

**Rancang Bangun Sistem Manajemen Dokumen Berbasis
Web Menggunakan Framework Laravel dengan Integrasi
AWS S3**

Proyek Akhir

Disusun Oleh:
Maulana Malik Ibrahim
3312301120



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM

2026

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Proyek Akhir ini dengan baik. Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Ahli Madya pada Program Studi Teknik Informatika, Politeknik Negeri Batam.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. **Abah dan Umma**, Selaku orang tua yang selalu mensupport dan juga memberikan bimbingan serta nasehat
2. **Marlia Dwiyantri** yang selalu memberikan support dan juga memperjuangkan lulus bersama
3. **Ibu Cyntia Lasmi Andesti, S.Kom., M.Kom**, selaku dosen pembimbing yang telah memberikan bimbingan, arahan, serta dukungan selama proses pengerjaan Proyek Akhir ini.
4. **Bapak/Ibu Dosen Penguji**, yang telah memberikan masukan dan saran yang konstruktif untuk penyempurnaan laporan ini.
5. **Seluruh teman-teman seperjuangan**, yang telah memberikan semangat dan bantuan baik secara langsung maupun tidak langsung.
6. **Keluarga tercinta**, yang senantiasa memberikan dukungan moral, spiritual, dan materiil selama penulis menempuh pendidikan hingga menyelesaikan Proyek Akhir ini.

Laporan Proyek Akhir ini berjudul “Rancang Bangun Sistem Manajemen Dokumen Berbasis Web Menggunakan Framework Laravel dengan Integrasi AWS S3”. Proyek ini dilatarbelakangi oleh tantangan dalam pengelolaan dokumen secara konvensional yang seringkali tidak efisien dan rentan terhadap kerusakan atau kehilangan data. Oleh karena itu, sistem ini dirancang untuk membantu para staf *document control* dalam menerapkan budaya kerja *paperless* (nirkertas). Dengan memanfaatkan layanan penyimpanan cloud AWS S3 (Simple Storage Service) yang skalabel dan andal, aplikasi ini bertujuan untuk menciptakan alur kerja manajemen dokumen PDF yang lebih terpusat, aman, dan mudah diakses.

Batam, 17,Oktober,2025

Maulana Malik Ibrahim

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	4
DAFTAR TABEL.....	5
DAFTAR GAMBAR.....	6
DAFTAR LAMPIRAN.....	7
ABSTRAK.....	8
BAB I PENDAHULUAN.....	9
1.1. Latar Belakang.....	9
1.2. Rumusan Masalah.....	10
1.3. Tujuan.....	10
1.4. Batasan Masalah.....	11
1.5. Manfaat.....	12
BAB II TINJAUAN PUSTAKA.....	13
2.1. Penelitian Terkait.....	13
2.2. Landasan Teori.....	16
2.2.1 Sistem Manajemen Dokumen (<i>Document Management System</i>).....	16
2.2.2 Framework Laravel.....	15
2.2.3 <i>Cloud Storage</i>	17
2.2.4 API (<i>Application Programming Interface</i>).....	17
2.2.5 Keamanan Cloud Storage AWS S3 dan IAM	17
2.2.6 Mekanisme Keamanan <i>Presigned URL</i>	18
2.2.7 Bucket Policy & Encryption.....	18
2.2.8 Dokumentasi Metadata Dokumen.....	19
2.3. Metode Pengembangan Produk.....	20
BAB III ANALISIS DAN PERANCANGAN.....	23
3.1. Analisis Kebutuhan.....	23
3.1.1 Proses Analisis Kebutuhan.....	23
3.1.2 Gambaran Umum Sistem.....	23
3.2. Perancangan.....	25
3.2.1 Kebutuhan Fungsional.....	25
3.2.2 Kebutuhan Non-Fungsional.....	26
3.2.3 Diagram <i>Use Case</i>	27
3.2.4 Skenario <i>Use Case</i>	28
3.2.5 <i>Activity Diagram</i>	44
3.2.6 <i>ER Diagram</i> (ERD).....	49
3.2.7 Perancangan Antarmuka (<i>Wireframe</i>).....	54
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	60
4.1 Hasil Implementasi.....	60

4.1.1 Implementasi Lingkungan Pengembangan.....	60
4.1.2 Implementasi Fitur Sistem.....	60
4.2 Pengujian <i>Black Box Testing</i>	71
4.3 Pengujian <i>System Usability Scale</i> (SUS).....	78
BAB V KESIMPULAN.....	81
5.1 Kesimpulan.....	81
5.2 Saran.....	82
DAFTAR PUSTAKA.....	83
DAFTAR LAMPIRAN.....	84

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sebelumnya.....	14
Tabel 2.2 <i>Sprint Planning</i> Pengembangan Sistem.....	21
Tabel 3.1 Kebutuhan Fungsional Sistem ARSIPIN.....	25
Tabel 3.2 Kebutuhan Non-Fungsional Sistem.....	26
Tabel 3.3 Skenario <i>Use Case</i> Melakukan Login.....	28
Tabel 3.4 Skenario <i>Use Case</i> Mengunggah Dokumen.....	29
Tabel 3.5 Skenario <i>Use Case</i> Mengedit Metadata Dokumen.....	31
Tabel 3.6 Skenario <i>Use Case</i> Mengajukan Hapus Dokumen.....	32
Tabel 3.7 Skenario <i>Use Case</i> Menyetujui dan Menolak Ajuan Penghapusan.....	33
Tabel 3.8 Skenario <i>Use Case</i> Menghapus Dokumen Mutlak.....	34
Tabel 3.9 Skenario <i>Use Case</i> Melihat Log Aktivitas (Audit Trail System).....	35
Tabel 3.10 Skenario <i>Use Case</i> Melihat Daftar Dokumen.....	36
Tabel 3.11 Skenario <i>Use Case</i> Melihat Detail Dokumen.....	37
Tabel 3.12 Skenario <i>Use Case</i> Mencari Dokumen.....	38
Tabel 3.13 Skenario <i>Use Case</i> Mengelola Akun Anggota (Kepala BU).....	39
Tabel 3.14 Skenario <i>Use Case</i> Mengelola Akun Pegawai (Admin).....	40
Tabel 3.15 Skenario <i>Use Case</i> Logout (Keluar Sistem).....	41
Tabel 3.16 Skenario <i>Use Case</i> Mengelola Kategori Dokumen.....	42
Tabel 4.1 Implementasi Lingkungan Pengembangan	60
Tabel 4.2 Hasil Pengujian Black Box Testing - Modul Manajemen Pengguna	71
Tabel 4.3 Hasil Pengujian Black Box Testing - Modul Dokumen & Integrasi AWS S3.....	72
Tabel 4.4 Hasil Pengujian Black Box Testing - Modul Keamanan Otorisasi & Pencegahan Broken Access Control.....	73
Tabel 4.5 Hasil Pengujian Black Box Testing - Modul Autentikasi Sesi Pengguna.....	74
Tabel 4.6 Hasil Pengujian Black Box Testing - Modul Manipulasi Metadata & Pencarian Berkas.....	74
Tabel 4.7 Hasil Pengujian Black Box Testing - Modul Struktur.....	75
Tabel 4.8 Hasil Pengujian Black Box Testing - Kebutuhan Non-Fungsional.....	75
Tabel 4.9 Hasil Pengujian Black Box Testing - Modul Audit Trail & Log Aktivitas.....	77
Tabel 4.10 Hasil Pengujian Black Box Testing - Modul Kelola Jenis Dokumen Keuangan....	77
Tabel 4.11 Pertanyaan Kuesioner.....	78
Tabel 4.12 Rekapitulasi Hasil Pengujian SUS.....	79
Tabel 4.13 Interpretasi Hasil Pengujian SUS.....	80

DAFTAR GAMBAR

Gambar 2.1 Tahapan Metode <i>Agile Software Development</i>	20
Gambar 3.1 Diagram Alir Gambaran Umum Sistem ARSIPIN.....	24
Gambar 3.2 Diagram <i>Use Case</i> Sistem.....	27
Gambar 3.3 <i>Activity Diagram</i> Pengguna Login.....	44
Gambar 3.4 <i>Activity Diagram</i> Admin Kelola Akun.....	45
Gambar 3.5 <i>Activity Diagram</i> Pengguna Unggah Dokumen.....	46
Gambar 3.6 <i>Activity Diagram</i> Pengguna Edit Dokumen.....	47
Gambar 3.7 <i>Activity Diagram</i> Alur Pengajuan dan Penyetujuan Penghapusan Dokumen.....	48
Gambar 3.8 <i>Entity Relationship Diagram</i> (ERD) Basis Data MySQL.....	49
Gambar 3.9 <i>Wireframe Login Page</i>	54
Gambar 3.10 <i>Wireframe Dashboard Admin</i>	55
Gambar 3.11 <i>Wireframe Kelola Akun Admin</i>	55
Gambar 3.12 <i>Wireframe Bisnis Unit - Retail Admin</i>	56
Gambar 3.13 <i>Wireframe Komersial Admin</i>	56
Gambar 3.14 <i>Wireframe Audit Trail Sistem Global</i>	57
Gambar 3.15 <i>Wireframe Dashboard Pegawai - Retail</i>	57
Gambar 3.16 <i>Wireframe Log Aktivitas Pegawai - Retail</i>	58
Gambar 3.17 <i>Wireframe Dashboard Pegawai - Komersial</i>	58
Gambar 3.18 <i>Wireframe Log Aktivitas Pegawai - Komersial</i>	59
Gambar 4.1 Implementasi Antarmuka Halaman Login Utama.....	61
Gambar 4.2 Implementasi Antarmuka Halaman Dashboard Admin.....	61
Gambar 4.3 Implementasi Antarmuka Halaman Kelola Akun Pegawai.....	62
Gambar 4.4 Implementasi Antarmuka Halaman Bisnis Unit - Retail.....	63
Gambar 4.5 Implementasi Antarmuka Halaman Daftar Perusahaan - Retail.....	63
Gambar 4.6 Implementasi Antarmuka Halaman Bisnis Unit - Komersial.....	64
Gambar 4.7 Implementasi Antarmuka Halaman Daftar Perusahaan - Komersial.....	65
Gambar 4.8 Implementasi Antarmuka Halaman <i>Audit Trail</i> Sistem.....	65
Gambar 4.9 Implementasi Antarmuka Halaman Dashboard Kepala BU.....	66
Gambar 4.10 Implementasi Antarmuka Halaman Kelola Staf Unit.....	66
Gambar 4.11 Implementasi Antarmuka Halaman Persetujuan Pemusnahan Dokumen.....	67
Gambar 4.12 Implementasi Antarmuka Halaman Timeline Aktivitas Unit.....	67
Gambar 4.13 Implementasi Antarmuka Halaman Dashboard Pegawai - Retail.....	68
Gambar 4.14 Implementasi Antarmuka Halaman Timeline Kerja Pegawai - Retail.....	69
Gambar 4.15 Implementasi Antarmuka Halaman Dashboard Pegawai - Komersial.....	69
Gambar 4.16 Implementasi Antarmuka Halaman Timeline Kerja Pegawai - Komersial.....	70

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement.....	74
Lampiran B. Link Produk.....	74
Lampiran C. Dokumen Pengujian UAT.....	74
Lampiran D. Dokumen Pengujian SUS.....	74

ABSTRAK

Pengelolaan dokumen secara konvensional seringkali dihadapkan pada berbagai tantangan, seperti inefisiensi dalam pencarian data, risiko kerusakan atau kehilangan dokumen fisik, serta kebutuhan ruang penyimpanan yang besar. Tantangan ini menghambat penerapan budaya kerja nirkertas (*paperless*) yang efisien. Oleh karena itu, proyek ini bertujuan untuk merancang dan membangun sebuah Sistem Manajemen Dokumen berbasis web yang modern dan terpusat. Sistem ini dikembangkan menggunakan kerangka kerja (*framework*) PHP Laravel yang dikenal dengan struktur dan keamanannya. Untuk penyimpanan file, sistem ini memanfaatkan integrasi dengan AWS S3 (Simple Storage Service), yang memungkinkan dokumen PDF disimpan secara aman, skalabel, dan andal di *cloud storage*. Sementara itu, metadata dokumen seperti judul, deskripsi, dan informasi terkait lainnya dikelola dalam database MySQL untuk memastikan akses data yang cepat dan terstruktur. Hasil dari perancangan ini adalah sebuah aplikasi web fungsional yang memungkinkan pengguna, khususnya staf *document control*, untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap dokumen digital. Dengan implementasi sistem ini, diharapkan proses manajemen dokumen menjadi lebih efisien, terpusat, aman, dan mendukung transformasi digital dengan mendorong penerapan budaya kerja *paperless*.

Kata Kunci: Sistem Manajemen Dokumen, Laravel, AWS S3, *Paperless*, *Cloud Storage*

BAB I PENDAHULUAN

1.1. Latar Belakang

Pengelolaan dokumen secara konvensional berbasis kertas menghadapi kendala serius di era digital. Proses ini tidak efisien, memakan biaya tinggi untuk pencetakan dan penyimpanan, serta sangat rentan terhadap risiko kerusakan atau kehilangan dokumen fisik. Ketergantungan pada arsip fisik juga menghambat aksesibilitas data, yang menjadi masalah krusial di tengah meningkatnya tren kerja jarak jauh.

Pengelolaan dokumen di perusahaan saat ini masih mengandalkan metode konvensional berbasis kertas, yang menimbulkan berbagai masalah signifikan. Proses pencarian arsip menjadi lambat dan tidak efisien karena penelusuran manual dan penyimpanan yang seringkali tidak terorganisir, menghambat produktivitas. Selain itu, ketergantungan pada dokumen fisik membawa risiko tinggi terhadap kerusakan akibat faktor lingkungan atau bencana, serta kehilangan data karena salah penempatan atau kelalaian. Sistem manual ini juga membutuhkan biaya operasional yang tinggi untuk kertas, pencetakan, dan ruang penyimpanan fisik yang luas, sekaligus membatasi aksesibilitas informasi hanya di lokasi kantor, yang tidak lagi relevan dengan kebutuhan kerja modern yang fleksibel.

Konsep kantor nirkertas (paperless) yang didukung oleh teknologi penyimpanan awan (cloud storage) menawarkan alternatif unggul sebagai solusi atas permasalahan tersebut. Pemanfaatan layanan seperti AWS S3 (Simple Storage Service) memungkinkan penyimpanan dokumen PDF secara digital yang sangat skalabel, aman, dan dapat diakses dari mana saja, sekaligus mengatasi keterbatasan ruang penyimpanan fisik dan menawarkan durabilitas data yang tinggi.

Namun, sekadar menyimpan file di cloud tidak cukup tanpa adanya sistem pengelolaan metadata yang terstruktur. Oleh karena itu, proyek ini mengusulkan “Rancang Bangun Sistem Manajemen Dokumen Berbasis Web Menggunakan Framework Laravel dengan Integrasi AWS S3”. Sistem ini akan dikembangkan menggunakan framework Laravel dan mengadopsi arsitektur hybrid, di mana metadata dokumen dikelola dalam database MySQL untuk pencarian cepat, sementara file fisik (khususnya PDF) disimpan secara aman melalui integrasi dengan API AWS S3. Kombinasi ini akan menghasilkan aplikasi yang efisien dan menjadi solusi konkret untuk mendukung transformasi digital menuju lingkungan kerja yang modern dan paperless.

Penelitian dan implementasi sistem ini difokuskan pada unit Document Control Finansial internal organisasi. Sebelum adanya sistem ARSIPIN, unit ini menangani volume dokumen keuangan yang masif—mencapai 150 hingga 300 berkas PDF baru per bulan. Dokumen-dokumen finansial tersebut secara spesifik mencakup jenis dokumen seperti Invoice, Voucher, Pajak, dan Rekening Koran Bank. Masalah riil yang dihadapi di lapangan adalah tingginya risiko kerusakan berkas harian, hilangnya data akibat salah penempatan file lokal, lambatnya pencarian data manual yang memakan waktu hingga hitungan jam, serta tidak adanya pembatasan wilayah kerja antar-divisi operasional yang berpotensi memicu kebocoran data akuntansi korporat.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang akan dibahas dalam proyek ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang sebuah sistem manajemen dokumen berbasis web yang dapat mengatasi inefisiensi dalam pencarian dan pengelolaan arsip PDF secara konvensional?
2. Bagaimana memanfaatkan teknologi cloud storage AWS S3 untuk mengurangi resiko kehilangan dan kerusakan dokumen fisik serta menyediakan penyimpanan digital yang aman?
3. Bagaimana membangun aplikasi terpusat yang memungkinkan akses aman dan terkontrol terhadap dokumen PDF dari berbagai lokasi, sesuai dengan peran pengguna?

1.3. Tujuan

Adapun tujuan yang ingin dicapai dari perancangan sistem ini disusun untuk menjawab rumusan masalah secara langsung, yaitu:

1. Menghasilkan aplikasi manajemen dokumen berbasis web untuk meningkatkan efisiensi dalam proses pencarian dan pengelolaan dokumen PDF secara konvensional, didukung dengan database MySQL untuk mengelola metadata secara terstruktur agar seluruh operasi CRUD dapat berjalan dengan cepat.
2. Membangun sistem penyimpanan yang aman dan andal dengan mengintegrasikan API *Cloud Storage* AWS S3 untuk meminimalkan risiko kehilangan serta kerusakan dokumen fisik finansial.

3. Menyediakan sebuah platform repositori terpusat yang memudahkan pengguna untuk mengakses dokumen PDF kapan saja dan di mana saja secara aman serta terkontrol sesuai dengan batasan wewenang dan peran aksesnya (*Role-Based Access Control*).

1.4. Batasan Masalah

Untuk memastikan pengembangan proyek ini fokus dan terarah, ruang lingkungannya dibatasi sebagai berikut:

1. Sistem hanya akan diimplementasikan sebagai aplikasi berbasis web. Pengembangan aplikasi *mobile* atau *desktop* tidak termasuk dalam ruang lingkup ini.
2. Integrasi API difokuskan pada AWS S3 API untuk fungsi penyimpanan (mengunggah, mendapatkan URL akses, menghapus) file PDF. Fitur kolaborasi atau anotasi *real-time* tidak akan dikembangkan.
3. Hak akses pengguna dibagi menjadi dua peran utama: Admin (hak akses penuh) dan Pegawai (*Filing Officer* dengan hak akses terbatas sesuai *scope* - misal: Retail/Commercial).
4. Format file yang dapat diunggah dan dikelola oleh sistem dibatasi hanya untuk dokumen .pdf.
5. Sistem tidak menyediakan fitur pendaftaran mandiri (*self-registration*) bagi pengguna umum. Seluruh pembuatan akun Pegawai dilakukan secara terpusat oleh Admin melalui halaman manajemen pengguna.
6. Sistem difokuskan secara spesifik pada pengelolaan Dokumen Keuangan (Finance)

1.5. Manfaat

Proyek ini diharapkan dapat memberikan manfaat dari beberapa aspek, yaitu:

1. Teoritis/Ilmiah

Memberikan studi implementasi mengenai arsitektur *hybrid storage*, di mana metadata dikelola secara relasional (MySQL) sementara file fisik (PDF) disimpan pada *cloud storage* AWS S3, sebagai model yang efektif untuk sistem manajemen informasi modern yang memanfaatkan layanan *cloud* skalabel.

2. Praktis

Menyediakan alat bantu kerja (*tool*) bagi staf *document control* untuk memusatkan dan menyederhanakan alur kerja pengarsipan dokumen digital PDF, sehingga proses pencarian dan pengelolaan menjadi lebih cepat dan dapat diakses dari mana saja

3. Praktis

Menyediakan alat bantu kerja (*tool*) bagi staf *document control* untuk memusatkan dan menyederhanakan alur kerja pengarsipan dokumen digital PDF, sehingga proses pencarian dan pengelolaan menjadi lebih cepat dan dapat diakses dari mana saja melalui integrasi dengan AWS S3.

4. Manajerial

Mendukung tujuan transformasi digital organisasi dengan menyediakan platform yang aman dan terkontrol untuk aset dokumen PDF, mengurangi ketergantungan pada arsip fisik, dan menjadi fondasi dalam implementasi kantor nirkertas (*paperless office*).

5. Sustainable Development Goals (SDGs)

a. **SDGs Poin 12 (Konsumsi dan Produksi yang Bertanggung Jawab):**

Berkontribusi langsung pada gerakan ramah lingkungan melalui pengurangan penggunaan kertas secara masif (*paperless office*) dalam operasional harian industri.

b. **SDGs Poin 9 (Industri, Inovasi, dan Infrastruktur):** Mendorong modernisasi digitalisasi infrastruktur arsip konvensional menuju arsitektur penyimpanan berbasis teknologi *cloud computing* yang efisien, aman, dan skalabel.

BAB II TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian terdahulu memberikan gambaran perkembangan sistem arsip digital dan menjadi dasar kebaruan penelitian ini.

Armansyah (2025) merancang *website* arsip digital menggunakan *framework* Laravel untuk membantu dokumentasi laporan dan mempermudah pengolahan data dokumen. Sistem ini berhasil meningkatkan efisiensi, namun penyimpanan filenya masih bersifat lokal (*on-premise*) dan belum terintegrasi dengan *cloud storage* eksternal untuk skalabilitas.

Makalalag dan Maramis (2025) mengembangkan sistem manajemen dokumen *cloud* yang diimplementasikan langsung menggunakan AWS S3 dan *pre-signed URL* untuk keamanan. Sistem ini terbukti meningkatkan aksesibilitas dan keamanan, tetapi dibangun menggunakan *framework* Flask dan PostgreSQL, serta sistem otorisasi hak aksesnya masih terbatas pada level admin/staf sederhana.

Putra, dkk. (2025) juga mengembangkan E-Arsip berbasis web menggunakan *framework* Laravel yang efektif mengurangi risiko kehilangan dokumen penting. Akan tetapi, penelitian ini masih berfokus pada solusi pengganti arsip manual dan belum mengimplementasikan penyimpanan *cloud* yang skalabel.

Alexandes, dkk. (2022) mengembangkan sistem e-document menggunakan metode Agile. Sistem ini berhasil membantu pengelolaan dan pencarian data, namun belum terintegrasi dengan layanan *cloud storage* eksternal dan masih bergantung pada kapasitas server aplikasi.

Selain itu, Lukitasari & Utomo (2024) serta Kurniawan & Amalia (2020) telah menunjukkan kelayakan penggunaan teknologi AWS dan *framework* Laravel, meskipun dalam konteks sistem informasi yang berbeda (pengolahan data nilai dan e-learning).

Berdasarkan tinjauan tersebut, dapat disimpulkan bahwa penelitian sebelumnya belum mengintegrasikan *framework* Laravel, penyimpanan *cloud* AWS S3, dan pemodelan struktur organisasi yang detail (Kategori, Bisnis Unit, Perusahaan) secara terpadu. Oleh sebab itu, proyek ini memiliki kebaruan berupa pengembangan aplikasi web manajemen dokumen

berbasis Laravel yang skalabel dengan AWS S3, yang dilengkapi *Role-Based Access Control* (RBAC) berdasarkan struktur organisasi yang kompleks.

