

**TUGAS PENGANTI UTS
SISTEM INFORMASI AKUNTANSI
APLIKASI SISTEM LAYANAN RUMAH SAKIT**



Dosen pengampu:

Dr. Totok Dewayanto, S.E., M.Si., Akt.

Disusun oleh:

Dandy Rizkiansah (12030124130135)

**PRODI S-1 AKUNTANSI
FAKULTAS EKONOMIKA DAN BISNIS
UNIVERSITAS DIPONEGORO**

Laporan Proyek Akhir: MHO - Sistem Agen Operasional Rumah Sakit

1. Pendahuluan: Konteks dan Tantangan Sistem Manajemen Rumah Sakit

1.1. Konteks Strategis Digitalisasi Rumah Sakit

Sistem Informasi Manajemen Rumah Sakit (SIMRS), atau yang dikenal secara global sebagai *Hospital Information Management Systems (HIMS)*, berfungsi sebagai infrastruktur data transaksional dan operasional bagi institusi pelayanan kesehatan modern. Sistem ini mengintegrasikan berbagai data dan proses dalam satu ekosistem terpadu. Ekosistem ini mencakup serangkaian komponen vital, mulai dari *Electronic Medical Records (EMR/EHR)* sebagai repositori data klinis, *Laboratory Information Systems (LIS)* untuk manajemen data laboratorium, *Radiology Information Systems (RIS)* yang mengelola data pencitraan medis, hingga modul manajemen farmasi, sistem penagihan (*billing*), dan penjadwalan pasien. Kehadirannya tidak hanya mengotomatisasi tugas administratif, tetapi juga menjadi fondasi bagi pengambilan keputusan berbasis data untuk meningkatkan kualitas layanan.

1.2. Mandat Regulasi di Indonesia

Di Indonesia, penyelenggaraan SIMRS bukan lagi pilihan, melainkan sebuah kewajiban hukum. Sesuai amanat Pasal 52 ayat (1) Undang-Undang Nomor 44 Tahun 2009 tentang Rumah Sakit, setiap rumah sakit diwajibkan untuk melakukan pencatatan dan pelaporan seluruh kegiatan penyelenggaraannya dalam bentuk sistem informasi manajemen rumah sakit. Ketentuan ini dipertegas lebih lanjut melalui Peraturan Menteri Kesehatan (Permenkes) Nomor 82 Tahun 2013, yang bertujuan untuk mendorong peningkatan efisiensi, efektivitas, dan mutu pelayanan kesehatan bagi masyarakat.

1.3. Tantangan Sistem Tradisional dan Era Kecerdasan Buatan (AI)

Sistem manajemen rumah sakit tradisional menghadapi berbagai tantangan fundamental, termasuk *Manual Record-Keeping* yang rentan kesalahan, *Inefficient Appointment Scheduling* yang menyebabkan waktu tunggu pasien yang panjang, *Lack of Real-Time Data* yang menghambat pengambilan keputusan klinis, dan *Inventory Management Issues* yang berisiko pada ketersediaan pasokan medis. Keterisolasian data dalam silo-silo departemen (*data silos*) semakin memperburuk masalah ini.

Seiring perkembangan teknologi, adopsi Kecerdasan Buatan (AI) menjanjikan solusi, namun justru melahirkan tantangan baru. Berdasarkan temuan penelitian sistematis, tantangan ini

mencakup isu **teknis** (kualitas data dan integrasi), **adopsi oleh tenaga medis** (kepercayaan dan pemahaman), serta masalah **keandalan dan validitas** (*reliability and validity*) dari sistem AI itu sendiri. Dengan demikian, sistem tradisional menciptakan tantangan data dan efisiensi, sementara solusi AI yang naif justru melahirkan tantangan baru terkait kepercayaan dan integrasi. Proyek MHO dirancang untuk mengatasi *kedua* lapisan tantangan ini secara simultan melalui sebuah pendekatan arsitektur yang inovatif.

