**PROPOSAL TUGAS AKHIR**



Diajukan untuk Memenuhi Sebagian Persyaratan

Guna Memperoleh Gelar Sarjana

Disusun Oleh:

**MUHAMMAD WAHYU ILHAM**

**NIM: 202131097**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TELEMATIKA ENERGI**

**INSTITUT TEKNOLOGI PLN**

**JAKARTA 2025**

# **LEMBAR PENGESAHAN**

# **ABSRAK**

MUHAMMAD WAHYU ILHAM

Dibimbing oleh Bapak Abdurrasyid, S.Kom., MMSI dan

Bapak Rakhmadi Irfansyah Putra, S.Kom., MMSI

Kata kunci:

# **ABSTRACT**

MUHAMMAD WAHYU ILHAM

Supervised by Mr. Abdurrasyid, S.Kom., MMSI and   
Mr. Rakhmadi Irfansyah Putra, S.Kom., MMSI

Keywords:

# **DAFTAR ISI**

[**LEMBAR PENGESAHAN** i](#_Toc204049607)

[**ABSRAK** ii](#_Toc204049608)

[**ABSTRACT** i](#_Toc204049609)

[**DAFTAR ISI** i](#_Toc204049610)

[**DAFTAR TABEL** iii](#_Toc204049611)

[**DAFTAR GAMBAR** iv](#_Toc204049612)

[**DAFTAR LAMPIRAN** v](#_Toc204049613)

[**BAB I PENDAHULUAN** 1](#_Toc204049614)

[**1.1 Latar belakang** 1](#_Toc204049615)

[**1.2 Rumusan Masalah** 2](#_Toc204049616)

[**1.3 Tujuan Penelitian** 2](#_Toc204049617)

[**1.4 Manfaat Penelitian** 2](#_Toc204049618)

[**1.5 Ruang Lingkup Masalah** 2](#_Toc204049619)

[**1.6 Sistematika Penulisan** 3](#_Toc204049620)

[**BAB II TINJAUAN PUSTAKA** 4](#_Toc204049621)

[**2.1 Kajian Relevan** 4](#_Toc204049622)

[**2.2 Landasan Teori** 12](#_Toc204049623)

[**2.2.1 Sistem Informasi** 13](#_Toc204049624)

[**2.2.2 Metode *Research and Development* (R&D)** 13](#_Toc204049625)

[**2.2.3 *Web - Application*** 15](#_Toc204049626)

[**2.2.4 *CodeIgniter*** 15](#_Toc204049627)

[**2.2.5 PHP (*Hypertext Preprocessor*)** 16](#_Toc204049628)

[**2.2.6 MySQL** 16](#_Toc204049629)

[**2.2.7 XAMPP** 17](#_Toc204049630)

[**2.2.8 *Unified Modelling Language (UML)*** 18](#_Toc204049631)

[**2.2.9** ***Black-Box Testing*** 22](#_Toc204049632)

[**2.3 Kerangka Pemikiran** 23](#_Toc204049633)

[**BAB III METODE PENELITIAN** 1](#_Toc204049634)

[**3.1 Waktu dan Tempat Penelitian** 1](#_Toc204049635)

[**3.2 Desain Penelitian** 1](#_Toc204049636)

[**3.3 Metode Pengumpulan Data** 16](#_Toc204049637)

[**3.4 Metode Analisis Data** 17](#_Toc204049638)

[**DAFTAR PUSTAKA** 18](#_Toc204049639)

# **DAFTAR TABEL**

[Tabel 2. 1 Jurnal Relevan 15](#_Toc195891010)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Kerangka Pemikiran 19](#_Toc195973835)

[Gambar 3. 1 Flowchart Desain Penelitian 22](#_Toc195973841)

# **DAFTAR LAMPIRAN**

[Lampiran 1 Daftar Riwayat Hidup 28](#_Toc195979997)

[Lampiran 2 Lembar Evaluasi 29](#_Toc195979998)

[Lampiran 3 Indeks Kemiripan ( Turnitin) 30](#_Toc195979999)

# **BAB I PENDAHULUAN**

## **1.1 Latar belakang**

Proses administrasi dalam Kementerian Agama RI menghadapi beberapa tantangan, antara lain kesalahan input data serta ketidakteraturan dalam klasifikasi tingkat urgensi, yang menyebabkan permohonan yang seharusnya menjadi prioritas justru tertunda. Sebagai instansi pemerintah yang memiliki cakupan luas dan berperan langsung dalam pembinaan keagamaan, Kementerian Agama RI perlu mengadopsi teknologi informasi dalam mendukung efektivitas dan transparansi layanan publik. Saat ini, Kementerian Agama masih menggunakan framework CodeIgniter versi 1 untuk sistem administrasinya, yang membuat website terasa berat dan lambat. Selain itu, belum adanya sistem prioritas yang terintegrasi menyebabkan pengelolaan urgensi permohonan masih dilakukan secara manual oleh para staf, yang berujung pada ketidakefisienan.

Berdasarkan hasil wawancara yang saya lakukan, terdapat sekitar 100 permohonan per bulan, dengan volume permohonan yang meningkat pesat pada awal tahun ajaran baru dan momentum keagamaan. Tanpa adanya sistem yang mampu mengklasifikasikan urgensi pengajuan secara otomatis, pengajuan yang lebih mendesak bisa tertunda, menghambat kelancaran proses administrasi di kementerian ini.

Pentingnya mengalihkan sistem administrasi dari metode manual ke sistem berbasis TI menjadi semakin jelas dalam konteks peningkatan efisiensi dan mengurangi potensi kesalahan manusia(Ugochukwu Francis Ikwuanusi et al., 2024), sehingga diperlukan sistem berbasis IT untuk meningkatkan akurasi dan efisiensi kerja. Oleh karena itu, dibutuhkan solusi digital yang lebih efisien untuk meningkatkan akurasi dan kecepatan dalam memproses permohonan yang masuk.

Penelitian ini bertujuan untuk mengembangkan aplikasi pengajuan berbasis web yang dapat mengotomatisasi pengklasifikasian urgensi permohonan dan memprioritaskan pemrosesan sesuai dengan tingkat urgensinya. Aplikasi ini diharapkan dapat memberikan kontribusi nyata dalam meningkatkan kualitas layanan administratif di lingkungan Kementerian Agama RI, sekaligus menjadi referensi bagi instansi pemerintah lain yang menghadapi tantangan serupa.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah dikemukakan di atas, dapat diperoleh rumusan permasalahan sebagai berikut :

1. Bagaimana merancang sistem pengajuan berbasis web yang dapat mengklasifikasikan tingkat urgensi pengajuan secara otomatis?
2. Bagaimana mengembangkan sistem yang dapat memproses pengajuan dan mengatur prioritas secara efisien?

## **1.3 Tujuan Penelitian**

Adapun tujuan yang diharapkan dalam penelitian yang akan dilaksanakan adalah :

1. Merancang sistem pengajuan berbasis web yang dapat mengklasifikasikan tingkat urgensi pengajuan secara otomatis dan memastikan pengajuan masuk ke dalam kategori yang tepat.
2. Mengembangkan sistem yang dapat memproses pengajuan dan mengatur prioritas pemrosesan secara efisien untuk meningkatkan kecepatan dan akurasi dalam proses administrasi.

## **1.4 Manfaat Penelitian**

Manfaat yang diharapkan saat penelitian dilaksanakan adalah :

1. Membangun sistem pengajuan berbasis web yang dapat mengklasifikasikan tingkat urgensi pengajuan secara otomatis.
2. Mengembangkan sistem yang dapat memproses pengajuan dan mengatur prioritas pemrosesan secara efisien.

## **1.5 Ruang Lingkup Masalah**

Ruang lingkup penelitian yang akan dilaksanakan adalah sebagai berikut :

1. Penelitian difokuskan pada pengembangan sistem pengajuan berbasis web untuk permohonan dan gangguan, yang merupakan pengembangan dari aplikasi lama, SIGAP (Sistem Informasi Layanan Gangguan dan Permohonan), menggunakan framework CodeIgniter 4 (CI4).
2. Fokus penelitian hanya pada pengajuan permohonan dan gangguan yang masuk, tanpa melibatkan jenis pengajuan lainnya.
3. Penelitian ini menggunakan metode *Research and Development* (R&D), yang berfokus pada pengembangan prototipe sistem secara cepat dan iteratif untuk menyesuaikan dengan kebutuhan pengguna.
4. Sistem ini akan mengklasifikasikan tingkat urgensi pengajuan secara otomatis dan mengatur prioritas pemrosesan pengajuan dengan efisien, dengan memastikan bahwa pengajuan masuk ke dalam kategori yang tepat sesuai dengan judul pengajuan yang telah ditentukan.

## **1.6 Sistematika Penulisan**

Secara sistematis, proposal skripsi ini diuraikan menjadi 3 bab yang disusun secara ringkas, padat, dan jelas. Adapun uraian dari penulisan skripsi ini adalah sebagai berikut:

**BAB 1 PENDAHULUAN**

Bab ini menerangkan tentang usulan penelitian yang meliputi latar belakang masalah, menentukan rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup masalah yang menjadi batasan dalam penelitian, serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini menguraikan penelitian terdahulu yang relevan serta teori-teori yang menjadi dasar pendukung dalam penyusunan penelitian.

