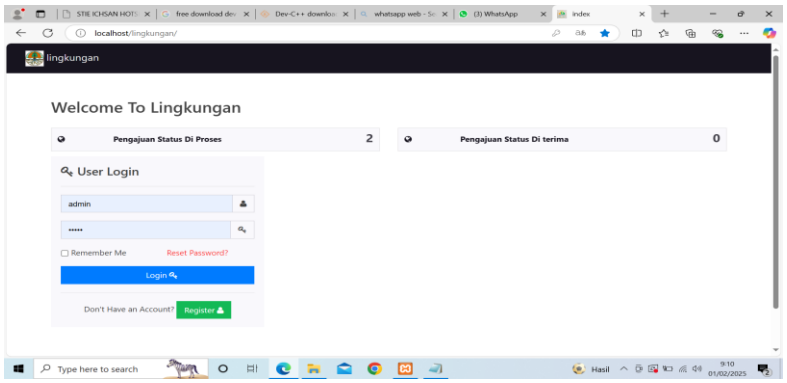
f. *All Pair Testing*

Berdasarkan parameter inputnya, setiap pasangan kasus uji dalam teknik ini dirancang untuk menjalankan setiap kombinasi diskrit yang memungkinkan dari pasangan-pasangan tersebut. Tujuan pengujian ini adalah memiliki dua kasus uji yang mencakup setiap pasangan tersebut.

g. *State Transition*

Pengujian ini berguna untuk mengevaluasi kesehatan mesin dan navigasi antarmuka pengguna grafis.

Tabel 2.9 Contoh Penggunaan *Black Box*

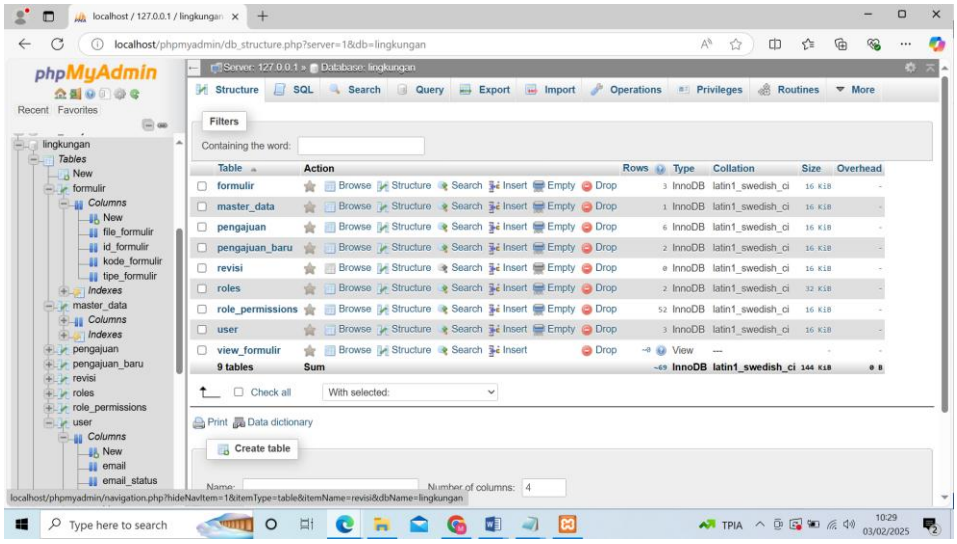
Test Faktor	**Hasil**	**Kesimpulan**
Pengguna menjalankan aplikasi pertama kali	✓	Informasi, tampil halaman utama
Screen Shot		
		

12. *Mysql*

Sistem manajemen basis data relasional sumber terbuka disebut *MySQL*. Basis data dan isinya dapat dibuat dan dikelola dengannya. Menambah, mengedit,

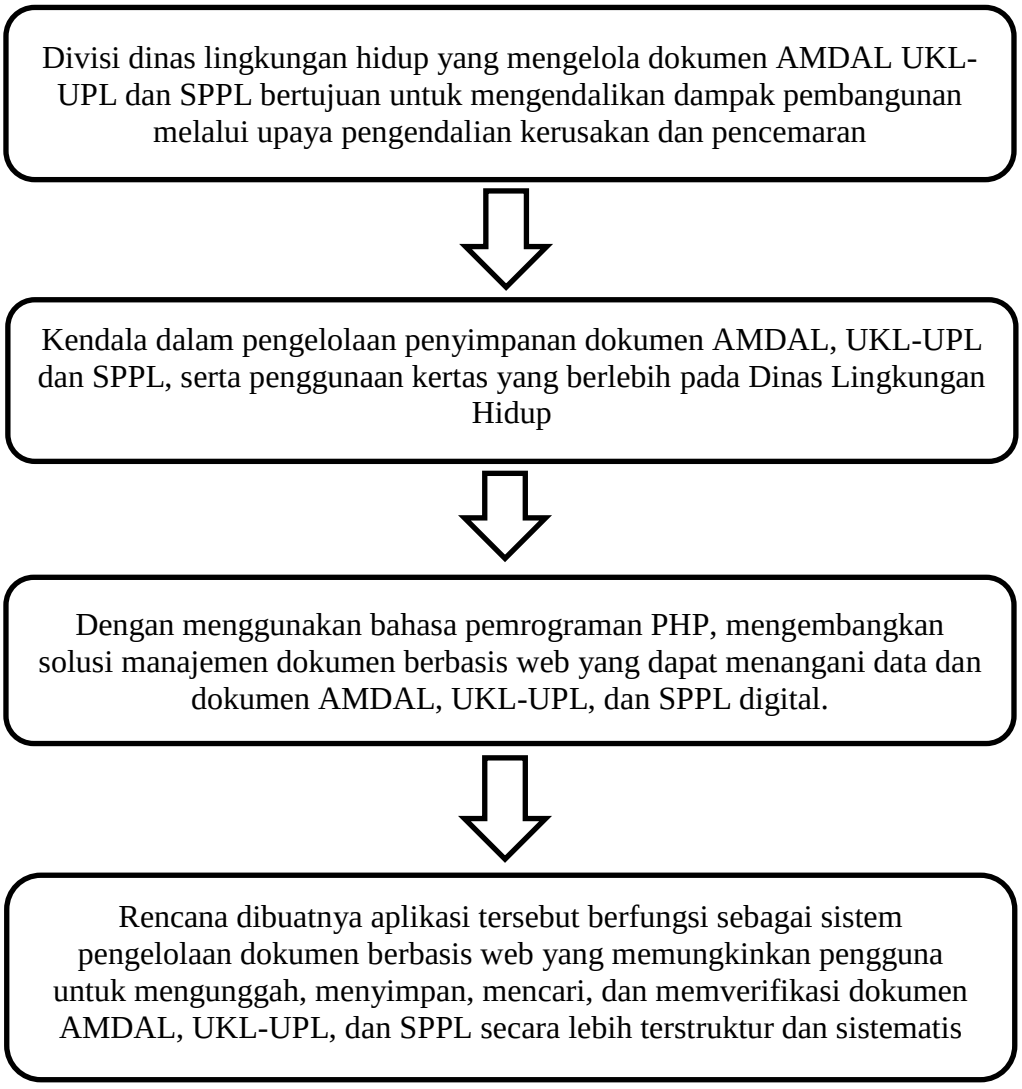
dan menghapus data dari basis data dimungkinkan dengan *MySQL*. Salah satu contoh sistem manajemen basis data relasional adalah *MySQL*. Hal ini memungkinkan manipulasi data yang lebih cepat karena data yang dikelola basis data disimpan dalam beberapa tabel berbeda. Berikut beberapa contoh perintah sederhana yang sering digunakan yakni:

a. *CREATE DATABASE*, untuk membangun database baru.
b. *SELECT*, untuk mengambil informasi dari basis data.
c. *UPDATE*, untuk melakukan pembaruan data basis data.
d. *DELETE*, untuk menghapus informasi dari database.
e. *INSERT INTO*, untuk menyisipkan data baru.
f. *CREATE TABLE*, untuk membuat tabel baru.
g. *DROP TABLE*, untuk menghapus tabel.
h. *ALTER DATABASE*, untuk memodifikasi database.
i. *ALTER TABLE*, untuk memodifikasi tabel.



Gambar 2.3 Contoh Tampilan *Mysql*

C. Kerangka Berpikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Pendekatan yang digunakan dengan teknik klasifikasi untuk mengelompokkan data sehingga menghasilkan nilai yang lebih stabil.

B. Waktu dan Tempat Penelitian

Penelitian ini memakan waktu sekitar 2 bulan yakni bulan September 2024 s.d Oktober 2024 dan dilaksanakan di Dinas Lingkungan Hidup Kota Parepare. Progres penelitian:

Tabel 3.1 Kegiatan Penelitian

BULAN SEPTEMBER **MINGGU 1-2**	
KEGIATAN	**HASIL**
- Melakukan studi literatur mengenai pengelolaan dokumen dan aplikasi yang relevan.	- Mengumpulkan referensi dari jurnal, buku, dan artikel terkait.
- Mengidentifikasi masalah yang ada di Dinas Lingkungan Hidup Kota Parepare terkait pengelolaan dokumen	- Menyusun kerangka pemikiran awal mengenai pentingnya aplikasi pengelolaan dokumen.
MINGGU 3-4	
KEGIATAN	**HASIL**
- Melakukan wawancara dengan staf Dinas Lingkungan Hidup untuk memahami proses pengelolaan dokumen yang ada	- Mendapatkan informasi mendalam mengenai alur kerja dan masalah yang ada.

Lanjutan **Tabel 3. 1**

- Mengumpulkan data mengenai jenis dokumen yang dikelola dan kendala yang dihadapi.	- Menyusun laporan awal mengenai kondisi pengelolaan dokumen saat ini.
BULAN OKTOBER **MINGGU 1-2**	
KEGIATAN	**HASIL**
- Merancang prototipe aplikasi pengelolaan dokumen berdasarkan data yang telah dikumpulkan.	- Prototipe awal aplikasi selesai dan siap untuk diuji coba.
- Mengembangkan fitur-fitur yang diperlukan dalam aplikasi, seperti penyimpanan, pencarian, dan pengarsipan dokumen.	- Mendapatkan umpan balik dari pengguna potensial mengenai desain dan fitur aplikasi.
MINGGU 3-4	
KEGIATAN	**HASIL**
- Melakukan uji coba prototipe aplikasi di Dinas Lingkungan Hidup Kota Parepare.	- Mengidentifikasi area yang perlu diperbaiki dan fitur tambahan yang diinginkan oleh pengguna.
- Mengumpulkan umpan balik dari pengguna mengenai kelebihan dan kekurangan aplikasi.	- Menyusun laporan akhir penelitian yang mencakup rekomendasi untuk pengembangan lebih lanjut.

C. Metode Pengumpulan Data

Penulis menggunakan beberapa strategi pengumpulan data dalam penelitian ini, termasuk:

1. Studi lapangan

   Studi lapangan, khususnya wawancara mendalam dan tidak terstruktur dengan informan kunci yang relevan dengan isu penelitian, digunakan untuk memperoleh data primer. Tujuan metode ini adalah untuk mengumpulkan

informasi yang menyeluruh dan terperinci. Meskipun wawancara bersifat fleksibel, prosesnya tetap mengikuti panduan dan sistematika tertentu. Informasi yang diperoleh dari informan dicatat sebagai data utama dan berfungsi sebagai acuan selama proses wawancara.

2. Studi pustaka

   Penulis melakukan studi literatur untuk mengumpulkan sumber informasi yang relevan dengan subjek yang dibahas dalam tesis ini dan untuk lebih memahami undang-undang dan peraturan terkait.

D. Alat dan Bahan Penelitian

Untuk mendukung kegiatan penelitian ini dan menyelesaikan proses penelitian, diperlukan alat dan sumber daya penelitian—seperti perangkat keras dan perangkat lunak. Adapun alat dan bahan sebagai berikut:

1. Alat
   a. Perangkat Keras (Hardware) yakni Laptop *Toshiba* dengan spesifikasi:
      1) *Device name Laptop*
      2) *Intel Processor Core i3*
      3) *Installed RAM 4.00 GB DDR 4 Memory*
      4) *512 GB HDD*
      5) *Sistem type 64-bit operating sistem, x64-based processor*
   b. Perangkat Lunak (Software)
      1) Sistem Operasi : Windows 10
      2) Editor : Visual Studio Code
      3) WEBserver : XAMPP

4) WEBBrowser : *Firefox, Microsoft Edge, dan Google Crome*

2. Bahan

   a. *MySQL*

   b. *HTML*

   c. *JavaScript*

   d. Dokumen Formulir (AMDAL, UKL-UPL, dan SPPL)

E. Tahapan Penelitian

Perencanaan penelitian, pengumpulan data, analisis data, desain, pengujian, dan implementasi adalah langkah-langkah dalam proses ini. Berikut penjelasan tahapan-tahapan tersebut:

1. Persiapan penelitian

   Pada tahap ini, peneliti mempersiapkan penelitian. Hal ini mencakup penulisan buku dan artikel tentang subjek penelitian serta perangkat lunak yang digunakan.

2. Studi *literature*

   Pada tahap ini, peneliti menyusun bibliografi, yang merupakan tinjauan pustaka. Hal ini mencakup penelusuran bahan referensi dan temuan penelitian terdahulu terkait yang telah dilakukan oleh pihak lain. Tujuannya adalah membangun landasan teoretis untuk isu yang diteliti. Peneliti dapat memahami situasi secara akurat dan ilmiah dengan menggunakan hipotesis ini sebagai landasan.

3. Pengumpulan data

   Selain mengumpulkan catatan dan melakukan pengamatan langsung di lokasi penelitian, peneliti sekarang juga mewawancarai partisipan penelitian.

4. Analisis

   Untuk mengembangkan solusi alternatif terhadap masalah tersebut, peneliti pertama-tama menilai sistem yang sedang digunakan sebelum merumuskan pertanyaan studi utama penelitian.

5. Perancangan

   Atas dasar berbagai jawaban atas kesulitan yang dirumuskan, peneliti kemudian mengembangkan desain untuk aplikasi yang ingin dikembangkannya.

6. Pengujian

   Peneliti menguji temuan setelah merancang. Mereka kembali ke langkah analisis jika terdapat kekurangan atau kelemahan dalam desain.

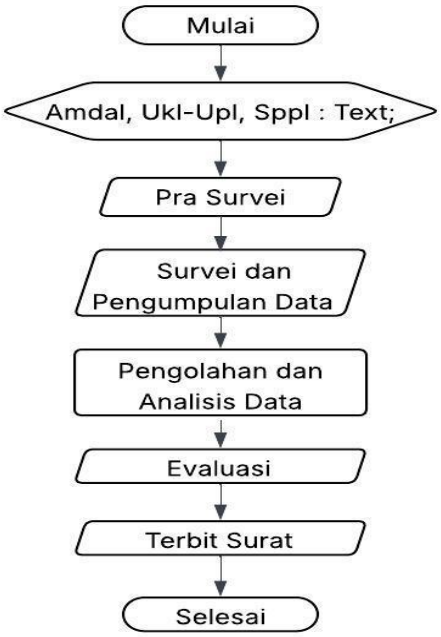
7. Implementasi

   Aplikasi disiapkan untuk penggunaan pengguna setelah semua kekurangan desain telah diperbaiki.