Tabel 2.1. Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Fokus	Metode/Teknologi	Kelebihan	Kekurangan/Celah
Anisah, dkk. (2021)	Merancang Sistem Informasi Manajemen Arsip Digital untuk mengatasi masalah pencarian manual dan penumpukan arsip fisik di kantor Perwakilan BPKP Provinsi Kepulauan Bangka Belitung.	Metode: Waterfall. Teknologi: PHP, MySQL	- Sistem mampu mempercepat proses pencarian arsip dari hitungan hari menjadi hitungan detik. - Memudahkan monitoring pergerakan arsip antar bidang	Sistem masih sepenuhnya <i>on-premise</i> (lokal). File arsip disimpan langsung di server lokal, sehingga tidak memanfaatkan skalabilitas dan fleksibilitas <i>cloud storage</i> .
Alexandes, dkk. (2022)	Mengembangkan sistem <i>e-document</i> untuk Suku Dinas Bina Marga Jakarta Barat guna mengatasi penumpukan dokumen di gudang penyimpanan dan mempermudah pengelolaan.	Metode: Agile. Teknologi: PHP.	- Memiliki fitur hak akses untuk menjaga kerahasiaan dokumen antar departemen. - Terdapat fitur "dokumen proyek" untuk memantau progres dan masa berlaku dokumen.	- Meskipun sudah digital, sistem ini belum terintegrasi dengan layanan <i>cloud storage</i> eksternal seperti Google Drive, sehingga masih bergantung pada kapasitas penyimpanan server aplikasi.
Armansyah, J. (2025)	Merancang dan membangun	Metode: Waterfall. Teknologi: Laravel, MySQL.	- Dibangun menggunakan <i>framework</i>	Penyimpanan file masih bersifat lokal di

	website arsip digital sebagai proyek kerja praktik, dengan studi kasus rekap data laporan polisi untuk mempermudah pengolahan data dokumen.		modern (Laravel) yang memiliki struktur dan keamanan yang baik. - Terdapat fitur pembatasan hak akses antara <i>admin</i> dan <i>staff</i> (<i>staff</i> tidak bisa edit & hapus).	dalam folder <i>storage</i> aplikasi. Belum memanfaatkan API eksternal untuk penyimpanan, sehingga skalabilitasnya terbatas.
Amalia, dkk. (2025)	Mengoptimalkan manajemen data dan sinkronisasi otomatis pada aplikasi web dengan mengintegrasikan Google Drive API sebagai solusi penyimpanan <i>cloud</i> .	Metode: Literasi. Teknologi: Laravel, PHP, Google Drive API.	- Berhasil mengintegrasikan Google Drive API untuk penyimpanan file. - Menunjukkan proses sinkronisasi data yang cepat (di bawah 60 detik) - Mengatasi limitasi kapasitas penyimpanan pada hosting.	- Penelitian ini tidak membahas pengelolaan metadata secara terstruktur dalam database relasional (MySQL) yang terpisah. - Fokus utamanya pada sinkronisasi file, belum pada sistem manajemen dokumen yang komprehensif (CRUD metadata).
Yazid, A., dkk. (2025)	Memberikan pelatihan digitalisasi manajemen arsip dan administrasi bagi lembaga pendidikan nonformal, dengan fokus pada penggunaan aplikasi <i>cloud storage</i> seperti	Metode: Pelatihan & Bimbingan Teknis. Teknologi: Google Drive, Dropbox (sebagai <i>tools</i>)	- Membuktikan bahwa penggunaan <i>cloud storage</i> dapat meningkatkan efisiensi, keamanan, dan kolaborasi dalam pengelolaan arsip.	- Penelitian ini berupa kegiatan pengabdian masyarakat (pelatihan), bukan membangun sebuah sistem/aplikasi terintegrasi. Mereka mengajarkan cara menggunakan Google Drive secara manual, bukan mengintegrasika

	Google Drive.			nya melalui API ke dalam aplikasi custom.
Penelitian ini	Rancang Bangun Sistem Manajemen Dokumen Berbasis Web Menggunakan Framework Laravel dengan Integrasi AWS S3	Metode: Agile. Teknologi: Laravel, MySQL, AWS S3 API.	- Menggunakan Laravel untuk pengembangan terstruktur & aman. - Menerapkan arsitektur <i>hybrid</i> (MySQL + Cloud). - Integrasi spesifik dengan AWS S3 untuk skalabilitas & durabilitas penyimpanan PDF. - Implementasi kontrol akses berbasis peran (Admin/Pegawai).	- Fokus hanya pada format PDF - Fungsionalitas terbatas pada CRUD dasar (MVP). - Tidak mencakup fitur kolaborasi <i>real-time</i>.

2.2. Landasan Teori

2.2.1 Sistem Manajemen Dokumen (*Document Management System*)

Sistem Manajemen Dokumen (SMD) digital, sering disebut juga sistem e-arsip, adalah solusi berbasis teknologi informasi yang dirancang untuk menggantikan pengelolaan arsip konvensional berbasis kertas. Tujuannya adalah untuk mengelola dokumen elektronik secara terpusat, mengatasi berbagai kendala seperti kesulitan dan lamanya waktu pencarian data pada arsip fisik, resiko kehilangan atau kerusakan dokumen (Putra et al., 2025)

2.2.2 Framework Laravel

Framework Laravel adalah kerangka kerja (*framework*) aplikasi web berbasis PHP yang bersifat *open-source* dan populer digunakan untuk mempercepat proses pengembangan aplikasi yang terstruktur. Laravel menerapkan arsitektur Model-View-Controller (MVC), yang memisahkan logika aplikasi (Model), tampilan antarmuka pengguna (View), dan penghubung antara keduanya (Controller). Penggunaan *framework* seperti Laravel

memberikan struktur dasar dan berbagai fitur bawaan (misalnya *routing*, sistem *template*, ORM untuk interaksi database), sehingga pengembang dapat lebih fokus pada pengembangan fungsionalitas spesifik aplikasi, seperti sistem E-Arsip, tanpa harus menulis semua kode dari awal. Keunggulan Laravel meliputi kemudahan pengembangan, pengelolaan kode yang terstruktur, keamanan, performa, serta kemudahan pemeliharaan dan pembaruan aplikasi web (Putra et al., 2025)

2.2.3 Cloud Storage

Cloud Storage (Penyimpanan Awan) adalah model layanan di mana data disimpan di server jarak jauh yang dikelola oleh pihak ketiga dan diakses melalui internet. Alih-alih menyimpan data di hard drive lokal atau server internal, pengguna dapat mengunggah dan mengakses file mereka dari mana saja dan kapan saja. Layanan ini menawarkan beberapa keuntungan utama seperti skalabilitas (kapasitas penyimpanan dapat ditambah sesuai kebutuhan), aksesibilitas tinggi, dan pengurangan biaya infrastruktur karena tidak perlu membeli dan memelihara perangkat keras server sendiri. (Salsabila, Z., & Syarif, A., 2022).

2.2.4 API (Application Programming Interface)

API adalah sekumpulan aturan dan protokol yang memungkinkan satu aplikasi perangkat lunak berkomunikasi dan bertukar data dengan aplikasi lain. API bertindak sebagai "jembatan" yang memungkinkan integrasi fungsionalitas antar sistem yang berbeda tanpa perlu mengetahui detail internal masing-masing.

Dengan mengintegrasikan AWS S3 API (biasanya melalui AWS SDK atau *library* seperti Flysystem di Laravel), aplikasi Anda dapat memanfaatkan keandalan, skalabilitas, dan keamanan AWS S3 untuk menangani penyimpanan file PDF secara efisien (Makalalag & Maramis, 2025)

2.2.5 Keamanan Cloud Storage AWS S3 dan IAM (Identity and Access Management)

Amazon Simple Storage Service (AWS S3) menyediakan infrastruktur penyimpanan berbasis objek yang memiliki tingkat durabilitas tinggi, namun keandalan tersebut harus didukung oleh konfigurasi kontrol akses yang ketat. Otorisasi keamanan terpusat pada AWS S3 dikelola melalui layanan Identity and Access Management (IAM). IAM menerapkan prinsip least privilege, di mana pengguna (IAM Users) atau peran (IAM Roles) diberikan hak akses minimum berupa format kebijakan berbasis data JSON (JSON Policy) untuk berinteraksi dengan API S3. Koneksi aman antara backend framework Laravel pada aplikasi

ARSIPIN dengan objek bucket AWS dikunci menggunakan kredensial privat, yaitu `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` yang disimpan dalam berkas konfigurasi `.env`. Konfigurasi ini memastikan setiap request manipulasi data dari luar sistem yang tidak membawa token enkripsi yang valid akan secara otomatis ditolak oleh gerbang IAM. (Sahyudi & Susanto, 2025)

2.2.6 Mekanisme Keamanan Presigned URL

Presigned URL merupakan fitur keamanan tingkat lanjut dari AWS SDK yang digunakan untuk memfasilitasi akses pratinjau (preview) dokumen privat tanpa mengekspos visibilitas objek bucket secara publik ke internet. Secara arsitektur, berkas PDF finansial yang diunggah ke dalam sistem ARSIPIN disimpan dengan status Private. Ketika seorang pengguna memilih aksi untuk melihat dokumen, backend controller Laravel akan melakukan handshake dengan API AWS S3 untuk membuat Presigned URL yang dibubuhi dengan tanda tangan kriptografi (cryptographic signature) dan waktu kedaluwarsa (expiration time) yang sangat singkat, misalnya 5 menit. Setelah batas waktu tersebut terlewati, token URL otomatis hangus dan tidak dapat diakses kembali, sehingga efektif menangani risiko eksploitasi data akibat pencurian tautan (Broken Access Control / URL Hijacking). (Makalagal & Maramis, 2025)

2.2.7 Bucket Policy & Encryption

Bucket Policy adalah kebijakan berbasis resource yang diterapkan langsung pada tingkat bucket AWS S3 untuk mengatur hak akses secara global, seperti memblokir seluruh akses publik secara mutlak (Block All Public Access). Untuk melengkapi perlindungan data finansial yang tersimpan, diterapkan sistem keamanan Server-Side Encryption (SSE-S3). Setiap berkas PDF yang masuk ke server AWS S3 akan secara otomatis dienkrpsi di sisi server sebelum ditulis ke dalam disk penyimpanan cloud menggunakan algoritma kriptografi canggih Advanced Encryption Standard dengan panjang kunci 256-bit (AES-256). Proses dekripsi hanya akan terjadi secara transparan jika request penarikan file datang dari entitas sistem yang memiliki kunci otorisasi yang valid, sehingga data tetap aman meskipun hardware fisik server AWS mengalami gangguan. (Fauzi et al., 2024)

2.2.8 Dokumentasi *Metadata* Dokumen (*Hybrid Storage Architecture*)

Sistem manajemen dokumen yang dirancang untuk menangani data masif idealnya menerapkan pola arsitektur *hybrid storage*. Pola ini memisahkan secara tegas antara penyimpanan file fisik dengan pengelolaan data identitas pendukungnya (*metadata*). Dokumen fisik asli berformat PDF didelegasikan ke media *cloud storage* AWS S3 dan diidentifikasi menggunakan string alamat unik berupa `s3_object_key`. Sementara itu, *metadata* dokumen terstruktur—yang meliputi `no_dokumen` hasil rumusan otomatis JavaScript, `nama_dokumen`, `tipe_keuangan` (Voucher/Invoice/Pajak), `rentang_bulan_buku`, `tahun_buku`, `urutan_jilid_buku`, serta log rekam jejak pengunggah (`user_id`)—didokumentasikan dan diindeks di dalam tabel basis data relasional MySQL. Pemisahan *metadata* ini bertujuan agar operasi pencarian data harian (*searching*) dan pemfilteran query data dapat dieksekusi secara instan di sisi database tanpa perlu membebani performa jaringan dengan membaca file fisik PDF secara langsung. (Armansyah, 2025)

2.3. Metode Pengembangan Produk

Metode yang digunakan untuk pengembangan Sistem Manajemen Dokumen Finance ini adalah Agile. Pendekatan Agile dipilih karena sangat cocok untuk proyek yang membutuhkan fleksibilitas, pengembangan bertahap (iteratif), dan adaptasi cepat terhadap perubahan. Metode ini memungkinkan sistem untuk dibangun dalam siklus-siklus pendek yang disebut sprint, dimana setiap sprint menghasilkan bagian fungsional dari aplikasi. Dengan Agile, tim dapat merespons umpan balik secara cepat, menyesuaikan prioritas fitur, dan mengirimkan pembaruan secara berkala.



Gambar 2.1 Metode Agile

sumber : BINAR

Tahapan dari metode Agile sebagai berikut (Handayani et al., 2023) :

a. Perencanaan (Planning/Requirement Gathering)

Tahap awal ini melibatkan kolaborasi dengan pemangku kepentingan (calon pengguna: Admin/Bos dan Pegawai/*Filing Officer*) untuk mengidentifikasi dan memahami kebutuhan

sistem. Fokusnya adalah mendefinisikan tujuan utama sistem manajemen dokumen, fitur-fitur krusial seperti CRUD dokumen, manajemen pengguna dan peran (*role*), serta integrasi dengan AWS S3

b. Perancangan dan Desain (Design)

Berdasarkan kebutuhan yang terkumpul, tim membuat rancangan teknis dan visual. Ini mencakup perancangan arsitektur sistem menggunakan pola MVC pada *framework* Laravel, desain skema database di MySQL, serta pembuatan *wireframe* dan *mockup* antarmuka pengguna (UI/UX) agar aplikasi intuitif dan mudah digunakan.

c. Implementasi dalam Siklus Sprint (Sprint Planning)

Dalam Agile, pekerjaan dibagi ke dalam sprint, yang biasanya berlangsung antara 1 hingga 4 minggu. Setiap sprint berfokus pada bagian-bagian tertentu dari aplikasi yang akan dibangun. Pada tahap Sprint Planning, tim akan merencanakan apa yang akan dikerjakan dalam sprint tersebut dan menetapkan task yang akan diselesaikan.

Tabel 2.2 sprint Planning

Fase	Durasi	Deskripsi
<i>Requirements</i>	2 – 3 Minggu	Melakukan analisis kebutuhan fungsional bersama pengguna kunci (Admin pusat dan Filing Officer unit keuangan) guna merumuskan batasan masalah, spesifikasi Role-Based Access Control (RBAC), serta integrasi cloud storage AWS S3. Hasil dari tahap ini dituangkan ke dalam Product Backlog.
<i>Design</i>	3 – 4 Minggu	Merancang arsitektur sistem berbasis Model-View-Controller (MVC) pada kerangka kerja Laravel 11, mendesain skema basis data relasional MySQL (ERD), serta membuat wireframe antarmuka pengguna (UI/UX) untuk memastikan tata letak navigasi intuitif.
<i>Development</i>	20 – 22 Minggu	Pengkodean dibagi menjadi beberapa modul increment (Sprint Backlog). Sprint 1 berfokus pada modul autentikasi dan manajemen pengguna; Sprint 2 berfokus pada integrasi API AWS S3 dan otomatisasi nomor dokumen; Sprint 3 berfokus pada alur workflow pemusnahan berkas dan sistem Audit Trail (Log Aktivitas).
<i>Testing</i>	4 Minggu	Setiap fitur yang selesai pada tiap sprint langsung diuji menggunakan metode Black Box Testing guna memvalidasi kesesuaian fungsionalitas input dan output, serta mendeteksi dini adanya celah keamanan seperti Broken Access Control.
<i>Review</i>	Berjalan Paralel / 2 Minggu	Demonstrasi berkala dilakukan kepada stakeholder (Kepala BU) di akhir siklus untuk mengevaluasi hasil kerja sistem. Masukan dari pengujian langsung dijadikan catatan perbaikan pada

		backlog berikutnya (misalnya penyempurnaan label status dan filter unit).
--	--	---

<i>Deployment</i>	3 Minggu	Setelah seluruh siklus sprint terpenuhi, stabil, dan lolos pengujian System Usability Scale (SUS), aplikasi di-deploy secara lokal dan repositori kodingannya disinkronkan ke platform GitHub sebagai dokumentasi final produk.
-------------------	----------	---

d. Pengujian (Testing)

Setelah fitur dikembangkan dalam satu *sprint*, tahap pengujian dilakukan untuk memastikan kualitas dan fungsionalitasnya. Metode yang digunakan adalah Black Box Testing, dimana penguji berinteraksi dengan aplikasi dari sudut pandang pengguna (misalnya mencoba login, mengunggah file, menghapus dokumen) untuk memverifikasi apakah output sesuai harapan, tanpa perlu melihat kode internalnya.

e. Evaluasi/Umpan Balik (Evaluation/Feedback)

Di akhir setiap *sprint*, versi aplikasi yang sudah berfungsi (meskipun belum lengkap) didemonstrasikan kepada pemangku kepentingan untuk mendapatkan umpan balik. Masukan ini sangat penting untuk mengevaluasi apakah hasil pengembangan sudah sesuai kebutuhan dan untuk merencanakan perbaikan atau penyesuaian pada *sprint* selanjutnya.

f. Peluncuran/Deployment (Launch/Deployment)

Setelah melalui beberapa siklus *sprint* dan pengujian, serta mendapatkan persetujuan dari pemangku kepentingan, versi aplikasi yang dianggap stabil dan memenuhi kebutuhan utama akan diluncurkan atau di-*deploy* agar dapat digunakan oleh pengguna akhir (Admin dan Pegawai). Proses Agile memungkinkan peluncuran fitur secara bertahap.

BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi masalah pada sistem pengelolaan dokumen yang berjalan saat ini dan menentukan spesifikasi yang harus dimiliki oleh sistem digital yang diusulkan.

3.1.1 Proses Analisis Kebutuhan

Proses analisis kebutuhan dilakukan untuk memahami masalah dan menentukan fitur yang diperlukan sistem. Proses ini menggunakan tiga metode utama:

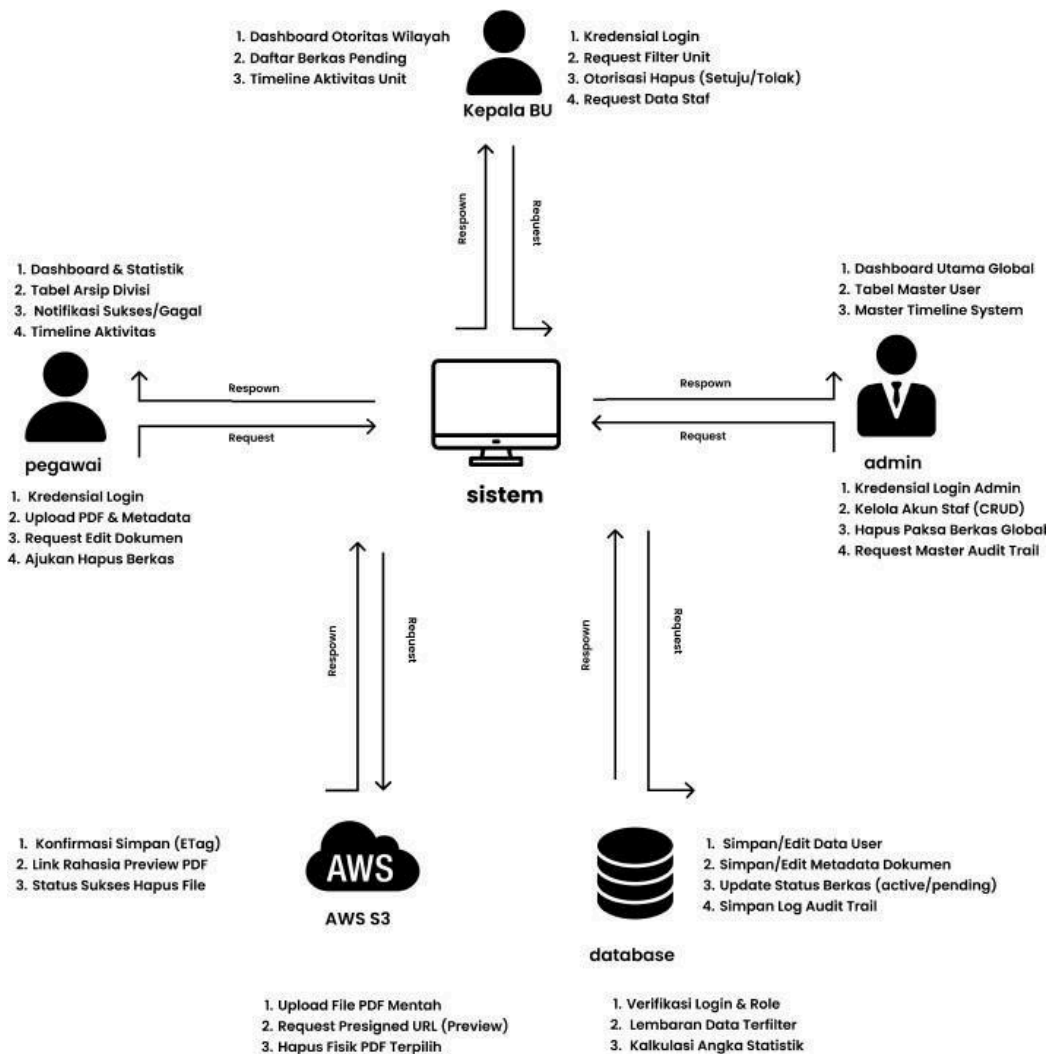
1. Studi Pustaka, yaitu Mengkaji jurnal, laporan, dan penelitian terkait untuk mempelajari solusi manajemen dokumen digital yang sudah ada, memahami teknologi relevan seperti *framework* Laravel dan layanan *cloud storage* AWS S3, serta mengidentifikasi celah atau area pengembangan yang dapat diisi oleh proyek ini.
2. Observasi, yaitu Mengamati alur kerja manual pengelolaan dokumen saat ini untuk mengidentifikasi inefisiensi secara langsung, seperti proses pencarian dokumen yang lambat, penumpukan arsip fisik, dan risiko kehilangan atau kerusakan data.
3. Wawancara Konseptual, yaitu Menggali kebutuhan spesifik dari calon pengguna (Admin dan Pegawai *document control*) untuk memahami kesulitan operasional yang mereka hadapi sehari-hari dan fitur-fitur apa saja yang paling diharapkan dari sebuah sistem digital terintegrasi.

3.1.2 Gambaran Umum Sistem

Sistem pengelolaan dokumen konvensional berbasis arsip fisik yang umum digunakan menghadapi berbagai kendala signifikan. Proses pencarian dokumen seringkali lambat dan tidak efisien karena memerlukan penelusuran manual, diperparah oleh penyimpanan yang mungkin tidak terorganisir. Selain itu, ketergantungan pada kertas membawa risiko tinggi terhadap kerusakan atau kehilangan data, membutuhkan ruang fisik yang besar serta biaya pemeliharaan, dan membatasi akses informasi hanya di lokasi kantor, yang menghambat fleksibilitas kerja modern.

Sebagai solusi, sistem manajemen dokumen digital berbasis web ini dirancang untuk mengatasi tantangan tersebut dengan mengadopsi konsep *paperless*. Memanfaatkan

framework Laravel dan integrasi penyimpanan awan AWS S3 untuk file PDF, sistem ini bertujuan menciptakan pengelolaan arsip yang terpusat, efisien, aman, dan mudah diakses dari mana saja. Dengan arsitektur *hybrid* yang mengelola metadata di database MySQL dan file fisik di AWS S3, solusi ini mendukung penuh transformasi digital organisasi.



Gambar 3.1 Gambaran Umum Sistem

Gambar 3.1 ini menjelaskan Sistem Pengarsipan Dokumen Finansial Terintegrasi (ARSIPIN) memproses interaksi data secara dinamis melalui tiga aktor utama, yaitu Pegawai, Kepala Business Unit (BU), dan Admin, yang menjalin komunikasi dua arah dengan Sistem sebelum diteruskan ke penyimpanan eksternal. Alur dimulai ketika pengguna mengirimkan *request* kredensial untuk autentikasi login, di mana Pegawai dapat mengunggah file PDF, memperbarui metadata, atau mengajukan permohonan hapus berkas, sementara Kepala BU

secara khusus mengirimkan perintah pemantauan berkas terfilter serta otorisasi validasi hapus dokumen staf, dan Admin berfokus pada manajemen akun kredensial serta penarikan seluruh data log audit global. Setiap permintaan teks dan manipulasi status berkas dari ketiga aktor tersebut akan diproses oleh Sistem dengan melempar query *request* ke Database MySQL untuk dikelola dan menerima *response* balik berupa data terfilter atau verifikasi hak akses, sedangkan file fisik PDF aslinya didelegasikan secara terpisah melalui protokol API ke AWS S3 Storage guna proses penyimpanan objek fisik mentah, penarikan *Presigned URL* untuk pratinjau aman, atau pemusnahan file fisik secara permanen dari *cloud*. Setelah seluruh rangkaian sinkronisasi internal antara Database dan AWS S3 selesai dieksekusi, Sistem akan mengembalikan *response* akhir secara *real-time* kepada masing-masing aktor berupa tampilan dashboard statistik yang adaptif, pembaruan tabel repositori dokumen, maupun notifikasi konfirmasi status tindakan yang sah sesuai dengan batasan hak akses role masing-masing.

3.2. Perancangan

Perancangan sistem mendefinisikan fungsionalitas dan fitur yang harus dimiliki oleh Sistem Manajemen Dokumen berbasis web ini agar dapat memenuhi tujuan yang telah ditetapkan pada tahap analisis kebutuhan. Kebutuhan fungsional berfokus pada tindakan-tindakan spesifik yang dapat dilakukan oleh pengguna (Admin dan Pegawai) dalam sistem.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah bagian penting dari perancangan sistem yang mendefinisikan apa yang harus dilakukan oleh sistem untuk memenuhi tujuan yang telah ditetapkan dalam tahap analisis kebutuhan. Dalam konteks aplikasi pembelajaran matematika berbasis gamifikasi ini, kebutuhan fungsional berfokus pada fitur dan fungsionalitas utama yang perlu ada untuk memberikan pengalaman pengguna yang optimal.