2. Solusi yang Diusulkan: Sistem Agen Operasional Rumah Sakit (MHO)

2.1. Visi dan Filosofi Inti Proyek MHO

Sistem Agen Operasional Rumah Sakit (MHO) diposisikan bukan sekadar sebagai SIMRS generasi baru, melainkan sebagai sebuah **prototipe arsitektur** untuk sistem kesehatan cerdas di masa depan. Proyek ini merupakan demonstrasi kelayakan sebuah paradigma baru yang disebut *agentic workflow*. Sebagai sebuah aplikasi web progresif (*Progressive Web App - PWA*), MHO dirancang untuk mengotomatisasi dan mengelola alur kerja rumah sakit dengan memanfaatkan *Generative AI*. Inovasi utamanya bukanlah pada aplikasi itu sendiri, melainkan pada pola arsitektur yang ditawarkannya: sebuah alternatif yang terukur, transparan, dan dapat diskalakan dibandingkan integrasi AI monolitik konvensional.

2.2. Konsep Kunci: "Agentic Workflow"

Inti dari inovasi MHO terletak pada konsep "**Agentic Workflow**". Arsitektur ini, yang dianalogikan dengan model *Hub-and-Spoke*, secara fundamental berbeda dari sistem AI monolitik. Di pusatnya terdapat **Orchestrator**, yang berfungsi sebagai **mesin pengenalan niat** (*intent-recognition engine*). Orchestrator menerima permintaan pengguna dalam bahasa alami dan, alih-alih memprosesnya sendiri, mendelegasikan tugas kepada serangkaian **agen spesialis** (*Sub-Agents*). Setiap sub-agen merupakan **layanan mikro** (*microservice*) **spesifik domain dengan logika dan tools yang terkapsulasi**. Pendekatan ini memecah masalah kompleks menjadi komponen-komponen yang dapat dikelola, meningkatkan akurasi, skalabilitas, dan keandalan.

2.3. Tujuan Proyek

Pengembangan Sistem MHO didasarkan pada tiga tujuan utama yang saling terkait untuk meningkatkan efektivitas operasional rumah sakit:

- **Mengurangi Beban Administrasi Manual:** Mengotomatisasi tugas-tugas repetitif seperti pendaftaran pasien, penjadwalan, dan pembuatan dokumen.
- **Mempercepat Proses Layanan Pasien:** Menyederhanakan alur kerja mulai dari admisi hingga penagihan, yang pada akhirnya mengurangi waktu tunggu pasien.
- **Memastikan Kepatuhan Terhadap Standar Medis:** Merancang sistem dengan keamanan dan privasi sebagai fondasi untuk memastikan kepatuhan terhadap standar

regulasi data kesehatan seperti HIPAA (*Health Insurance Portability and Accountability Act*).

Pencapaian tujuan ini dimungkinkan melalui implementasi arsitektur teknis yang terdefinisi secara presisi, yang akan dibahas pada bagian selanjutnya.

3. Desain Arsitektur dan Rasional Teknologi

3.1. Arsitektur Mikro-servis Berbasis AI

Arsitektur MHO mengadopsi pola *Hub-and-Spoke*, yang pada dasarnya adalah implementasi modern dari arsitektur mikro-servis berbasis AI. Pendekatan ini merupakan jawaban langsung terhadap tantangan integrasi AI yang diidentifikasi sebelumnya. Dengan memecah logika menjadi agen-agen independen, arsitektur ini mengatasi **tantangan teknis** dalam mengintegrasikan AI ke infrastruktur yang ada. Lebih jauh lagi, pendekatan ini menjawab kekhawatiran mengenai **keandalan dan validitas** (*reliability and validity*). Jika sistem AI monolitik sering beroperasi sebagai 'kotak hitam' (*black box*), sistem agen dengan peran spesifik secara inheren lebih dapat diaudit, dikelola, dan diandalkan.