F. Desain Sistem

1. Sistem yang berjalan

   Adapun sistem yang berjalan saat ini yaitu:

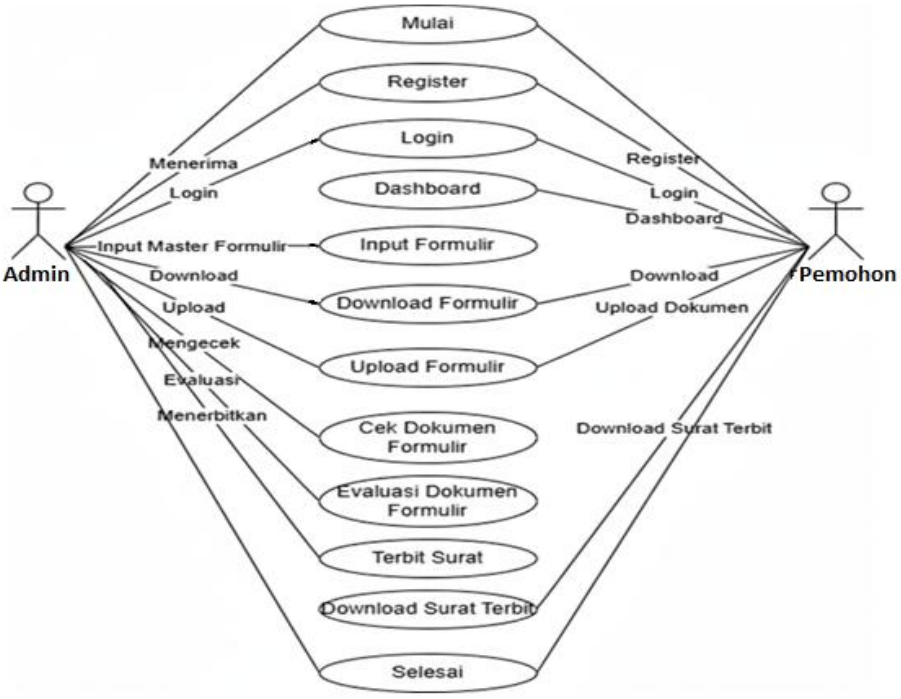


Gambar 3. 1 *Flowchart* Sistem Yang Berjalan

Sistem yang berjalan saat ini yakni pemohon mengajukan surat permohonan persetujuan lingkungan dan pemeriksaan dokumen terkait. Setelah itu dilakukan pemeriksaan terhadap dokumen dengan tahapan survei, pengumpulan data, pengolahan data dan analisis data untuk menganalisis DPSIR (*Driving Force-Pressure-State-Impact-Respon*). Pemeriksaan ini memerlukan beberapa hari kerja yang cukup lama tergantung jenis dokumen yang diperiksa. Pada tahap ini pula bisa saja terjadi penumpukan dokumen yang menyebabkan dokumen tercecer. Tahap selanjutnya dilakukan evaluasi terhadap dokumen, apabila dinyatakan memerlukan perbaikan maka dokumen akan dikembalikan kepada pemohon, Saat tahap perbaikan dan evaluasi ini tentunya membutuhkan banyak kertas untuk mencetak dokumen, yang nantinya dapat menyebabkan banyak sampah kertas. Kemudian apabila sudah tidak memerlukan perbaikan, DLH menerbitkan surat persetujuan dan surat izin lingkungan hidup.

2. Sistem yang diusulkan

Adapun sistem yang diusulkan dengan UML yaitu:



Gambar 3.2 *UML* Sistem Yang Diusulkan

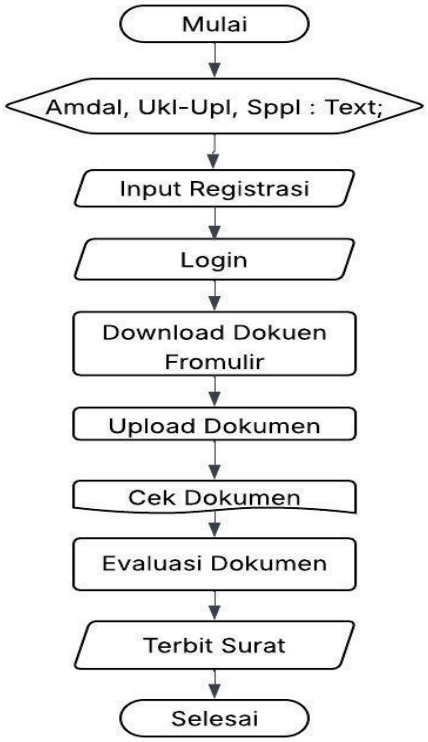
Tabel 3. 2 Keterangan gambar UML dan Uraian

NO	SIMBOL USECASE	URAIAN
1	Admin	Admin ialah staf dinas lingkungan hidup yang menerima dan mengonfirmasi dokumen yang masuk
2	Pemohon	Pemohon ialah orang yang mengajukan dokumen untuk meminta izin
3	Registrasi	-Admin menerima registrasi yang diisikan oleh pemohon -Pemohon mengisi registrasi sebelum melakukan login
4	Download formulir	-Pemohon mendownload formulir dokumen terlebih dahulu, setelah itu mengisi semua data yang diperlukan guna kelengkapan berkas.
5	Upload Formulir	-Pemohon meng-upload formulir dokumen yang telah diisi dan menunggu hasil pengecekan admin apakah statusnya diterima, ditolak, di pending atau di proses.

Lanjutan **Tabel 3. 2**

NO	SIMBOL USECASE	URAIAN
6	Mengecek Dokumen	-Admin mendata dokumen yang disetor oleh pemohon -Pemohon menyetor dokumen untuk dilakukan pengecekan
7	Evaluasi Dokumen	-Admin mengevaluasi dan mengomentari dokumen yang kurang lengkap -Pemohon melakukan revisi dari evaluasi yang diarahkan oleh admin
8	Terbit Surat	-Admin menerbitkan surat izin -Pemohon menerima surat izin

Flowchart Sistem yang di usulkan:



Gambar 3. 3 *Flowchart* Sistem Yang Diusulkan

Sistem yang diusulkan yakni ada 2 *aktor* masing-masing memliki peran dimana admin (operator) dapat login dan mengelola data registrasi dimana sebelumnya pemohon terlebih dahulu daftar untuk registrasi (yang mencakup nama, email, username, dan password). Kemudian tahap berikutnya pemohon menyetor

dokumen dan dokumen pelengkap lainnya, dan admin dapat mendata atau mengecek kelengkapan dokumen pemohon terlebih dulu. Lalu ada tahap evaluasi dokumen, admin memiliki wewenang untuk mengevaluasi dokumen yang apabila ada perbaikan akan ditambahkan komentar pada dokumen tersebut, sedangkan pemohon dapat mengecek apa saja yang harus diperbaiki. Pada tahap akhir admin akan mengevaluasi atau memverifikasi dokumen baru yang sudah di upload kembali oleh pemohon dan begitu surat izin statusnya di terima maka dokumen tersebut dinyatakan sudah terbit.

Contohnya dalam registrasi perizinan AMDAL (Analisis Mengenai Dampak Lingkungan), pemohon perlu menginputkan beberapa informasi penting, antara lain:

1. Identitas Pemohon: Informasi pribadi dan kontak pemohon seperti nama, alamat, nomor telepon, dan email.
2. Data Proyek: Informasi mengenai proyek yang akan dilakukan, termasuk lokasi proyek, deskripsi proyek, tujuan proyek, dan skala proyek.
3. Konsep Pengelolaan Lingkungan: Rencana mengenai bagaimana proyek akan dikelola dan pengendalian dampak lingkungan yang diantisipasi.
4. Rencana Konsultasi Publik: Rencana pemohon untuk melibatkan masyarakat dalam proses pengambilan keputusan mengenai dampak lingkungan proyek.
5. Rencana Pemantauan Lingkungan: Rencana yang menjelaskan bagaimana pemantauan dampak lingkungan proyek akan dilakukan secara berkala.

6. Dokumen Pendukung: Pemohon juga perlu melampirkan dokumen pendukung seperti peta lokasi proyek, studi kelayakan, dan analisis dampak lingkungan terkait proyek tersebut.

Proses registrasi ini bertujuan untuk memastikan bahwa pemohon telah menyediakan informasi yang diperlukan untuk melakukan analisis dampak lingkungan secara komprehensif dan mematuhi peraturan yang berlaku.

Form Registrasi
Nama User: Alamat Email: Username: Password: Konfirmasi Password:
Simpan

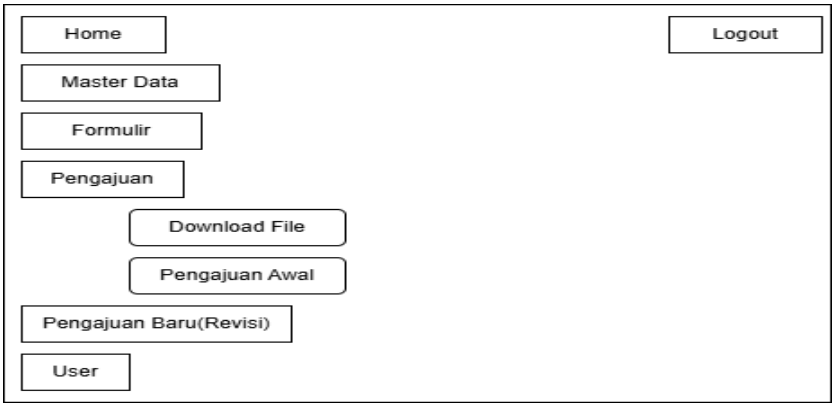
Gambar 3. 4 Form Registrasi

Setelah melakukan registrasi user baru boleh melakukan Login untuk masuk ke menu utama.

Form Login
Username: Password:
login

Gambar 3. 5 Form Login

Selanjutnya setelah user dan password sudah benar akan lanjut ke halaman menu utama untuk memilih pengisian form.



Gambar 3. 6 Tampilan Menu Setelah Login

Setelah memilih menu pengisian form kemudian mengisi form dokumen yang dipilih, diantaranya AMDAL, UKL-UPL atau SPPL.

Form Pengisian Dokumen
Nama Proyek: Alamat Proyek: Nomor Izin Lingkungan: Pemrakarsa Proyek: Konsultan Amdal: Tanggal Pengisian: A. Identitas Proyek: Deskripsi Proyek: Lokasi Proyek: a. Kabupaten/Kota: b. Kecamatan: c. Desa/Kelurahan: B. Tinjauan Awal Lingkungan: Iklim: a. Curah Hujan: b. Suhu Rata-rata: c. Kelembaban Rata-rata: Geologi: a. Jenis Tanah: b. Topografi: Hidrologi: a. Kondisi Drainase:

b. Sumber Air:
c. Kualitas Air:
Ekosistem:
a. Tipe Vegetasi:
b. Keanekaragaman Hayati:
c. Jenis-jenis Satwa:

C. Tinjauan Dampak Lingkungan:
Dampak Terhadap Udara:
a. Emisi Gas:
b. Debu Partikulat:
c. Kualitas Udara Sekitar:
Dampak Terhadap Air:
a. Kuantitas Air Permukaan:
b. Kualitas Air Permukaan:
c. Kualitas Air Tanah:
Dampak Terhadap Tanah:
a. Kualitas Tanah:
b. Penggundulan Lahan:
c. Erosi Tanah:
Dampak Terhadap Vegetasi dan Satwa:
a. Kerusakan Habitat:
b. Pemusnahan Spesies:
c. Gangguan pada Ekosistem:

D. Upaya Pengelolaan dan Penanggulangan Dampak:
Rancangan Pengelolaan Dampak Lingkungan:
Langkah Pemantauan dan Evaluasi:
Rencana Tindakan Darurat:

Simpan

Gambar 3. 7 Form Pengisian Dokumen

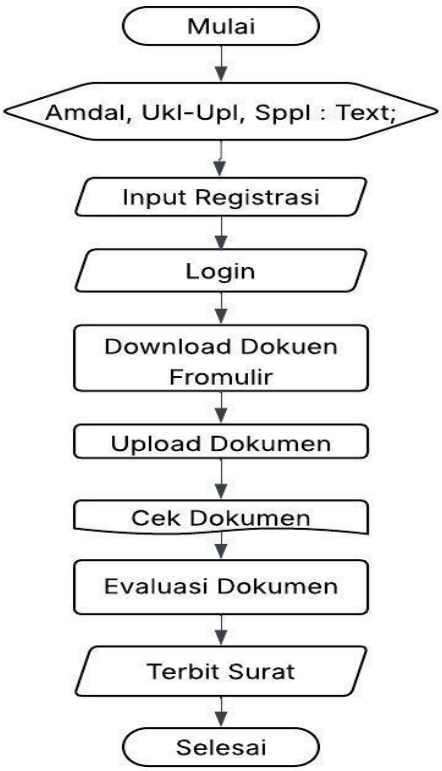
Pada form ini pemohon mengisi informasi-informasi mengenai hal yang diperlukan sebagai bahan pertimbangan dampak yang akan timbul dari adanya proyek/usaha yang akan dibuat.