Tabel 3.1 Kebutuhan Fungsional

NO	Kebutuhan Fungsional
FR-01	user (admin, kepala bu, pegawai) dapat melakukan login
FR-02	user (admin, kepala bu, pegawai) dapat mengunggah dokumen baru

FR-03	user (admin, kepala bu, pegawai) dapat melihat daftar dokumen
FR-04	user (admin, kepala bu, pegawai) dapat mengedit metadata dokumen
FR-05	user (admin, kepala bu, pegawai) dapat melihat detail dokumen
FR-06	Pegawai dapat mengajukan hapus dokumen
FR-07	user (admin, kepala bu, pegawai) dapat mencari dokumen
FR-08	kepala BU dapat mengelola akun anggota
FR-09	kepala BU dapat menyetujui dan menolak ajuan penghapusan
FR-10	admin dapat mengelola akun sistem
FR-11	admin dapat menghapus dokumen
FR-12	user (admin, kepala bu, pegawai) dapat melihat log aktivitas
FR-13	Admin dapat mengelola jenis dokumen (Kategori Keuangan)
FR-14	user (admin, kepala bu, pegawai) dapat melakukan logout

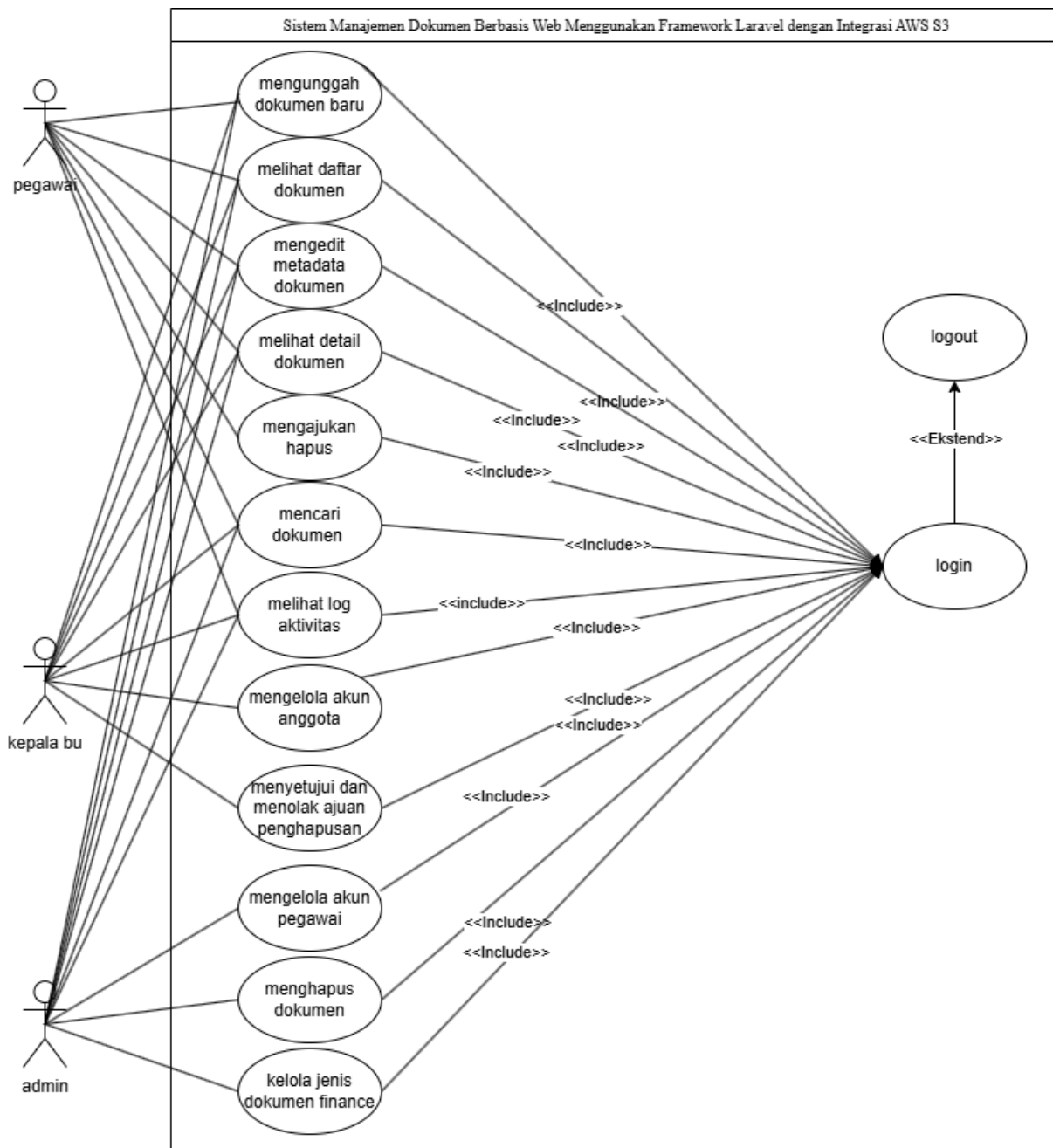
3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional berfokus pada kualitas dan karakteristik sistem yang mendukung kinerja dan pengalaman pengguna secara keseluruhan yang mencakup hal-hal yang tidak terkait langsung dengan fungsi utama sistem, namun sangat penting untuk memastikan sistem dapat berjalan dengan baik dan sesuai harapan.

Tabel 3.2 Kebutuhan Non Fungsional

NO	Non Fungsional
NFR-01	Sistem harus memastikan hanya pengguna terotentikasi dan berwenang (sesuai <i>role</i>) yang dapat mengakses data dan fitur. Data sensitif seperti <i>password</i> harus dienkripsi
NFR-02	Antarmuka pengguna harus intuitif, mudah dipelajari, dan konsisten di seluruh halaman agar pengguna dapat mengoperasikannya dengan minimal pelatihan.
NFR-03	Kode sumber harus ditulis dengan baik, terstruktur (mengikuti standar Laravel), dan didokumentasikan secukupnya agar mudah dipelihara dan dikembangkan lebih lanjut.

3.2.3 Diagram Use Case



Gambar 3.2 Use case diagram

Diagram use case di atas menggambarkan interaksi dua peran utama, Pegawai dan Admin, dengan Sistem Manajemen Dokumen. Pegawai dapat melakukan fungsi operasional dasar seperti Login, Mengunggah Dokumen Baru, Melihat Daftar Dokumen, Melihat Detail Dokumen, Mengedit Metadata (terbatas), Menghapus Dokumen (terbatas), Mencari Dokumen, dan Logout. Di sisi lain, Admin memiliki hak akses yang lebih luas, mencakup semua fungsi yang bisa dilakukan Pegawai, ditambah kemampuan administratif khusus yaitu

Mengelola Akun Pegawai dan Menetapkan Role Pegawai, menunjukkan perannya sebagai pengelola sistem secara keseluruhan.

3.2.4 Skenario Use Case

Menjelaskan detail langkah-langkah interaksi antara aktor dengan sistem untuk setiap use case. Disarankan menggunakan Tabel agar lebih rapi

Tabel 3.3. Skenario Use Case Melakukan Login

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna membuka browser dan mengakses halaman login aplikasi ARSIPIN .
Alur Normal	1. Pengguna memasukkan <i>username</i> atau email beserta <i>password</i> pada form login. 2. Pengguna mengeklik tombol "Masuk ke Sistem". 3. Sistem melakukan validasi kecocokan data kredensial dengan database MySQL lokal. 4. Sistem mengenali role pengguna dan mengarahkan (<i>redirect</i>) ke halaman dashboard sesuai wilayah otoritas kerja.

Alur Alternatif	Kredensial Salah: 1. Sistem mendeteksi data inputan tidak cocok dengan database. 2. Sistem membatalkan proses masuk dan menampilkan pesan eror merah di atas form.
Postkondisi	Pengguna berhasil masuk ke sistem dan mendapatkan <i>session</i> aktif untuk mengakses menu utama.

Tabel 3.4 Skenario Use Case Mengunggah Dokumen Baru

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna sudah login dan berada di halaman form tambah dokumen unit bisnisnya.
Alur Normal	1. Pengguna mengisi data dokumen (Nama, Jenis Dokumen, Bulan Buku, Tahun Buku, Jilid Buku). 2. JavaScript di <i>frontend</i> mendeteksi input secara <i>real-time</i> dan melakukan <i>auto-generate</i> nomor dokumen resmi. 3. Pengguna memilih file PDF resmi (maksimal 3MB). 4. Pengguna mengklik tombol "Simpan ke Arsip Digital".

	5. Sistem memvalidasi seluruh input data dan format berkas 6. Sistem mengunggah file fisik PDF mentah secara langsung ke Amazon S3 Cloud Storage 7. Amazon AWS S3 mengembalikan respons berupa status sukses dan <i>S3 Object Key</i> 8. Sistem meng-inject metadata dokumen beserta S3 Object Key tersebut ke dalam database MySQL lokal. 9. Sistem mencatat transaksi ini ke dalam jurnal log aktivitas audit trail dan menampilkan notifikasi sukses.
Alur Alternatif	Format File Bukan PDF atau Ukuran > 3MB: 1. JavaScript mendeteksi ekstensi file palsu atau ukuran gajah. 2. Sistem membekukan (<i>disable</i>) tombol submit dan menampilkan kotak pengumuman peringatan merah. 3. File di-reset otomatis agar user memilih ulang berkas yang valid.
Postkondisi	Berkas PDF berhasil diarsipkan di server dan nomor dokumen resmi terkunci aman di tabel database.

Tabel 3.5 Skenario Use Case Mengedit Metadata Dokumen

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna memilih salah satu dokumen aktif dan mengeklik tombol "Edit" (ikon pensil).
Alur Normal	1. Sistem memicu fungsi <i>explode()</i> PHP untuk memecah nomor dokumen lama ke kotak input Bulan, Tahun, dan Jilid secara otomatis. 2. Pengguna mengubah data metadata yang salah (misal: mengubah Jilid Buku dari 01 ke 02). 3. JavaScript merakit ulang nomor dokumen resmi yang baru di boks <i>readonly</i> secara <i>real-time</i>. 4. Pengguna mengeklik "Simpan Perubahan". 5. Controller mengeksekusi fungsi "updateDokumen" ke database, mencatat log audit, dan melakukan <i>redirect</i> sukses membawa alert hijau.
Alur Alternatif	Nomor Hasil Racikan Kembar (Duplikat di DB): 1. Backend mendeteksi nomor hasil edit ternyata kembar dengan dokumen milik berkas lain. 2. Sistem membatalkan proses simpan dan memunculkan

	pengumuman eror:
--	------------------

	<i>"The no dokumen has already been taken."</i>
Postkondisi	Metadata dokumen berhasil diperbarui di database tanpa merusak data file PDF utama jika tidak diganti.

Tabel 3.6 Skenario Use Case Mengajukan Hapus Dokumen

Aktor	Pegawai
Prekondisi	Pegawai berada di halaman daftar dokumen aktif milik unit bisnisnya.
Alur Normal	1. Pegawai memilih dokumen yang ingin dimusnahkan lalu mengeklik tombol "Hapus". 2. Sistem menangkap perintah dan tidak menghapus file, melainkan mengubah atribut status dokumen di database menjadi "pending_hapus". 3. Sistem mencatat aksi pengajuan ini ke dalam tabel "log_aktivitas" untuk timeline Kepala BU
Alur Alternatif	-

Postkondisi	Dokumen berpindah status menjadi antrean pemusnahan (<i>pending verification</i>) dan menghilang dari tabel dokumen aktif pegawai. .
-------------	--

Tabel 3.7 Skenario Use Case Menyetujui dan Menolak Ajuan Penghapusan

Aktor	Kepala BU
Prekondisi	Kepala BU masuk ke halaman menu Verifikasi Pemusnahan Dokumen.
Alur Normal	1. Kepala BU melihat daftar berkas berstatus “pending_hapus”. 2. Kepala BU mengklik tombol icon cheklist 3. Sistem memproses perintah, mengeksekusi perintah delete() permanen pada baris MySQL, menghapus fisik file PDF dari server lokal, dan memicu catatLog sukses.
Alur Alternatif	Kepala BU Menolak Hapus: 1. Kepala BU mengklik tombol icon silan. 2. Sistem memulihkan status dokumen dari “pending_hapus” kembali menjadi “active” (berkas aman tidak jadi musnah).
Postkondisi	Berkas benar-bener lenyap dari sistem jika disetujui, atau kembali utuh ke meja arsip jika ditolak.

Tabel 3.8 Skenario Use Case Menghapus Dokumen Mutlak

Aktor	Admin
Prekondisi	Admin masuk dengan akun ber-role 'admin' dan melihat tabel dokumen global.
Alur Normal	1. Admin memilih berkas apa saja di seluruh unit bisnis dan mengeklik tombol "Hapus Permanen". 2. Sistem memeriksa keamanan role di backend 3. Sistem langsung melompati gerbang approval, menghapus baris data di database, mendepak file PDF dari harddisk, dan mencatat log tindakan darurat pusat.
Alur Alternatif	Bukan Akun Admin (Hacker Iseng): 1. Sistem mendeteksi bypass rute oleh role non-admin. 2. Sistem memblokir paksa tindakan dan melemparkan halaman eror 403 Forbidden.

Postkondisi	Dokumen musnah secara mutlak seketika atas perintah Admin.
-------------	--

Tabel 3.9 Skenario Use Case Melihat Log Aktivitas (Audit Trail System)

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna telah berhasil <i>login</i> ke dalam sistem ARSIPIN dan memiliki sesi aktif.
Alur Normal	1. Pengguna mengeklik menu "Audit Trail Sistem" (Admin) atau "Timeline Aktivitas Unit / Kerja Saya" (Kepala BU / Pegawai) pada <i>sidebar</i> navigasi. 2. Sistem melakukan verifikasi <i>role</i> pengguna dan melempar <i>query</i> penarikan data ke tabel <i>log_aktivitas</i> di basis data MySQL. 3. Sistem menyajikan antarmuka <i>timeline</i> rekam jejak digital secara kronologis <i>real-time</i> sesuai batasan wilayah kerja (Admin: <i>global</i>, Kepala BU: <i>unit bisnis</i>, Pegawai: <i>log mandiri & atasan</i>). 4. (Opsional) Pengguna memilih kriteria filter spesifik (seperti filter nama staf atau jenis tindakan) lalu mengeklik tombol "Terapkan Filter". 5. Sistem memproses <i>filter</i> dan memperbarui tampilan daftar <i>log</i> aktivitas secara instan.

Alur Alternatif	Log Aktivitas Belum Tertera (<i>Empty State</i>): 1. Backend mendeteksi belum ada rekaman aksi operasional yang tersimpan di tabel log_aktivitas. 2. Sistem menampilkan informasi <i>empty state</i> dengan pesan: "<i>Belum ada riwayat aktivitas digital yang terekam pada sistem.</i>"
-----------------	---

Postkondisi	Seluruh kronologi transaksi pengarsipan, pembaruan metadata, dan riwayat persetujuan dokumen tersaji secara transparan dan dapat diaudit.
-------------	---

Tabel 3.10 Skenario Use Case Melihat Daftar Dokumen

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna telah berhasil login dan memilih salah satu unit bisnis serta perusahaan tujuan.
Alur Normal	1. Pengguna mengeklik menu navigasi arsip perusahaan pada sidebar. 2. Sistem memproses request parameter URL kluster divisi (Retail/Komersial) dan nama PT terkait. 3. Sistem melempar query penarikan data ke tabel dokumens di MySQL dengan filter unit bisnis yang sesuai. 4. Sistem menampilkan tabel daftar berkas digital secara lengkap beserta metadata, ukuran file, dan status sirkulasinya.

Alur Alternatif	Data Kosong (<i>Empty State</i>): 1. Sistem mendeteksi belum ada arsip dokumen yang diunggah untuk PT tersebut. 2. Sistem menampilkan tabel kosong disertai pemberitahuan informasi bahwa belum ada berkas yang diarsipkan
-----------------	--

Postkondisi	Pengguna berhasil melihat keseluruhan daftar arsip digital finansial yang terdaftar pada entitas perusahaan terkait.
-------------	--

Tabel 3.11 Skenario Use Case Melihat Detail Dokumen

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna berada di halaman daftar dokumen dan memilih salah satu berkas arsip.
Alur Normal	1. Pengguna mengeklik tombol pratinjau ("<i>Lihat</i>" atau ikon PDF) pada baris dokumen. 2. Sistem memvalidasi eksistensi fisik objek berkas di dalam penyimpanan cloud AWS S3. 3. Sistem memicu fungsi <i>Presigned URL</i> untuk menghasilkan tautan sementara berdurasi 5 menit dengan tanda tangan kriptografi. 4. Sistem mengalihkan (<i>redirect</i>) pengguna untuk membuka dokumen PDF di tab pratinjau terpisah secara

	aman.
Alur Alternatif	Berkas Fisik Tidak Ditemukan di Cloud S3: 1. Sistem mendeteksi <i>object key</i> gagal divalidasi pada direktori cloud. 2. Sistem menghentikan proses dan memunculkan <i>response</i> error 404 (<i>File Not Found</i>).
Postkondisi	Pengguna dapat melihat isi dokumen PDF secara jelas tanpa mengekspos visibilitas <i>bucket</i> S3 secara publik ke internet.

Tabel 3.12 Skenario Use Case Mencari Dokumen

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
Prekondisi	Pengguna berada di halaman tabel repositori dokumen suatu perusahaan.
Alur Normal	1. Pengguna mengetik kata kunci (nama berkas atau nomor dokumen resmi) pada kolom pencarian (search bar). 2. Pengguna menekan tombol Enter atau membiarkan sistem menyaring data secara real-time. 3. Sistem menjalankan fungsi pencarian berbasis klausa LIKE pada basis data relasional MySQL. 4. Sistem menyaring dan menyajikan baris data dokumen yang cocok secara instan pada antarmuka tabel.

Alur Alternatif	Kata Kunci Tidak Ditemukan: 1. Sistem mendeteksi tidak ada kesesuaian string pencarian pada database. 2. Sistem menampilkan tampilan tabel kosong yang menginformasikan bahwa data pencarian tidak ditemukan.
-----------------	---

Postkondisi	Pengguna berhasil menemukan dokumen spesifik secara cepat tanpa harus melakukan penelusuran manual satu per satu.
-------------	---

Tabel 3.13 Skenario Use Case Mengelola Akun Anggota (Kepala BU)

Aktor	Kepala Business Unit (Kepala BU)
Prekondisi	Pengguna login sebagai Kepala BU dan mengakses menu "Kelola Staf Unit".
Alur Normal	1. Kepala BU membuka akordeon daftar kandidat pegawai baru (pending). 2. Kepala BU memeriksa profil akun staf yang didaftarkan oleh Admin untuk unit bisnis yang sama. 3. Kepala BU mengeklik tombol "+ Rekrut" pada baris kandidat pegawai pilihan. 4. Sistem memperbarui status kolom keaktifan akun dari pending menjadi approved di database MySQL.

	5. Sistem menampilkan notifikasi sukses dan akun pegawai resmi bergabung ke dalam staf aktif unit bisnis terkait.
Alur Alternatif	Keluaran Staf dari Unit: 1. Kepala BU mengeklik tombol hapus/keluarkan pada baris staf aktif. 2. Sistem mengubah status akun pegawai kembali menjadi pending dan memutus akses operasionalnya dari unit tersebut.

Postkondisi	Kepala BU berhasil memvalidasi dan mengontrol struktur keanggotaan staf operasional di bawah wilayah wewenangnya.
-------------	---

Tabel 3.14 Skenario Use Case Mengelola Akun Pegawai (Admin)

Aktor	Admin
Prekondisi	Pengguna login sebagai Admin dan mengakses menu "Kelola Akun Pegawai".

Alur Normal	1. Admin mengeklik tombol "+ Tambah Anggota" atau tombol "Edit" pada tabel pengguna. 2. Admin mengisi formulir (Nama, Username, Email, Password, Role, dan Penempatan Bisnis Unit) atau memperbarui datanya. 3. Sistem memvalidasi keunikan data (unique check untuk email/username) dan mencocokkan format input. 4. Sistem mengenkripsi password menggunakan algoritma hashing, lalu menyimpan atau memperbarui data akun ke tabel users di MySQL.
Alur Alternatif	Validasi Ganda / Duplikat Gagal: 1. Admin memasukkan email atau username yang sudah terdaftar sebelumnya. 2. Sistem menggagalkan penyimpanan dan memunculkan pesan peringatan error validasi.

Postkondisi	Admin berhasil melakukan manajemen data kredensial seluruh pengguna sistem secara terpusat.
-------------	---

Tabel 3.15 Skenario Use Case Logout (Keluar Sistem)

Aktor	Admin, Kepala BU, Pegawai Retail, Pegawai Komersial
-------	---

Prekondisi	Pengguna berada di dalam sistem dan memiliki sesi aktif (logged in).
Alur Normal	1. Pengguna mengeklik tombol "Keluar" (ikon merah) pada bagian bawah menu sidebar. 2. Sistem menangkap permintaan pengakhiran sesi (logout). 3. Sistem menghapus token sesi aktif (session invalidation) dari memori server. 4. Sistem mengalihkan (redirect) pengguna kembali ke halaman utama login dengan membawa pesan notifikasi keluar yang aman.
Alur Alternatif	Tidak ada alur alternatif; proses penghancuran token sesi bersifat mutlak.

Postkondisi	Sesi pengguna terputus secara total, sehingga peramban (browser) terkunci dan tidak dapat mengakses halaman internal tanpa login ulang.
-------------	---

Tabel 3.16 Skenario Use Case Mengelola Kategori Dokumen

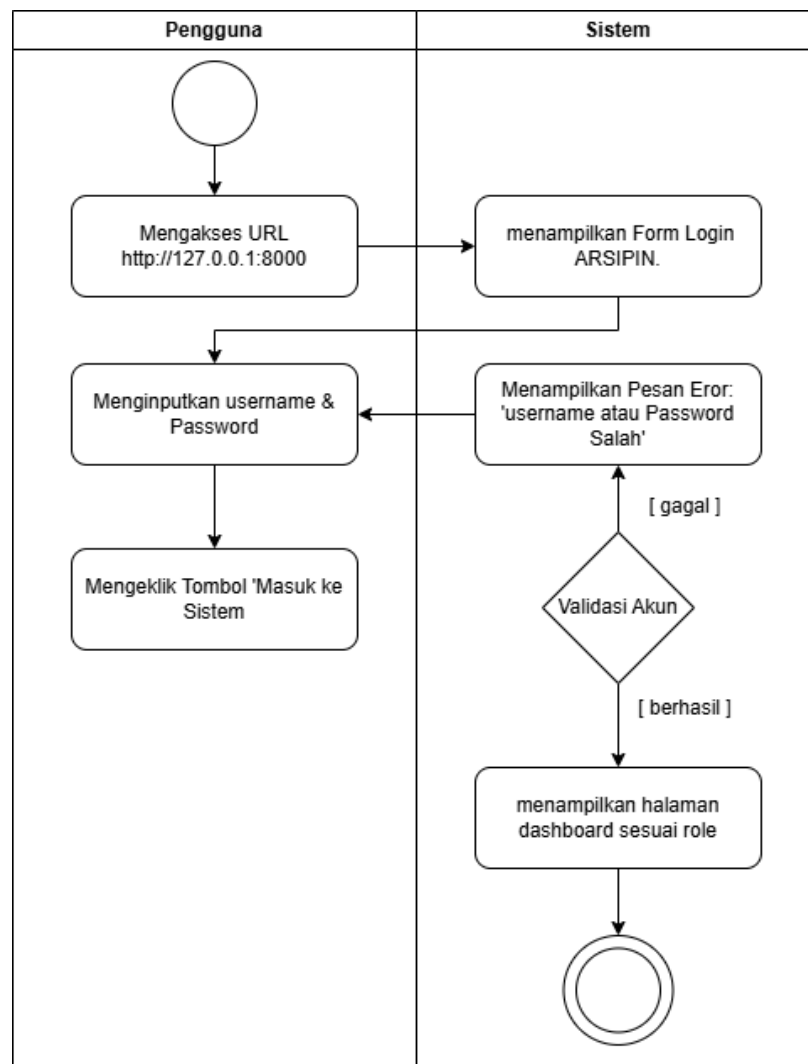
Aktor	Admin
-------	-------

Prekondisi	Admin telah <i>login</i> ke sistem dan mengakses menu "Kelola Kategori".
Alur Normal	1. Admin mengeklik menu kategori pada sidebar. 2. Sistem menampilkan tabel daftar kategori yang tersedia (misal: Rekening Koran, Slip Gaji, Pajak, Voucher). 3. Admin mengisi nama kategori dan singkatan (misal: RK, SG, TX, VC). 4. Admin mengeklik tombol "Tambah Kategori". 5. Sistem memvalidasi data dan menyimpannya ke tabel kategoris di database. 6. Sistem menampilkan notifikasi sukses dan memperbarui tabel secara real-time.
Alur Alternatif	Data Kategori Kosong: 1. Admin mengeklik tombol simpan tanpa mengisi form. 2. Sistem menampilkan pesan peringatan field required agar Admin melengkapi data.