**BAB III METODE PENELITIAN**

Bab ini menjelaskan langkah-langkah yang diambil dalam penelitian untuk menyelesaikan masalah dan mencapai tujuan yang ditetapkan. Pada bab ini juga dibahas mengenai lokasi dan waktu penelitian, desain penelitian, metode yang digunakan, teknik pengumpulan data, serta analisis data yang berkaitan dengan proses perhitungan algoritma yang diterapkan dalam penelitian.

# **BAB II TINJAUAN PUSTAKA**

Bab tinjauan pustaka ini akan membahas perbandingan berbagai penelitian sebelumnya yang memiliki topik dan metode serupa, serta menguraikan teori-teori yang berkaitan langsung dengan penelitian ini.

## **2.1 Kajian Relevan**

Penggunaan teknologi informasi dalam administrasi pemerintahan sangat penting untuk meningkatkan efisiensi dan akurasi pengolahan data permohonan. Kementerian Agama RI menghadapi tantangan dalam sistem administrasi yang masih manual, seperti kesalahan input data dan ketidakteraturan pengklasifikasian urgensi permohonan. Hal ini menyebabkan penundaan pemrosesan permohonan yang seharusnya menjadi prioritas. Oleh karena itu, dibutuhkan sistem berbasis TI yang dapat mengotomatisasi pengklasifikasian urgensi permohonan dan memprioritaskan pemrosesan sesuai tingkat urgensinya. Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis web yang dapat meningkatkan efisiensi, akurasi, dan kualitas layanan administratif di Kementerian Agama RI.

Tabel 2. 1 Studi Literatur

|  |  |
| --- | --- |
| No | 1 |
| Nama Peneliti | Kinanti Wijaya, Sutrisno, Nono Sebayang, Harvan Ardiansyah |
| Tahun | 2022 |
| Judul | *Development of Video Tutorials AutoCAD Learning in Software Applications and Building Interior Design Lessons With 4D Development Model* |
| Metode | *Research and Development (R&D),* 4D |
| Kesimpulan | Penelitian ini berhasil mengembangkan media pembelajaran berupa video tutorial AutoCAD untuk siswa SMK dengan pendekatan model 4D (Define, Design, Develop, Disseminate). Hasil validasi menunjukkan bahwa media sangat layak digunakan, dengan nilai rata-rata 4,19 dari skala 5. Media ini dinilai praktis oleh siswa dan dapat membantu meningkatkan pemahaman siswa terhadap materi praktikum. Namun, fokus utamanya hanya pada pengembangan media audiovisual tanpa mengembangkan sistem digital berbasis aplikasi atau layanan internal. |
|  |  |
| No | 2 |
| Nama Peneliti | Yose Indarta, Alfi Syahrin, Wendi Syafutra |
| Tahun | 2022 |
| Judul | *Web Application Development for Syar'i Online Shop Using 4D Model* |
| Metode | *Research and Development (R&D),* dengan model 4D dan Framework CodeIgniter 4 |
| Kesimpulan | Penelitian ini mengembangkan aplikasi toko online berbasis web menggunakan framework CodeIgniter 4 dengan model pengembangan 4D. Aplikasi yang dihasilkan memudahkan proses transaksi, pengelolaan stok, dan pelaporan penjualan. Sistem ini mengatasi masalah manualitas pencatatan yang sebelumnya dilakukan oleh toko dan memberikan solusi digital yang efisien. Namun, sistem tidak mencakup fitur klasifikasi tingkat urgensi atau alur layanan permohonan. |
|  |  |
| No | 3 |
| Nama Peneliti | M. Zikrul Ansori, Delsina Faiza, Efrizon, Thamrin |
| Tahun | 2024 |
| Judul | *Development of Digital Circuits Learning Media Using Articulate Storyline with the 4D Model* |
| Metode | *Research and Development (R&D), 4D Model* |
| Kesimpulan | Penelitian ini mengembangkan media pembelajaran interaktif berbasis Articulate Storyline untuk materi Rangkaian Digital di SMK Negeri 1 Padang, menggunakan *model 4D (Define, Design, Develop, Disseminate).* Hasil validasi dari ahli materi dan ahli media menunjukkan bahwa media ini sangat valid, dengan skor 91.67% untuk materi dan 94.16% untuk media. Uji coba kepraktisan dengan 15 siswa menghasilkan nilai 87.90%, menunjukkan media ini sangat praktis digunakan. |
|  |  |
| No | 4 |
| Nama Peneliti | David Sarpong, Derrick Boakye, George Ofosu |
| Tahun | 2022 |
| Judul | *The Three Pointers of Research and Development (R&D) for Growth-Boosting Sustainable Innovation System* |
| Metode | *Research and Development (R&D)* |
| Kesimpulan | Penelitian ini membahas pentingnya tiga komponen utama dalam pengembangan sistem inovasi berkelanjutan yang dapat mendorong pertumbuhan ekonomi: investasi R&D, talenta, dan lembaga pembelajaran (seperti universitas). Peneliti mengajukan model baru untuk meningkatkan sistem inovasi berkelanjutan melalui penyelarasan ketiga komponen tersebut. Penelitian ini juga menyoroti tantangan dalam memastikan investasi R&D dapat menghasilkan inovasi yang dapat diterapkan secara luas dan berkelanjutan |
|  |  |
| No | 5 |
| Nama Peneliti | Nur Wildayani Kahar, Abdul Malik, M. Hasanuddin, Muh. Hajar Harike, Suharsono Bantun |
| Tahun | 2024 |
| Judul | *Design and Development of a Web-Based Cashier and Accounting System Using CodeIgniter 4* |
| Metode | *Rapid Application Development (RAD)* |
| Kesimpulan | Penelitian ini berhasil merancang dan mengembangkan sebuah sistem pembukuan dan kasir berbasis web untuk CV Sinar Situju dengan menggunakan framework CodeIgniter 4 dan metode *Rapid Application Development* (RAD). Tujuan utama dari pengembangan sistem ini adalah untuk mengatasi permasalahan yang dihadapi oleh perusahaan, yaitu ketidakefisienan dan rawannya kesalahan dalam pengelolaan data yang dilakukan secara manual. |
|  |  |
| No | 6 |
| Nama Peneliti | Nova Selajar Da Costa, Panser Karo–Karo, NM Faizah |
| Tahun | 2024 |
| Judul | Perancangan Sistem Informasi Penerimaan Siswa Baru di SMK Harapan Bangsa Berbasis Web Menggunakan CodeIgniter 4 |
| Metode | *Waterfall* |
| Kesimpulan | Penelitian ini mengembangkan sebuah sistem informasi penerimaan siswa baru berbasis web untuk SMK Harapan Bangsa, yang bertujuan untuk meningkatkan efisiensi dan mengurangi kesalahan dalam proses pendaftaran siswa baru yang sebelumnya dilakukan secara manual menggunakan dokumen fisik dan aplikasi Microsoft Excel. Dalam penelitian ini, digunakan framework CodeIgniter 4 dan basis data MySQL untuk menciptakan sistem yang lebih terstruktur dan efektif. Peneliti juga memilih untuk menggunakan model Waterfall dalam pengembangan sistem, yang melalui tahapan analisis, desain, implementasi, dan pengujian. |
|  |  |
| No | 7 |
| Nama Peneliti | Alifah Alfiatur Rohmah, Dedi Gunawan |
| Tahun | 2023 |
| Judul | Implementasi Algoritma Priority Scheduling Sistem Informasi Pelayanan Administrasi Kependudukan Desa |
| Metode | *Priority Scheduling Algorithm* |
| Kesimpulan | Penelitian ini mengembangkan sistem informasi untuk pelayanan administrasi kependudukan desa yang mengimplementasikan algoritma priority scheduling dalam proses verifikasi surat yang diajukan oleh masyarakat. Tujuan utama dari penelitian ini adalah untuk meningkatkan efisiensi dan mengurangi keterlambatan dalam pengolahan surat administrasi di kantor desa, yang sebelumnya dilakukan secara manual, dengan membebankan proses pengurutan surat berdasarkan prioritas. |
|  |  |
| No | 8 |
| Nama Peneliti | Tati Harihayat M., Riani Lubis, Mahfudz Abdulloh |
| Tahun | 2025 |
| Judul | Keperawatan Rumah Sakit Menggunakan Algoritma Priority Scheduling |
| Metode | *Priority Scheduling Algorithm* |
| Kesimpulan | Penelitian ini berfokus pada desain sistem penjadwalan tenaga keperawatan di Unit Rawat Jalan Rumah Sakit, dengan menggunakan algoritma priority scheduling. Penjadwalan tenaga keperawatan sangat penting untuk memastikan ketersediaan tenaga medis yang cukup, serta menjamin pelayanan kesehatan yang optimal kepada pasien. Dalam penelitian ini, penjadwalan dilakukan dengan mempertimbangkan beberapa keterbatasan dan kebutuhan khusus tenaga perawat, seperti perawat yang sedang hamil, sedang studi lanjut, atau perawat yang membutuhkan jadwal tertentu karena alasan kesehatan atau sosial |
|  |  |
| No | 9 |
| Nama Peneliti | Muhammad Iqbal Afrianto, Fauziah, Yunan Fauzi Wijaya |
| Tahun | 2024 |
| Judul | Kombinasi *Algoritma Priority Scheduling* dan *Earliest Due Date* untuk Sistem Penjadwalan Slitting Produk Berbasis Web |
| Metode | *Priority Scheduling, Earliest Due Date* (EDD) |
| Kesimpulan | Penelitian ini mengusulkan kombinasi algoritma *Priority Scheduling* dan *Earliest Due Date* (EDD) untuk meningkatkan efisiensi penjadwalan slitting produk. Priority Scheduling digunakan untuk mengurutkan pekerjaan berdasarkan prioritas yang telah ditentukan, sementara EDD digunakan untuk memastikan bahwa pekerjaan yang memiliki tenggat waktu lebih dekat akan diproses lebih dulu. Gabungan dua algoritma ini bertujuan untuk mengurangi keterlambatan dan meminimalkan waktu tunggu. |
|  |  |
| No | 10 |
| Nama Peneliti | Herman Bedi Agtriadi, Hendra Jatnika, Yessy Fitriani, Zakiya Viantika Sihabudin, Muhammad Nur Khanib |
| Tahun | 2021 |
| Judul | *Implementation of WYSIWYG in the Development of ITCC ITPLN Letter Management Information System* |
| Metode | *Rapid Application Development (RAD), WYSIWYG* |
| Kesimpulan | Sistem manajemen surat berbasis web yang menggunakan teknologi WYSIWYG (*What You See Is What You Get*) berhasil meningkatkan efisiensi pengelolaan surat di ITPLN. Sistem ini menyederhanakan proses pembuatan dan pengeditan dokumen serta mempermudah penyimpanan surat dalam format digital. Dengan menggunakan WYSIWYG, dokumen dapat dikelola secara lebih efektif tanpa memerlukan keterampilan teknis yang mendalam. |
|  |  |
| No | 11 |
| Nama Peneliti | Dewi Ayu Nur Wulandari, Asyil Alfin Halani Bahar, Muhammad Ghifari Arfananda, Helina Apriyani |
| Tahun | 2021 |
| Judul | *PROTOTYPING MODEL IN INFORMATION SYSTEM DEVELOPMENT OF AL-RUHAMAA' BOGOR YATIM CENTER FOUNDATION* |
| Metode | *Model Prototyping* |
| Kesimpulan | Penggunaan *model prototyping* dalam pengembangan sistem distribusi donasi di Yayasan Yatim Al-Ruhaamaa' Bogor terbukti meningkatkan efisiensi dalam proses transaksi dan pembuatan laporan donasi. Sistem ini mengotomatisasi proses yang sebelumnya dilakukan secara manual, sehingga memberikan kemudahan dalam pelaporan dan pengelolaan donasi. |
|  |  |
| No | 12 |
| Nama Peneliti | Rima Rizqi Wijayanti, Wada Kaligula Budiargo, Abdurrasyid |
| Tahun | 2020 |
| Judul | RANCANG BANGUN SISTEM INFORMASI BORESCOPE INSPECTION REPORT PADA ENGINE MAINTENANCE PT GMF AEROASIA, TBK |
| Metode | *System Development Life Cycle (SDLC), BlackBox Testing* |
| Kesimpulan | Pengembangan sistem berbasis web untuk laporan inspeksi borescope di PT GMF AeroAsia berhasil mengurangi penggunaan sistem manual yang memakan waktu lama. Sistem ini mempermudah pengolahan dan pelaporan hasil inspeksi mesin, meningkatkan efisiensi dan akurasi dalam pelaporan. |
|  |  |
| No | 13 |
| Nama Peneliti | Hendra Jatnika, M. Farid Rifai, Yudhi S. Purwanto, Sely Karmila, Yessy Fitriani, Dine Tiara Kusuma, M Yoga Distra Sudirman |
| Tahun | 2022 |
| Judul | Implementasi Teknologi Informasi dan Pelatihan Pengelolaan Website di SMAN 3 Garut |
| Metode | *Solution Plan* |
| Kesimpulan | Implementasi teknologi pengelolaan website di SMAN 3 Garut berhasil meningkatkan kehadiran digital sekolah, memperbaiki aksesibilitas, dan memperkuat keterlibatan pengguna (siswa, guru, dan orang tua) dalam mengakses informasi yang relevan. Pelatihan yang diberikan kepada staf dan guru juga mempercepat adaptasi terhadap teknologi informasi,. |
|  |  |
| No | 14 |
| Nama Peneliti | Yessy Fitriani, Mochamad Farid Rifai, Yudhy S. Purwanto, Hendra Jatnika, M. Yoga Distra Sudirman, Dine Tiara Kusuma, Sely Karmila |
| Tahun | 2024 |
| Judul | Digitalisasi Informasi Sekolah Berbasis Web pada Madrasah Ibtidaiyah Darul Hasan |
| Metode | *Participatory Action Research (PAR)* |
| Kesimpulan | Pembuatan website untuk Madrasah Ibtidaiyah Darul Hasan berhasil meningkatkan aksesibilitas dan efisiensi pengelolaan data sekolah. Dengan website ini, informasi seperti profil sekolah, kegiatan, dan pengumuman dapat diakses kapan saja dan dari mana saja. Selain pengembangan website, pelatihan pengelolaan website dilakukan untuk memastikan bahwa staf sekolah dapat mengelola dan memperbarui informasi di website dengan baik. |