BAB IV

HASIL DAN PEMBAHASAN

A. Sistem Yang Diusulkan

1. *Flowchart*



Gambar 4. 2 *Sistem Yang Diusulkan*

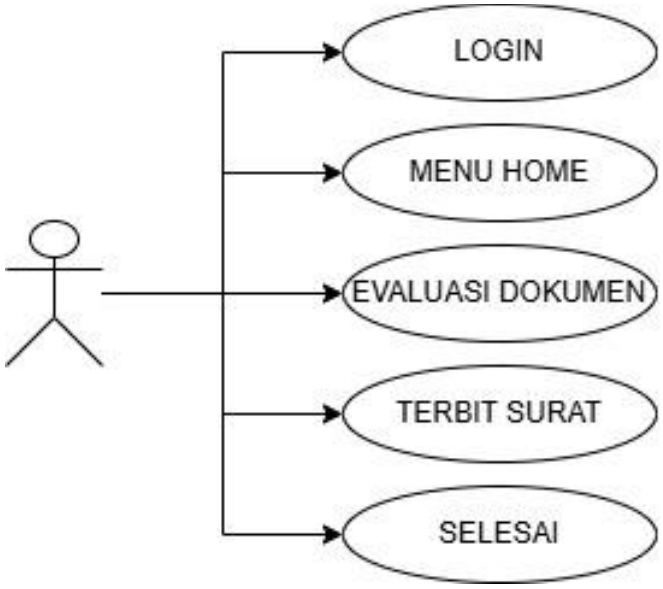
Tabel 4.1 Penjelasan *Use Case* Diagram Aplikasi

Nama	Deskripsi
Admin, User	Aktor
Login	*Use Case* yang menjelaskan proses login sebagai Admin dan User
Input Registrasi	*Use Case* yang menjelaskan proses input data, edit, hapus data
Evaluasi dokumen	*Use Case* menjelaskan Dokumen Keseluruhan

Lanjutan **Tabel 4. 1**

Nama	Deskripsi
Terbit Surat	*Use Case* Menjelaskan proses Penerbitan Surat
Selesai	*Use Case* menjelaskan selesainya proses input data

2. *Use Case* Diagram dengan *Login* Admin



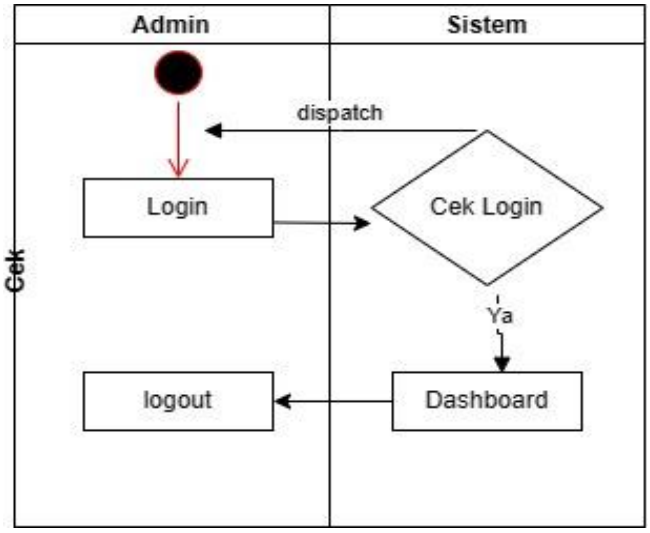
Gambar 4. 3 *Login Admin*

Tabel 4.2 Penjelasan *Use Case* Diagram *Login* Admin Input Dokumen

Nama *UC* Aplikasi	**Deskripsi *UC* Aplikasi**
Admin	Aktor
Login	*Use Case* yang menjelaskan proses *login* sebagai Admin
Input Data dokumen	*Use Case* yang menjelaskan *input* data dokumen
Selesai	*Use Case* menjelaskan selesainya proses *login* Admin

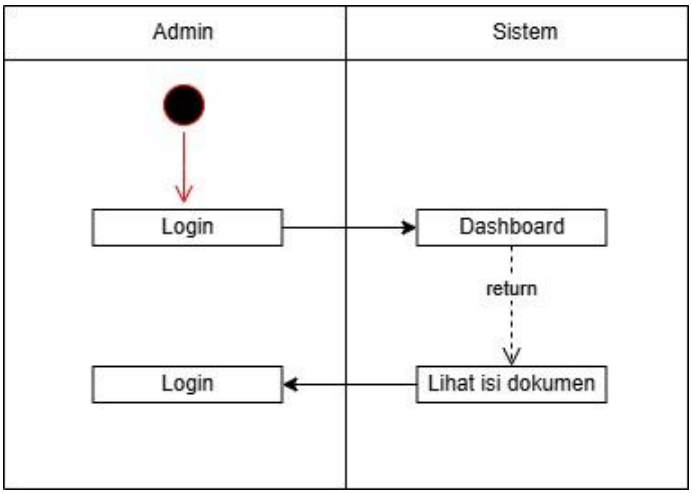
3. *Activity Diagram*

   a. Admin *Login*



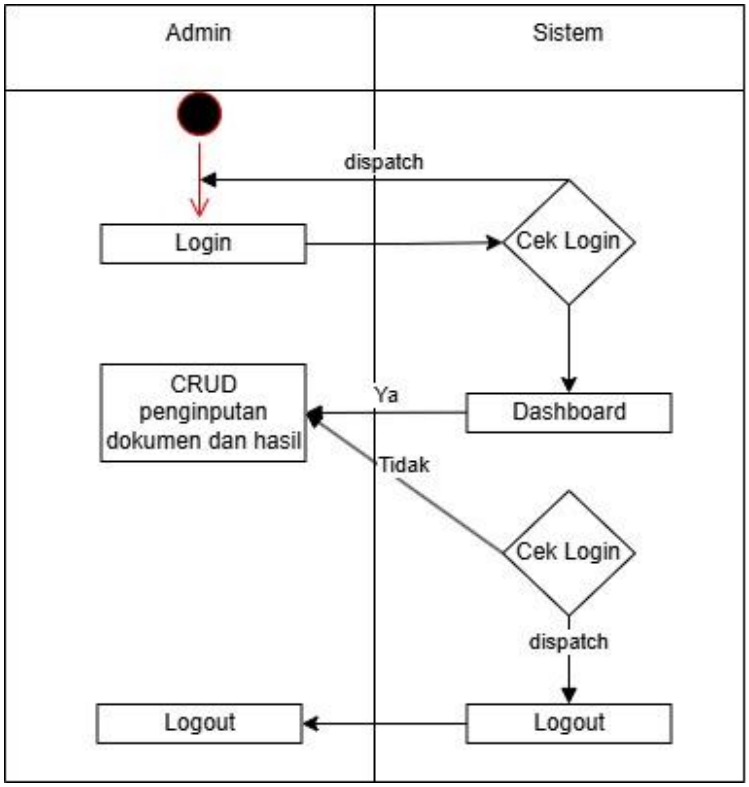
Gambar 4. 4 *Activity diagram login admin*

b. User *Login*



Gambar 4. 5 *Activity diagram login user*

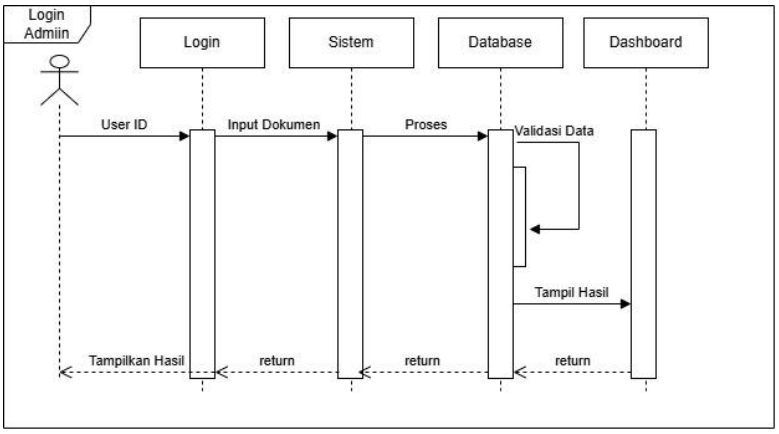
c. *Admin Input*



Gambar 4. 6 *Activity diagram input dokumen*

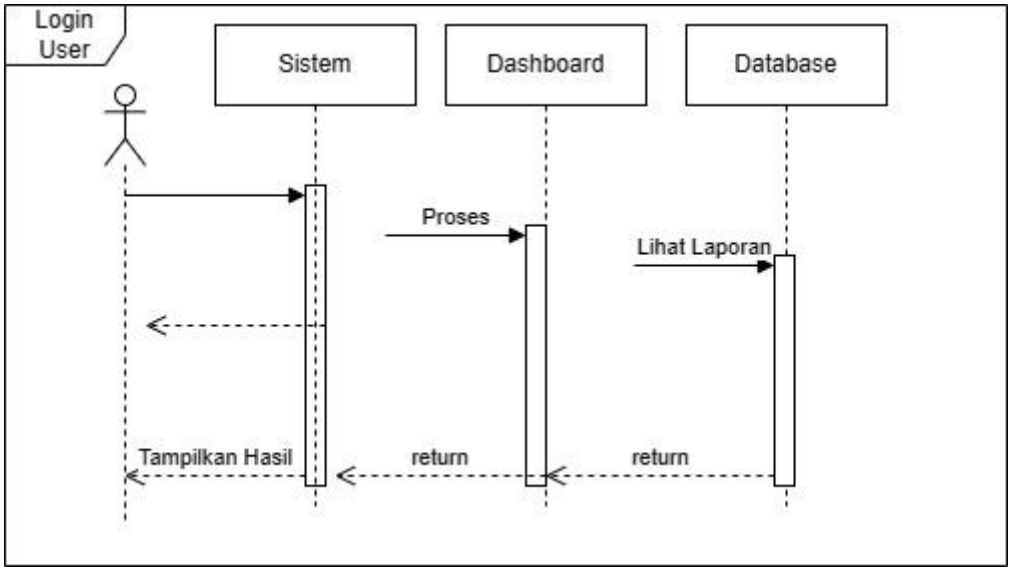
4. *Diagram Sequence*

a. *Admin Login*



Gambar 4. 7 *Sequence Diagram Login Admin*

b. *Login* User



Gambar 4. 8 *Sequence Diagram Login User*

B. Kamus Data

Tabel 4. 3 Tabel User

Nama Tabel		*user*		
No	**Nama**	***Type***	**Size**	**Keterangan**
1	id_user	*Int*	11	*Primary key*
2	username	*varchar*	255	
3	password	*varchar*	255	
4	nama_pengguna	*varchar*	255	
5	email	*varchar*	50	
6	foto	*varchar*	250	
7	login_session_key	*varchar*	255	
8	email_status	*varchar*	255	
9	password_reset_key	*varchar*	255	
10	roles	*varchar*	255	
11	user_role_id	*Int*	11	
12	id_jabatan	*Int*	11	
13	id_departemen	*int*	11	

Tabel 4. 4 Pengajuan

Nama Tabel		*pengajuan*		
No	**Nama**	***Type***	**Size**	**Keterangan**
1	id_pengajuan	*Int*	11	*Primary key*
2	tipe_pengajuan	*varchar*	50	
3	file_upload	*varchar*	255	
4	status	*varchar*	255	
5	pemohon	*varchar*	255	
6	tgl_pengajuan	*datetime*	-	
7	isi_revisi	*varchar*	255	
8	id_user	*varchar*	150	
9	nama_instansi	*varchar*	255	
10	penanggungjawab	*varchar*	255	
11	jenis_usaha	*varchar*	150	
12	persyaratan	*varchar*	150	
13	lokasi	*varchar*	255	
14	luas_lahan	*double*	-	
15	dampak_signifikan	*varchar*	255	
16	potensi	*varchar*	255	
17	upaya_pengelolaan	*varchar*	255	
18	pernyataan_kesangupan	*varchar*	255	
19	tgl_pernyataan	*datetime*	-	
20	kategori	*varchar*	255	
21	ktp	*varchar*	50	
22	npwp	*varchar*	50	
23	akte_perusahaan	*varchar*	150	
24	amdal	*varchar*	150	
25	surat_izin_usaha	*varchar*	150	
26	sertifikat_keahlian	*varchar*	150	
27	rencana_pengelolaan	*varchar*	150	
28	surat_pernyataan	*varchar*	255	

Tabel 4. 5 Pemohon

Nama Tabel		*pemohon*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_pemohon	*Int*	11	*Primary key*
2	id_user	*Int*	11	
3	nama_pemohon	*varchar*	255	
4	jenis_pemohon	*varchar*	150	
5	nik_npwp	*varchar*	150	
6	alamat	*varchar*	255	
7	telepon	*float*	-	
8	email	*varchar*	255	
9	kontak_person	*float*	-	
10	created_at	*timestamp*	-	
11	updated_at	*timestamp*	-	

Tabel 4. 6 Formulir

Nama Tabel		*formulir*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_formulir	*Int*	11	*Primary key*
2	kode_formulir	*varchar*	50	
3	tipe_formulir	*varchar*	255	
4	file_formulir	*varchar*	255	

Tabel 4. 7 Kategori_dokumen

Nama Tabel		*kategori_dokumen*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_kategori	*Int*	11	*Primary key*
2	nama_kategori	*varchar*	255	
3	deskripsi	*varchar*	255	
4	created_at	*timestamp*	-	
5	updated_at	*timestamp*	-	
6	kode_kategori	*varchar*	20	

Tabel 4. 7 Jenis_dokumen

Nama Tabel		*jenis_dokumen*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_jenis	*Int*	11	*Primary key*
2	nama_jenis	*varchar*	50	
3	persyaratan	*text*	-	
4	template_dokumen	*varchar*	255	
5	id_kategori	*int*	11	
6	created_at	*timestamp*	-	
7	update_at	*timestamp*	-	

Tabel 4. 9 Arsip

Nama Tabel		*arsip*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_arsip	*Int*	11	*Primary key*
2	id_dokumen	*Int*	11	*Secondary key*
3	klasifikasi_arsip	*varchar*	150	
4	retensi_tahun	*varchar*	150	
5	lokasi_fisik	*varchar*	255	
6	kondisi_dokumen	*varchar*	255	
7	tanggal_arsip	*date*	-	
8	tanggal_musnah	*date*	-	
9	status_arsip	*varchar*	255	
10	created_at	*timestamp*	-	
11	update_at	*timestamp*	-	

Tabel 4. 10 Departemen

Nama Tabel		*departemen*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_departemen	*Int*	11	*Primary key*
2	nama_departemen	*varchar*	255	
3	kepala_departemen	*varchar*	255	
4	deskripsi	*varchar*	255	
5	created_at	*timestamp*	-	
6	updated_at	*timestamp*	-	

Tabel 4. 11 Disposisi

Nama Tabel		*disposisi*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_disposisi	*Int*	11	Primary key
2	id_dokumen	*Int*	11	
3	id_user_pengirim	*Int*	11	
4	id_user_penerima	*Int*	11	
5	id_departemen	*Int*	11	
6	instruksi_disposisi	*varchar*	255	
7	prioritas	*varchar*	150	
8	batas_waktu	*datetime*	-	
9	status_disposisi	*varchar*	255	
10	tangggal_disposisi	*datetime*	-	
11	tanggal_selesai	*datetime*	-	
12	created_at	*timestamp*	-	
13	updated_at	*timestamp*	-	

Tabel 4. 12 Dokumen

Nama Tabel		*dokumen*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_dokumen	*Int*	11	*Primary key*
2	id_user	*Int*	11	*Secondary key*
3	id_jenis	*Int*	11	*Secondary key*
4	nomor_dokumen	*varchar*	50	
5	judul_dokumen	*varchar*	255	
6	file_path	*varchar*	255	
7	ukuran_file	*varchar*	255	
8	fomat_file	*varchar*	255	
9	tanggal_upload	*date*	-	
10	status_dokumen	*varchar*	50	
11	versi_dokumen	*varchar*	50	
12	keterangan	*varchar*	255	
13	created_at	*timestamp*	-	
14	updated_at	*timestamp*	-	