Postkondisi	Jenis dokumen baru tersedia di dropdown form upload dokumen (baik di retail maupun komersial) untuk digunakan oleh seluruh pegawai.
-------------	---

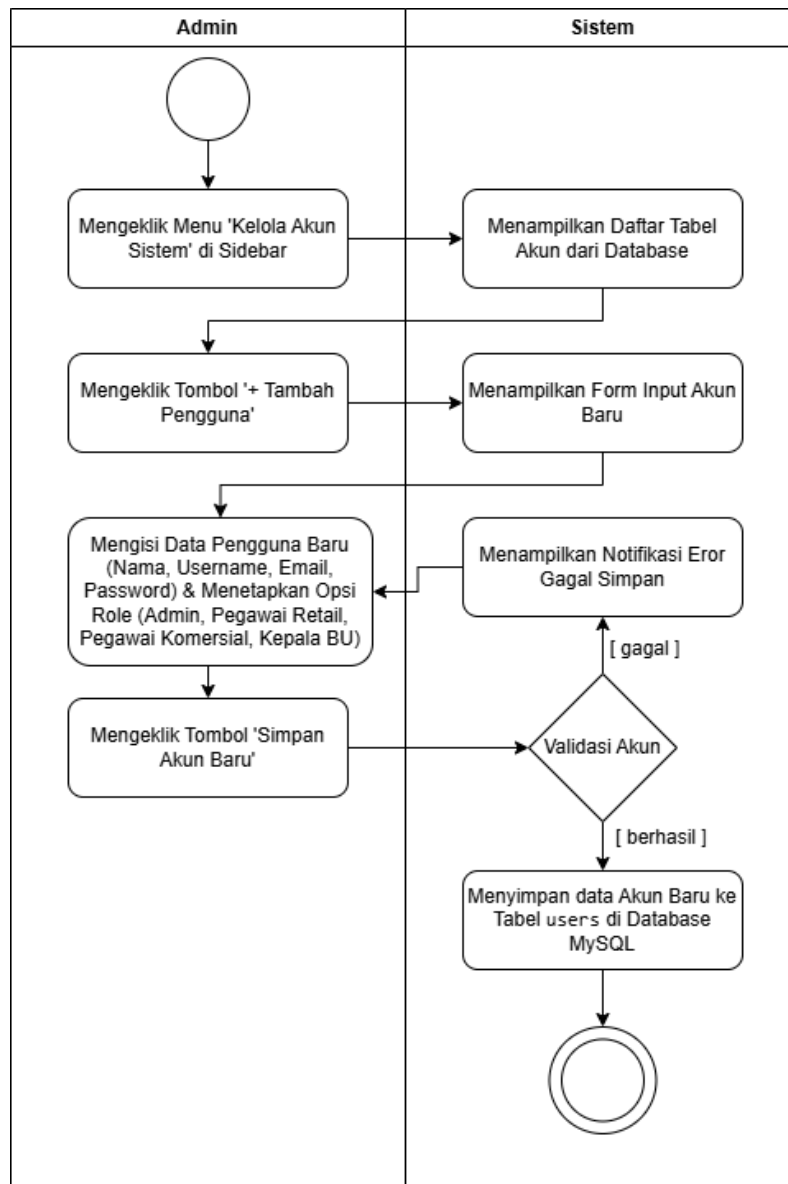
3.2.5 Activity Diagram

Activity diagram adalah salah satu jenis diagram dalam UML (Unified Modeling Language) yang digunakan untuk menggambarkan alur aktivitas atau proses kerja dalam suatu sistem secara berurutan



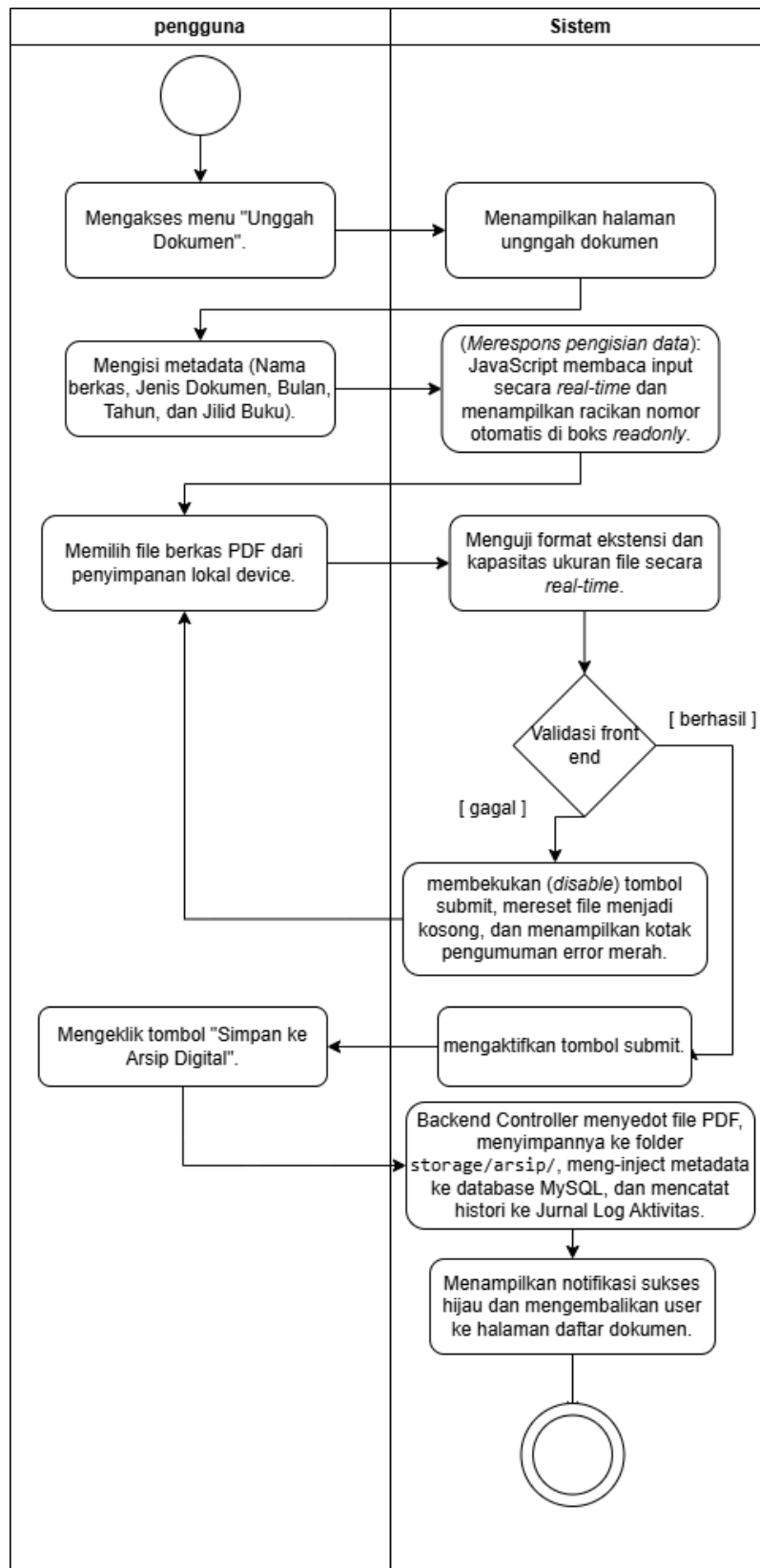
Gambar 3.3 pengguna login

Gambar 3.3 Diagram ini menggambarkan alur kerja sistem login dari sistem ARSIPIN, proses ini mencakup pengguna untuk mengisi username dan password yang sudah sesuai dengan apa yang telah didaftarkan oleh admin, ketika pengguna berhasil maka akan diarahkan ke halaman dashboard, adapun jika gagal maka akan mendapatkan pesan error



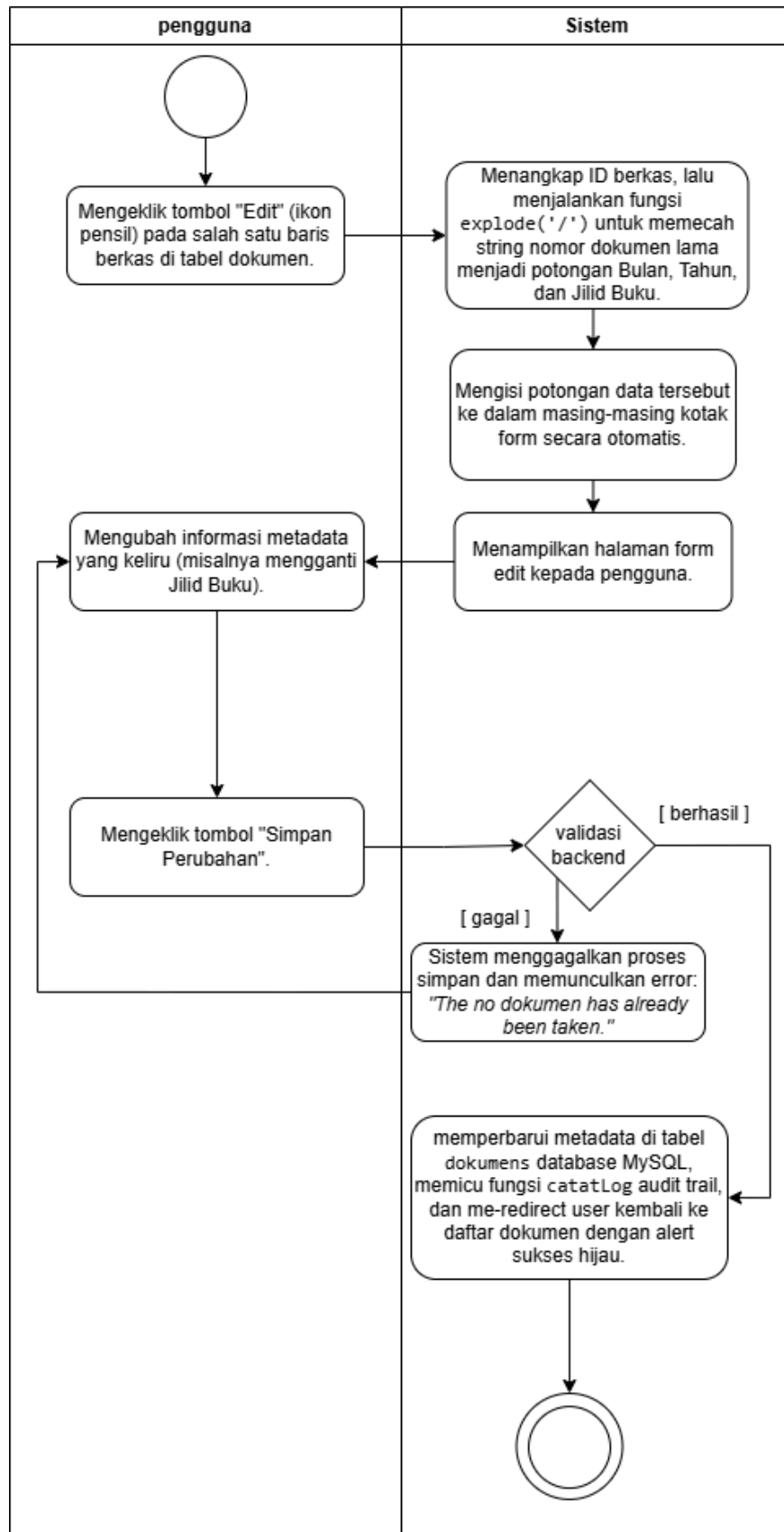
Gambar 3.4 Admin Kelola Akun

gambar 3.4 Diagram ini menggambarkan alur kerja Admin dalam memanipulasi data kredensial pegawai pada sistem ARSIPIN. Proses mencakup penampilan daftar akun, penambahan akun baru (nama, email, password, *role*, unit bisnis) disertai validasi sistem, serta pembaruan (*update*) dan penghapusan (*delete*) data pegawai yang sudah tidak aktif ke dalam database.



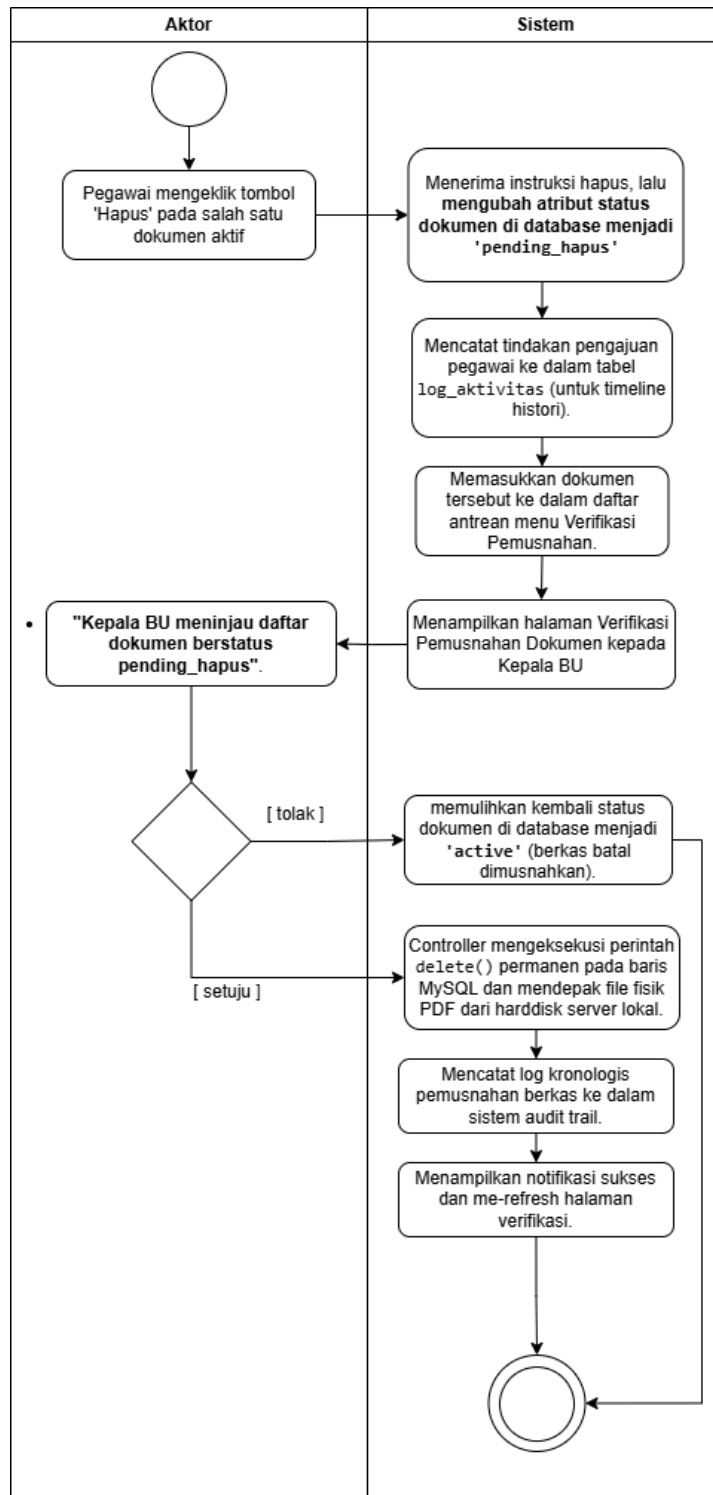
Gambar 3.5 Pengguna Unggah Dokumen

Gambar 3.5 Diagram ini menggambarkan alur kerja ketika pengguna melakukan Upload Dokumen pada sistem ARSIPIN, proses ini mencakup pengguna untuk melakukan upload Dokumen, pada saat pengguna mengisi metadata dan memilih format file harus dalam bentuk PDF, jika tidak dalam bentuk PDF maka JS akan mengunci tombol submit dan menampilkan pesan error merah, tetap jika itu PDF maka dapat menekan tombol submit



Gambar 3.6 Pengguna Edit Dokumen

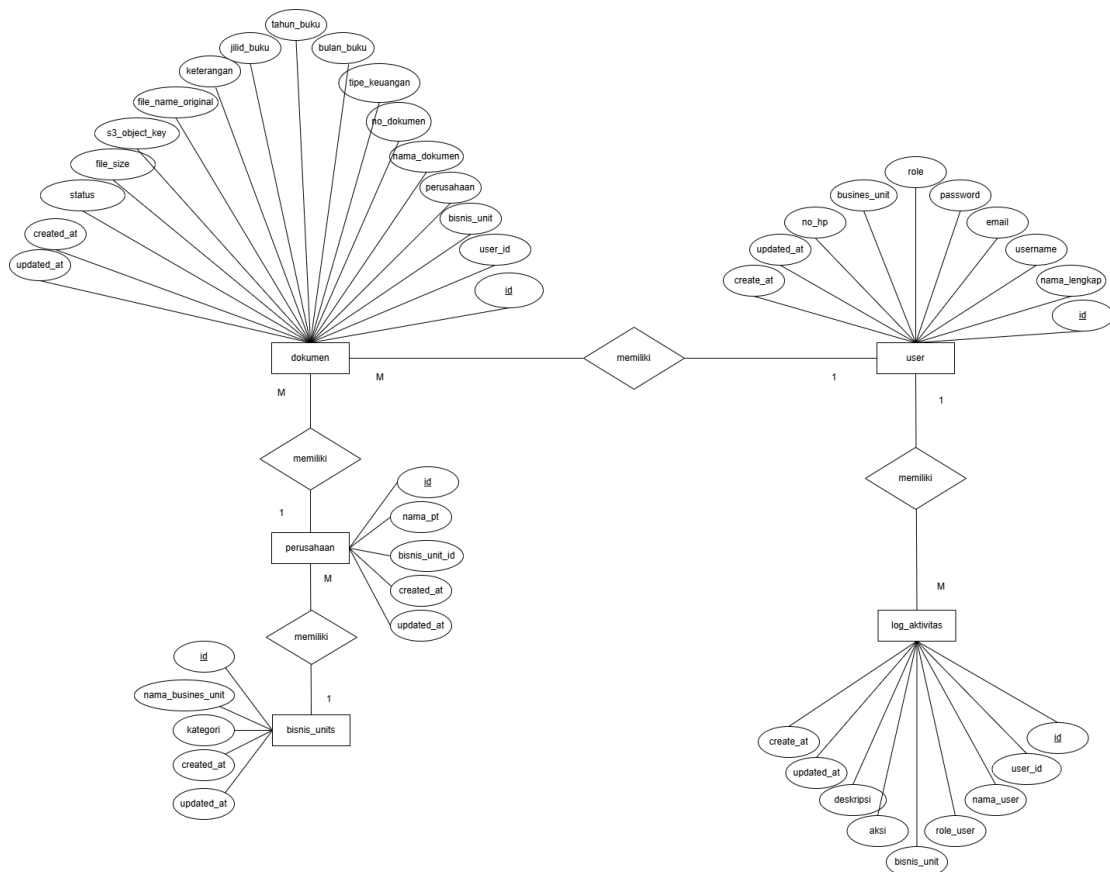
Gambar 3.6 Diagram ini menggambarkan alur kerja ketika pengguna melakukan edit Dokumen dan yang berubah hanyalah meta datanya saja, ketika pada tahap edit ini jika nomor dokumen terdetekt nomor dokumen yang sama maka sistem akan menggagalkan perubahan dan menampilkan pesan error



Gambar 3.7 Alur Pengajuan Dan Penyetujuan Penghapusan Dokumen

Gambar 3.7 Diagram ini menggambarkan alur kerja ketika pegawai ingin memusnahkan atau menghapus suatu dokumen pada sistem ARSIPIN, ketika pegawai melakukan pengajuan penghapusan maka hal itu akan masuk dalam log aktivitas pegawai, kemudian dokumen tadi akan masuk pada halaman peninjauan dokumen yang akan dimusnahkan, setelah itu jika admin setuju maka akan terhapus dan jika tidak setuju maka dokumen akan kembali seperti semula

3.2.6 ER Diagram



Gambar 3.8 ERD

Gambar 3.8 ini adalah ERD yang sudah sesuai dengan database mysql dari website ARSIPIN, berikut adalah penjelasan nya

1.user

Pengguna adalah entitas yang merepresentasikan individu yang memiliki hak akses dan berinteraksi langsung dengan sistem ARSIPIN. Entitas ini mencakup berbagai tingkatan hak akses, mulai dari Admin Utama (Mata Elang Pusat), Kepala Unit Bisnis (Otoritas Approval), hingga Pegawai Staf (Filing Officer Divisi Retail/Komersial).

a. **id (PK)**

ID unik numerik (Auto Increment) untuk setiap pengguna yang terdaftar. Bertindak sebagai *Primary Key*. .

b. **nama_lengkap**

Nama lengkap pengguna.

c. **username**

Nama unik pengguna tanpa spasi yang digunakan sebagai kredensial keamanan saat proses login.

d. **email**

Alamat surat elektronik unik milik pengguna yang juga berfungsi sebagai ID login sistem.

e. **password**

Kata sandi rahasia pengguna yang disimpan dalam bentuk terenkripsi aman (*hashing system*)

f. **role**

Tingkatan hak akses pengguna dalam sistem, bernilai: “admin”, “pegawai-retail”, “pegawai-komersial” dan “kepala_bu”

g. **bisnis_unit**

Stempel waktu (timestamp) kapan akun dibuat.

h. **no_hp**

Nomor telepon seluler aktif milik pengguna.

i. **created_at**

Stempel waktu otomatis (*timestamp*) saat data akun pengguna pertama kali dibuat.

j. **updated_at**

Stempel waktu otomatis saat data akun pengguna mengalami perubahan terakhir.

2.bisnis_unit

Bisnis Unit adalah entitas yang merepresentasikan pembagian kluster atau pos unit usaha spesifik yang dinaungi oleh perusahaan induk. Unit usaha ini dipisah secara tegas berdasarkan kategori rumpun operasionalnya.

a. **id (PK)**

ID unik numerik untuk setiap Unit Bisnis yang bertindak sebagai *Primary Key*.

b. nama_bisnis_unit

Nama unik dari pos unit bisnis spesifik (contoh: 'spbu', 'lpg-pso', 'shipyard').

c. kategori

Pengelompokan jenis rumpun operasional bisnis unit, bernilai: 'retail' atau 'commercial'.

d. created_at

Stempel waktu otomatis saat pos unit bisnis pertama kali ditambahkan ke sistem.

e. updated_at

Stempel waktu otomatis saat ada perubahan informasi data pada unit bisnis terkait.

3.perusahaan

Perusahaan adalah entitas yang merepresentasikan nama anak perusahaan atau PT (*Perseeroan Terbatas*) yang bernaung secara resmi di bawah otoritas Unit Bisnis tertentu. Entitas ini menjadi wadah pengelompokan sekunder sebelum berkas dokumen diarsipkan.

a. id (PK)

ID unik numerik untuk setiap data nama entitas PT yang bertindak sebagai *Primary Key*.

b. nama_pt

Nama resmi dari perseeroan terbatas yang dikelola (contoh: 'PT SCK', 'PT SBS').

c. bisnis_unit_id(FK)

Kunci asing (*Foreign Key*) yang merujuk pada id di entitas “bisnis_units”, berfungsi mengunci di bawah Unit Bisnis mana PT tersebut beroperasi.

d. created_at

Stempel waktu otomatis saat data nama PT berhasil didaftarkan ke sistem.

e. updated_at

Stempel waktu otomatis saat data nama PT mengalami pembaruan informasi.

4.dokumen

Dokumen adalah entitas inti (*core entity*) dari sistem ARSIPIN yang berfungsi menyimpan seluruh informasi data (*metadata*) dari berkas arsip fisik berbentuk PDF yang telah diunggah dan disimpan ke dalam sistem penyimpanan digital.

a. **id (PK)**

ID unik numerik dari setiap arsip digital yang masuk ke sistem. Bertindak sebagai *Primary Key*.

b. **user_id(FK)**

Kunci asing (*Foreign Key*) yang merujuk pada id di entitas “users”, sebagai perekam jejak digital siapa akun pegawai yang mengunggah berkas tersebut.

c. **perusahaan_id(FK)**

Kunci asing (*Foreign Key*) yang merujuk pada id di entitas “perusahaans”, sebagai penanda mutlak dokumen tersebut merupakan hak milik dari berkas PT mana. Merujuk ke id pada entitas *bisnis_unit*

d. **bisnis_unit**

Salinan data string nama unit bisnis dokumen untuk mempercepat proses pencarian query dashboard global.

e. **perusahaan**

Salinan data string nama PT dokumen guna mempermudah penampilan visual tabel daftar berkas.

f. **nama_dokumen**

Judul atau penamaan deskriptif berkas yang diinput oleh pengguna.

g. **no_dokumen**

Nomor identifikasi resmi dokumen hasil rumusan rakitan otomatis sistem menggunakan template JavaScript: “JENIS/PT/BULAN/TAHUN/JILID”. Bersifat unik (*Unique*).

h. **tipe_keuangan**

Klasifikasi rumpun jenis dokumen keuangan, bernilai: 'Invoice', 'Voucher', 'Pajak', atau 'Rekening Koran' .

i. **bulan_buku**

Data string bulan buku dokumen, mendukung format bulan tunggal maupun rentang pembukuan

j. **tahun_buku**

Data string tahun buku dokumen, mendukung format tahun tunggal maupun lintas tahun anggaran

k. **jilid_buku**

Data string penunjuk nomor urut jilid bundel fisik dokumen

l. keterangan

Catatan ringkas tambahan mengenai isi berkas dokumen pembukuan (bersifat opsional).

m. file_name_original

Nama berkas file asli berformat PDF saat pertama kali dipilih dari device pengguna.

n. s3_object_key

Kunci alamat objek (*file path string*) unik yang mengidentifikasi letak fisik berkas PDF di dalam direktori penyimpanan server lokal maupun cloud AWS S3.

o. file_size

Keterangan ukuran kapasitas data file yang disimpan

p. status

Status sirkulasi aktif berkas arsip di dalam sistem

q. created_at

Stempel waktu otomatis saat metadata arsip pertama kali disimpan (waktu unggah).

r. updated_at

Stempel waktu otomatis saat metadata berkas mengalami perubahan data atau status.