## **2.2 Landasan Teori**

Landasan teori merupakan elemen krusial dalam sebuah penelitian yang berperan sebagai pijakan atau rujukan ilmiah untuk mendukung argumen, analisis, serta interpretasi data. Bagian ini mencakup konsep, prinsip, dan teori yang berkaitan dengan topik penelitian, diambil dari literatur yang sudah ada. Dengan landasan teori, peneliti dapat memperkuat justifikasi penelitian, menghubungkan penelitian dengan studi sebelumnya, serta menjelaskan bagaimana penelitian tersebut memperluas pengetahuan atau memperdalam pemahaman terkait isu yang diteliti.

### **2.2.1 Sistem Informasi**

Sistem informasi adalah sekumpulan komponen yang terintegrasi, yang dirancang untuk mengumpulkan, menyimpan, memproses, dan mendistribusikan data dan informasi. Dalam konteks organisasi, sistem informasi memainkan peranan penting dalam mendukung operasi sehari-hari, pengambilan keputusan, dan pengendalian manajemen(Rouf, 2019) .Komponen ini terdiri dari perangkat keras dan perangkat lunak, serta mencakup proses, orang, dan hubungan yang dihasilkan dari interaksi antar komponen ((Noerdin, 2011)

Pembangunan sistem informasi melibatkan berbagai metodologi dan pendekatan yang bergantung pada kebutuhan spesifik dari instansi atau organisasi tersebut. Metode pengembangan, seperti Waterfall atau Agile, menentukan bagaimana sistem informasi tersebut akan dirancang, dikembangkan, dan diimplementasikan (Putri & Suharso, 2023) .Dalam penelitian oleh Putri dan Suharso, dikemukakan bahwa pemilihan dan penerapan metodologi yang tepat sangat krusial dalam pengembangan sistem informasi agar dapat memastikan keberhasilan dan efektivitas sistem yang dihasilkan (Putri & Suharso, 2023)

### **2.2.2 Metode *Research and Development* (R&D)**

Metode *Research and Development* (R&D) adalah suatu pendekatan sistematis yang digunakan untuk merancang, menciptakan, dan menguji produk, layanan, atau proses baru. Metode ini berfokus pada inovasi dan perbaikan berkelanjutan, serta memfasilitasi penciptaan solusi yang relevan terhadap kebutuhan pasar dan masyarakat. Dalam konteks penelitian, R&D ini sering melibatkan langkah-langkah yang berurutan mulai dari pengumpulan data, desain eksperimen, hingga pengujian produk akhir (mufadhol, 2017)

Setelah memahami konsep dasar *Research and Development* (R&D), penelitian ini mengadopsi model 4D dalam tahap pengembangan sistem. Model 4D adalah pendekatan yang sederhana namun efektif untuk merancang dan mengembangkan produk atau sistem, yang terdiri dari empat tahap utama, yaitu *Define, Design, Develop, and Dissaminate*. Lihat pada gambar 2.1

A diagram of a design

AI-generated content may be incorrect.

Gambar 2. 1 Tahapan Pengembangan 4D

Berikut tahap – tahapan metode *Research and Development* menggunakan model 4D, terdapat 4 bagian dari model 4D. Berikut penjelasan secara rinci dari model 4D.