Tabel 4. 13 Jabatan

Nama Tabel		*jabatan*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_jabatan	*int*	11	*Primary key*
2	deskripsi_jabatan	*varchar*	255	
3	level_akses	*varchar*	50	
4	nama_jabatan	*varchar*	50	
5	created_at	*timestamp*	-	
6	update_at	*timestamp*	-	

Tabel 4. 14 Komentar

Nama Tabel		*komentar*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_komentar	*Int*	11	*Primary key*
2	id_user	*Int*	11	*Secondary key*
3	isi_komentar	*varchar*	255	
4	lampiran_komentar	*varchar*	255	
5	tanggal_komentar	*datetime*	-	
6	created_at	*timestamp*	-	
7	updated_at	*timestamp*	-	

Tabel 4. 15 Notifikasi

Nama Tabel		*notifikasi*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_notifikasi	*Int*	11	*Primary key*
2	id_user	*Int*	11	*Secondary key*
3	id_dokumen	*Int*	11	*Secondary key*
4	jenis_notifikasi	*varchar*	255	
5	judul_notifikasi	*varchar*	255	
6	pesan	*varchar*	255	
7	status_baca	*varchar*	150	
8	tanggal_notifikasi	*date*	-	
9	updated_at	*timestamp*	-	

Tabel 4. 16 Roles

Nama Tabel		*roles*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	role_id	*Int*	11	*Primary key*
2	role_name	*varchar*	255	*Secondary key*

Tabel 4. 17 Role_permissions

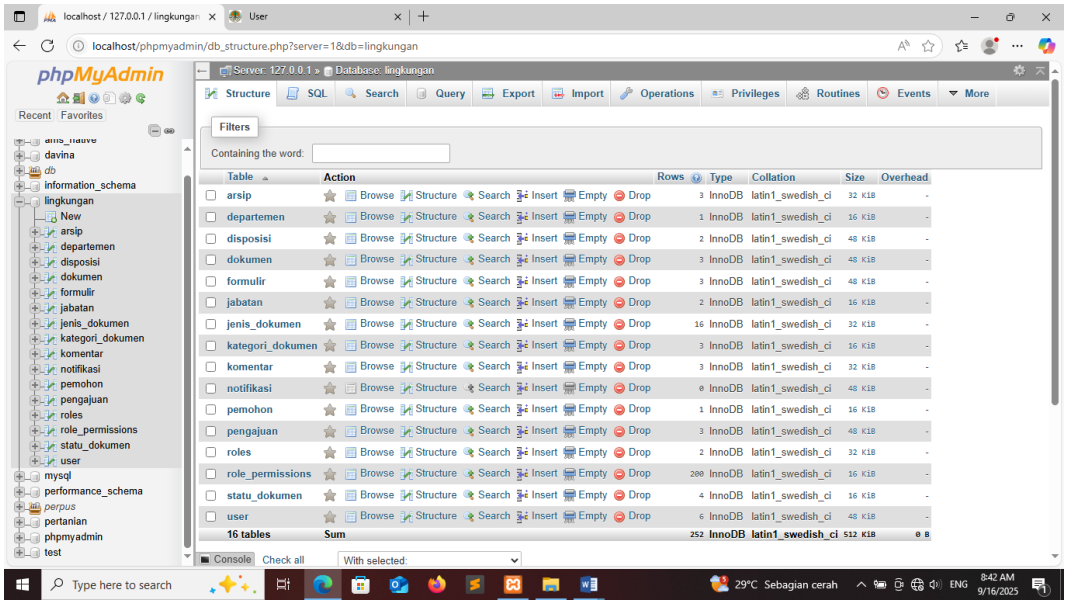
Nama Tabel		*role_permissions*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	permission_id	*Int*	11	*Primary key*
2	role_id	*Int*	11	
3	page_name	*varchar*	255	
4	action_name	*varchar*	255	

Tabel 4. 18 Status_dokumen

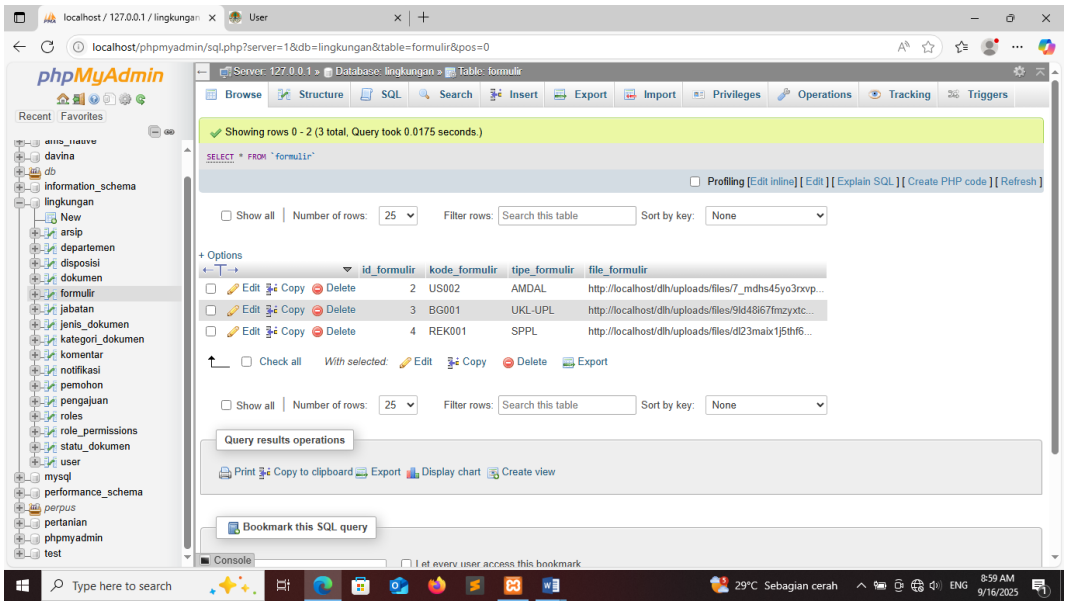
Nama Tabel		*status_dokumen*		
No	**Nama**	**Type**	**Size**	**Keterangan**
1	id_status	*Int*	11	*Primary key*
2	nama_status	*varchar*	150	
3	deskripsi_status	*varchar*	255	
4	warna_status	*varchar*	150	
5	cerated_at	*timestamp*	-	
6	updated_at	*timestamp*	-	

C. Rancangan Database

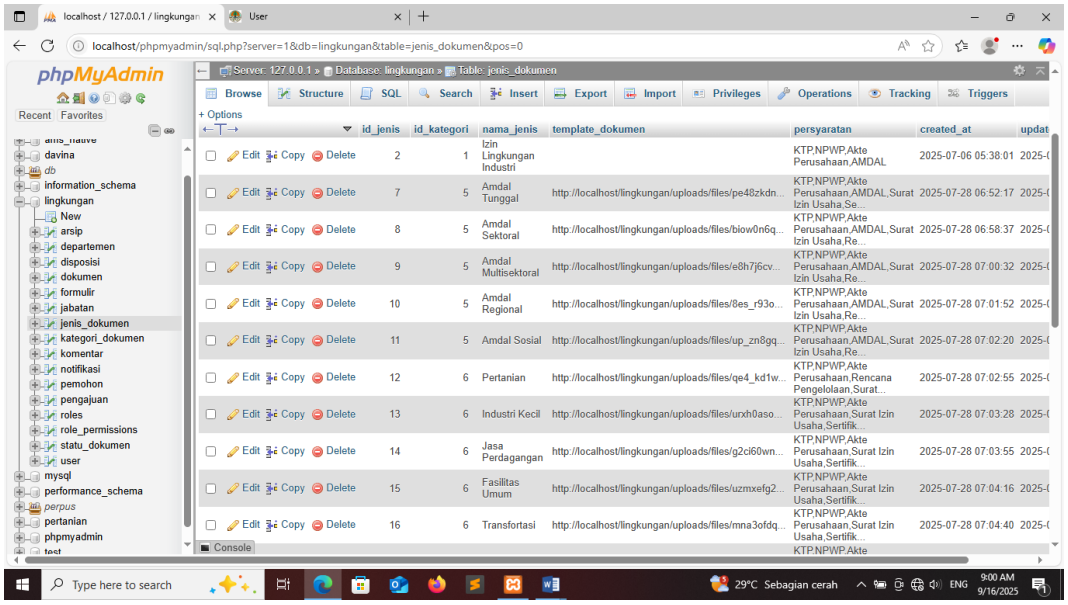
Berikut ini rancangan *database* untuk membuat APLIKASI PENGELOLAAN DOKUMEN PADA DINAS LINGKUNGAN HIDUP KOTA PAREPARE.



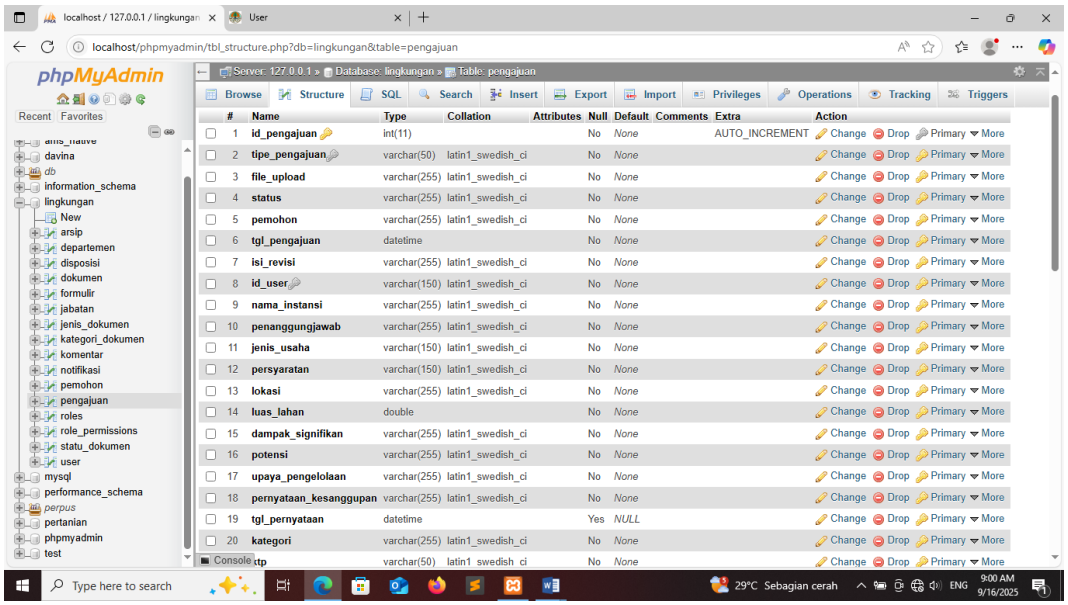
Gambar 4. 9 Tampilan Database Lingkungan



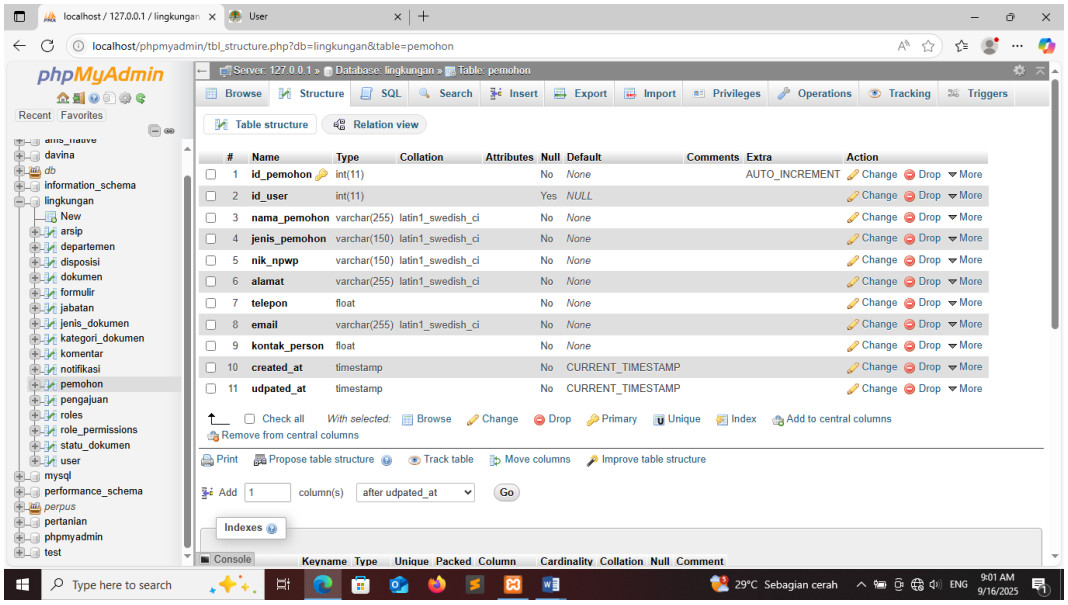
Gambar 4. 10 Tampilan Tabel Formulir



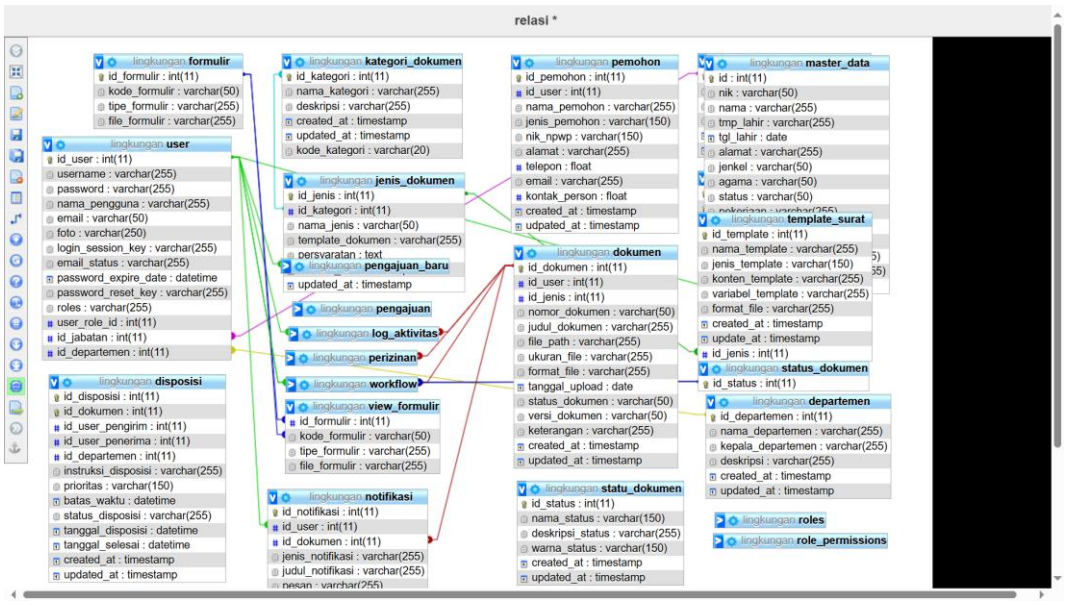
Gambar 4. 11 Tampilan Tabel Jenis Dokumen