5.log_aktivitas

Log Aktivitas adalah entitas baru yang bertindak sebagai mesin perekam (*Audit Trail*) berkas digital secara transparan. Entitas ini mencatat seluruh rekam jejak aksi operasional yang dilakukan oleh seluruh aktor pengguna untuk disajikan kembali dalam bentuk timeline kronologis.

a. id (PK)

ID unik numerik dari setiap baris rekaman aktivitas sistem yang bertindak sebagai *Primary Key*.

b. user_id(FK)

Kunci asing (*Foreign Key*) yang merujuk pada id di entitas “users”, mencatat siapa aktor utama yang melakukan tindakan tersebut.

c. nama_user

Duplikasi string nama lengkap aktor saat melakukan tindakan demi menjaga stabilitas data histori jika akun di masa depan terhapus.

d. role_user

Keterangan jabatan peran aktor saat beraktivitas.

e. bisnis_unit

Kunci string nama unit bisnis lokasi tempat terjadinya aktivitas, berfungsi menyaring penampilan timeline agar sesuai dengan wilayah kerja Kepala BU terkait.

f. aksi

Nama kategori tindakan sistem yang dijalankan

g. deskripsi

Kalimat narasi naratif panjang yang menjelaskan detail aktivitas kerja secara transparan.

h. created_at

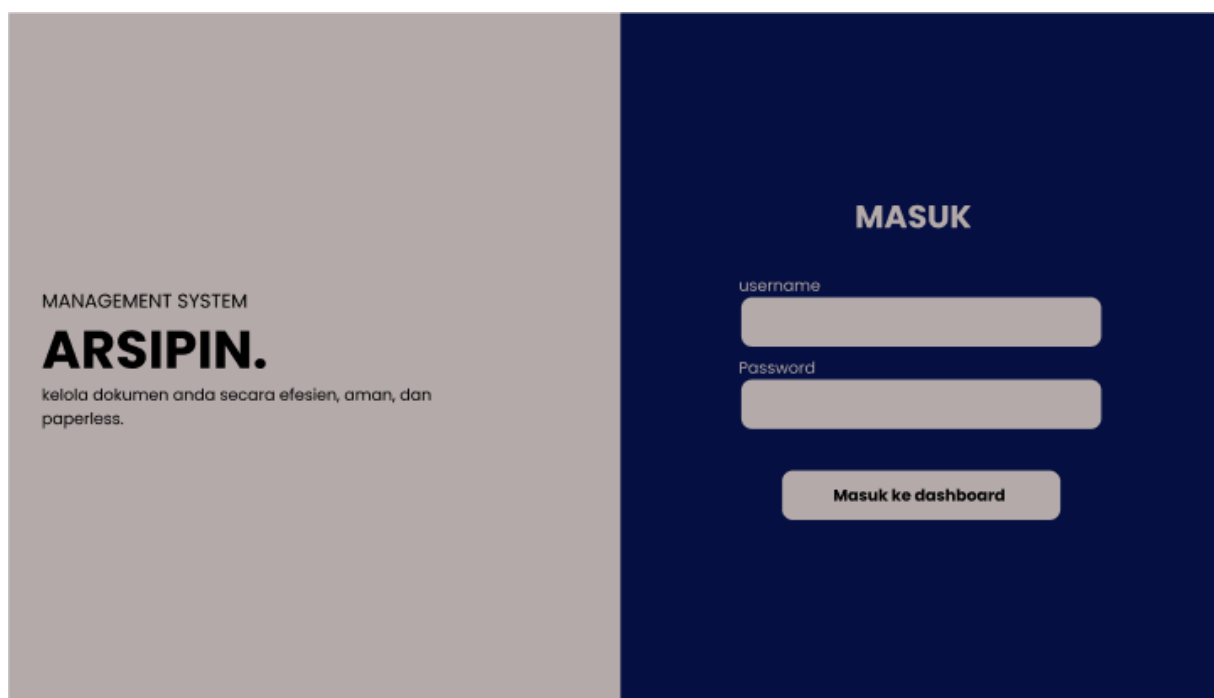
Stempel waktu otomatis (*timestamp*) kapan aktivitas digital itu terjadi di dalam sistem ARSIPIN.

i. updated_at

Stempel waktu otomatis perubahan data log

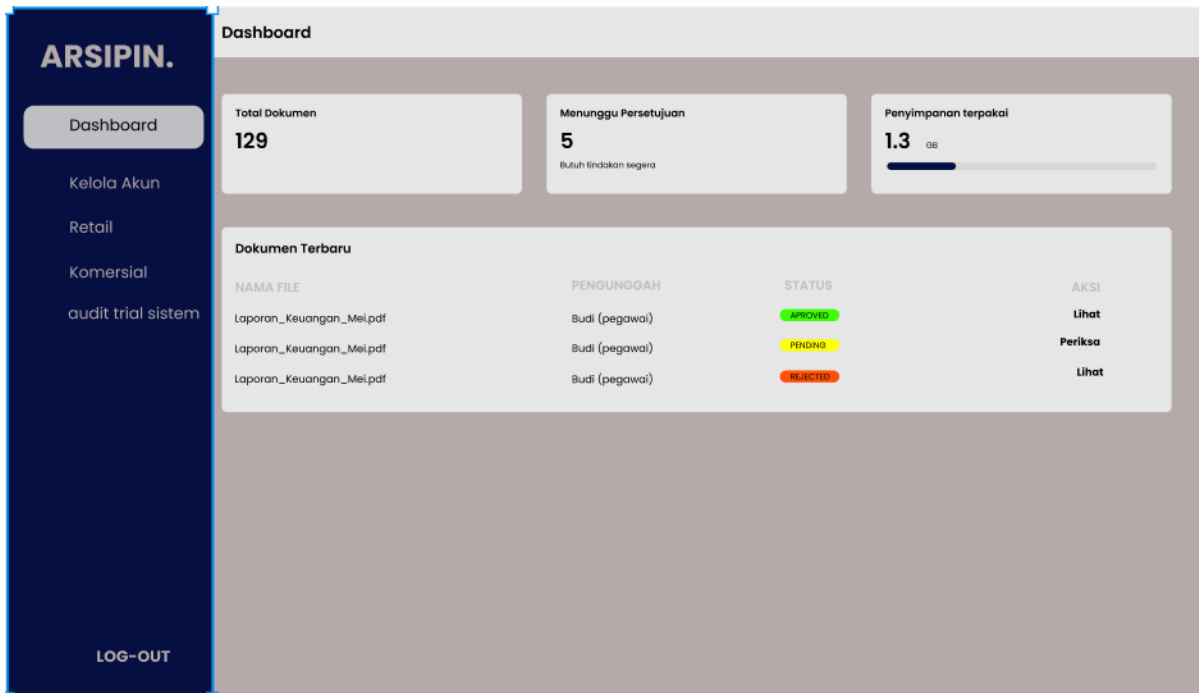
3.2.7 Perancangan Antarmuka (wireframe)

Berikut merupakan rancangan wireframe antarmuka sistem yang digunakan sebagai acuan pada tahap implementasi.



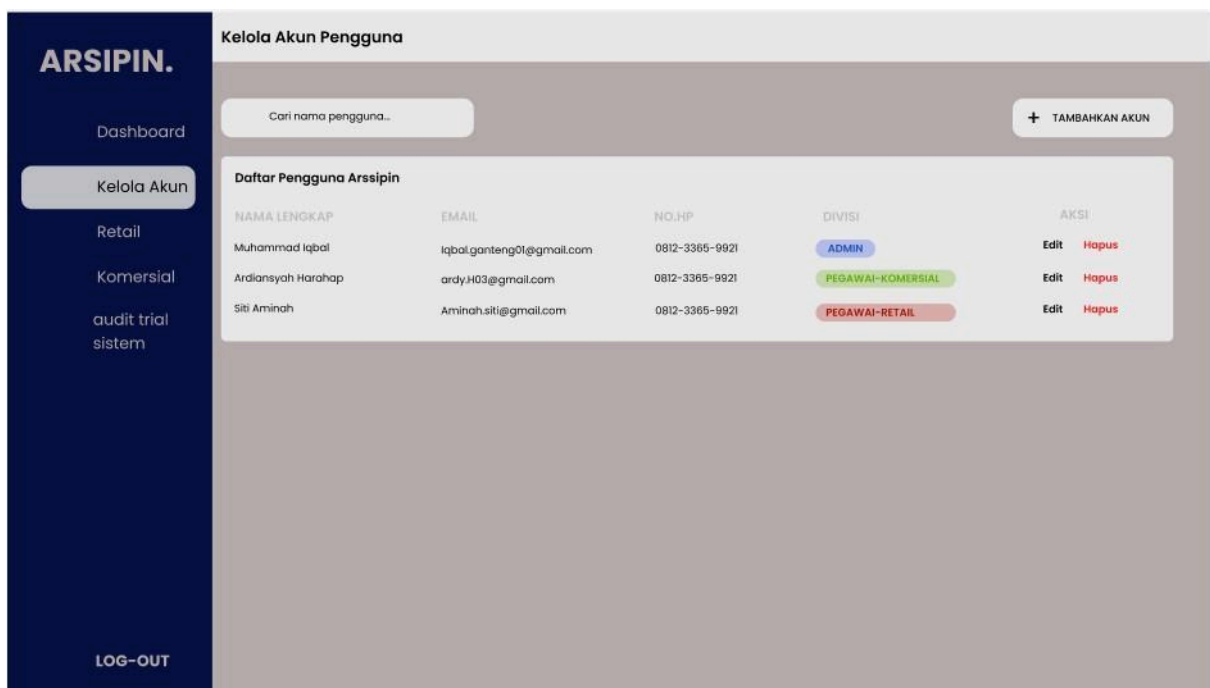
Gambar 3.9 Wireframe Login Page

Gambar 3.9 menunjukkan rancangan wireframe tampilan login pada website Arsipin sebelum tahap implementasi dilakukan.



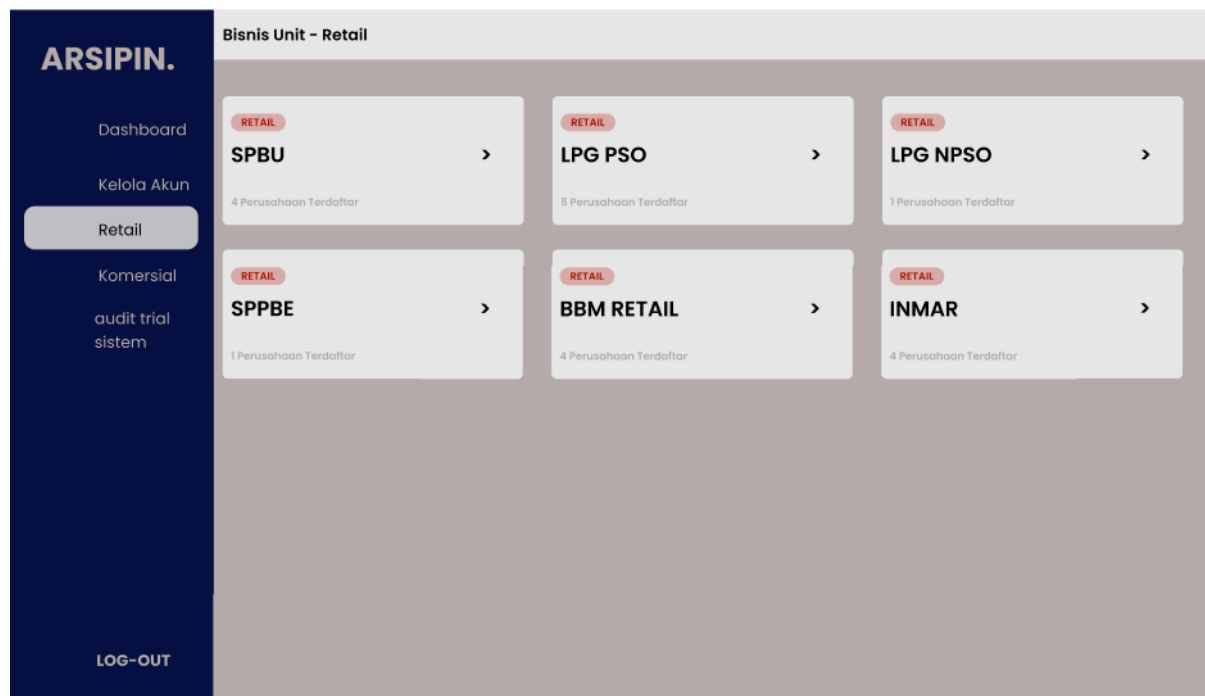
Gambar 3.10 Wireframe Dashboard Admin

Gambar 3.10 menunjukkan rancangan wireframe tampilan dashboard admin pada website Arsipin sebelum tahap implementasi dilakukan.



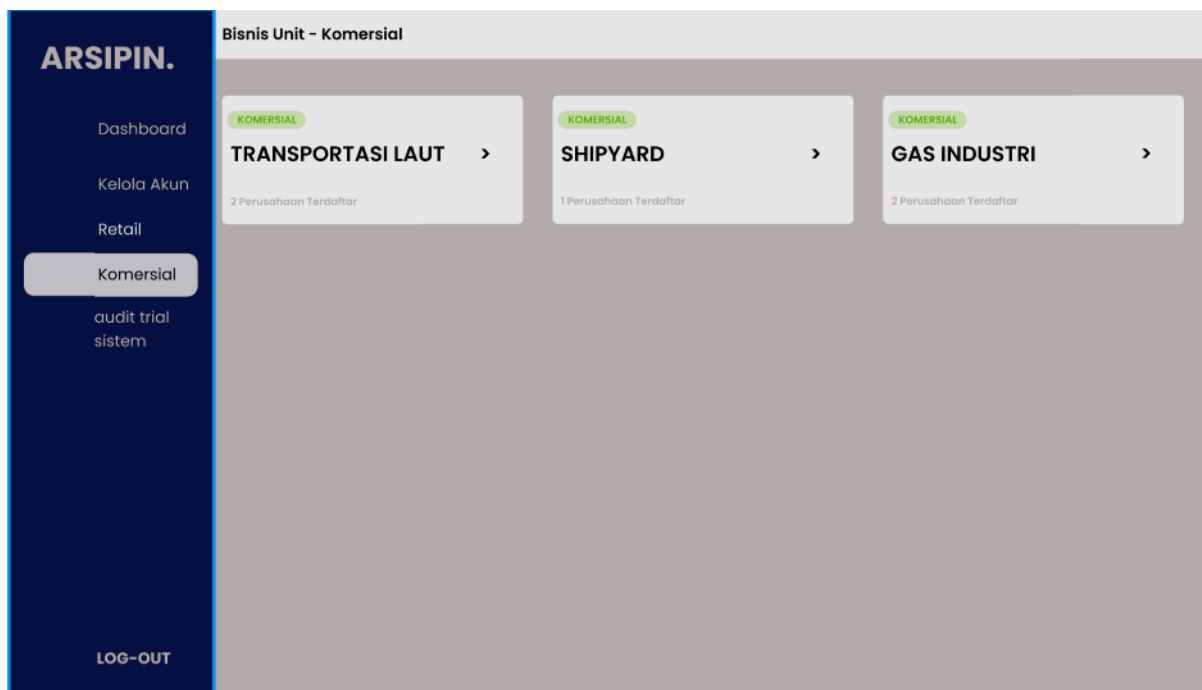
Gambar 3.11 Wireframe Kelola Akun Admin

Gambar 3.11 menunjukkan rancangan wireframe tampilan kelola akun pengguna di bagian admin pada website Arsipin sebelum tahap implementasi dilakukan



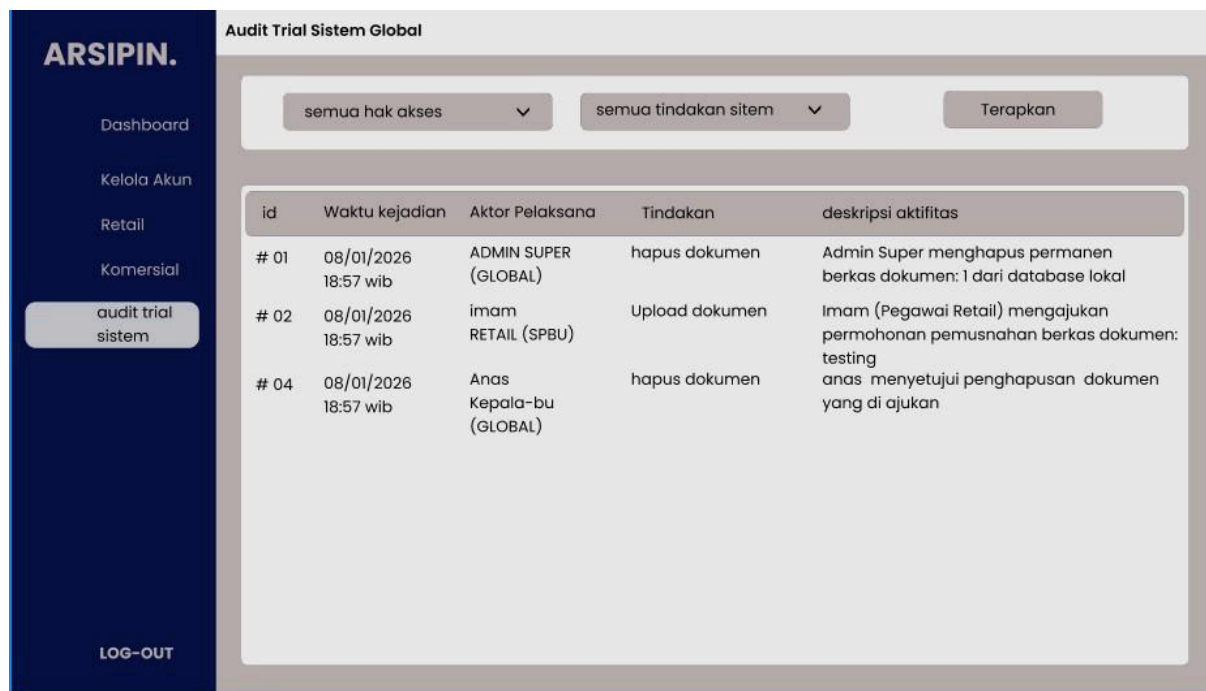
Gambar 3.12 Wireframe Bisnis Unit-Retail Admin

Gambar 3.12 menunjukkan rancangan wireframe tampilan Bisnis Unit-Retail di bagian admin pada website Arsipin sebelum tahap implementasi dilakukan.



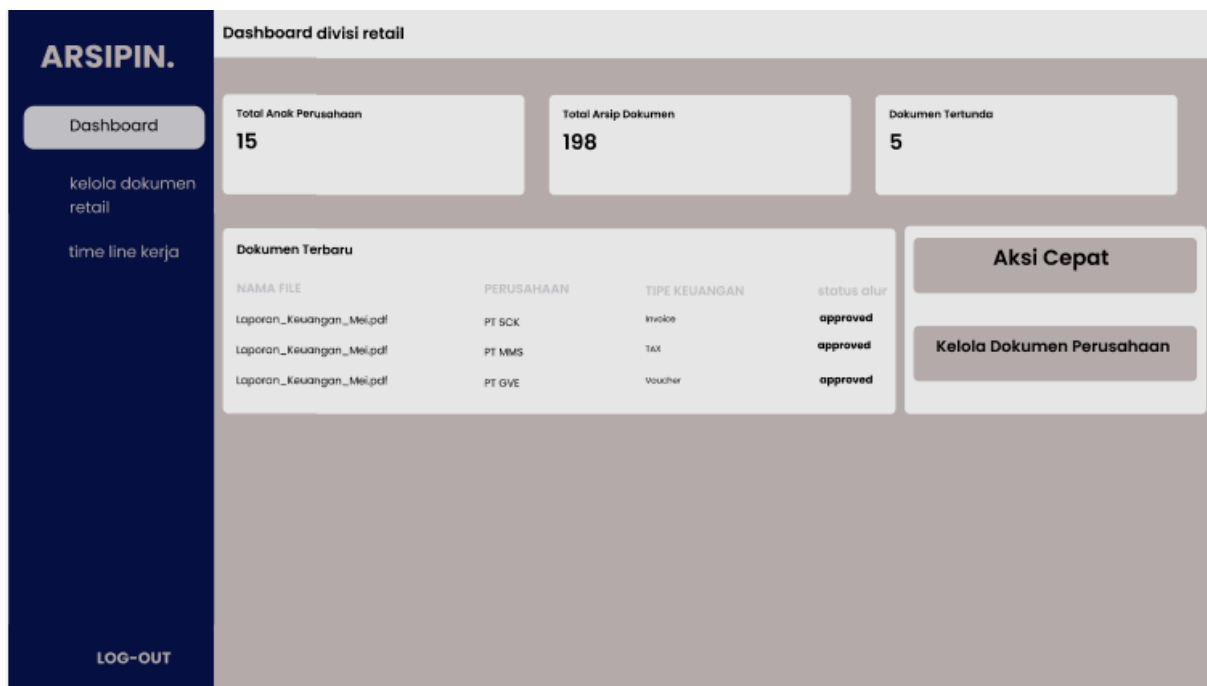
Gambar 3.13 Wireframe Komersial Admin

Gambar 3.13 menunjukkan rancangan wireframe tampilan Bisnis Unit-komersial di bagian admin pada website Arsipin sebelum tahap implementasi dilakukan.



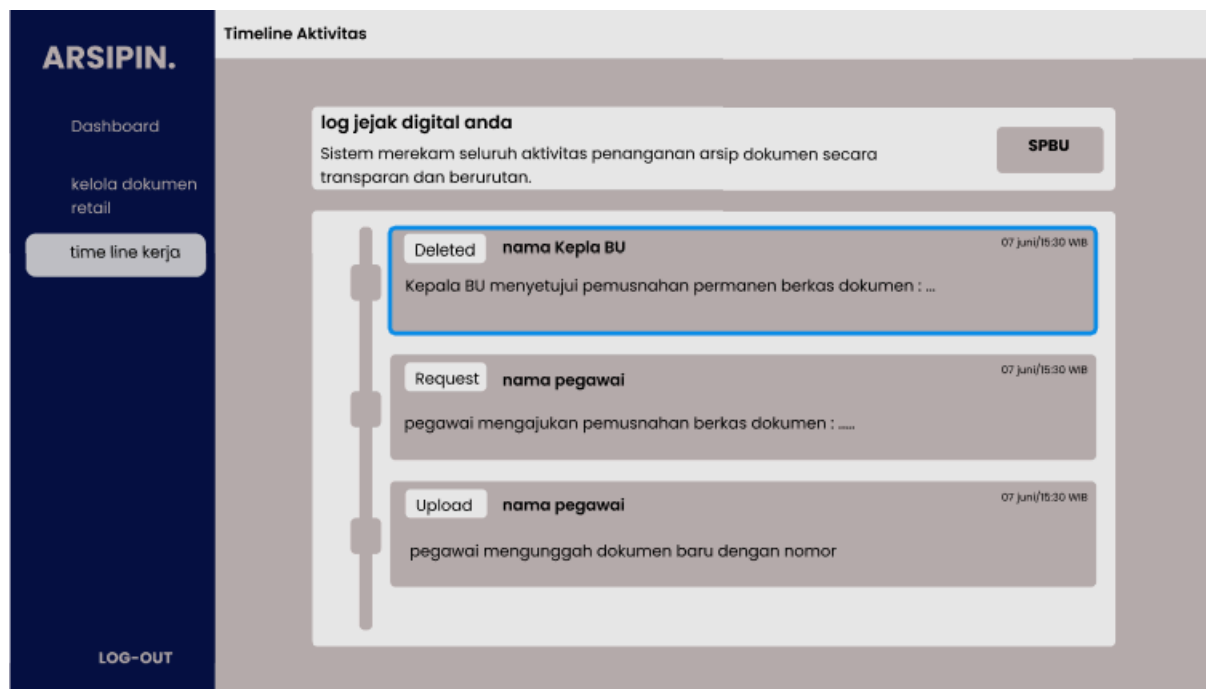
Gambar 3.14 Wireframe audit trail sistem (log aktivitas)

Gambar 3.14 menunjukkan rancangan wireframe tampilan log aktivitas di bagian admin pada website Arsipin sebelum tahap implementasi dilakukan.