1. *Define* (Pendefinisian)

Tahap ini membantu dalam menentukan dan menjelaskan kebutuhan serta mengumpulkan informasi terkait hal-hal yang akan dikembangkan dalam produk yang akan di buat. Tahap ini meliputi analisis awal-akhir, analisis pembelajar, analisis tugas, analisis konsep, dan tujuan-tujuan instruksional khusus.

1. *Design* (Perancangan)

Tahap ini membantu menentukan desain yang akan diterapkan. Pada tahap ini dapat dilakukan pemilihan media, pemilihan format, dan pembuatan rancangan awal.

1. *Develop* (Pengembangan)

Tahap ini bertujuan untuk menghasilkan produk. Dalam tahap ini, produk yang telah dibuat harus melewati beberapa tahap perbaikan dari ahli atau validator dan diuji terhadap konsumen sebagai pengguna. Tahap uji pengembang dilakukan jika proses pembuatan produk telah selesai dan siap diuji kevalidanya oleh para ahli. Proses pengujian dilakukan bertahap, mulai dari uji validasi produk oleh para ahli, uji coba pada skala individu, uji coba pada kelompok kecil, dan terakhir uji coba pada kelompok besar. Apabila produk terbukti tidak atau kurang valid saat di uji oleh ahli, pengujian ini akan diulang untuk memastikan hasil yang sesuai dengan harapan.

1. *Disseminate* (Penyebaran)

pada tahap ini produk dapat disebarkan dan dikenal

kan kepada masyarakat luas melampaui lingkup pengembangan itu sendiri. Beberapa faktor yang harus dipertimbangkan saat melakukan penyebarluasan adalah analisis pengguna, strategi dan tema, timing penyebaran, dan pemilihan media penyebaran(Riani Johan et al., n.d.)

### **2.2.3 *Web - Application***

Aplikasi web (*web app*) adalah program aplikasi yang disimpan di server jarak jauh dan dikirimkan melalui internet melalui antarmuka browser. Layanan web adalah aplikasi web menurut definisinya dan banyak, meskipun tidak semua, situs web berisi aplikasi web.

Pengembang merancang aplikasi web untuk berbagai macam penggunaan dan pengguna, dari organisasi hingga individu karena berbagai alasan. Aplikasi web yang umum digunakan dapat mencakup webmail, kalkulator online, jejaring sosial , atau toko e-commerce . Meskipun pengguna hanya dapat mengakses beberapa aplikasi web melalui browser tertentu, sebagian besar tersedia tanpa memandang browser [(”*Techtarget”,2024)*.](https://www.techtarget.com/searchsoftwarequality/definition/Web-application-Web-app)

### **2.2.4 *CodeIgniter***

*CodeIgniter* adalah sebuah *framework* yang dikembangkan pada tahun 2006 oleh Rick Ellis. *CodeIgniter* berfungsi untuk web dan *application development* yang hadir dalam bentuk platform *open-source*. Framework ini diciptakan untuk para developer yang hendak membangun situs web maupun aplikasi menggunakan bahasa pemrograman PHP. Kehadiran *CodeIgniter* ini bertujuan agar para developer dapat mengembangkan proyek mereka dengan jauh lebih cepat dan mudah. Sebab, menurut developer *CodeIgniter*, kinerja seorang developer terkadang menjadi lebih lama karena mereka harus menulis kodenya sendiri dari awal.

Melalui *CodeIgniter* ini, mereka menyediakan built-in-library di mana terdapat ratusan template dan solusi untuk pekerjaan yang sering kali dikerjakan oleh para developer. Selain itu, *CodeIgniter* juga menyediakan *user interface* (UI) dan struktur yang sederhana serta logis untuk mengakses *built-in-library*.[(](https://www.biznetgio.com/news/apa-itu-codeigniter)*[BizNetgio)](https://www.biznetgio.com/news/apa-itu-codeigniter)*

.

### **2.2.5 PHP (*Hypertext Preprocessor*)**

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman open-source yang umumnya digunakan untuk membangun aplikasi web dinamis dan interaktif. PHP dapat dijalankan pada server web dan dikombinasikan dengan HTML, CSS, dan JavaScript untuk membuat halaman web yang dinamis. Saat ini, PHP sangat populer di kalangan web developer karena mudah dipelajari dan memiliki kemampuan yang cukup kuat. PHP juga mendukung banyak jenis database, seperti MySQL, PostgreSQL, dan Oracle sehingga memungkinkan pengembang untuk membuat aplikasi web yang lebih kompleks dan fungsional.

Tidak hanya itu, bahasa pemrograman ini juga memiliki banyak kerangka kerja (framework) yang dapat digunakan oleh pengembang untuk mempercepat proses pembuatan aplikasi web. Beberapa kerangka kerja PHP yang populer antara lain *Laravel*, *CodeIgniter*, dan *Symfony*. Sebagai bahasa penulisan skrip atau bahasa yang mengotomatiskan eksekusi task, PHP sebenarnya mirip dengan JavaScript dan Python. Namun yang membedakannya adalah PHP digunakan untuk komunikasi di sisi server. Sedangkan, JavaScript digunakan untuk *frontend* dan *backend*, sera Python hanya untuk sisi server (*backend*). ([*BizNetgio*](https://www.biznetgio.com/news/apa-itu-php/)*)*

### **2.2.6 MySQL**

MySQL adalah RDBMS (*relational database management system*) sumber terbuka yang menggunakan SQL (*Structured Query Language*) untuk membuat dan mengelola basis data. Sebagai basis data relasional, MySQL menyimpan data dalam bentuk tabel baris dan kolom yang disusun dalam skema. Skema menentukan cara data disusun dan disimpan serta menggambarkan hubungan antara berbagai tabel. Dengan format ini, pengembang dapat dengan mudah menyimpan, mengambil, dan menganalisis berbagai jenis data, termasuk teks sederhana, angka, tanggal, waktu, dan, yang terbaru, JSON dan vektor .

Karena MySQL bersifat sumber terbuka, MySQL mencakup banyak fitur yang dikembangkan melalui kerja sama erat dengan komunitas pengguna selama hampir 30 tahun. Dua kemampuan yang diandalkan oleh para pengembang adalah dukungan MySQL untuk transaksi ACID dan kemampuan MySQL untuk melakukan penskalaan. ACID merupakan singkatan dari “ *atomicity, consistency, isolation, and durability*,” yaitu empat properti yang memastikan transaksi basis data diproses secara andal dan akurat. Dengan transaksi ACID, MySQL dapat menjamin bahwa semua modifikasi data dilakukan secara konsisten dan andal, bahkan jika terjadi kegagalan sistem. MySQL dapat ditingkatkan skalanya untuk mendukung basis data yang sangat besar, dan dapat menangani sejumlah besar koneksi bersamaan. Performa MySQL, kemudahan penggunaan, dan biaya rendah yang dipadukan dengan kemampuannya untuk ditingkatkan secara andal seiring pertumbuhan bisnis telah menjadikannya basis data sumber terbuka paling populer di dunia.[(](https://www.oracle.com/mysql/what-is-mysql/)*[Oracle)](https://www.oracle.com/mysql/what-is-mysql/)*

### **2.2.7 XAMPP**

XAMPP merupakan *software* yang dikembangkan oleh sekelompok tim *Apache Friend* pada 2002 dan bisa didapatkan secara gratis dengan label *General Public License* (GNU). Sebagai *software open source* berbasis web server, XAMPP ini memiliki berbagai program dan mendukung berbagai sistem operasi yang umum digunakan, seperti Linux, Windows, MacOS, dan Solaris. Aplikasi ini berfungsi sebagai *server* lokal yang sudah mencakup program Apache, MySQL, dan PHP.

XAMPP disebut juga sebagai *standalone server* atau *server* yang dapat berdiri sendiri sehingga memudahkan pengguna saat menjalankan proses pengeditan, desain, dan pengembangan aplikasi. Penggunaan XAMPP dirasa mampu menghemat anggaran karena dapat menggantikan peran web hosting dengan cara menyimpan file website ke dalam localhost agar bisa dipanggil atau dihubungkan melalui browser. Adapun XAMPP memiliki kepanjangan X (*cross platform*), A (*Apache*), M (MySQL/MariaDB), P (PHP), dan P (Perl). Kepanjangan XAMPP tersebut merupakan program yang tersedia pada *software* ini.(*[BizNetgio](https://www.biznetgio.com/news/apa-itu-xampp))*

### **2.2.8 *Unified Modelling Language (UML)***

*(UML) Unified Modelling Language* adalah salah satu alat yang membantu pemodelan secara visual untuk membantu proses perancangan sistem sehingga meminimalisir kesalahan dalam membuat program(Voutama, 2022). UML dapat didefinisikan sebagai bagan untuk memvisualisasikan perancarangan atau gambaran alur sebuah sistem. Dengan UML diharapkan dapat mempermudah pengembangan sistem, serta memenuhi kebutuhan *user* dengan efektif, lengkap, dan tepat.