Gambar 4. 12 Tampilan Tabel Pengajuan



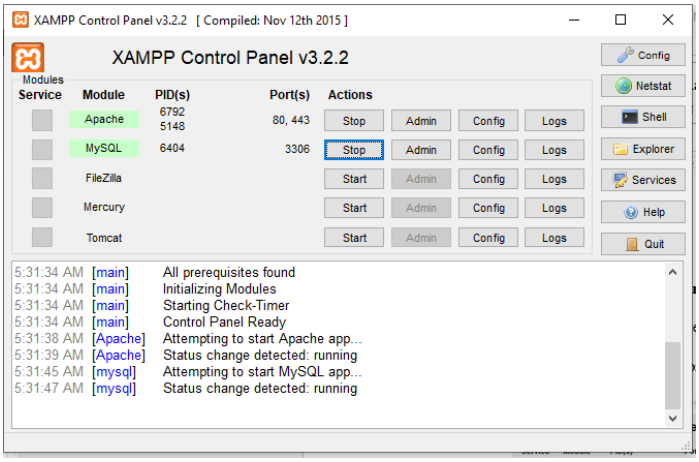
Gambar 4. 13 Tampilan Tabel Pemohon



Gambar 4. 14 Tampilan Relasi Antar Tabel

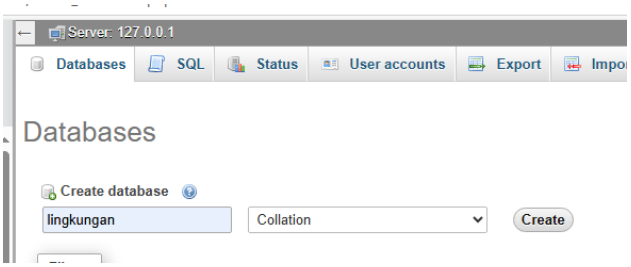
D. Pembuatan Aplikasi

1. Buka aplikasi *XAMPP*, kemudian aktifkan *module Apache*, *MySQL*, dengan menekan tombol *Start*.



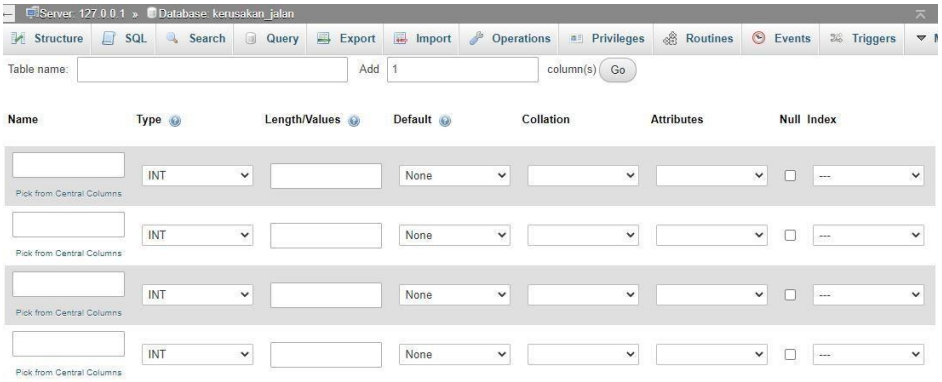
Gambar 4. 14 Mengaktifkan *XAMPP*

2. Membuat *database* dengan cara membuka *Google Chrome*, kemudian masukkan *url* *http://localhost/phpmyadmin/.* Setelah itu pilih *New*, kemudian isi nama *database* pada kolom *Create Database*.



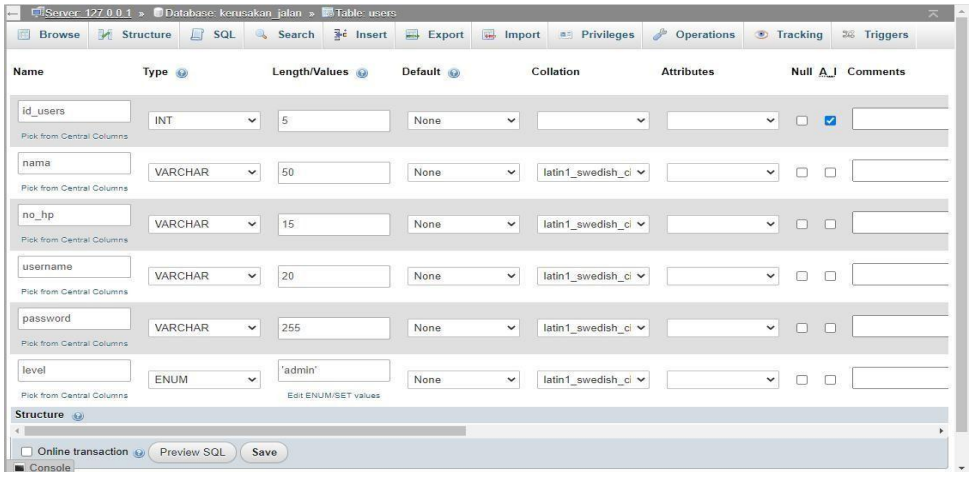
Gambar 4. 15 Membuat *Database*

3. Selanjutnya membuat tabel dengan cara memasukkan nama tabel yang ingin dibuat pada kolom *Create Table*. Kemudian pilih *Go*.



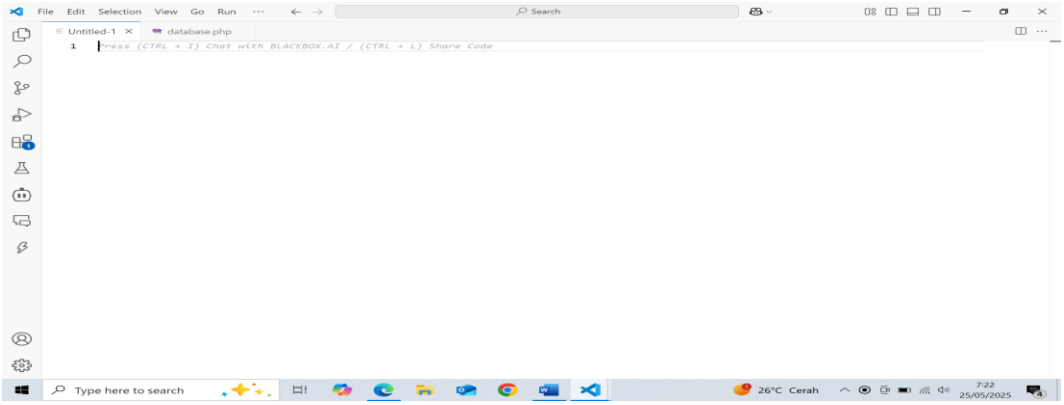
Gambar 4. 16 Membuat Tabel

4. Masukkan nama kolom sesuai yang diinginkan, kemudian tentukan *type* data dan jumlah *valuenya*, serta tentukan *primary key* pada data tersebut. Aktifkan A_I (*Auto_Increment*) untuk membuat nomor secara berurutan pada *Primary Key* lalu tekan *Save* untuk menyimpan.



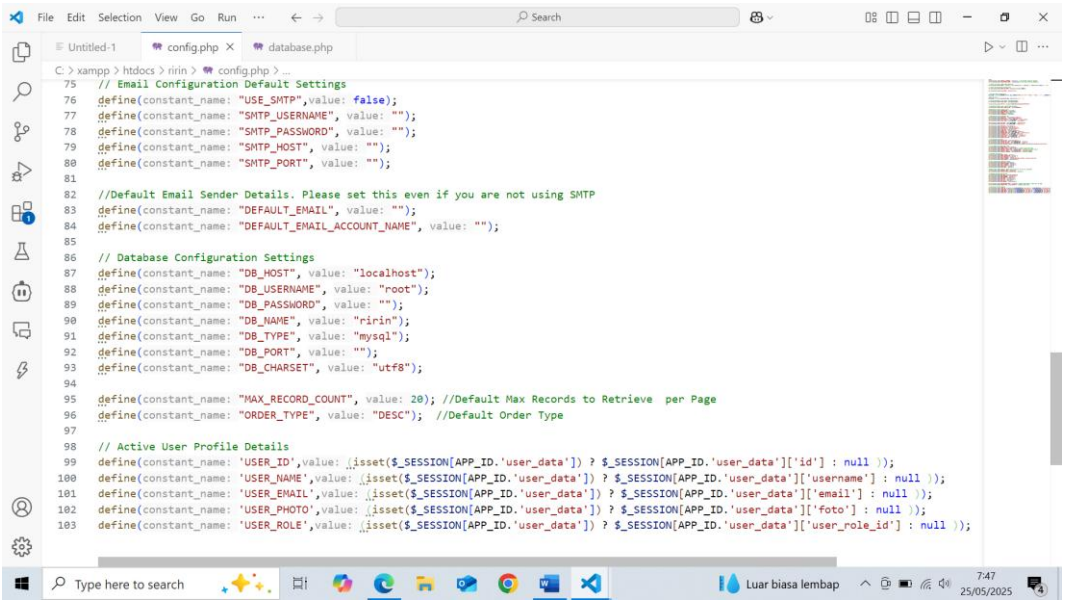
Gambar 4. 17 Membuat Isi Tabel

5. Setelah membuat *database* dan *table* selanjutnya membuat *folder* Raport di *C:\xampp\htdocs* untuk tempat menyimpan *script* aplikasi yang akan dibuat.
6. Selanjutnya buka aplikasi *visual studio code* lalu pilih *File*, *Open Folder* dan pilih folder *yang* telah dibuat, kemudian Select *Folder*.



Gambar 4. 18 Menentukan folder

7. Langkah selanjutnya adalah membuat *file* baru pada *folder conf* dengan cara (*ctrl + n*) dengan nama *file* koneksi.php untuk dihubungkan ke *database* yang telah dibuat



```
// Email Configuration Default Settings
define("USE_SMTP",false);
define("SMTP_USERNAME", "");
define("SMTP_PASSWORD", "");
define("SMTP_HOST", "");
define("SMTP_PORT", "");

//Default Email Sender Details. Please set this even if you are not using SMTP
define("DEFAULT_EMAIL", "");
define("DEFAULT_EMAIL_ACCOUNT_NAME", "");

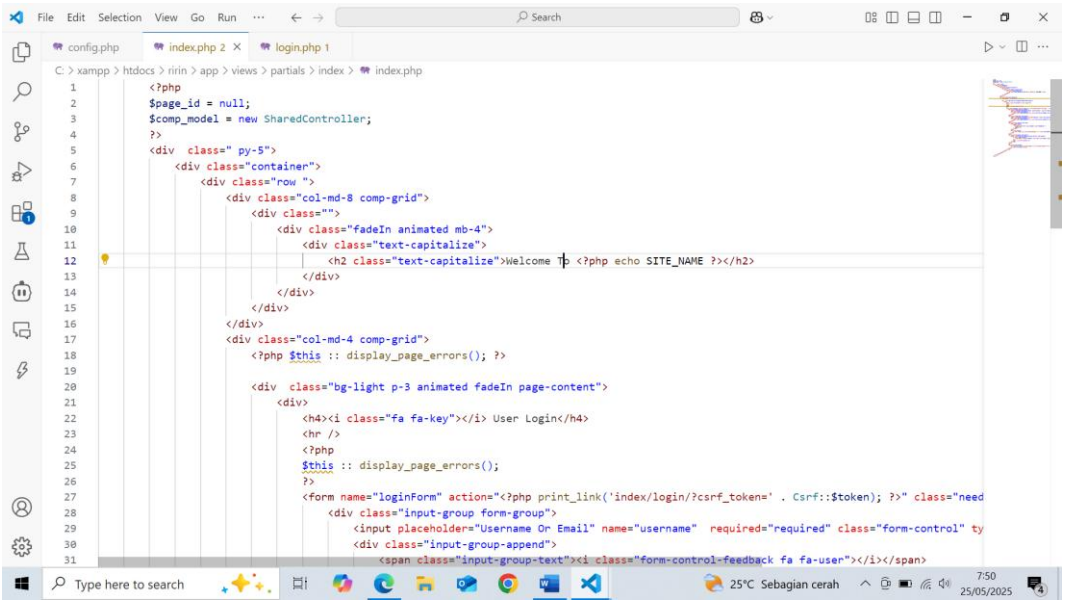
// Database Configuration Settings
define("DB_HOST", "localhost");
define("DB_USERNAME", "root");
define("DB_PASSWORD", "");
define("DB_NAME", "ririn");
define("DB_TYPE", "mysql");
define("DB_PORT", "");
define("DB_CHARSET", "utf8");

define("MAX_RECORD_COUNT", 20); //Default Max Records to Retrieve  per Page
define("ORDER_TYPE", "DESC");  //Default Order Type

// Active User Profile Details
define('USER_ID',(isset($_SESSION[APP_ID.'user_data']) ? $_SESSION[APP_ID.'user_data']['id'] : null ));
define('USER_NAME',(isset($_SESSION[APP_ID.'user_data']) ? $_SESSION[APP_ID.'user_data']['username'] : null ));
define('USER_EMAIL',(isset($_SESSION[APP_ID.'user_data']) ? $_SESSION[APP_ID.'user_data']['email'] : null ));
define('USER_PHOTO',(isset($_SESSION[APP_ID.'user_data']) ? $_SESSION[APP_ID.'user_data']['foto'] : null ));
define('USER_ROLE',(isset($_SESSION[APP_ID.'user_data']) ? $_SESSION[APP_ID.'user_data']['user_role_id'] : null ));
```

Gambar 4. 19 Membuat File conf.php

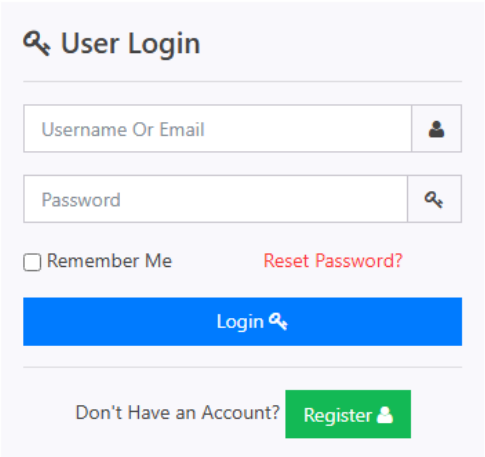
8. Membuat *file index.php* yang akan menampilkan *form login* dengan menuju *link home.php*



Gambar 4. 20 Membuat File login.php

E. Implementasi

1. Tampilan *Login*

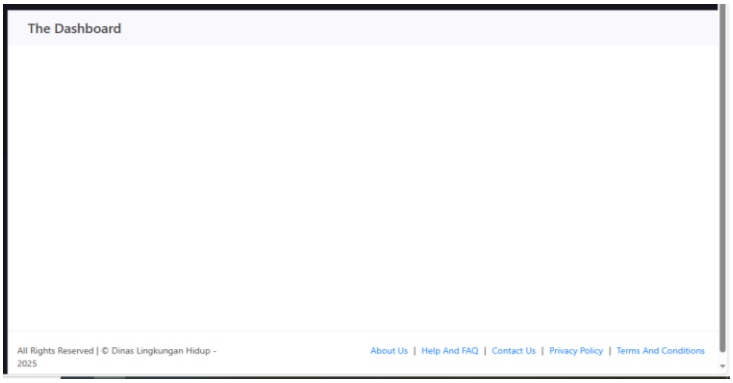


Gambar 4. 21 Tampilan Login

```
function login($formdata = null){
            if($formdata){
                  $modeldata = $this->modeldata = $formdata;
                  $username = trim($modeldata['username']);
                  $password = $modeldata['password'];
                  $rememberme                                    =
(!empty($modeldata['rememberme']) ? $modeldata['rememberme']
: false);
                  $this->login_user($username,     $password,
$rememberme);
            }
            else{
                  $this->set_page_error("Invalid request");
                  $this->render_view("index/login.php");
            }
      }
```

Tampilan yang terlihat di atas adalah halaman *login*, di mana *admin dan user* perlu memasukkan *username* dan *password* untuk mendapat akses masuk ke halaman beranda pada *website*.