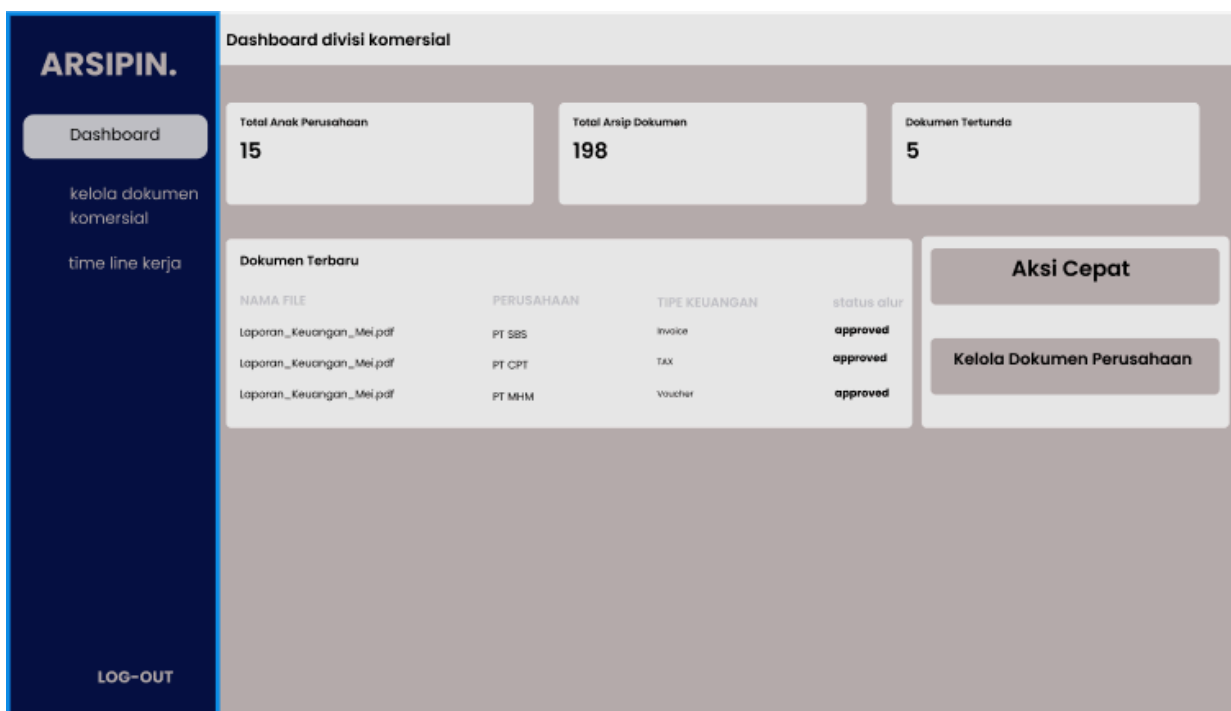
Gambar 3.15 Wireframe Dashboard Pegawai-Retail

Gambar 3.15 menunjukkan rancangan wireframe tampilan dashboard pegawai retail pada website ARSIPIN sebelum tahap implementas



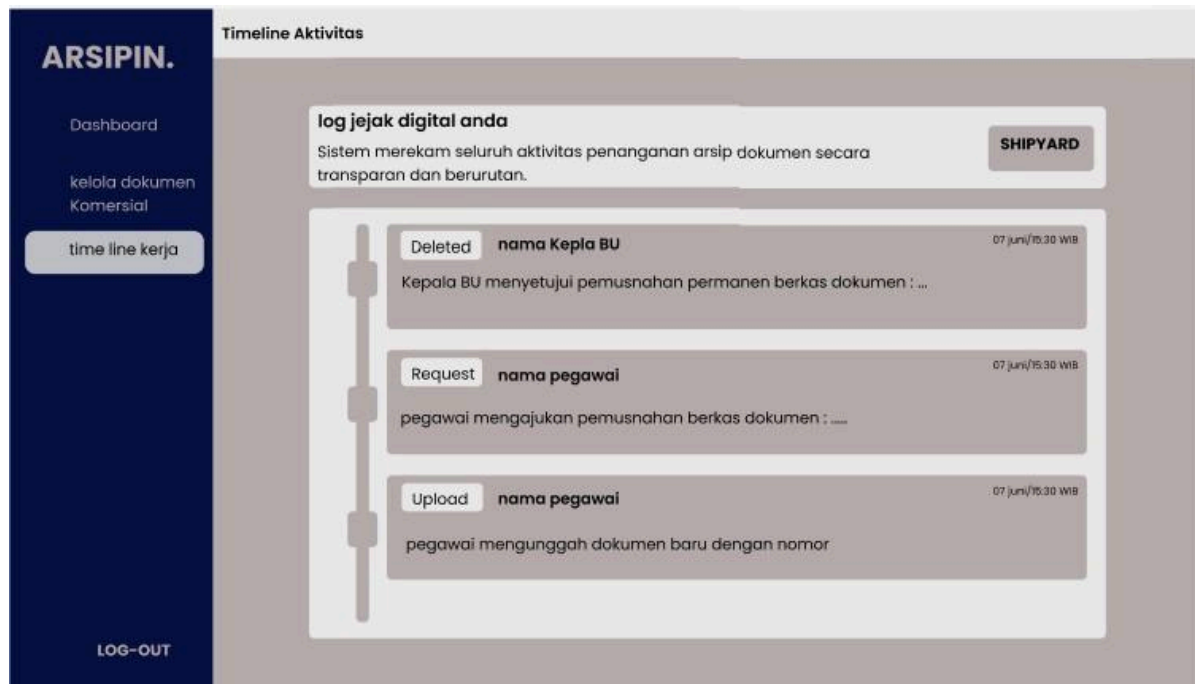
Gambar 3.16 Wireframe Log Aktivitas Pegawai-Retail

Gambar 3.16 menunjukkan rancangan wireframe tampilan Log aktivitas atau timeline pegawai retail pada web ARSIPIN sebelum tahap implementasi



Gambar 3.17 Wireframe Dashboard Pegawai-Komersial

Gambar 3.17 menunjukkan rancangan wireframe tampilan dashboard pegawai Komersial pada website ARSIPIN sebelum tahap implementasi



Gambar 3.18 Wireframe Log Aktivitas Pegawai-Komersial

Gambar 3.18 menunjukkan rancangan wireframe tampilan Log aktivitas atau timeline pegawai komersial pada web ARSIPIN sebelum tahap implementasi

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

Tahap implementasi merupakan proses penerapan hasil perancangan sistem ke dalam bentuk aplikasi yang dapat digunakan. Sistem yang dibangun pada Proyek Akhir ini berupa website Rancang Bangun Sistem Manajemen Dokumen Berbasis Web Menggunakan Framework Laravel dengan Integrasi AWS S3

4.1.1 Implementasi Lingkungan Pengembangan

Lingkungan pengembangan sistem yang digunakan meliputi:

Tabel 4.1 Implementasi Lingkungan Pengembangan

Framework	Laravel 11
Bahasa Pemrograman	PHP
Database	MySql
Frontend	Blade template, Tailwind CSS, JS
Web Server	Apache

4.1.2 Implementasi Fitur Sistem

Implementasi sistem dibagi menjadi tiga bagian utama, yaitu sisi pegawai baik retail dan komersial kemudian kepala BU dan admin

a. Implementasi Halaman Login

Selamat Datang Kembali

Silakan masukkan kredensial akun Anda untuk mengakses pusat data pembukuan dan audit finansial.

USERNAME / EMAIL PENGGUNA

test@example.com

KATA SANDI

MASUK KE SISTEM

© 2025 ARSIPIN, Politeknik Negeri Batam. All Rights Reserved.

INTEGRASI TATA KELOLA DOKUMEN FINANSIAL PUSAT.

Pusat manajemen dan pelacakan arsip digital pembukuan yang terintegrasi. Dirancang khusus untuk memfasilitasi audit keuangan perusahaan melalui validasi berkas yang akurat, transparan, dan terlacak secara kronologis ("audit trail tracing").

100%
INTEGRITAS AUDITING

< 3 MB
RESTRIKSI BERKAS

Activate Windows
Go to Settings to activate Windows.

Gambar 4.1 Halaman Login

Gambar 4.1 menunjukkan tampilan halaman Login untuk website Arsipin, disana terdapat username dan password yang harus diinputkan oleh pengguna nantinya sesuai dengan yang di buatkan oleh admin

b. Implementasi Halaman Dashboard Admin

ARSIPIN.

Dashboard Admin

KONTROL ADMIN

Kelola Akun Pegawai

Arsip Retail (All)

Arsip Komersial (All)

Audit Trail Sistem

Admin Super

ADMIN

KELUAR

Dashboard Utama Admin

Pusat Otoritas Dokumen

MATRIKS DAN LOG PENUTUPAN BUKU FINANSIAL

08 Jun 2026 | 18:56:04 WIB

TOTAL ARSIP BERKAS

4

PENGGUNA AKTIF

5

ANTRIAN PENGHAPUSAN

0

ENTITAS PERUSAHAAN

24

Jurnal Riwayat Berkas Terbaru

Arsip data yang baru saja ter-upload masuk ke server

Semua

Retail

Komersial

INFORMASI BERKAS	UNIT OPERASIONAL	ENTITAS	STATUS	KLASIFIKASI	AKSI
testing 200/SCX/06-11/2025/02	SPBU	PT SCK	ACTIVE	Invoice	Buka
tes 01 1m/sbs/2022/01	SHEPHERD	PT SBS	ACTIVE	Invoice	Buka
tes 02 1m/cpt/2020/01	TRANSPORTASI LAUT	PT CPT	ACTIVE	Invoice	Buka
tes 01 1m/cpt/2020/01	TRANSPORTASI LAUT	PT CPT	ACTIVE	Pajak	Buka

Volume Trafik Berkas Finansial

Gambarkan jumlah arsip penutupan buku berdasarkan tahun

Skala: Tahunan

Activate Windows

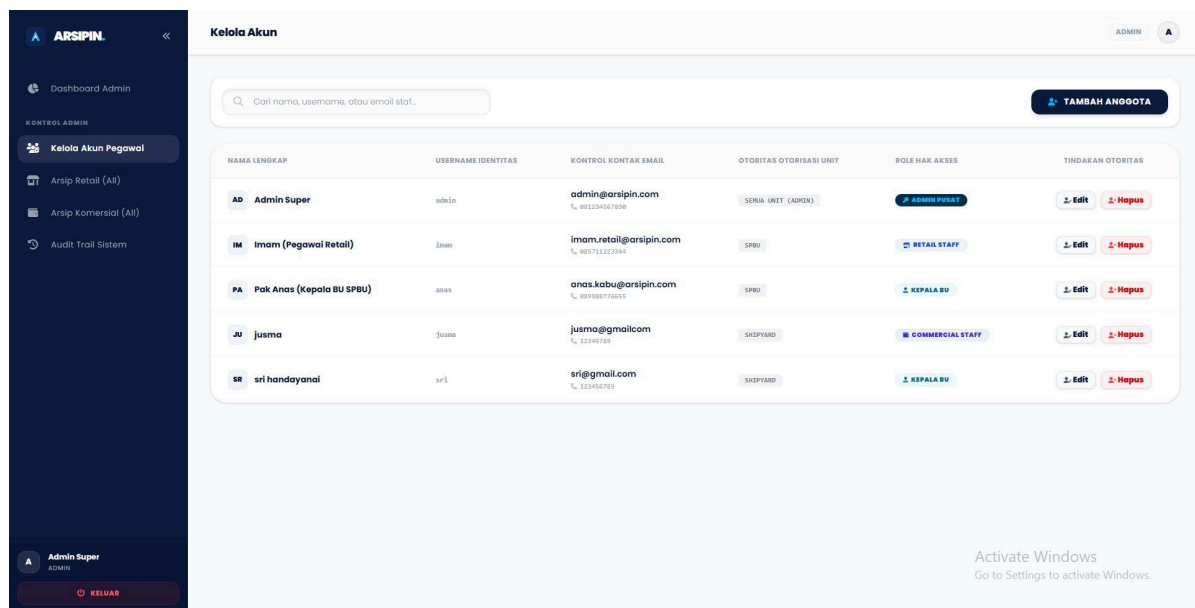
Integritas Penyimpanan AWS S3

Status area kapasitas ruang penyimpanan cloud storage

Gambar 4.2 Halaman Login

Gambar 4.2 menunjukkan tampilan halaman dashboard admin untuk website Arsipin, disana terdapat ringkasan dari total berkas dokumen arsip, pengguna aktif, antrian pengajuan hapus, total perusahaan, kemudian juga terdapat daftar dokumen yang baru-baru masuk serta terdapat chart visual untuk total dokumen yang masuk dalam beberapa tahun, bulan, minggu terakhir dan juga chart total penyimpanan yang telah digunakan di aws s3

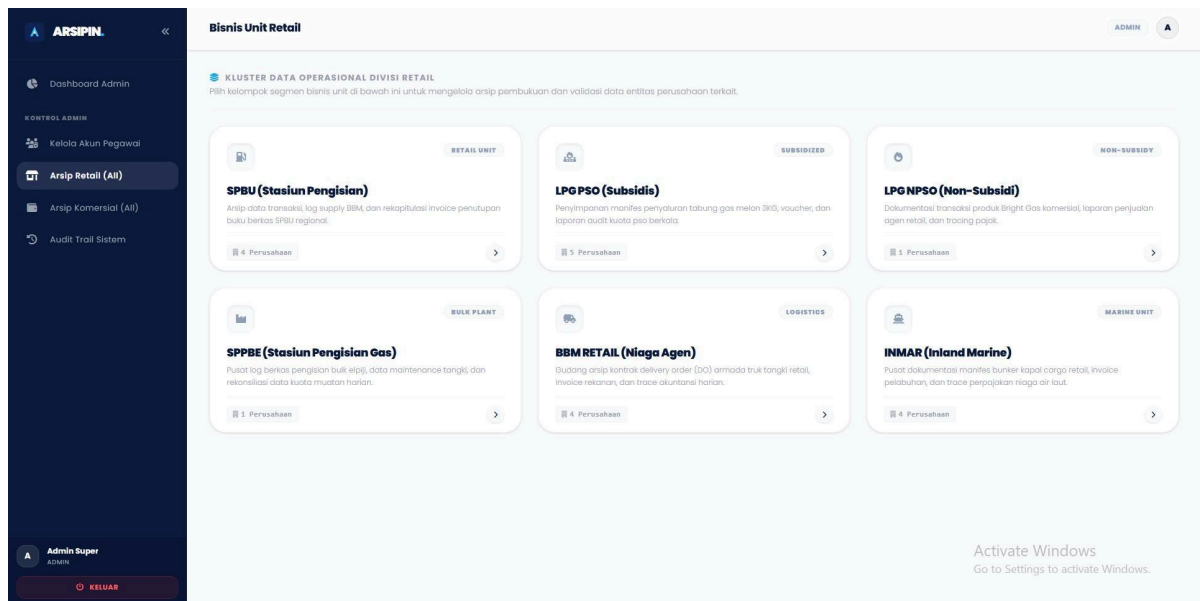
c. Implementasi Halaman Kelola Akun Pegawai



Gambar 4.3 Halaman Kelola Akun Pegawai

Gambar 4.3 menunjukkan tampilan halaman kelola akun pegawai yang dikelola oleh Admin untuk website Arsipin, disana terdapat form tambah pengguna, juga terdapat informasi detail dari setiap pengguna seperti nama lengkap, username identitas, kontrol kontak email, otoritas otoritas unit, role hak akses, serta aksi berupa edit dan hapus untuk akun pengguna

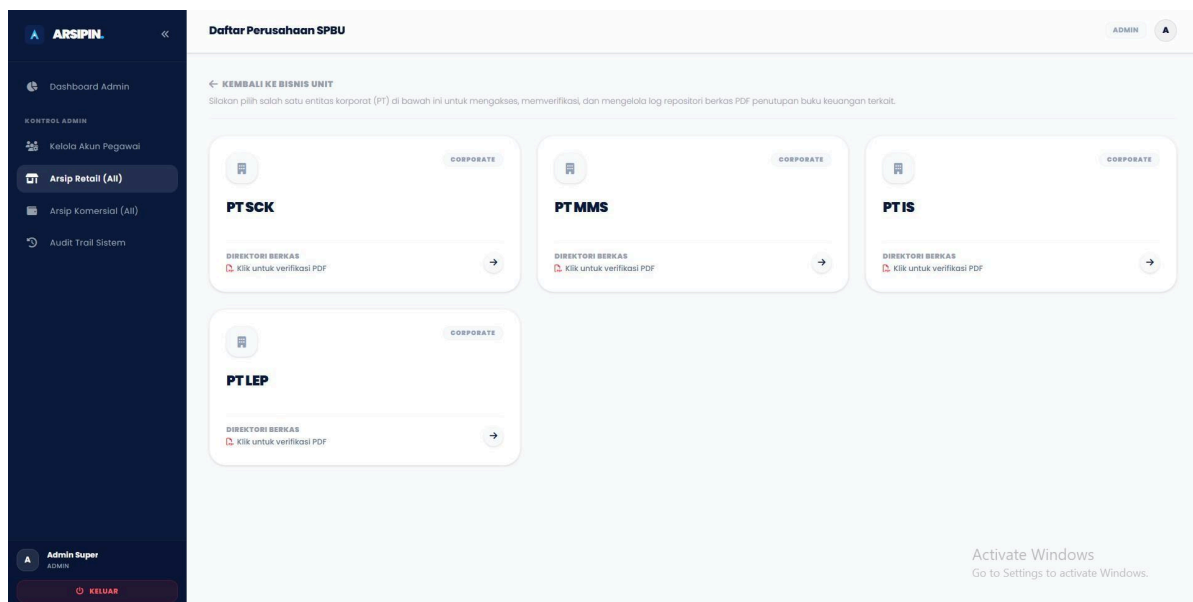
d. Implementasi Halaman Bisnis Unit-Retail



Gambar 4.4 Halaman Bisnis Unit-Retail

Gambar 4.4 menunjukkan tampilan halaman Bisnis Unit dari Retail untuk website Arsipin, disana terdapat Bisnis Unit yang terdaftar untuk bagian retail, kemudian 1 Bisnis Unit menaungi beberapa perusahaan di dalamnya, tetapi juga ada yang hanya menaungi 1 perusahaan

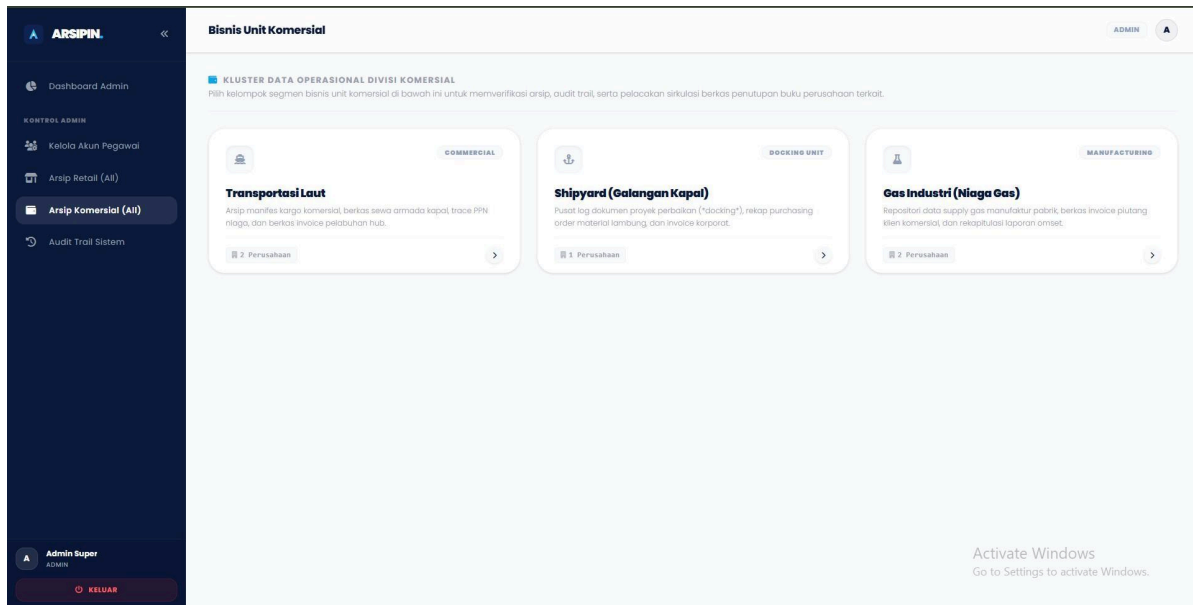
e. Implementasi Halaman Daftar Perusahaan-Retail



Gambar 4.5 Halaman Daftar Perusahaan-Retail

Gambar 4.5 menunjukkan tampilan halaman Perusahaan-Retail, disana terdapat perusahaan perusahaan yang dinaungi oleh Bisnis Unit-Retail sebelumnya

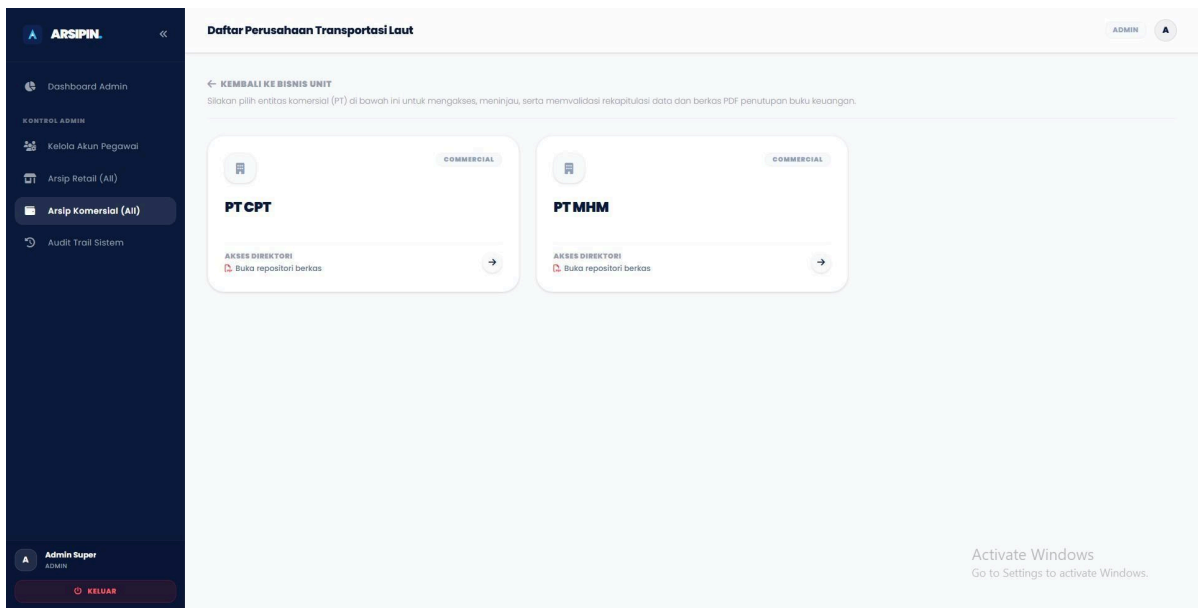
f. Implementasi Halaman Bisnis Unit-Komersial



Gambar 4.6 Halaman Bisnis Unit-Komersial

Gambar 4.5 menunjukkan tampilan halaman Bisnis Unit dari komersial untuk website Arsipin, disana terdapat Bisnis Unit yang terdaftar untuk bagian komersial, kemudian 1 Bisnis Unit menaungi beberapa perusahaan di dalamnya, tetapi juga ada yang hanya menaungi 1 perusahaan

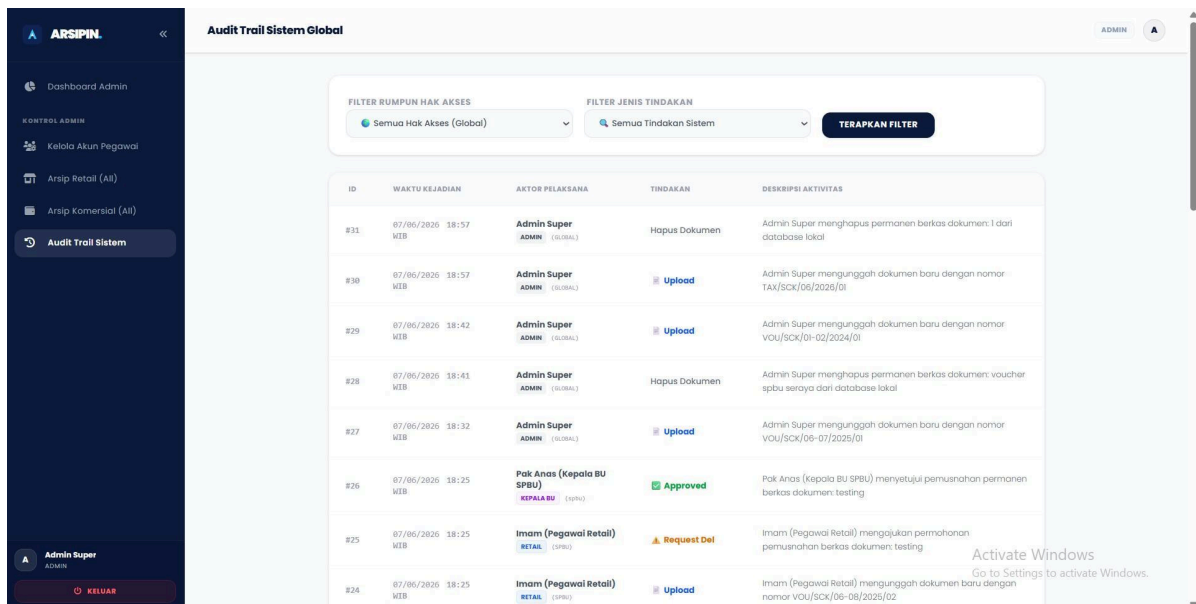
g. Implementasi Halaman Daftar Perusahaan-Komersial



Gambar 4.7 Halaman Daftar Perusahaan-Komersial

Gambar 4.7 menunjukkan tampilan halaman Perusahaan-Komersial, disana terdapat perusahaan perusahaan yang dinaungi oleh Bisnis Unit-Retail sebelumnya