#### **Use Case diagram**

*use case diagram* adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Diagram ini terdiri dari Aktor ( entitas yang berada diluar sistem), dan use-case (gambaran umum fungsional sistem).(*[SekawanMedia](https://www.sekawanmedia.co.id/blog/use-case-diagram/))*

| Simbol | Keterangan |
| --- | --- |
|  | Aktor: Mewakili peran seseorang, sistem lain, atau alat ketika berinteraksi dengan sistem. |
|  | Use Case : Sebuah abstraksi dari interaksi antara sistem dan aktor. |
|  | Association : Sebuah abstraksi dari hubungan antara aktor dan use case. |
| <<*extend>>* | Kondisi Ini menunjukkan bahwa sebuah use case merupakan fungsionalitas tambahan dari use case lain jika kondisi tertentu terpenuhi. |
| <<*include>>* | Kondisi ini juga menunjukkan bahwa sebuah use case sepenuhnya mencakup fungsionalitas dari use case lain. |

#### **Activity diagram**

*Activity Diagram* adalah rancangan aliran kerja sistem, *Activity Diagram* digunakan untuk menjelaskan atau mengelompokkan alur sistem.([*Binus*](https://socs.binus.ac.id/2019/11/22/uml-diagram-activity-diagram/)*)*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Status Awal | Diagram aktivitas dimulai dengan keadaan awal. |
|  | Aktivitas | Tindakan yang dilakukan oleh sistem, biasanya dimulai dengan kata kerja. |
|  | Percabangan/ Decision | Titik-titik di mana terdapat beberapa aktivitas yang mungkin dilakukan. |
|  | Penggabungan/join | Titik-titik di mana beberapa aktivitas bergabung menjadi satu. |
|  | Swimlane | Ini memisahkan organisasi bisnis yang berbeda yang bertanggung jawab atas aktivitas yang terjadi di dalam diagram. |

#### **Class diagram**

*Class Diagram* digunakan untuk menggambarkan hubungan sistem serta menampilkan struktur statisnya. *Class Diagram* merepresentasikan kelas-kelas didalam sistem beserta atribut dan operasinya.(*[SekawanMedia](https://www.sekawanmedia.co.id/blog/class-diagram/))*

|  |  |
| --- | --- |
| Attribute | Penjelasan |
| Asosiasi | Hubungan umum antar kelas, sering kali disertai dengan multiplisitas. |
| Asosiasi berarah | Hubungan yang menunjukkan bahwa satu kelas menggunakan kelas lain, biasanya disertai dengan multiplisitas. |
| Generalisasi | Hubungan yang menunjukkan hierarki atau pewarisan, di mana satu kelas adalah versi yang lebih spesifik dari kelas lainnya. |
| Kebergantungan | Hubungan yang menandakan hubungan seluruh bagian antar kelas. |
| Apresiasi | Hubungan yang menunjukkan bahwa satu kelas sepenuhnya terdiri dari bagian-bagian kelas lain. |

#### **Sequence diagram**

*Sequence diagram* menunjukkan bagaimana operasi dalam sistem dilakukan, bentuknya berupa pesan (*message*) dengan apa yang dikirim. *Sequence diagram* menampilkan interaksi antar objek dalam dua dimensi.(*[SekawanMedia](https://www.sekawanmedia.co.id/blog/sequence-diagram/)*)

|  |  |
| --- | --- |
| Atribut | Penjelasan |
| nama\_aktor | *actor*, Individu, proses, atau sistem lain yang berinteraksi dengan dan mendapatkan manfaat dari sistem. |
| Objek  nama\_aktor | Terlibat dalam proses dengan mengirim atau menerima pesan secara berurutan. |
| Garis hidup objek | Mewakili tanda hidup objek selama interaksi, diakhiri dengan tanda 'X' pada titik di mana objek tidak lagi berinteraksi. |
| Objek sedang aktif berinteraksi | Menunjukkan kapan suatu objek mengirim atau menerima pesan. |
| Pesan  pesan() | Satu objek mengirim pesan ke objek lain. |
| <<create>> | Sebuah objek membuat objek lain, ditunjukkan oleh panah yang menunjuk ke objek yang baru dibuat. |
| Masukan | Sebuah objek mengirimkan input ke objek lain, ditunjukkan dengan tanda panah yang mengarah ke objek penerima. |
| keluaran | Sebuah objek atau metode mengembalikan nilai ke objek tertentu, ditunjukkan dengan tanda panah yang mengarah ke objek penerima. |
| destroy()  X | Sebuah objek mengakhiri keberadaan objek lain, ditunjukkan dengan tanda panah yang mengarah ke objek yang dihancurkan. Idealnya, jika ada penciptaan, harus ada juga *destroy*. |

### ***Black-Box Testing***

Black box testing atau dapat disebut juga Behavioral Testing adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, penguji tidak harus memiliki kemampuan menulis kode program. Pengujian ini dapat dilakukan oleh siapa saja.

A diagram of a black box

AI-generated content may be incorrect.

Sumber :<https://itbox.id/wp-content/uploads/2023/03/thumbnail_Black-box.jpeg>

Gambar 2. 2 Black-Box Testing

## **2.3 Kerangka Pemikiran**

Kerangka pemikiran Ini berfungsi sebagai panduan dasar untuk melakukan penelitian dengan memberikan struktur dan pemahaman yang jelas tentang proses penelitian.

A diagram of a pyramid

AI-generated content may be incorrect.

Gambar 2. 3 Kerangka Berpikir

1. Input

Pada bagian input, dijelaskan komponen data yang menjadi dasar penelitian. Input yang dimaksud di sini berupa data pengajuan yang terdiri dari dua tipe, yaitu pengajuan permohonan dan pengajuan gangguan.

1. Proses

Pada bagian proses, terdapat tahap pengumpulan data dan klasifikasi urgensi. Tahap pengumpulan data memastikan bahwa data yang diajukan oleh pengguna telah terdaftar dalam sistem. Tahap klasifikasi urgensi adalah tahap di mana data yang telah dikumpulkan diklasifikasikan berdasarkan tingkat urgensinya. Sistem akan memprioritaskan pengajuan berdasarkan indikator yang telah ditentukan.

1. Output

Pada bagian ini, akan ada dua *output* yang dihasilkan. *Output* pertama menampilkan status pengajuan, apakah telah diproses, ditunda (*pending*), atau ditolak. Sistem akan menampilkan status terkini mengenai pengajuan yang masuk, sehingga memudahkan pengguna untuk mengetahui perkembangan pengajuan mereka. *Output* kedua menampilkan urutan pemrosesan pengajuan berdasarkan tingkat urgensi yang telah ditetapkan oleh sistem. Pengajuan yang lebih mendesak akan diproses lebih cepat dibandingkan dengan pengajuan yang memiliki prioritas lebih rendah. Setelah diproses, status pengajuan pada halaman pengguna akan berubah menjadi selesai.

**BAB III   
METODE PENELITIAN**

**3.1 Waktu dan Tempat Penelitian**

Tempat dan lokasi penelitian Pengembangan Aplikasi Sistem Informasi Layanan Permohonan dan Gangguan pada Pusat Data dan Teknologi Informasi Kementerian Agama adalah sebagai berikut.

Waktu : Maret 2025 – Juli 2025

Tempat : Jl. Lap. Banteng Barat No.3, Ps. Baru, Kecamatan Sawah Besar, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta 10710

**3.2 Desain Penelitian**

Desain tahapan penelitian ini disusun dengan tujuan untuk menjelaskan secara rinci alur dan tahapan yang akan diikuti dalam proses penelitian. Untuk memberikan gambaran yang lebih jelas mengenai alur penelitian ini, dapat dilihat pada Gambar 3.1 sebagai berikut:

A diagram of a company

AI-generated content may be incorrect.

Gambar 3. 1 Desain penelitian

1. *Define* (Pendefinisian)

Pada tahap ini dilakukan analisis untuk mengidentifikasi dam mendefinisikan permasalahan yang ada dalam proses pengajuan aplikasi Sistem Informasi Layanan Gangguan dan Permohonan (SIGAP) di Kementerian Agama Republik Indonesia. Identifikasi dilakukan melalui dua metode utama, yaitu wawancara dan obeservasi.

1. Wawancara

Wawancara dilakukan dengan Bapak Irfan Sembiring S.T, M.T.I yang merupakan Kepala Pusat Data dan Informasi di Kementerian Agama Republik Indonesia. Pemilihan narasumber ini dilakukan karena beliau memiliki pemahaman yang mendalam tentang sistem yang sedang dikembangkan dan terlibat langsung dalam pengelolaan administrasi pengajuan di Kementerian Agama. Walaupun hanya satu orang yang diwawancarai, wawancara ini memberikan wawasan yang sangat berharga mengenai permasalahan yang ada dalam sistem yang sedang berjalan. Informasi yang didapatkan digunakan untuk memahami masalah yang ada dalam sistem lama, seperti kesalahan input, keterlambatan proses dan ketidakteraturan pengklasifikasian urgensi. Wawancara dilakukan selama tujuh menit lima puluh detik. Berikut penulis melampirkan data pendukung yang bisa dilihat pada tabel .... dibawah ini

(Isi wawancara)

Hasil wawancara menunjukkan bahwa sistem yang ada saat ini masih manual, dimana pengklasifikasiannya masih dilakukan secara manual yang berisiko tinggi terhadap keefektifan sistem pengajuan administrasi dan juga sistem lama masih menggunakan *CodeIgniter* versi 1 yang sudah lemot, sehingga terkadang ada pengajuan yang telat atau tidak dilakukan sama sekali.