2. Tampilan *Dashboard*

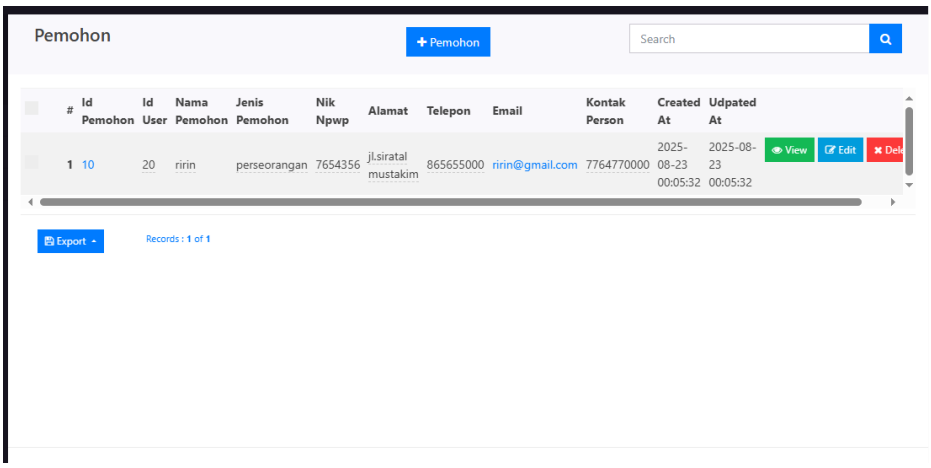


Gambar 4. 22 Tampilan Beranda

```
<div            class="collapse            navbar-collapse"
id="navbarTogglerDemo01">
                        <ul class="navbar-nav mr-auto mt-
2 mt-lg-0">
                                <li class="nav-item active">
                                        <a    class="nav-link"
href="<?php print_link(HOME_PAGE) ?>">Home</a>
                                </li>
                                <li class="nav-item active">
                                        <a    class="nav-link"
href="<?php print_link('info/about') ?>">About us</a>
                                </li>
                                <li class="nav-item active">
                                        <a    class="nav-link"
href="<?php print_link('info/help') ?>">Help and FAQ</a>
                                </li>
                                <li class="nav-item active">
                                        <a    class="nav-link"
href="<?php print_link('info/contact') ?>">Contact us</a>
                                </li>
                        </ul>
                </div>
```

Pada tampilan dashboard menampilkan halaman dashboard setelah berhasil login.

3. Tampilan Form Pemohon

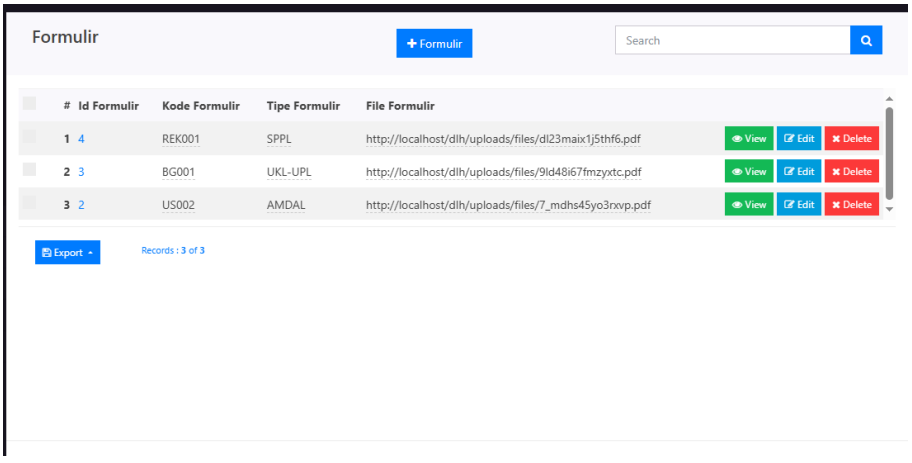


Gambar 4. 23 Form Pemohon

```
class PemohonController extends SecureController{
	function __construct(){
		parent::__construct();
		$this->tablename = "pemohon";
	}
	function index($fieldname = null , $fieldvalue = null){
		$request = $this->request;
		$db = $this->GetModel();
		$tablename = $this->tablename;
		$fields = array("id_pemohon",
			"id_user",
			"nama_pemohon",
			"jenis_pemohon",
			"nik_npwp",
			"alamat",
			"telepon",
			"email",
			"kontak_person",
			"created_at",
			"udpated_at");
		$pagination =$this->get_pagination(MAX_RECORD_COUNT);
```

Setelah pemohon berhasil login, terlebih dahulu pemohon mengisi form pemohon. Adapun yang diisi yaitu nama pemohon, jenis pemohon, nik, npwp, alamat, telepon, email, dan kontak person.

4. Tampilan View Format Dokumen



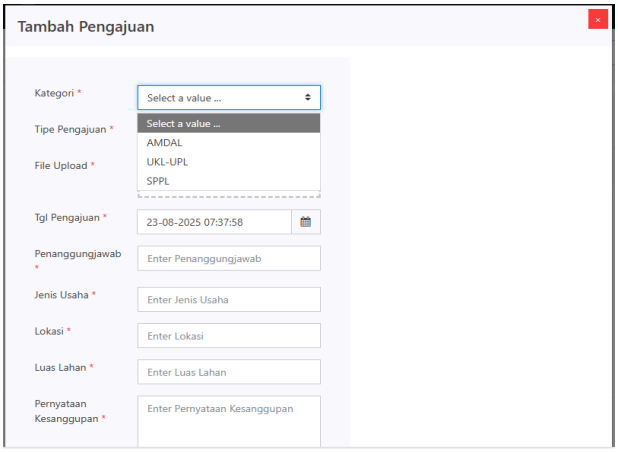
Gambar 4. 24 Tampilan view Format Dokumen

```
function index($fieldname = null , $fieldvalue = null){
            $request = $this->request;
            $db = $this->GetModel();
            $tablename = $this->tablename;
            $fields = array("kode_formulir",
                  "tipe_formulir",
                  "file_formulir");
            $pagination                =                $this-
>get_pagination(MAX_RECORD_COUNT); // get current pagination
e.g array(page_number, page_limit)
            //search table record
            if(!empty($request->search)){
                  $text = trim($request->search);
                  $search_condition = "(
                        view_formulir.id_formulir LIKE ? OR
                        view_formulir.kode_formulir LIKE ? OR
                        view_formulir.tipe_formulir LIKE ? OR
                        view_formulir.file_formulir LIKE ?
                  )";
                  $search_params = array(

      "%$text%","%$text%","%$text%","%$text%"
                  );
                  //setting search conditions
                  $db->where($search_condition,
$search_params);
                   //template to use when ajax search
                  $this->view->search_template                =
"view_formulir/search.php";
            }
```

Sebelum pemohon mengisi form pengajuan, pemohon harus mendownload format dokumen. Format dokumen yang ada yaitu AMDAL, UKL-UPL, dan SPPL.

5. Tampilan Form Pengajuan

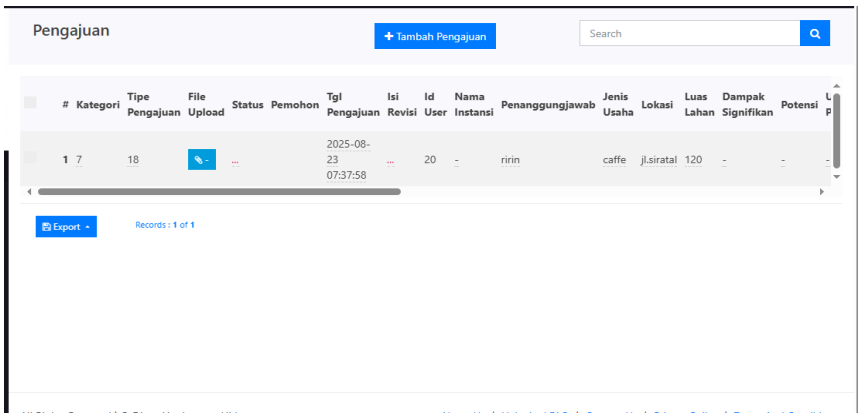


Gambar 4. 23 Tampilan Form Pengajuan

```
function __construct(){
            parent::__construct();
            $this->tablename = "pengajuan";
      }
      function index($fieldname = null , $fieldvalue = null){
            $request = $this->request;
            $db = $this->GetModel();
            $tablename = $this->tablename;
            $fields = array("id_pengajuan",
                  "kategori",
                  "tipe_pengajuan",
                  "file_upload",
                  "pemohon",
                  "tgl_pengajuan",
                  "id_user",
                  "nama_instansi",
                  "penanggungjawab",
                  "jenis_usaha",
                  "lokasi",
                  "luas_lahan",
                  "dampak_signifikan",
                  "potensi",
                  "upaya_pengelolaan",
                  "tgl_pernyataan",
                  "ktp",
                  "npwp",
                  "akte_perusahaan",
                  "amdal",
                  "surat_izin_usaha",
                  "sertfikat_keahlian",
                  "rencana_pengelolaan",
                  "surat_pernyataan");
            $pagination                =                $this-
>get_pagination(MAX_RECORD_COUNT);
```

Pada tampilan diatas user mengisi form pengajuan, diantaranya kategori, tipe pengajuan, pemohon, tanggal pengajuan, nama instansi, penanggungjawab, jenis usaha, lokasi, luas lahan, dampak signifikan, potensi, upaya pengelolaan, tanggal pernyataan, ktp, npwp, akte perusahaan, surat izin usaha, sertifikat keahlian, rencana pengelolaan, dan surat pernyataan.

6. Tampilan setelah pengajuan



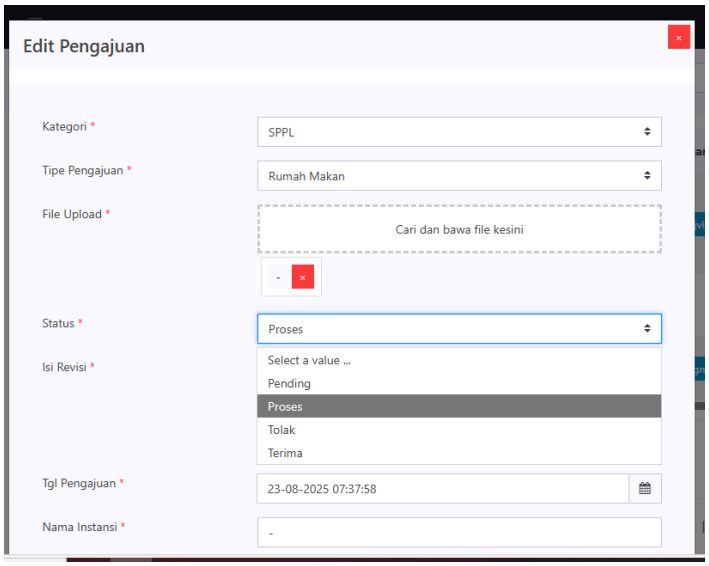
Gambar 4. 24 Tampilan Setelah Pengajuan

```
function view($rec_id = null, $value = null){
            $request = $this->request;
            $db = $this->GetModel();
            $rec_id = $this->rec_id = urldecode($rec_id);
            $tablename = $this->tablename;
            $fields = array("id_pengajuan",
                  "kategori",
                  "tipe_pengajuan",
                  "file_upload",
                  "status",
                  "pemohon",
                  "tgl_pengajuan",
                  "isi_revisi",
                  "id_user",
                  "nama_instansi",
                  "penanggungjawab",
                  "jenis_usaha",
                  "lokasi",
                  "luas_lahan",
                  "dampak_signifikan",
                  "potensi",
                  "upaya_pengelolaan",
                  "tgl_pernyataan",
                  "ktp",
                  "npwp",
                  "akte_perusahaan",
                  "amdal",
                  "surat_izin_usaha",
                  "sertfikat_keahlian",
                  "rencana_pengelolaan",
                  "surat_pernyataan");
```

```
            $allowed_roles = array ('administrator');
            if(!in_array(strtolower(USER_ROLE),
$allowed_roles)){
            $db->where("pengajuan.id_user",
get_active_user('id_user') );
            }
```

Setelah melakukan pengajuan user akan masuk pada tampilan pengajuan baru. User dapat melihat tipe pengajuan, file yang di upload, tanggal pengajuan, status, isi revisi dan pemohon. User juga dapat melihat dokumen file dengan memilih tombol view. Dan juga user dapat menghapus pengajuan dengan memlih tombol delete.