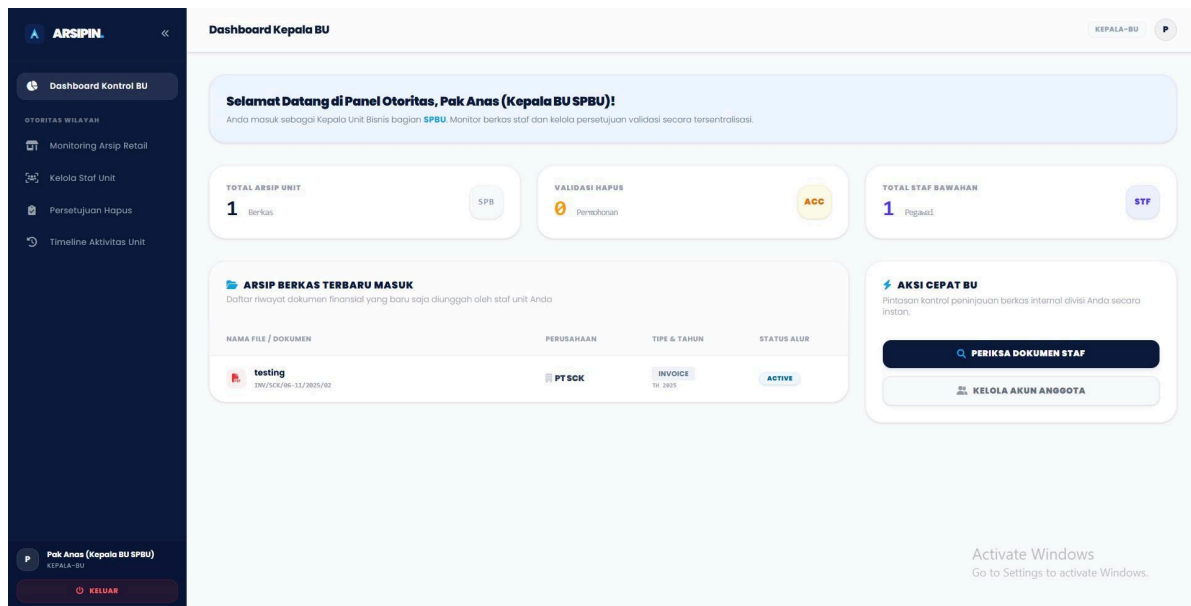
h. Implementasi Halaman Audit Trail Sistem



Gambar 4.7 Halaman Audit Trail Sistem

Gambar 4.7 menunjukkan tampilan halaman Audit Trail Sistem dari sisi seorang admin, disana admin dapat melihat riwayat aktivitas keseluruhan pengguna secara realtime, juga terdapat filtering ketika admin ingin memperkecil lingkup nya

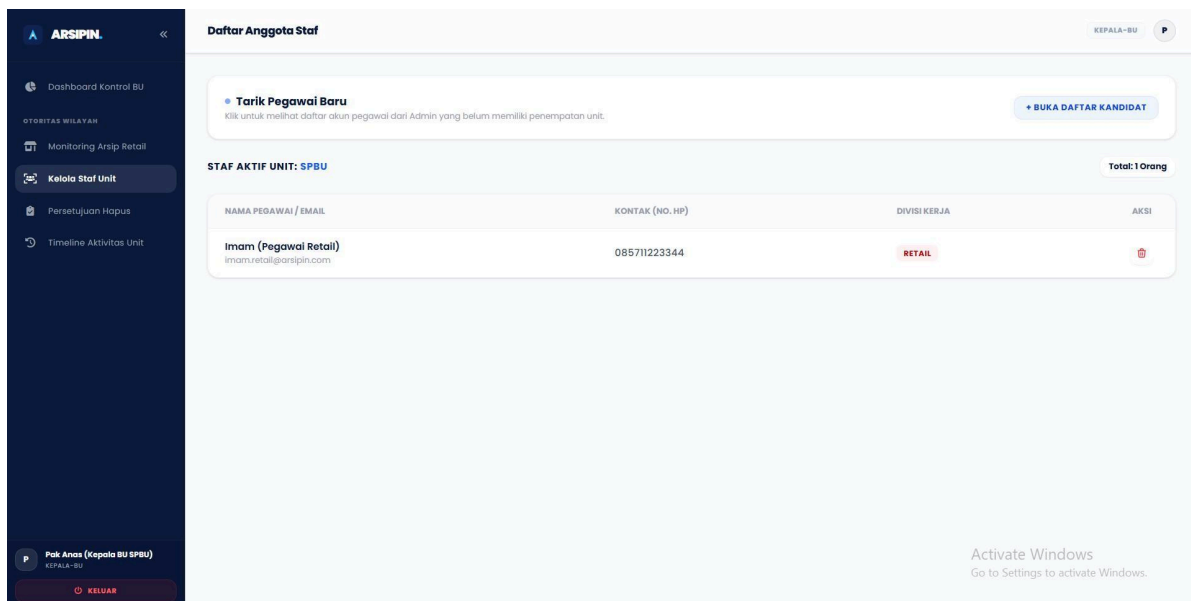
i. Implementasi Halaman Dashboard Kepala BU



Gambar 4.8 Halaman Dashboard Kepala BU

Gambar 4.8 menunjukkan tampilan halaman Dashboard dari sisi seorang Kepala BU, disana terdapat total arsip unit yang dinaungi, total dokumen yang membutuhkan validasi untuk dimusnahkan, dan total dari pegawai atau staf yang dinaungi oleh Kepala BU tersebut di bawah nya juga terdapat aktivitas dokumen yang baru baru masuk

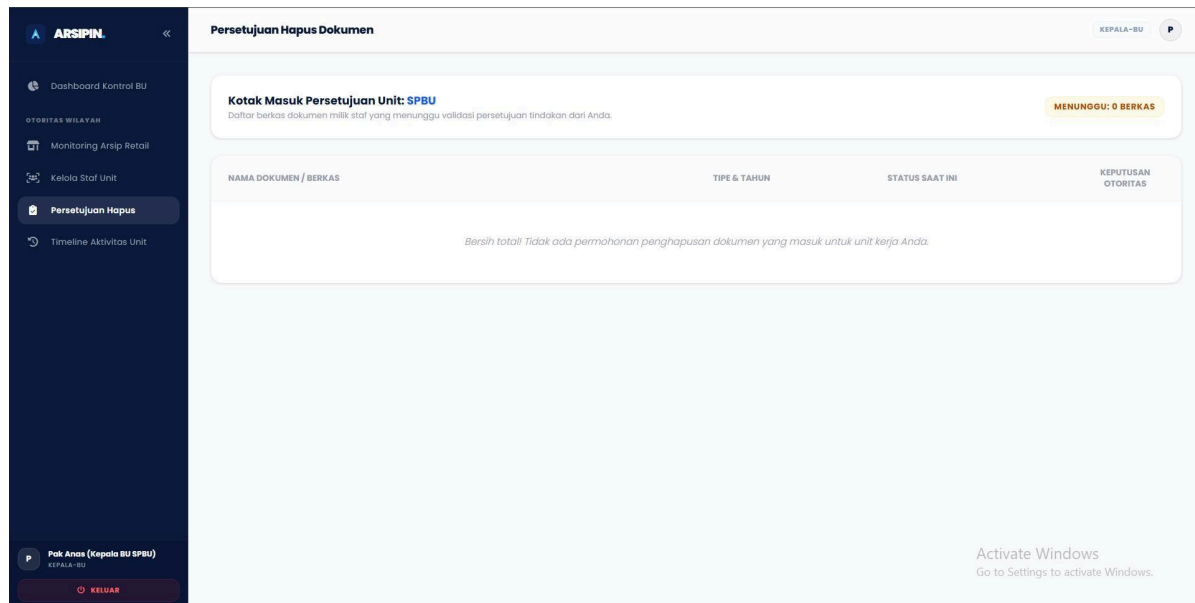
j. Implementasi Halaman Kelola Staf Unit



Gambar 4.9 Halaman Kelola Staf Unit

Gambar 4.9 menunjukkan tampilan halaman Kelola Staf Unit oleh kepala BU, disana terdapat total staff atau pegawai yang dinaungi, kemudian juga pada bagian terdapat tarik pegawai baru, yaitu jika admin membuatkan sebuah akun untuk pegawai sesuai dengan role dan BU, maka nanti kepala bu dapat menarik akun tersebut untuk menjadi staff yang dinaungi oleh Kepala BU tersebut

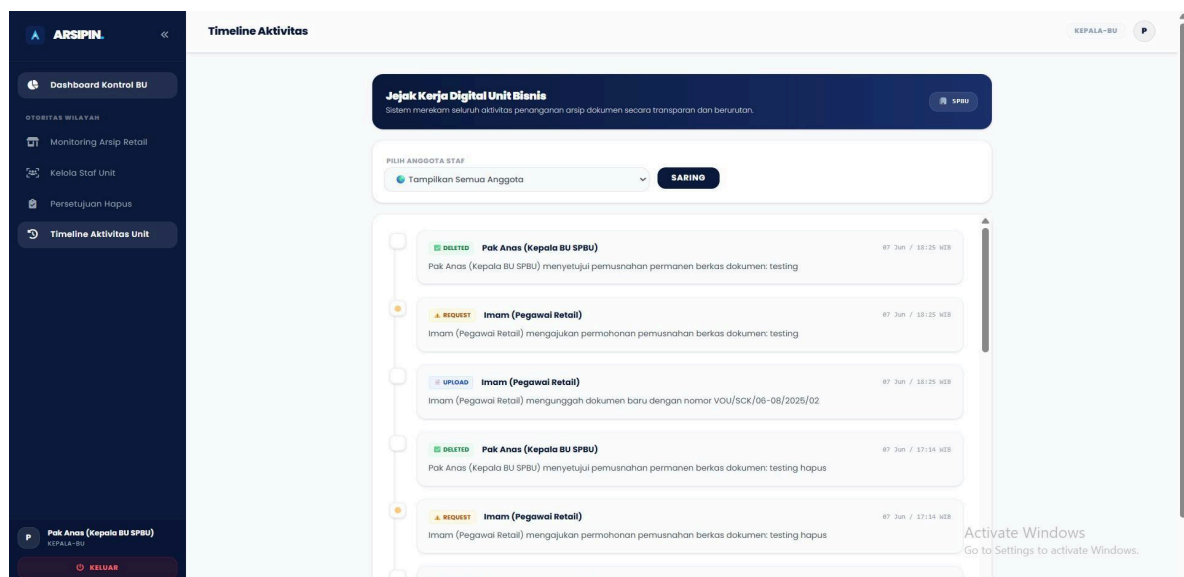
k. Implementasi Halaman Persetujuan Pemusnahan Dokumen



Gambar 4.10 Halaman Persetujuan Pemusnahan Dokumen

Gambar 4.10 menunjukkan tampilan halaman Persetujuan Pemusnahan Dokumen yang dikelola oleh Kepala BU, disana terdapat checklist dan silang jika ada pegawai yang mengajukan pemusnahan dan juga total dokumen yang membutuhkan validasi persetujuan pemusnahan dokumen

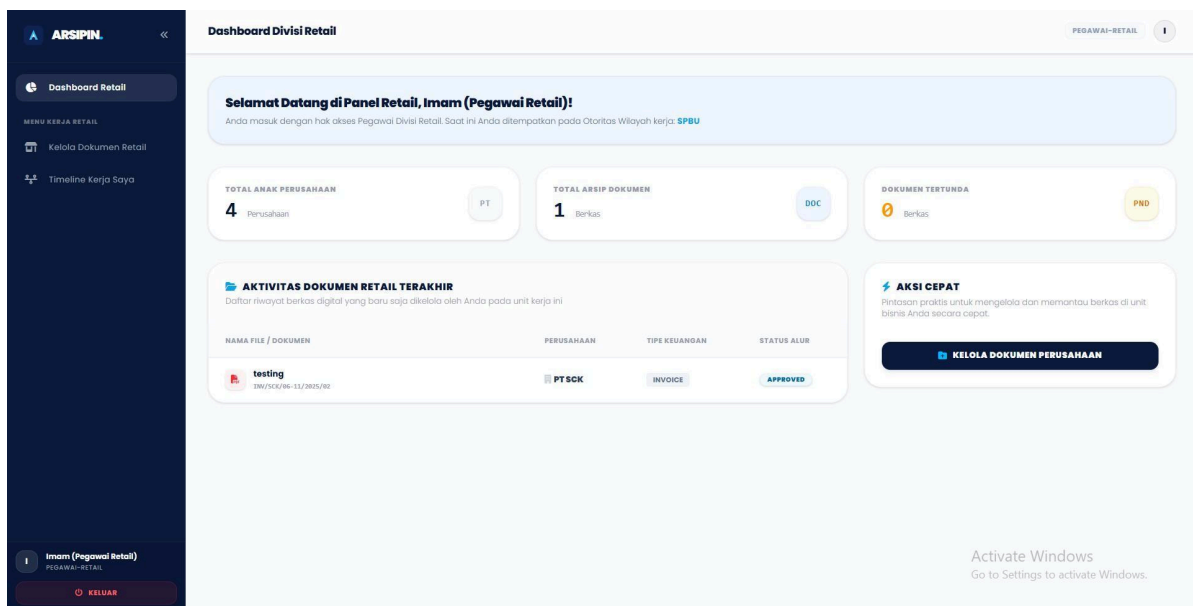
l. Implementasi Halaman Timeline Aktivitas Unit pada Kepala BU



Gambar 4.11 Halaman Timeline Aktivitas Unit pada Kepala BU

Gambar 4.11 menunjukkan tampilan log aktivitas kepala BU, disana terdapat aktivitas apa saja yang dilakukan oleh staf yang dinaungi oleh kepala BU tersebut, kemudian juga terdapat filtering jika kepala BU ingin memperkecil log aktivitas berdasarkan per anggota

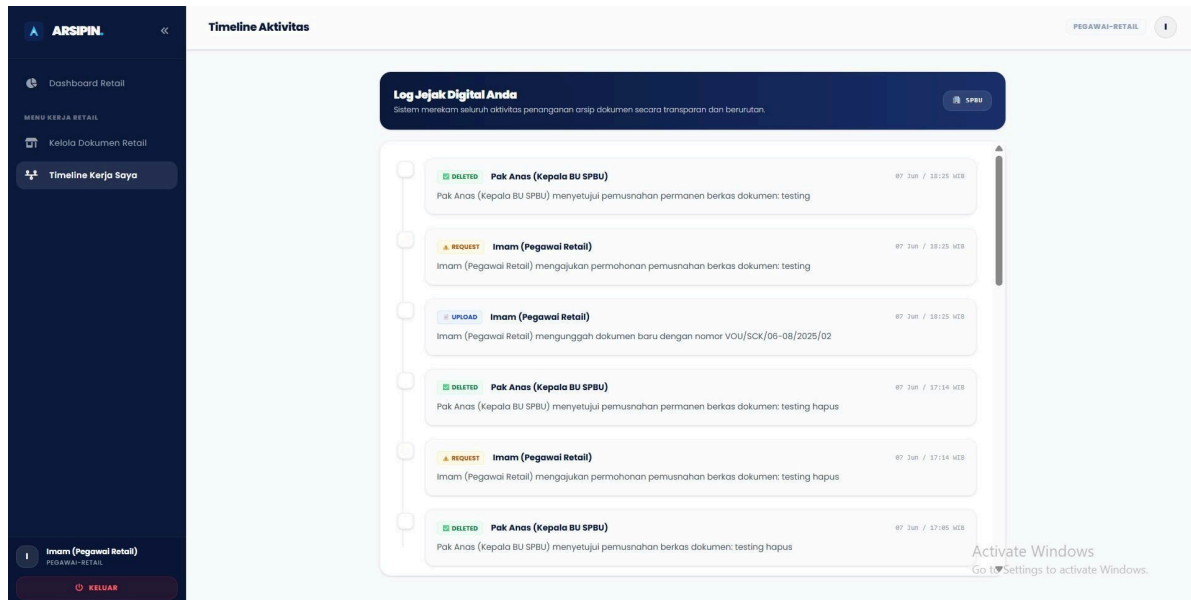
m. Implementasi Halaman Dashboard Pada Pegawai-Retail



Gambar 4.12 Halaman Dashboard Pada Pegawai-Retail

Gambar 4.12 menunjukkan tampilan Dashboard pada pegawai-retail, disana terdapat jumlah total perusahaan yang bergerak pada Bu pegawai tersebut, kemudian juga total arsip dokumen nya, serta jumlah dokumen yang membutuhkan validasi pemusnahan di bawah nya juga terdapat aktivitas dokumen yang baru baru masuk

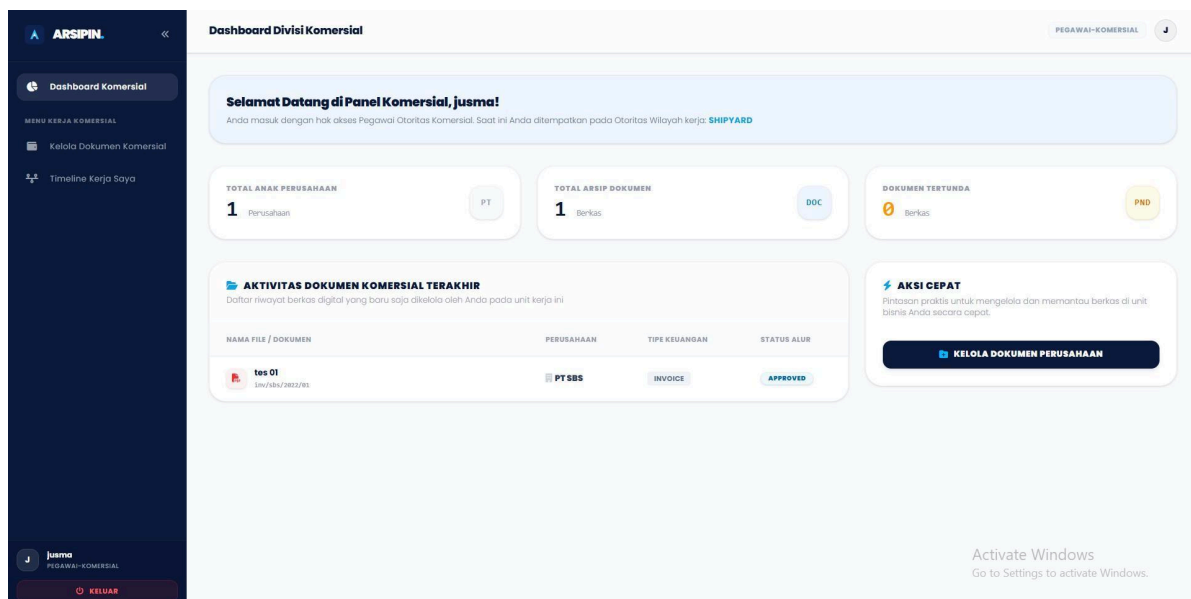
n. Implementasi Halaman Timeline Kerja Pegawai-Retail



Gambar 4.13 Halaman Timeline Kerja Pegawai-Retail

Gambar 4.13 menunjukkan tampilan Timeline Kerja dari seorang pegawai-retail, disana terdapat log aktivitas apa saja yang ia lakukan, kemudian juga dapat melihat apakah suatu dokumen yang membutuhkan validasi pemusnahan itu disetujui oleh kepala Bu atau ditolak

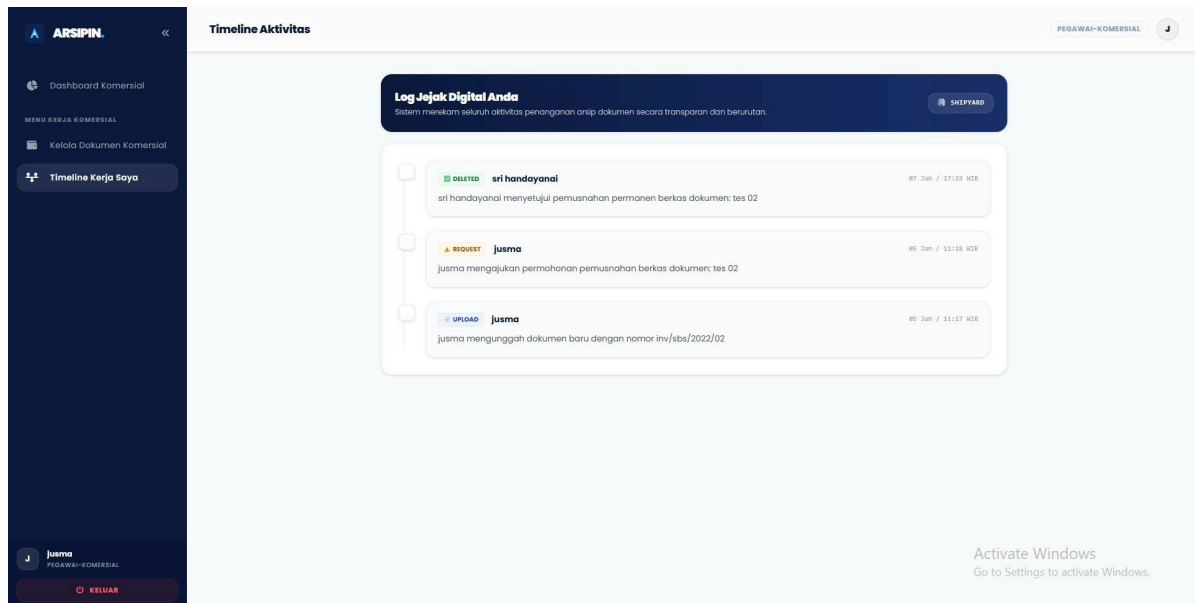
o. Implementasi Halaman Dashboard Pegawai-Komersial



Gambar 4.14 Halaman Dashboard Pada Pegawai-Komersial

Gambar 4.14 menunjukkan tampilan Dashboard pada pegawai-komersial, disana terdapat jumlah total perusahaan yang bergerak pada Bu pegawai tersebut, kemudian juga total arsip dokumen nya, serta jumlah dokumen yang membutuhkan validasi pemusnahan di bawah nya juga terdapat aktivitas dokumen yang baru baru masuk

p. Implementasi Halaman Timeline Kerja Pegawai-Komersial



Gambar 4.15 Halaman Timeline Kerja Pegawai-Komersial

Gambar 4.15 menunjukkan tampilan Timeline Kerja dari seorang pegawai-komersial, disana terdapat log aktivitas apa saja yang ia lakukan, kemudian juga dapat melihat apakah suatu dokumen yang membutuhkan validasi pemusnahan itu disetujui oleh kepala Bu atau ditolak

4.2 Pengujian Black Box Testing

Pengujian sistem dilakukan menggunakan metode Black Box Testing. Metode ini berfokus pada pengujian fungsi perangkat lunak berdasarkan masukan (input) dan keluaran (output) yang dihasilkan tanpa memperhatikan struktur kode program yang digunakan. Tujuan dari pengujian Black Box Testing adalah untuk memastikan bahwa setiap fitur sistem dapat berjalan sesuai dengan kebutuhan fungsional yang telah ditentukan serta mampu menghasilkan keluaran yang sesuai dengan harapan pengguna. Metode ini banyak digunakan dalam pengujian aplikasi karena efektif dalam mengidentifikasi kesalahan fungsi, validasi data, dan kesesuaian proses bisnis yang diimplementasikan pada sistem

Tabel 4.2 Hasil Pengujian Black Box Testing - Modul Manajemen Pengguna (FR-10)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-10-01	Tambah pengguna data lengkap	Isi nama, username, email unik, password, pilihan role (kepala-bu), dan wilayah unit kerja.	Akun baru tersimpan ke MySQL, password ter-hash otomatis, dan data tampil di tabel master pengguna.	Sesuai	Berhasil
BBT-10-02	Tambah pengguna email/username duplikat	Menginput email atau username yang sudah terdaftar sebelumnya di database.	Sistem menggagalkan proses simpan dan menampilkan pesan <i>alert error</i> validasi ganda.	Sesuai	Berhasil
BBT-10-03	Tambah pengguna field wajib kosong	Mengosongkan salah satu field wajib (misal: kolom password atau role).	Validasi wajib isi muncul pada layar dan sistem menolak eksekusi query simpan.	Sesuai	Berhasil
BBT-10-04	Perbaruan (<i>Update</i>) data akun pengguna	Mengubah data pengguna lama (misal: mengubah unit kerja dari LPG PSO menjadi LPG-PSO).	Jalur Penyelamat data <i>controller</i> mensterilkan spasi menjadi strip kapital, data MySQL ter-update.	Sesuai	Berhasil

BBT-10-05	Penghapusan (<i>Delete</i>) akun pengguna secara permanen	Mengklik tombol "Hapus" pada baris data baim dan menyetujui modal konfirmasi.	Baris data akun pengguna terhapus sepenuhnya dari tabel basis data MySQL.	Sesuai	Berhasil
-----------	---	---	---	--------	----------

Tabel 4.3 Hasil Pengujian Black Box Testing - Modul Dokumen & Integrasi AWS S3 (FR-02, FR-05, FR-11)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-02-01	Cloud Upload: Unggah berkas dokumen finansial valid	Mengisi metadata dokumen secara lengkap dan memilih file fisik berformat .pdf ukuran 1.2 MB.	File PDF berhasil terunggah ke bucket AWS S3 via API, metadata beserta s3_object_key ter-inject ke MySQL.	Sesuai	Berhasil
BBT-02-02	Failure Handling: Unggah file dengan format non-PDF	Memilih file dokumen dengan ekstensi palsu atau gambar berformat .jpg / .png.	JavaScript frontend mendeteksi ekstensi ilegal, membekukan tombol upload, dan memicu alert error merah.	Sesuai	Berhasil
BBT-02-03	Failure Handling: Unggah file dengan ukuran melebihi batas	Memilih file PDF dengan ukuran kapasitas data sebesar 7 MB (Batas maksimal sistem adalah 5 MB).	Sistem mendeteksi ukuran file <i>over-limit</i> , mengunci tombol submit, dan mereset file input menjadi kosong.	Sesuai	Berhasil
BBT-05-01	Cloud Download: Penarikan pratinjau berkas digital	Mengklik tombol "Lihat Berkas" pada baris dokumen finansial milik PT SCK yang berstatus aktif.	Laravel meminta token privat dinamis ke AWS S3, menghasilkan <i>Presigned URL</i> sementara, dan PDF tampil di tab baru.	Sesuai	Berhasil
BBT-11-01	Cloud Delete: Pemusnahan	Admin mengklik tombol "Hapus"	Baris data metadata di MySQL terhapus secara mutlak, dan objek berkas	Sesuai	Berhasil

	berkas mutlak oleh Admin	permanen pada baris berkas di tabel repositori global global.	fisik di bucket AWS S3 terhapus via API.		
--	--------------------------	---	--	--	--