1. Observasi

Observasi dilakukan selama pelaksanaan magang di Kementerian Agama, yang berlangsung selama 3 bulan. Tujuan dari observasi ini adalah untuk mengetahui alur proses pengajuan administrasi. Penulis memantau bagaimana pengajuan diterima dan diproses melalui sistem yang sedang berjalan. Observasi ini dilakukan untuk memahami secara praktis bagaimana alur sistem pengajuan berjalan dan untuk mengindetifikasikan masalah yang tidak terungkap dalam wawancara. Dalam observasi Penulis mencatat beberapa masalah utama seperti, judul pengajuan masih dilakukan mengetik secara manual sehingga berisiko tinggi terhadap kesalahan input yang bisa mengakibatkan pengkategorian pengajuan jadi rancu dan nggak sesuai. Yang kedua terlalu banyak kategori dalam sistem pengajuan yang lama sehingga membuat admin yang mengelola pengajuan administrasi jadi bingung dan mengakibatkan beberapa pengajuan terlambat dikerjakan.

1. Analisis Proses Bisnis Sistem Order SIGAP

|  |  |  |  |
| --- | --- | --- | --- |
| No | Proses Bisnis | Detail Aktivitas | Aktor Terkait |
| 1 | Pembuatan order | User Satker login, mengakses daftar order, dan mengisi form order | User Satker |
| 2 | Verifikasi order | Admin menerima, memeriksa, dan memutuskan menerima atau menolak order | Admin |
| 3 | Eksekusi order | Order yang diterima diteruskan ke Super Admin untuk diproses dan diselesaikan | SuperAdmin |
| 4 | Pengiriman Hasil order | Super Admin menyelesaikan dan mengirimkan hasil order kepada user | SuperAdmin |
| 5 | Penerimaan Hasil order | User Satker menerima hasil dan proses dianggap selesai | User Satker |

1. Analisis Aktor Sistem Order SIGAP

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi Peran |
| 1 | User Satker | Mengajukan order dan menerima hasil order |
| 2 | Admin | Memverifikasi dan menentukan apakah order diterima atau ditolak |
| 3 | SuperAdmin | Menangani order yang telah disetujui oleh Admin dan menyelesaikannya |

1. Analisis Aturan Bisnis

|  |  |
| --- | --- |
| No | Aturan Bisnis |
| 1 | Order hanya dapat diproses jika sudah disetujui oleh Admin |
| 2 | Admin memiliki hak mutlak untuk menerima atau menolak order |
| 3 | Super Admin tidak dapat memproses order yang belum di-ACC |
| 4 | Setiap order yang selesai wajib dikirimkan kembali ke user oleh Super Admin |
| 5 | Status order harus tercatat di sistem: "Pending", "Disetujui", "Ditolak", dll. |

1. Analisis Masalah dan Solusi Sistem Order SIGAP

|  |  |  |
| --- | --- | --- |
| No | Permasalahan Utama | Solusi yang diusulkan |
| 1 | Sistem SIGAP masih menggunakan *framework CodeIgniter* versi 1 yang sudah usang dan tidak optimal | *Rebuild* sistem menggunakan *framework modern* yaitu *Codeigniter* versi 4 |
| 2 | Penentuan tingkat urgensi masih dilakukan secara manual, sehingga rawan bias dan keterlambatan |  |
| 3 | Tampilan dan performa sistem lama lambat, tidak responsif, dan membingungkan pengguna | *Redesign* UI/UX dengan pendekatan *user-centered design* dan tampilan responsif |
| 4 | Tidak ada pemisahan peran yang jelas antara pemeriksa dan pelaksana order | Membedakan peran: Admin bertugas memverifikasi, Super Admin bertugas menyelesaikan order |
| 5 | Risiko overload pekerjaan jika semua order langsung ke Super Admin tanpa filter | Order hanya diteruskan ke Super Admin jika sudah di-ACC oleh Admin, sehingga workload lebih terkontrol |
| 6 | User tidak memiliki informasi real-time mengenai status order | Sistem menyediakan notifikasi status otomatis: “Dipending”, “Disetujui”, “Ditolak”, “Diproses”,  “Done” |
| 7 | Tidak ada jalur penolakan formal jika order tidak valid | Menyediakan opsi “Tolak Order” oleh Admin dan menyampaikan alasan penolakan kepada user secara langsung (email). |

Berdasarkan hasil wawancara dan observasi yang dilakukan oleh penulis, dapat disimpulkan bahwa pengembangan aplikasi Sistem Informasi Layanan Gangguan dan Permohonan (SIGAP) yang lebih sederhana, dengan kemampuan untuk mengklasifikasikan urgensi secara otomatis melalui sistem serta memprioritaskan pemrosesan pengajuan, sangat diperlukan guna meningkatkan efisiensi dan mengurangi kesalahan dalam pengelolaan pengajuan di Kementerian Agama.

1. *Design* (Perancangan)

Berdasarkan hasil analisis pada tahap Define, masalah utama dalam pengajuan administrasi adalah pengklasifikasian urgensi yang dilakukan secara manual, kesalahan input data, serta kebingungan akibat terlalu banyak kategori dalam sistem lama. Oleh karena itu, tujuan desain sistem ini adalah untuk mengembangkan aplikasi Sistem Informasi Layanan Gangguan dan Permohonan (SIGAP) yang dapat mengotomatisasi proses pengklasifikasian urgensi pengajuan, memprioritaskan pengolahan data, serta menyederhanakan kategori pengajuan untuk meningkatkan efisiensi dan mengurangi kesalahan.

1. *Context Diagram*

*Context Diagram* merupakan bagian dari perancangan sistem yang bertujuan untuk menggambarkan batasan sistem dan interaksi antar aktor eksternal dengan sistem informasi yang dikembangkan. Diagram ini berfungsi untuk memberikan pemahaman awal tentang bagaimana sistem menerima masukan, memproses data, dan menghasilkan keluaran kepada aktor terkait.

A diagram of a system

AI-generated content may be incorrect.

Gambar 3. 2 Context Diagram

Pada pengembangan Sistem Informasi Layanan Gangguan dan Permohonan (SIGAP) di Kementerian Agama Republik Indonesia, terdapat tiga aktor utama yang berinteraksi dengan sistem, yaitu: User Satker, Admin, dan Super Admin. Setiap aktor memiliki hak akses dan fungsi berbeda sesuai tanggung jawabnya. Berikut

|  |  |
| --- | --- |
| Aktor | Peran |
| User Satker | Aktor yang mengajukan permohonan layanan atau melaporkan gangguan melalui sistem SIGAP. User melakukan registrasi dan login untuk dapat mengakses fitur pengajuan dan menerima hasil atau notifikasi status pengajuan. |
| Admin | Bertugas melakukan verifikasi terhadap pengajuan yang masuk. Admin dapat memutuskan apakah pengajuan ditolak atau disetujui untuk dilanjutkan ke tahap proses. Admin juga melakukan login sebelum mengakses sistem. |
| SuperAdmin | Berperan dalam menindaklanjuti pengajuan yang telah diverifikasi dan disetujui oleh admin. Super Admin memproses pengajuan dan mengunggah hasilnya ke dalam sistem untuk dikirimkan kembali kepada user. Sama seperti aktor lainnya, Super Admin wajib login terlebih dahulu. |

1. *Use Case Diagram*

Use Case Diagram menggambarkan hubungan antara aktor eksternal dan fungsi-fungsi utama yang tersedia dalam sistem SIGAP. Model ini membantu menjelaskan bagaimana setiap peran (aktor) berinteraksi dengan sistem dan batasan-batasan akses yang dimiliki.

Sistem SIGAP memiliki tiga jenis pengguna utama, yaitu User Satker, Admin, dan Super Admin. Setiap pengguna memiliki fitur terbatas sesuai peran dan tanggung jawabnya.

1. *Use Case Diagram* Satker

A diagram of a user seker

AI-generated content may be incorrect.