7. Tampilan Revisi Pengajuan

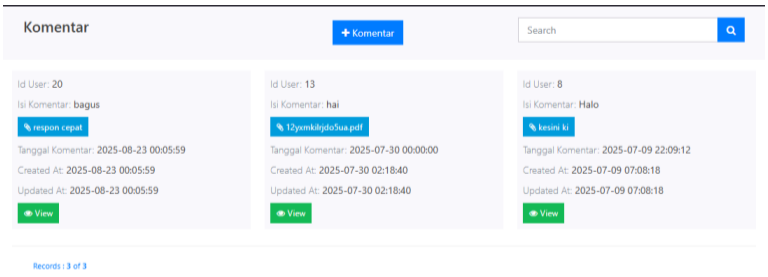


Gambar 4. 27 Tampilan Edit Pengajuan

```
if($formdata){
                    $postdata = array();
                    $fieldname = $formdata['name'];
                    $fieldvalue = $formdata['value'];
                    $postdata[$fieldname] = $fieldvalue;
                    $postdata            =            $this-
>format_request_data($postdata);
                    $this->rules_array = array(
                          'kategori' => 'required',
                          'tipe_pengajuan' => 'required',
                          'file_upload' => 'required',
                          'status' => 'required',
                          'isi_revisi' => 'required',
                          'tgl_pengajuan' => 'required',
                          'nama_instansi' => 'required',
                          'penanggungjawab' => 'required',
                          'jenis_usaha' => 'required',
                          'lokasi' => 'required',
                          'luas_lahan'                    =>
'required|numeric',
                          'dampak_signifikan' => 'required',
                          'potensi' => 'required',
                          'upaya_pengelolaan' => 'required',
                          'tgl_pernyataan' => 'required',
                          'ktp' => 'required',
                          'npwp' => 'required',
                          'akte_perusahaan' => 'required',
                          'amdal' => 'required',
                          'surat_izin_usaha' => 'required',
                          'sertfikat_keahlian'=> 'required',
                          'rencana_pengelolaan'=>'required',
                          'surat_pernyataan' => 'required',
                    );
```

Pada bagian ini admin dapat mengubah status dokumen, da memberikan revisi terkait pengajuan dari pemohon.

8. Tampilan Komentar



Gambar 4. 28 Tampilan Komentar

```
<?php
/**
 * Komentar Page Controller
 * @category  Controller
 */
class KomentarController extends SecureController{
      function __construct(){
            parent::__construct();
            $this->tablename = "komentar";
      }
      /**
     * List page records
     * @param $fieldname (filter record by a field)
     * @param $fieldvalue (filter field value)
     * @return BaseView
     */
      function  index($fieldname  =  null  ,  $fieldvalue  =
null){
            $request = $this->request;
            $db = $this->GetModel();
            $tablename = $this->tablename;
            $fields = array("id_komentar",
                  "id_user",
                  "isi_komentar",
                  "lampiran_komentar",
                  "tanggal_komentar",
                  "created_at",
                  "updated_at");
            $pagination              =              $this-
>get_pagination(MAX_RECORD_COUNT);
```

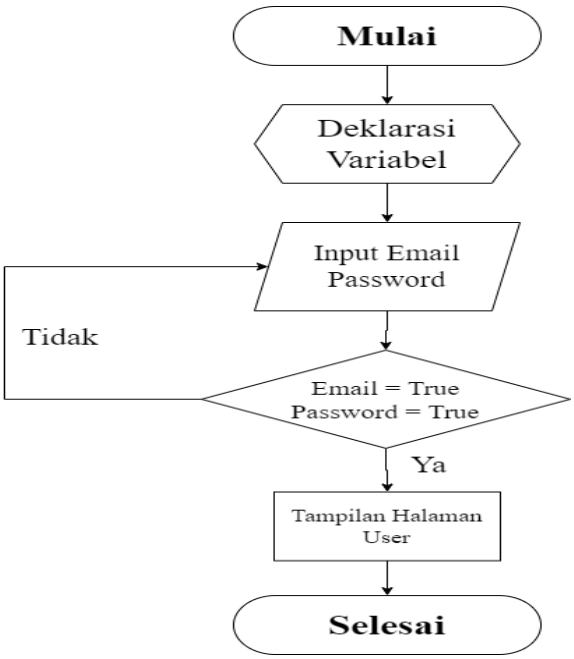
Pada bagian ini pemohon dapat meninggalkan komentar terkait aplikasi maupun pelayanan dan respon admin dalam mengelola dokumen pengajuan.

F. Pengujian Sistem

Pengujian aplikasi dengan pengujian *WhiteBox:*

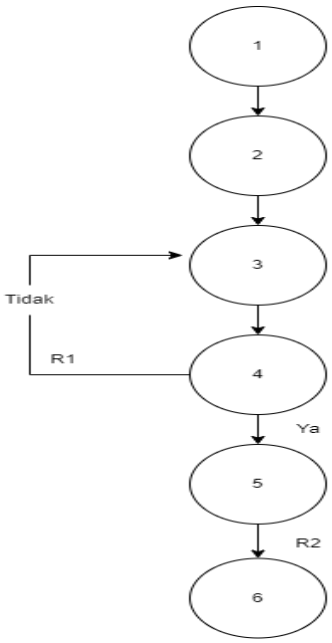
1. *WhiteBox*

   a. *Flowchart* dan *Flowgraph Login*



Gambar 4. 25 *Flowchart Aktivitas Login*

Flowgraph ditentukan berdasarkan Flowchart yang digunakan pada pengujian perangkat lunak berikut:



Gambar 4. 26 *Flowgraph Aktivitas Login*

Dari gambar flowgraph Menu Login di atas, proses perhitungan dapat dilakukan dengan cara berikut:

5) Menghitung *Cyclomatic ComplexcityV(G)* dari *Edge* dan *Node*:

   Dengan Rumus: V (G) = E – N + 2

   Dengan E (*edge*) = 6

N (*node*) = 6

Predikat *Node* (P) = 1

Penyelesaian: V(G) = E – N + 2

= 6 – 6 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

6) Berdasarkan hitungan dengan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki Region = 2

7) *Independent Path* pada *flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 3

   Path 2 = 1 – 2 – 3 – 4 – 5 – 6

8) Grafis *Matriks*

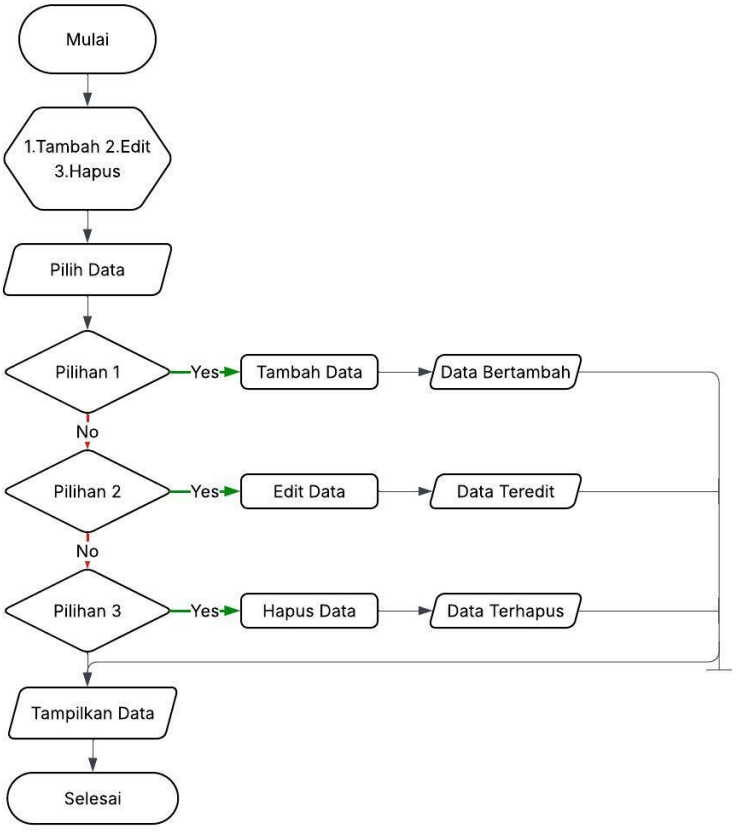
Tabel 4. 3 *Grafik Matriks dari Flowgraph Login*

	1	**2**	**3**	**4**	**5**	**6**	**E – 1**
1		1					1 – 1 = 0
2			1				1 – 1 = 0

Lanjutan **Tabel 4. 3**

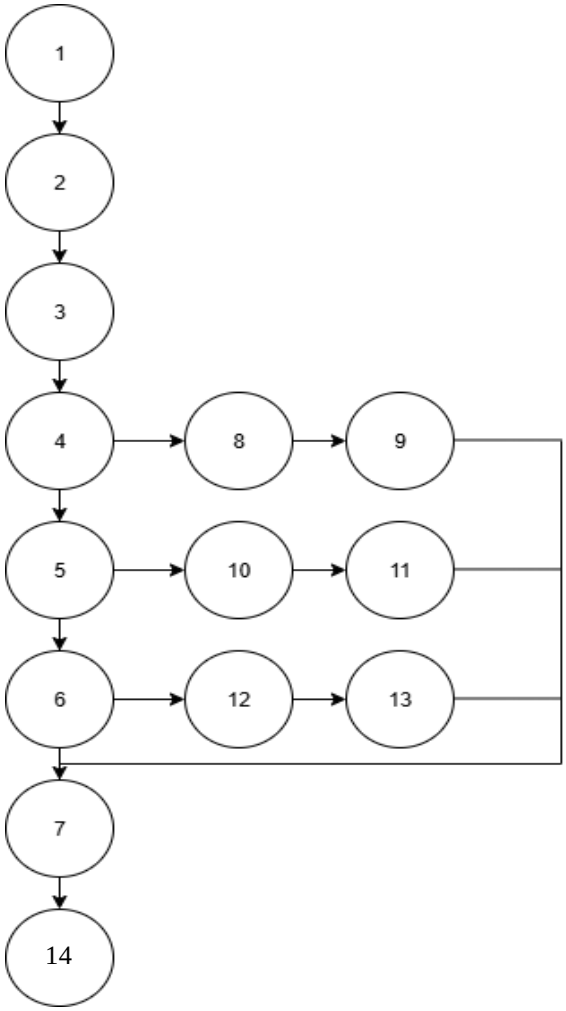
	1	**2**	**3**	**4**	**5**	**6**	**E - 1**
3				1			1 – 1 = 0
4			1		1		2 – 1 = 1
5						1	1 – 1 = 0
6							0
Sum (E) + 1							1 + 1 = 2

b. *Flowchart* dan *Flowgraph* Aktivitas *Web*



Gambar 4. 27 *Flowchart Aktivitas Web*

Dari *Flowchart* yang dipergunakan, dapat ditentukan *Flowgraph* sebagai berikut:



Gambar 4. 28 *Flowgraph Aktivitas Web*

Dari gambar *Flowgraph* diatas dilakukan perhitungan sebagai berikut:

1) Menghitung C*yclomatic Complexcity* V(G) dari *Egde* dan *Node*:

Dengan Rumus: V (G) = E – N + 2

N(*node*) = 14

E(*edge*) = 14

P (*predikat node*) = 3

Penyelesaian: V(G) = E – N + 2

= 14- 12 + 2 = 4

Predikat *Node* (N) = P + 1

= 3 + 1 = 4

2) Berdasarkan hitungan dengan C*yclomatic Complexcity* dari *Flowgraph* sebelumnya memiliki *Region* = 4

3) *Independent Path* pada *Flowgraph* diatas adalah:

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 14

Path 2 = 1 – 2 – 3 – 4 – 8 – 9 – 7 – 14

Path 3 = 1 – 2 – 3 – 4 – 5 – 10 – 11 – 7 – 14

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 12 – 13 – 7 – 14

4) Grafik *Matriks Web*

Tabel 4. 4 *Grafik Matriks Aktivitas Web*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**E-1**
1		1													1-1=0
2			1												1-1=0
3				1											1-1=0
4					1				1						2-1=1
5						1				1					2-1=1
6							1				1				2-1=1
7														1	1-1=0
8									1						1-1=0
9							1								1-1=0
10										1					1-1=0

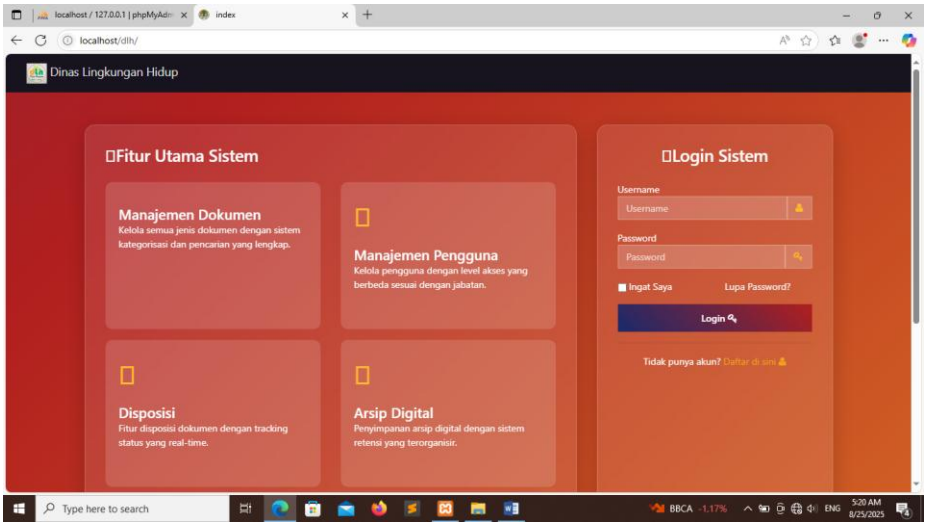
Lanjutan **Tabel 4. 4**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E - 1
11							1								1-1=0
12													1		1-1=0
13							1								1-1=0
14															
SUM(E+1)															3+1=4

2. *Blackbox*

Pengujian sistem menggunakan *BlackBox.*

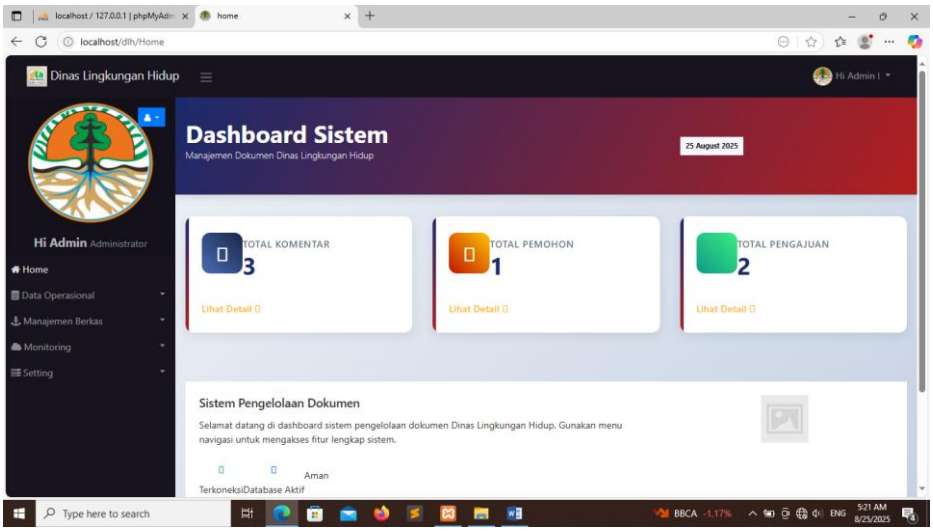
Tabel 4. 5 BlackBox *Halaman Utama*

Test Faktor	Hasil	Kesimpulan
Pengguna menjalankan aplikasi pertama kali	✓	Informasi, tampil halaman utama
Screen Shot		
		