Tabel 4.4 Hasil Pengujian Black Box Testing - Modul Keamanan Otorisasi & Pencegahan Broken Access Control (FR-03, FR-06, FR-09)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-03-01	Role Access: Isolasi menu kerja berbasis unit	User dengan akun Pegawai Retail unit SPBU melakukan login ke dalam sistem.	Sidebar dan URL otomatis dikunci ke /retail/spbu. Dokumen milik unit lain (LPG-PSO / Shipyard) tersembunyi total.	Sesuai	Berhasil
BBT-03-02	Broken Access Control: Bypass URL manual oleh peretas	Pegawai Komersial secara iseng mengetik manual URL kluster retail /retail/spbu pada address bar browser.	Middleware mendeteksi ketidakcocokan hak akses role/unit, memblokir request, dan melempar halaman Error 403 Forbidden.	Sesuai	Berhasil
BBT-06-01	Audit Log & Workflow: Pengajuan pemusnahan berkas harian	Staf mengklik tombol sampah merah "Ajukan Hapus Berkas" pada baris dokumen aktif.	Status dokumen di MySQL berubah menjadi pending_hapus, berkas gaib dari tabel aktif staf, dan log audit trail ter-record.	Sesuai	Berhasil
BBT-09-01	Role Access Isolation: Proteksi hak hapus sepihak Kepala BU	Kepala BU membuka menu pemantauan berkas aktif unit SPBU.	Sistem mendeteksi role Kepala BU dan otomatis menyembunyikan tombol "Edit" dan tombol sampah merah (Tampilan <i>Read Only</i>).	Sesuai	Berhasil
BBT-09-02	Otorisasi Approval: Validasi berjenjang oleh atasan	Kepala BU masuk ke menu <i>Persetujuan Hapus</i> dan mengklik tombol centang hijau (Setuju Hapus).	Perintah destroy() dieksekusi, metadata MySQL dilenyapkan, dan file fisik di AWS S3 terhapus permanen disertai log audit.	Sesuai	Berhasil

Tabel 4.5 Hasil Pengujian Black Box Testing - Modul Autentikasi Sesi Pengguna (FR-01 & FR-12)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-01-01	Login pengguna dengan kredensial valid	Memasukkan username dan password yang benar, lalu mengklik tombol "Masuk ke Sistem".	Sistem memverifikasi data akun di MySQL, membuat session, dan melakukan <i>redirect</i> ke dashboard sesuai role.	Sesuai	Berhasil
BBT-01-02	Login pengguna dengan kredensial salah	Memasukkan kombinasi username atau password yang asal-asalan/belum terdaftar.	Sistem menolak akses masuk dan menampilkan pesan eror merah "username atau Password Salah" di atas form.	Sesuai	Berhasil
BBT-01-02	Pengakhiran sesi pengguna (<i>Logout</i>)	Mengklik tombol merah "KELUAR" pada bagian bawah menu navigasi sidebar.	Sesi token pengguna dihancurkan dari memori, rute sistem dikunci kembali, dan user diarahkan ke halaman login utama.	Sesuai	Berhasil

Tabel 4.6 Hasil Pengujian Black Box Testing - Modul Manipulasi Metadata & Pencarian Berkas (FR-04 & FR-07)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-04-01	Mengedit informasi metadata dokumen	Mengklik ikon pensil (Edit), mengubah data yang keliru (misal: Jilid Buku), lalu klik "Simpan Perubahan".	PHP mengeksekusi fungsi <i>updateDokumen</i> di MySQL dan sistem menampilkan notifikasi sukses hijau secara real-time.	Sesuai	Berhasil
BBT-04-02	Mengedit metadata memicu duplikasi nomor	Mengubah isian metadata sehingga hasil rumusan nomor dokumen	Sistem membatalkan proses pembaruan data dan melempar pesan validasi "The no dokumen has already been taken".	Sesuai	Berhasil

	dokumen	kembar dengan berkas lain di database.			
BBT-07-01	Pencarian berkas berdasarkan nomor dokumen valid	Mengetik potongan string nomor dokumen resmi yang ada di database pada kolom pencarian, lalu tekan Enter.	Sistem menjalankan query LIKE ke MySQL dan menyajikan baris berkas yang dicari secara instan pada tabel.	Sesuai	Berhasil
BBT-07-02	Pencarian berkas dengan kata kunci tidak terdaftar	Mengetik karakter acak atau nomor dokumen palsu pada kolom input pencarian, lalu tekan Enter.	Tabel repositori dokumen mendadak kosong dan sistem menampilkan teks pemberitahuan data tidak ditemukan.	Sesuai	Berhasil

Tabel 4.7 Hasil Pengujian Black Box Testing - Modul Struktur Organisasi & Kelola Staf Unit (FR-08)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-08-01	Kepala BU memantau daftar staf aktif unit	Kepala BU masuk ke dalam menu navigasi khusus "Kelola Staf Unit".	Sistem melakukan query terfilter dan menyajikan daftar nama beserta kontak seluruh staf yang berada di bawah wewenang unitnya.	Sesuai	Berhasil
BBT-08-02	Kepala BU menarik akun pegawai baru ke dalam unit	Mengklik tombol "+ Buka Daftar Kandidat" dan menyetujui salah satu pengajuan akun pegawai berstatus pending.	Status keaktifan akun pegawai berubah menjadi approved di database dan resmi bergabung dalam rumpun unit bisnis terkait.	Sesuai	Berhasil

Tabel 4.8 Hasil Pengujian Black Box Testing - Kebutuhan Non-Fungsional (NFR-01 s.d NFR-03)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-NFR-01a	Keamanan Data: Enkripsi kata sandi pengguna di database	Membuat akun pegawai baru dan menginput kata sandi pada kolom password.	Sistem secara otomatis mengamankan kata sandi menggunakan algoritma <i>hashing</i> bawaan Laravel sebelum disimpan ke database MySQL.	Sesuai	Berhasil
BBT-NFR-01b	Keamanan Data: Enkripsi berkas fisik pada server cloud	Staf mengunggah file dokumen finansial berformat PDF ke dalam sistem ARSIPIN.	API AWS S3 merespons dengan mengaktifkan <i>Server-Side Encryption</i> (SSE-S3) menggunakan algoritma AES-256 sebelum file ditulis ke disk <i>cloud</i> .	Sesuai	Berhasil
BBT-NFR-02a	Usability & Responsivitas: Adaptasi visual antarmuka pengguna	Mengakses web ARSIPIN menggunakan berbagai perangkat (Desktop, Tablet, dan Smartphone).	Antarmuka pengguna yang dibangun dengan Tailwind CSS v4 merespons secara dinamis, seimbang, dan menyesuaikan dengan ukuran koordinat layar kerja.	Sesuai	Berhasil
BBT-NFR-02b	Usability & Navigasi: Efisiensi ruang menu navigasi	Mengklik tombol <i>toggle</i> pada komponen Mini-Sidebar Collapse di layar desktop.	Menu navigasi mengecil (<i>collapse</i>) secara responsif dan smooth, memberikan ruang layar yang lebih luas tanpa merusak rute rujukan utama.	Sesuai	Berhasil
BBT-NFR-03a	Maintainability: Pengujian struktur kode <i>backend</i>	Menjalankan perintah eksekusi rute (<i>routing</i>) harian dan pemanggilan <i>Controller</i> .	Kode sumber aplikasi yang terstruktur mengikuti standar arsitektur Model-View-Controller (MVC) Laravel berjalan lancar dan mudah dipelihara.	Sesuai	Berhasil
BBT-NFR-03b	Maintainability: Sinkronisasi integrasi kode repositori	Melakukan pembaruan atau perbaikan <i>bug</i> kecil pada kode sumber aplikasi.	Kode sumber yang terintegrasi dengan Git di repositori GitHub dapat dikelola versinya dengan baik demi menjaga skalabilitas sistem secara berkelanjutan.	Sesuai	Berhasil

Tabel 4.9 Hasil Pengujian Black Box Testing - Modul Audit Trail & Log Aktivitas (FR-12)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-12-01	Penarikan Riwayat Log Aktivitas Pengguna	Mengeklik menu "Timeline Aktivitas" / "Audit Trail Sistem" pada sidebar navigasi.	Sistem mengeksekusi query ke tabel log_aktivitas MySQL dan menyajikan lini masa riwayat tindakan secara transparan & kronologis sesuai role.	Sesuai	Berhasil
BBT-12-02	Pemfilteran Riwayat Aktivitas (Filtering Audit Trail)	Memilih opsi nama staf atau jenis tindakan pada dropdown filter, lalu mengeklik tombol "Terapkan Filter".	Sistem memperbarui tampilan timeline dan hanya menampilkan baris rekam jejak yang sesuai dengan kriteria filter yang dipilih.	Sesuai	Berhasil
BBT-12-03	Pencatatan Otomatis Sistem (Auto-Logging Trigger)	Mengunggah, mengedit metadata, atau menyetujui pemusnahan dokumen di dalam aplikasi.	Fungsi catatLog() ter-memicu secara otomatis dan menyuntikkan baris data log baru ke MySQL lengkap dengan username, role, aksi, dan timestamp.	Sesuai	Berhasil

Tabel 4.10 Hasil Pengujian Black Box Testing - Modul Kelola Jenis Dokumen Keuangan (FR-13)

No	Skenario Pengujian	Input	Expected Output	Actual Output	Status
BBT-14-01	Admin menambah kategori baru	Mengeklik menu "Timeline Aktivitas" / "Audit Trail Sistem" pada sidebar navigasi.	Data kategori tersimpan dan otomatis muncul di semua dropdown jenis dokumen pada form upload.	Sesuai	Berhasil
BBT-14-02	Admin menghapus kategori)	Mengeklik tombol hapus pada kategori yang tidak	Kategori hilang dari daftar pilihan sistem tanpa merusak integritas dokumen yang sudah ada.	Sesuai	Berhasil

		digunakan			
--	--	-----------	--	--	--

4.3 Pengujian System Usability Scale (SUS)

System Usability Scale (SUS) digunakan untuk mengukur tingkat kebergunaan (usability) aplikasi SIPB dari sudut pandang pengguna. Pengujian melibatkan 20 responden yang terdiri atas Cso dan Kkc dengan mengisi kuesioner berisi sepuluh pernyataan berskala 1 (sangat tidak setuju) hingga 5 (sangat setuju). Skor tiap responden dihitung menggunakan rumus standar SUS dengan rentang akhir 0 sampai 100, kemudian dirata-rata untuk menentukan tingkat kebergunaan sistem secara keseluruhan. Hasil perhitungan dan interpretasinya diuraikan pada bagian berikut.

Evaluasi tingkat kebergunaan (usability) antarmuka sistem ARSIPIN menggunakan metode standar System Usability Scale (SUS) yang dikembangkan oleh John Brooke pada tahun 1996 (Brooke, J., 1996. "SUS: a quick and dirty usability scale". Usability evaluation in industry, 189(194), pp.4-7). Kuesioner ini terdiri dari 10 pertanyaan baku (5 pertanyaan bernomor ganjil yang bersifat positif dan 5 pertanyaan bernomor genap yang bersifat negatif) untuk mengukur efektivitas, efisiensi, dan kepuasan subjektif pengguna secara menyeluruh.

Tabel 4.11 Pertanyaan Kuesioner

No	Pertanyaan
P1	Saya rasa saya akan sering menggunakan aplikasi ini.
P2	Saya merasa aplikasi ini terlalu rumit padahal bisa dibuat lebih sederhana.

P3	Saya rasa aplikasi ini mudah digunakan.
P4	Saya rasa saya membutuhkan bantuan orang teknis untuk dapat menggunakan aplikasi ini.
P5	Saya merasa berbagai fitur pada aplikasi ini berjalan dan terintegrasi dengan baik.
P6	Saya merasa ada terlalu banyak ketidaksesuaian (inkonsistensi) dalam aplikasi ini.
P7	Saya rasa kebanyakan orang akan dapat mempelajari aplikasi ini dengan cepat.

P8	Saya merasa aplikasi ini sangat membingungkan atau merepotkan untuk digunakan.
P9	Saya merasa sangat percaya diri saat menggunakan aplikasi ini.
P10	Saya perlu mempelajari banyak hal terlebih dahulu sebelum dapat menggunakan aplikasi ini dengan baik.

Jawaban dari seluruh responden kemudian diolah untuk memperoleh skor SUS setiap responden, lalu dirata-ratakan. Hasil rekapitulasi jawaban serta skor akhirnya disajikan pada tabel 4.10 dengan langkah-langkah pengolahan data kuesioner SUS dari 20 responden (Admin, Kepala BU, dan Pegawai) dilakukan melalui tahapan matematis berikut:

1. Penilaian Skor Item Ganjil (P1, P3, P5, P7, P9): Untuk setiap pertanyaan bernomor ganjil, skor kontribusi dikurangi 1 dari nilai skala yang dipilih responden ($\$Skor - 1\$$).
2. Penilaian Skor Item Genap (P2, P4, P6, P8, P10): Untuk setiap pertanyaan bernomor genap, skor kontribusi dihitung dengan mengurangi angka 5 dengan nilai skala yang dipilih responden ($\$5 - Skor\$$).
3. Akumulasi Skor Total: Jumlahkan seluruh nilai skor kontribusi dari kesepuluh pertanyaan untuk setiap responden.
4. Konversi ke Skala 100: Kalikan total jumlah skor responden dengan $\$2,5\$$ sehingga diperoleh skor akhir SUS dengan rentang nilai skala dari 0 hingga 100.
5. Perhitungan Rata-Rata (*Average SUS Score*): Seluruh skor akhir dari 20 responden dijumlahkan lalu dibagi 20 untuk mendapatkan nilai rata-rata keseluruhan sistem. Hasil rata-rata tersebut kemudian diinterpretasikan ke dalam tiga indikator mutu standar, yaitu: *Adjective Rating* (Peringkat Kualitatif), *Grade Scale* (Skala Huruf Mutu), dan *Acceptability Range* (Kelayakan Penerimaan Sistem).

Tabel 4.12 Rekapitulasi Hasil Pengujian SUS

Responden	Peran	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Skor SUS
R1	admin	3	2	5	2	5	1	5	1	5	1	87,5
R2	pegawai	2	2	4	3	5	1	4	1	4	3	65,0
R3	pegawai	5	2	5	1	5	1	4	1	5	1	85,0

R4	pegawai	5	1	5	1	5	1	5	1	5	1	100,0
R5	pegawai	5	2	4	3	4	2	4	1	4	4	62,5
R6	pegawai	3	3	3	4	4	2	2	4	3	5	45,0
R7	pegawai	5	1	4	4	4	1	4	1	5	4	72,5
R8	admin	5	1	5	1	5	1	4	2	3	3	85,0
R9	pegawai	5	1	5	1	5	1	5	1	5	4	85,0
R10	K.BU	5	1	5	1	5	1	5	1	5	1	100,0
R11	admin	5	1	5	2	5	1	5	1	5	2	90,0
R12	admin	5	1	5	1	5	1	5	1	5	1	100,0

R13	admin	3	3	4	4	4	2	4	3	4	5	45,0
R14	admin	4	2	5	3	4	3	5	2	5	3	67,5
R15	K.BU	5	2	4	1	5	2	5	1	4	2	80,0
R16	admin	5	1	5	1	4	2	4	3	5	2	85,0
R17	K.BU	5	2	5	2	4	3	5	1	4	1	82,5
R18	pegawai	5	3	5	2	4	1	5	3	5	2	77,5
R19	pegawai	5	2	5	1	4	3	4	2	5	1	85,0
R20	K.BU	5	2	5	1	4	3	4	2	5	3	77,5
Rata Rata Skor SUS												79,00

Berdasarkan hasil perhitungan pada Tabel x.x, diperoleh skor rata-rata SUS sebesar 79,00. Skor ini berada di atas ambang batas rata-rata SUS sebesar 68, sehingga aplikasi E-Checksheets termasuk dalam kategori dapat diterima (acceptable). Skor terendah dari responden adalah 45,0 dan tertinggi adalah 100,0.

Skor rata-rata tersebut selanjutnya diinterpretasikan berdasarkan tiga aspek penilaian standar SUS, yaitu *adjective rating*, *grade usability*, dan *acceptability range*

Tabel 4.13 Interpretasi Hasil Pengujian SUS

Aspek Interpretasi	Hasil	Keterangan
--------------------	-------	------------

Adjective Rating	Excellent	Menunjukkan pengalaman pengguna yang baik dan positif
Grade Usability	B	Mengindikasikan tingkat kebergunaan yang tinggi.
Acceptability Range	Acceptable	Aplikasi dinilai layak dan diterima oleh pengguna.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan keseluruhan rangkaian pengujian fungsionalitas yang telah dilakukan pada Sistem Manajemen Dokumen Finansial (ARSIPIN), maka dapat ditarik kesimpulan secara terstruktur guna menjawab seluruh rumusan masalah proyek akhir sebagai berikut:

1. Sistem repositori digital ARSIPIN berbasis web berhasil dibangun menggunakan *framework* Laravel dan database MySQL, di mana sistem terbukti mampu mendigitalisasi proses administrasi konvensional secara terpusat. Kehadiran rumus otomatisasi penomoran dokumen resmi berbasis JavaScript terbukti mempercepat efisiensi alur pencarian dan pengarsipan berkas finansial (Invoice, Voucher, Pajak, dan Rekening Koran Bank) dari yang semula membutuhkan waktu jam menjadi hitungan detik.
2. Sistem penyimpanan digital yang aman dan andal telah berhasil diwujudkan melalui integrasi protokol API *Cloud Storage* Amazon Web Services (AWS) S3. Hasil pengujian *Black Box Testing* membuktikan seluruh fungsi vital manajemen berkas (*cloud upload*, download via *presigned URL privat*, dan *permanent delete*) berjalan sukses tanpa kendala kegagalan sistem (*failure handling*), sehingga berhasil meminimalkan risiko kerusakan logistik dan kehilangan berkas fisik akuntansi korporat.
3. Platform arsip terpusat ini sukses memfasilitasi hak akses yang terkontrol dari berbagai lokasi bagi *multi-user* berbasis peran (*Role-Based Access Control*). Pembatasan rute lewat *Middleware* terbukti secara fungsional mampu membatasi wewenang tindakan secara berjenjang (Staf untuk unggah berkas, Kepala BU untuk monitoring wilayah dan validasi persetujuan hapus), serta terbukti sukses menangani celah kerentanan *Broken Access Control* dengan memblokir upaya manipulasi URL ilegal dan mengalihkannya langsung ke halaman pertahanan *Error 403 Forbidden* yang tercatat valid di jurnal *Master Audit Trail*.

5.2 Saran

Guna pengembangan, penyempurnaan, dan menjaga skalabilitas sistem ARSIPIN secara berkelanjutan pada masa yang akan datang, beberapa poin saran konstruktif yang dapat direkomendasikan adalah sebagai berikut:

1. Melakukan optimasi infrastruktur skala penuh pada sisi *backend* guna mengintegrasikan kalkulasi kapasitas ruang penyimpanan *cloud storage* AWS S3 secara dinamis ke visual grafik, serta mengganti seluruh komponen data tiruan grafik visual dengan penarikan data riil yang sinkron dari *database* utama MySQL.
2. Mengembangkan sistem lebih lanjut dengan menyuntikkan teknologi cerdas berupa *Optical Character Recognition* (OCR) untuk memfasilitasi ekstraksi nomor dokumen resmi secara otomatis langsung dari dalam pindaian berkas PDF yang diunggah staf, yang dikombinasikan dengan penguatan protokol enkripsi kriptografi pada file dokumen guna memperkuat benteng proteksi data finansial korporat dari risiko manipulasi pihak luar.
3. Melakukan pengujian beban (*load testing*) dan *stress testing* dengan volume lalu lintas data yang jauh lebih masif di kemudian hari, guna memastikan bahwa kinerja mesin pencarian berkas dan responsivitas fitur *scroll-bar* internal pada tabel riwayat dokumen tetap berjalan dengan sangat stabil, ringan, dan *smooth* ketika menangani lonjakan ribuan data jurnal keuangan baru secara simultan.

DAFTAR PUSTAKA

- Alexandes, D. L., Aditio, R., & Jumaryadi, Y. (2022). Implementasi Metode Agile dalam Pengembangan Sistem E-document. *Journal of Information System Research (JOSH)*, 4(1), 318–329.
- Amalia, P. S., Haerani, E., Romli, R., & Roshinta, T. A. (2025). Optimizing Data Management in Web Applications through Google Drive API Integration and Synchronization. *International Journal of Informatics and Computing*, 1(1), 26–33.
- Anisah, Wahyuningsih, D., Helmud, E., Suwanda, T., Romadiana, P., & Irawan, D. (2021). Rancang Bangun Sistem Informasi Manajemen Arsip Digital. *Jurnal SISFOKOM (Sistem Informasi dan Komputer)*, 10(3), 419–425.
- Armansyah, J. (2025). RANCANG BANGUN WEBISTE ARSIP DIGITAL MENGGUNAKAN FRAMEWORK LARAVEL. Laporan Kerja Praktek, Politeknik Negeri Bengkalis.
- Yazid, A., Khairunnisa, Suprihartini, Y., Hariyadi, A., & Shofwani, S. A. (2025). Pelatihan Digitalisasi Manajemen Arsip dan Administrasi bagi Lembaga Pendidikan Nonformal Menggunakan Aplikasi Cloud Storage. *JIPITI: Jurnal Pengabdian kepada Masyarakat*, 2(2), 106–110.
- Handayani, H., Faizah, K. U., Ayulya, A. M., Rozan, M. F., & Wulan, D. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(1), 29–40.
- Lukitasari, D., & Utomo, F. S. (2024). Sistem Informasi Pengolahan Data Nilai Siswa Menggunakan AWS Berbasis WEB. *Jurnal Informatika dan Teknologi Interaktif (JIITE)*, 1(3), 206–214.
- Baron, J., & Kotecha, S. (2013). *Storage Options in the AWS Cloud*. Amazon Web Services Whitepaper.
- Kurniawan, A. C., Tibyani, & Amalia, F. (2020). Implementasi Teknologi Cloud Computing untuk E-Learning berbasis Website dengan Framework Laravel (Studi Kasus: MAN 9 Jombang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(11), 3833–3844.
- Lutfiani, N., Harahap, E. P., Aini, Q., Rizki Ahmad, A. D. A., & Rahardja, U. (2020). Inovasi Manajemen Proyek I-Learning Menggunakan Metode Agile Scrumban. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 5(1), 96–101.
- Makalalag, R. R., & Maramis, G. D. P. (2025). Implementasi Sistem Manajemen Dokumen Cloud di Dinas Kebudayaan dan Pariwisata Kabupaten Minahasa. *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, 9(3), 1101–1109.
- Pertiwi, T. A., Luchia, N. T., Sinta, P., Aprinastya, R., Dahlia, A., & Fachrezi, I. R. (2023). Perancangan Dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(1), 53–66.
- Putra, C. H. A., Soni, A. A., Sudandi, I. G. A., Candra, I. K. A. S., & Mahesa, I. G. M. W. (2025). E-Arsip UKM Paskamras Berbasis Web Menggunakan Framework Laravel. *INNOVATIVE: Journal Of Social Science Research*, 5(2), 406–415.

DAFTAR LAMPIRAN

Daftar lampiran disesuaikan dengan ruang lingkup Proyek Akhir, yaitu apakah berfokus pada pengembangan perangkat lunak, implementasi jaringan, atau *artificial intelligence*. Pada template ini, contoh yang digunakan mengacu pada ruang lingkup pengembangan perangkat lunak. Untuk implementasi jaringan, lampiran memuat hasil pengujian serta tautan *repository* (jika tersedia). Sementara itu, untuk *artificial intelligence*, lampiran meliputi hasil pengumpulan data (dapat berupa tautan), hasil *preprocessing* data (dapat berupa tautan), serta tautan produk, yaitu *repository* GitHub.

Lampiran A. Dokumen Proses Pengumpulan Requirement

Instrumen, dokumentasi, dll

Lampiran B. Link Produk

https://github.com/baim110334/arsipin_aws_s3.git

Lampiran C. Dokumen Pengujian UAT

https://drive.google.com/drive/folders/1igZlHsXMvQLnUAMRVrT24y6uugP3uzFu?usp=drive_link

Lampiran D. Dokumen Pengujian SUS

https://drive.google.com/drive/folders/1igZlHsXMvQLnUAMRVrT24y6uugP3uzFu?usp=drive_link