Gambar 3. 3 Use Case Satker

User Satker berfungsi sebagai pengusul layanan atau pelapor gangguan. Adapun fitur yang tersedia :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Use Case | Deskripsi | Aktor | Kondisi Awal | Kondisi Akhir |
| 1 | *Register* | Proses pendaftaran pengguna baru dengan email dan password | User Satker | Pengguna belum memiliki akun | Pengguna berhasil terdaftar dan dapat login. |
| 2 | *Login* | Pengguna masuk ke sistem menggunakan akun terdaftar | User Satker | Pengguna berada di halaman login, sudah punya akun | Pengguna berhasil masuk dan diarahkan ke halaman Home |
| 3 | *Home* | Halaman awal setelah login, berisi ringkasan sistem dan navigasi utama | User Satker | Pengguna sudah login | Pengguna dapat mengakses fitur lainnya melalui menu |
| 4 | Daftar Pengajuan | Pengguna melihat daftar pengajuan yang telah diajukan sebelumnya | User Satker | Pengguna sudah login | Sistem menampilkan daftar pengajuan milik pengguna |
| 5 | Buat Pengajuan | |  | | --- | |  |  |  | | --- | | Pengguna membuat dan mengirim permohonan atau laporan baru | | User Satker | Pengguna sudah login dan berada di halaman pengajuan | Sistem menyimpan pengajuan baru dan menambah ke daftar |
| 6 | Profile | Menampilkan informasi akun pengguna | User Satker | Pengguna sudah login | Sistem menampilkan Informasi akun milik pengguna |
| 7 | Edit Profile | Pengguna mengubah informasi akunnya (nama, email, dan password) di halaman Profile | User Satker | Pengguna sudah login dan akses profile | Perubahan profil tersimpan |
| 8 | Logout | Pengguna keluar dari sesi penggunaan aplikasi | User Satker | Pengguna sudah login | Pengguna kembali ke halaman login |

1. *Use Case Diagram* Admin

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 3. 4 Use Case Diagram Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Use Case | Deskripsi | Aktor | Kondisi Awal | Kondisi Akhir |
| 1 | Login | Proses masuk ke sistem menggunakan akun yang valid. | Admin | Admin berada di halaman login. | Admin diarahkan ke halaman Home. |
| 2 | Home | Halaman utama setelah login, menampilkan navigasi utama sistem. | Admin | Admin telah berhasil login. | Admin dapat memilih fitur lain seperti Pengajuan Masuk |
| 3 | Pengajuan Masuk | Menampilkan daftar pengajuan dari user Satker yang harus ditinjau. | Admin | Admin memilih menu Pengajuan Masuk. | Admin dapat menyetujui, menolak, atau mengunduh file. |
| 4 | Setujui Pengajuan | Aksi menyetujui pengajuan yang dipilih. | Admin | Admin berada di halaman Pengajuan Masuk. | Status pengajuan berubah menjadi “Disetujui”. |
| 5 | Tolak Pengajuan | Aksi menolak pengajuan yang dipilih | Admin | Admin berada di halaman Pengajuan Masuk. | Status pengajuan berubah menjadi “Ditolak”. |
| 6 | Download File | Mengunduh file lampiran dari pengajuan untuk diperiksa. | Admin | Admin berada di halaman Pengajuan Masuk. | File berhasil diunduh ke perangkat admin. |
| 7 | Status Pengajuan | Menampilkan semua pengajuan yang telah ditinjau, baik diterima maupun ditolak. | Admin | Admin memilih menu Status Pengajuan. | Admin dapat melihat status akhir dari pengajuan yang disetujui. |
| 8 | |  | | --- | |  |  |  | | --- | | Pengajuan Ditolak | | Menampilkan daftar pengajuan yang telah ditolak. | Admin | Admin memilih menu Pengajuan Ditolak. | Admin dapat melihat detail pengajuan yang ditolak. |
| 9 | Logout | Keluar dari sistem dan mengakhiri sesi. | Admin | Admin dalam kondisi login. | Admin diarahkan kembali ke halaman login. |

1. *Use Case Diagram* Super Admin

A diagram of a diagram

AI-generated content may be incorrect.

Gambar 3. 5 Use Case Super Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Nama Use Case | Deskripsi | Aktor | Kondisi Awal | Kondisi Akhir |
| 1 | Login | Super Admin masuk ke sistem dengan kredensial | Super Admin | Halaman login | Berhasil masuk dan diarahkan ke Home |
| 2 | |  | | --- | |  |  |  | | --- | | Home | | Menampilkan menu utama system | Super Admin | Login berhasil | Menu dapat diakses |
| 3 | Pengajuan Masuk | Melihat daftar pengajuan baru dari admin | Super Admin | Data pengajuan tersedia | Tabel pengajuan ditampilkan |
| 4 | Pengajuan diselesaikan | Menandai pengajuan sebagai selesai | Super Admin | Pengajuan status process | Status menjadi “Selesai” |
| 5 | Pengajuan dipending | Memberi tanda bahwa pengajuan belum bisa disetujui | Super Admin | Pengajuan status process | Status menjadi “Pending” |
| 6 | Download File | Mengunduh file lampiran dari pengajuan | Super Admin | File tersedia dalam pengajuan | File berhasil diunduh |
| 7 | Pengajuan Selesai | Melihat daftar pengajuan yang telah disetujui | Super Admin | Ada pengajuan berstatus “Selesai” | Daftar ditampilkan |
| 8 | Logout | Keluar dari system | Super Admin | Dalam kondisi login | Kembali ke halaman login |

1. Activity diagram

1, Activity diagram login

A diagram of a computer

AI-generated content may be incorrect.

Pada gambar, User Satker memulai proses dengan menginputkan username dan password. Sistem kemudian melakukan validasi login untuk memastikan kredensial yang dimasukkan sesuai dengan data yang tersimpan. Jika login tidak valid, pengguna diminta untuk menginputkan kembali informasi login. Namun, jika login valid, sistem akan mengarahkan User Satker ke Halaman Home, yang menandakan bahwa pengguna berhasil masuk dan dapat mengakses fitur atau fungsi utama dari sistem.

1. Activity diagram Register

A diagram of a software system

AI-generated content may be incorrect.

Pada gambar, proses dimulai ketika User Satker mengisi form registrasi dengan data yang diperlukan. Setelah itu, sistem melakukan validasi registrasi untuk memastikan bahwa data yang dimasukkan sudah benar dan lengkap. Jika data tidak valid, sistem akan mengarahkan pengguna untuk memperbaiki form registrasi. Namun, jika data valid, pengguna akan diarahkan ke Halaman Login, menandakan bahwa registrasi berhasil dilakukan dan User Satker dapat melanjutkan untuk login ke sistem

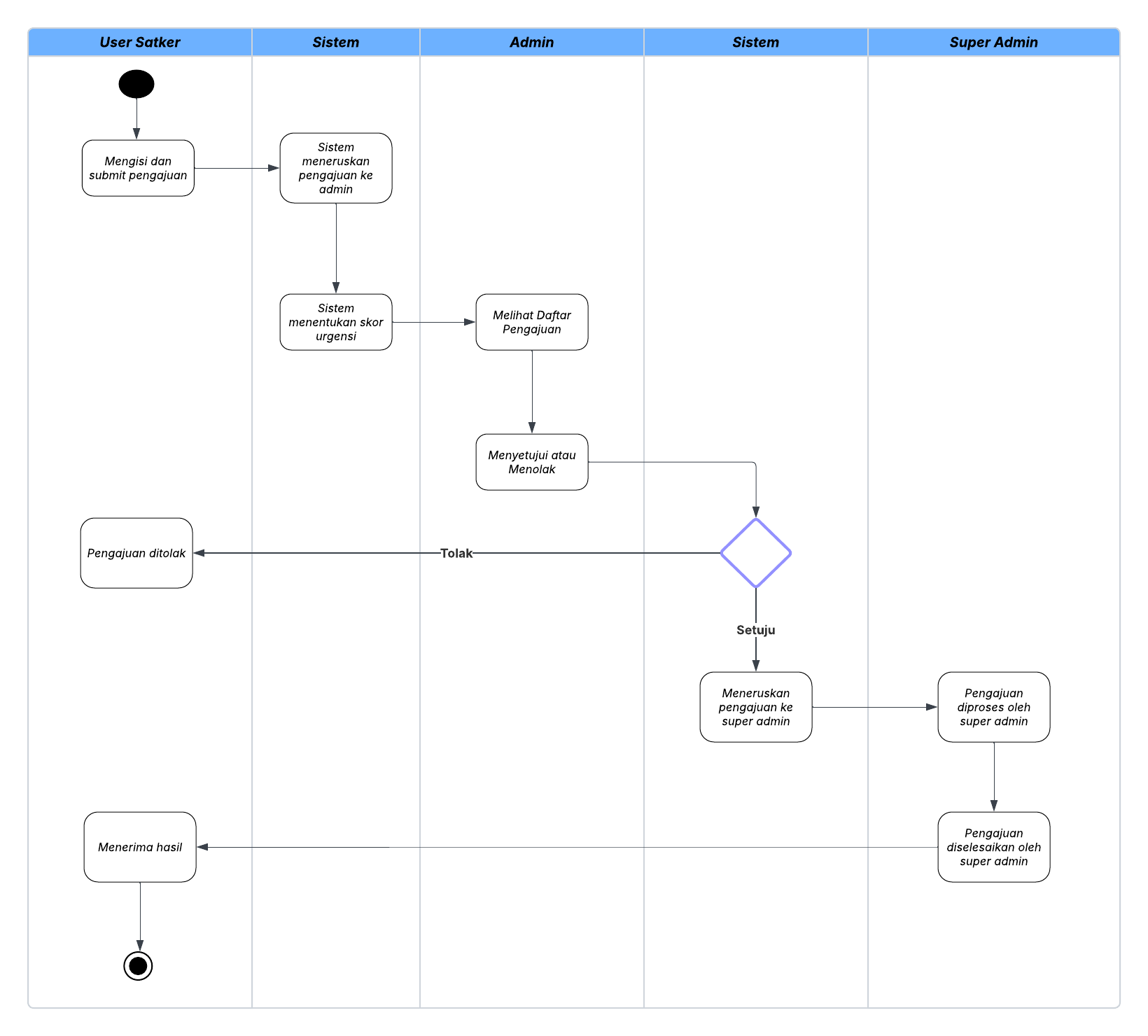
1. Activity diagram Buat Order

A diagram of a product

AI-generated content may be incorrect.