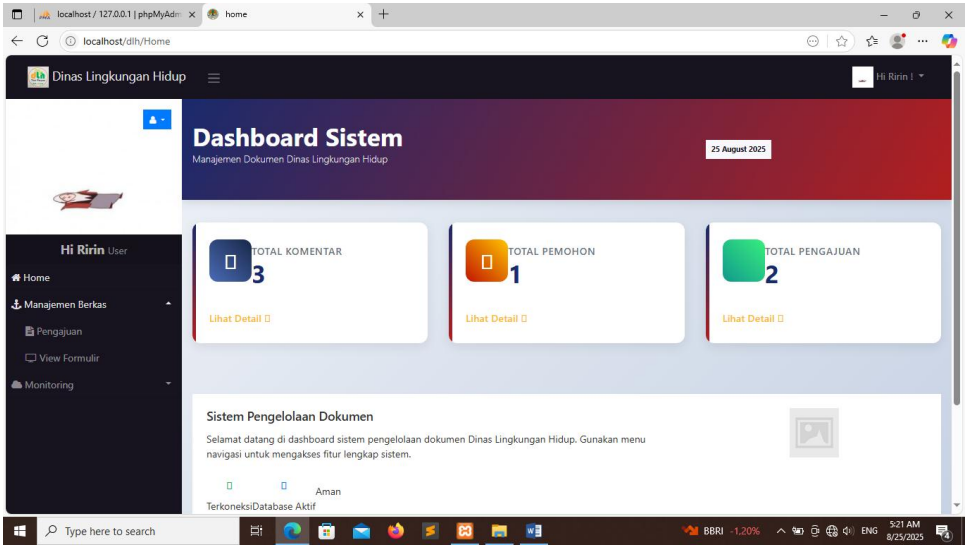
Tabel 4. 6 BlackBox *Halaman* Login *Admin*

Test Faktor	Hasil	Kesimpulan
Admin menekan tombol *Login*.	✓	Informasi, tampil halaman *Login admin*

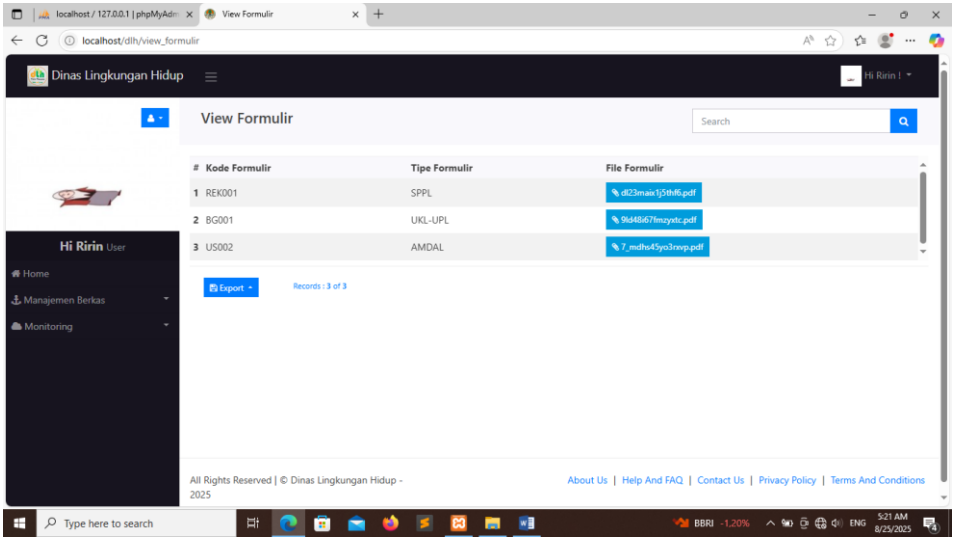
Lanjutan **Tabel 4. 6**

Screen Shot


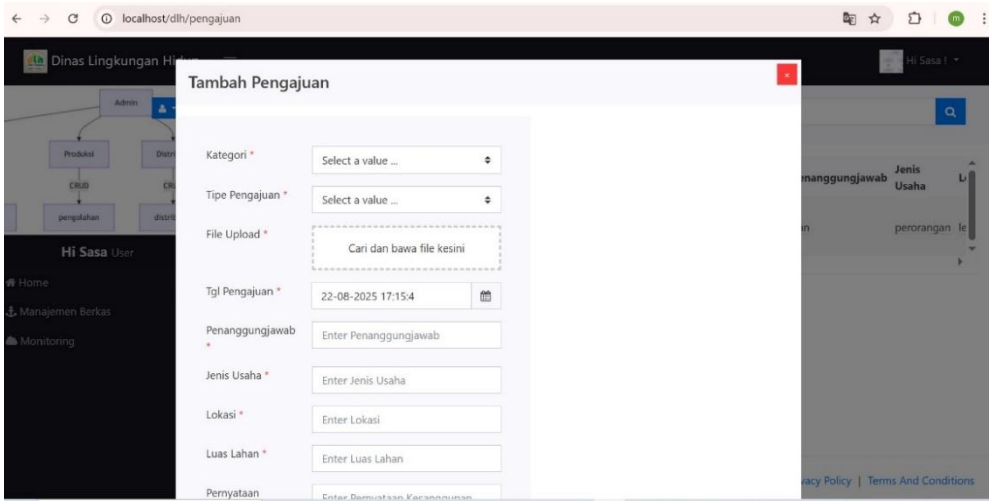
Tabel 4. 7 BlackBox *Halaman* Login *User*

Test Faktor	Hasil	Kesimpulan
User menekan tombol *Login*.	✓	Informasi, tampil halaman *Login user*
Screen Shot		
		

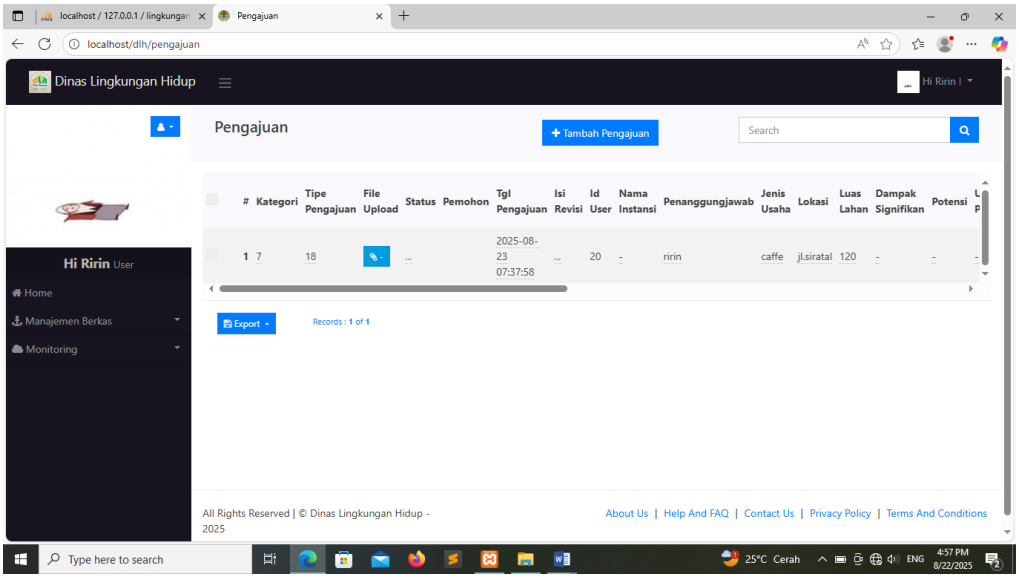
Tabel 4. 8 BlackBox *Halaman Download File Formulir*

Test Faktor	Hasil	Kesimpulan
User Menekan Tombol *Download File Formulir*	✓	Informasi, tampil halaman *Download File Formulir*
Screen Shot		
		

Tabel 4. 9 BlackBox *Halaman Pengajuan*

Test Faktor	Hasil	Kesimpulan
User Menekan Tombol *Pengujuan Awal*	✓	Informasi, tampil halaman *Pengajuan*
Screen Shot		
		

Tabel 4. 10 BlackBox *Halaman Submit*

Test Faktor	Hasil	Kesimpulan
User Menekan Tombol *Submit*	✓	Informasi, tampil halaman Telah Mengisi Pengajuan
Screen Shot		
		

BAB V

PENUTUP

A. KESIMPULAN

Berdasarkan hasil penelitian serta pengujian yang dilakukan penulis dalam pembuatan Aplikasi Pengelolaan Dokumen Pada Dinas Lingkungan Hidup Kota Parepare, dapat ditarik kesimpulan bahwa dalam aplikasi ini terdapat fitur pengisian formulir dan pengajuan dokumen yang lengkap disertai dengan persyaratannya. Selain itu dalam aplikasi ini juga dapat menampilkan revisi, yang memungkinkan pemohon dapat tahu kekurangan dari pengajuaanya. Dimana tentunya fitur-fitur ini memudahkan kepengurusan pengajuan.

B. SARAN

Penulis masih merasa banyak hal yang dapat dikembangkan. Pada aplikasi ini penulis hanya berfokus pada pengolaan dokumen terkhusus pengajuan formulir, diharapkan penulis berikutnya dapat mengembangkan aplikasi ini seperti dengan melakukan penilaian menggunakan metode yang mungkin lebih bagus lagi, akurat dan efesien.

Demikian saran yang dapat penulis berikan, semoga saran tersebut dapat dipertimbangkan sebagai bahan masukan untuk pembaca.

DAFTAR PUSTAKA

Adam Afriansyah dan Ade Oktarino. 2020. Perancangan Sistem Pelaporan Pengelolaan Lingkungan Di Dinas Lingkungan Hidup Kota Jambi Dengan Penerapan Metode Prototype.

Ade Hastuty (2021), Dasar Pemrograman, Cetakan 2 (revisi), Makassar: CV Bangun Bumitama, hal 18-23.

A. M. I. Fadhilah, O. Nurdiawan, and F. M. Basyisyar, "PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB SMART CONTRACT PADA BLOCKCHAIN BERBASIS NFT," 2023.

Bhuvan, *Unhelkar*. *2018*. *Software Engineering with UML*. Boca Raton: CRC. Press. Buttle, Francis and Stan Maklan. 2015.

Dermawan, J., & Hartini, S. (2017). Implementasi Model Waterfall Pada Pengembangan Berbasis Web Pada Sekolah Dasar Al-Azhar Syifa Budi Jatibening. Paradigma, 19(2), 142–147.

D. S. Purnia, A. Rifai, and S. Rahmatullah, Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android, 2019.

Diding Kusnady, & Ardiman Siregar. (2018). *Sistem Informasi Biaya Pendidikan (BPP) pada Politeknik Ganesha Medan Berbasis Web*. Jurnal Insitusi Politeknik Ganesha Medan, 1(1), 5.

Ginanjar Wiro Sasmito. 2017. Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal.

Hakim, D. A. (2016). Politik Hukum Lingkungan Hidup Di Indonesia Berdasarkan Undang-Undang Nomor 32 Tahun 2009 Tentang Perlindungan Dan Pengelolaan Lingkungan Hidup. FIAT JUSTISIA: Jurnal Ilmu Hukum, 9(2), 114–132. https://doi.org/10.25041/fiatjustisia.v9no2.592

Hatta, M., Anwar, M. M., Diana, I. N., & Amarul M, M. H. (2019). Perancangan Sistem Informasi Pengarsipan Dan Disposisi Surat Berbasis Web Dengan Menggunakan Framework Codeigniter. SCAN - Jurnal Teknologi Informasi Dan Komunikasi, 14(2). https://doi.org/10.33005/scan.v14i2.1481

Kartomiharjo, I. D., Hayuhardhika, W., Putra, N., & Purnomo, W. (2019). Analisis dan Perancangan Sistem Informasi Pelaporan Sampah (Studi Kasus pada Dinas Lingkungan Hidup dan Kebersihan Kabupaten Sidoarjo). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 3(11), 10756–10764.

Kent, R. (2021). Dampak positif perizinan pengelolaan lingkungan hidup menurut undang - undang nomor 11 tahun tahun 2020 tentang cipta kerja juncto peraturan pemerintah nomor 22 tahun 2021 tentang penyelenggaraan perlindungan dan pengelolaan lingkungan hidup.

Muchtar, E., &Effiyaldi, E. (2019). Analisis Dan Perancangan Sistem Infromasi Manajemen Surat Masuk Dan Keluar Pada Dinas Kependudukan Dan Pencatatan Sipil Kota Jambi. Jurnal Manajemen Sistem Informasi, 4(2),

193.https://doi.org/10.33998/jurnalmanajemensisteminformasi.2019.4.2.624

Priabudi, D. S., Rusdianto, D. S., Muhammad, E., & Jonemaro, A. (2021). Sistem Pelayanan Dokumen dan Pengaduan Lingkungan Hidup berbasis Web (Studi Kasus: Dinas Lingkungan Hidup Kota Malang). 5(1), 327–336.

Rohmat, C. L., Putri, D. E., Martanto., Prihartono, W. (2023). Rancang Bangun Sistem Informasi Arsip Surat Menggunakan Metode Waterfall Pada Dinas Lingkungan Hidup Kota Cirebon.

Syaputri, M. D., & Syaputri, M. D. (2017). Partisipasi Masyarakat Dalam Penyusunan Analisis Mengenai Dampak Lingkungan Berdasarkan Uu No 32 Tahun 2009. Varia Justicia, 13(2), 123–131. https://doi.org/10.31603/variajusticia.v13i2.1886.

Saifulloh, Ridho Pamungkas, Tobi Dwiongko Saputro, Farras Raihan Al-ayyubi (Januari – 2021). Perancangan Prototype Pengelolaan Arsip Surat di Dinas Lingkungan Hidup Kota Madiun.

T. Hidayat, O. Nurdiawan, and Y.A. Wijaya, "ANALISA WEBSITE PORTAL INFORMASI SEKOLAH DENGAN MENGGUNAKAN METODE HEURISTIC EVALUATION," 2023.

Tri Haryono Patria Mangambe. 2019. Pengawasan Oleh Dinas Lingkungan Hidup Terhadap Pelaku Usaha Dan/Atau Kegiatan Yang Memiliki Surat Pernyataan Kesanggupan Pengelolaan Dan Pemantauan Lingkungan Hidup (Sppl) Sebagai Upaya Perlindungan Dan Pengelolaan Lingkungan Hidup Di Kota Yogyakarta