Pada gambar, proses dimulai ketika User Satker pergi ke menu Buat Order dan menginputkan orderan baru. Setelah itu, sistem melakukan validasi untuk memastikan apakah semua input yang diperlukan telah terisi dengan benar. Jika ada input yang belum terisi, User Satker akan diarahkan kembali untuk melengkapi form orderan. Namun, jika semua input valid, sistem akan mengonfirmasi bahwa orderan baru telah dibuat dan proses order berhasil dan akan kembali ke menu Buat Order.

1. Activity diagram Proses persetujuan dan penolakan



Berdasarkan gambar ...., proses dimulai ketika User Satker mengisi dan submit pengajuan. Sistem kemudian meneruskan pengajuan ke Admin, yang selanjutnya melihat daftar pengajuan dan memproses pengajuan tersebut. Setelah itu, Sistem menentukan skor urgensi pengajuan. Admin kemudian memutuskan untuk menyetujui atau menolak pengajuan tersebut. Jika pengajuan ditolak, sistem menginformasikan kepada User Satker bahwa pengajuan mereka ditolak. Jika pengajuan disetujui, maka Sistem akan meneruskan pengajuan ke Super Admin. Super Admin kemudian memproses dan menyelesaikan pengajuan tersebut, yang mengakhiri alur proses. User Satker akhirnya menerima hasil dari pengajuan yang telah diproses dan diselesaikan oleh Super Admin.

1. Activity diagram penentu urgensi

A diagram with text and images

AI-generated content may be incorrect.

Gambar...... menggambarkan proses penentuan dan tampilan level urgensi dalam sistem. Proses dimulai ketika sistem menerima data pengajuan yang mencakup deadline dari pengguna. Setelah menerima data, sistem kemudian menghitung selisih hari antara tanggal pengajuan dengan deadline yang ditentukan. Berdasarkan selisih hari tersebut, sistem menentukan skor urgensi, yang dihasilkan dari perhitungan yang sudah ditetapkan. Setelah skor urgensi ditentukan, sistem menyimpan skor urgensi tersebut. Terakhir, sistem menampilkan level urgensi pada halaman Admin, memberikan admin informasi mengenai level urgensi (Tinggi, Sedang, atau Rendah) dari pengajuan tersebut. Diagram ini menggambarkan seluruh proses secara jelas, mulai dari penerimaan data hingga tampilan level urgensi di halaman admin.

1. Class diagram

A diagram of a computer

AI-generated content may be incorrect.

1. Sequence diagram
   * + 1. Sequence diagram Login

A diagram of a user login

AI-generated content may be incorrect.

Berdasarkan gambar, alur proses validasi login User Satker dijelaskan sebagai berikut. Proses dimulai ketika User Satker berada di halaman login dan menginputkan username dan password mereka. Data yang dimasukkan akan diterima oleh Sistem yang kemudian memvalidasi data tersebut. Sistem akan menghubungi MySQL untuk mengambil data pengguna yang sesuai dengan username yang dimasukkan. Jika username dan password yang dimasukkan valid, maka Sistem akan mengirimkan User Satker ke halaman Home (main page), yang menunjukkan bahwa login berhasil dan pengguna dapat melanjutkan akses ke sistem.

Namun, jika password yang dimasukkan salah atau data yang dimasukkan tidak sesuai dengan yang ada di MySQL, maka Sistem akan menampilkan pesan "Password salah" yang muncul sebagai *popup notification*. Pesan ini memberitahukan User Satker bahwa login tidak valid dan meminta mereka untuk mengulangi proses login dengan data yang benar. Dengan demikian, Sequence Diagram ini menggambarkan interaksi antara User Satker, Sistem, dan MySQL dalam memastikan bahwa data yang dimasukkan valid sebelum memberikan akses ke sistem.

1. sequence diagram Register

A diagram of a system

AI-generated content may be incorrect.

Berdasarkan gambar, berikut adalah penjelasan tentang proses registrasi pengguna. Proses dimulai ketika User Satker mengisi form registrasi dengan informasi seperti email dan password. Data yang dimasukkan kemudian dikirimkan ke Sistem untuk divalidasi. Jika data yang dimasukkan valid, sistem akan mengirimkan informasi tersebut ke MySQL untuk menyimpan data pengguna. Setelah data disimpan, sistem akan mengarahkan pengguna ke Login Page untuk login dengan akun yang baru didaftarkan.

Namun, jika data yang dimasukkan tidak valid (misalnya, email sudah terdaftar atau password tidak sesuai dengan kriteria), sistem akan menerima respons dari MySQL dan menampilkan pesan kesalahan kepada User Satker. Pengguna kemudian diminta untuk memperbaiki data yang salah dan mengisi form registrasi kembali dengan data yang benar. Dengan demikian, diagram ini menggambarkan alur registrasi pengguna yang melibatkan validasi data, penyimpanan di database, dan penanganan kesalahan ketika data tidak valid.

1. Sequence diagram Proses persetujuan dan penolakan

A diagram of a diagram

AI-generated content may be incorrect.

Berdasarkan gambar, proses persetujuan dan penolakan pengajuan/order. Proses dimulai ketika User Satker mengisi form pengajuan dan submit permohonan mereka. Data pengajuan ini kemudian diteruskan oleh Sistem ke Admin untuk dilakukan verifikasi dan penilaian. Setelah menerima pengajuan, Admin akan melakukan perhitungan skor prioritas berdasarkan kriteria yang sudah ditentukan dalam sistem. Jika pengajuan valid dan disetujui oleh Admin, maka Admin akan menyetujui pengajuan tersebut dan mengirimkan pengajuan ke Super Admin untuk diproses lebih lanjut. Setelah itu, Super Admin akan memproses pengajuan tersebut.

Namun, jika pengajuan ditolak oleh Admin (tidak memenuhi kriteria), Admin akan menolak pengajuan tersebut dan Sistem akan memberikan pesan kepada User Satker bahwa pengajuan mereka ditolak. Proses ini menggambarkan alur persetujuan dan penolakan pengajuan yang melibatkan User Satker, Admin, Sistem, dan Super Admin dengan langkah-langkah validasi dan keputusan berdasarkan prioritas pengajuan.

1. Entity relational Diagram

.

# **DAFTAR PUSTAKA**

mufadhol, mufadhol. (2017). *The Phenomenon of Research and Development Method in Research of Software Engineering*. https://doi.org/10.31219/osf.io/rj4ze

Noerdin, N. S. (2011). Penerapan Sistem Informasi dan Masalah Sosio-Teknis yang Ditimbulkannya. *Jurnal SIFO Mikroskil*, *12*(2), 105–114. https://doi.org/10.55601/jsm.v12i2.38

Putri, F. R., & Suharso, A. (2023). SYSTEMATIC LITERATURE REVIEW PENGGUNAAN METODOLOGI PENGEMBANGAN SISTEM INFORMASI. *INFOTECH Journal*, *9*(2), 377–382. https://doi.org/10.31949/infotech.v9i2.6270

Riani Johan, J., Iriani, T., & Maulana, A. (n.d.). Penerapan Model Four-D dalam Pengembangan Media Video Keterampilan Mengajar Kelompok Kecil dan Perorangan. In *Jurnal Pendidikan West Science* (Vol. 01, Issue 06). Juni.

Rouf, A. (2019). Pembuatan Aplikasi Manajemen Organisasi Swara (AMORA) Berbasis Desktop. *Technomedia Journal*, *4*(1), 1–14. https://doi.org/10.33050/tmj.v4i1.871

Ugochukwu Francis Ikwuanusi, Okeoma Onunka, Samuel Jesupelumi Owoade, & Abel Uzoka. (2024). Digital transformation in public sector services: Enhancing productivity and accountability through scalable software solutions. *International Journal of Applied Research in Social Sciences*, *6*(11), 2744–2774. https://doi.org/10.51594/ijarss.v6i11.1724

Voutama, A. (2022). Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. *Komputika : Jurnal Sistem Komputer*, *11*(1), 102–111. https://doi.org/10.34010/komputika.v11i1.4677

**LAMPIRAN**

Lampiran 1 Daftar Riwayat Hidup

**A person in a white shirt

AI-generated content may be incorrect.**

Lampiran 2 Lembar Evaluasi**A document with a signature

AI-generated content may be incorrect.**

Lampiran 3 Indeks Kemiripan ( Turnitin)

**A screenshot of a computer

AI-generated content may be incorrect.**