3.2. Tumpukan Teknologi (Technology Stack)

Pemilihan teknologi untuk MHO didasarkan pada kebutuhan akan performa *real-time*, aksesibilitas tinggi, dan skalabilitas. Berikut adalah rincian tumpukan teknologi yang digunakan:

Komponen	Teknologi	Rasional Pemilihan
Front-End	React (TypeScript) dengan Tailwind CSS	Membangun antarmuka yang responsif (<i>Mobile-First</i>) dan dapat diinstal sebagai <i>Progressive Web App (PWA)</i> untuk aksesibilitas tinggi di berbagai perangkat tanpa memerlukan instalasi dari <i>app store</i> .
AI Engine	Google Gemini API (Model: gemini-1.5-flash)	Dipilih karena latensi yang sangat rendah, yang krusial untuk aplikasi interaktif dan <i>real-time</i> , serta menawarkan kemampuan <i>reasoning</i> yang kuat untuk menganalisis niat pengguna secara akurat.

Deployment	Netlify	Platform ini memungkinkan konfigurasi <i>Continuous Deployment (CD)</i> yang efisien, sehingga setiap pembaruan sistem dapat dirilis dengan cepat dan otomatis, memastikan sistem selalu dalam kondisi terkini.
-------------------	---------	---

3.3. Mekanisme "Agentic Workflow"

Alur kerja sistem MHO dimulai ketika pengguna memasukkan permintaan dalam bahasa alami. Pesan ini diterima oleh **Orchestrator Agent**, yang menganalisis niat (*intent*) pengguna. Sebagai contoh, permintaan "Jadwalkan konsultasi dengan Dr. Budi untuk besok pagi" diubah menjadi *payload* JSON terstruktur yang berisi niat (`intent: 'schedule_appointment'`), entitas (`entities: {doctor: 'Dr. Budi', time: 'besok pagi'}`), dan metadata lainnya. *Payload* ini kemudian dirutekan ke agen yang sesuai, yaitu **AppointmentSchedulingAgent**. Penting dicatat, Orchestrator hanya bertindak sebagai perute cerdas dan tidak memproses data medis sensitif secara mendalam, sehingga membatasi permukaan serangan dan menjaga privasi data.

3.4. Profil Agen-Agen Spesialis (Sub-Agents)

Setiap sub-agen dirancang untuk menangani fungsi operasional yang spesifik dengan menggunakan *tools* yang telah ditentukan.

PatientAdmissionAgent (Agen Admisi) Agen ini bertanggung jawab atas semua proses yang terkait dengan data administratif pasien.

- **Tugas Utama:**
 - Mendaftarkan pasien baru ke dalam sistem.
 - Memperbarui informasi dan rekam medis pasien yang sudah ada.
 - Mengelola prosedur penerimaan (admisi) dan pemulangan (*discharge*).
- **Tools:**
 - **Generate Document:** Untuk membuat formulir pendaftaran atau catatan pemulangan.

AppointmentSchedulingAgent (Agen Penjadwalan) Agen ini mengelola seluruh siklus hidup janji temu antara pasien dan tenaga medis.

- **Tugas Utama:**
 - Memesan janji temu baru sesuai permintaan.
 - Menjadwalkan ulang atau membatalkan janji temu.
 - Memeriksa ketersediaan jadwal dokter atau departemen.
- **Tools:**

- **Google Search:** Untuk mencari informasi tambahan tentang dokter atau spesialisasi.

PharmacyManagementAgent (Agen Farmasi) Agen ini berfokus pada manajemen obat dan resep.

- **Tugas Utama:**
 - Memeriksa ketersediaan dan stok obat di farmasi.
 - Memberikan informasi dasar mengenai suatu obat.
 - Menghasilkan dokumen resep digital berdasarkan instruksi.
- **Tools:**
 - **Google Search:** Untuk mencari detail obat, seperti interaksi atau efek samping.
 - **Generate Document:** Untuk membuat dokumen resep dalam format standar.

BillingAndFinanceAgent (Agen Keuangan/RCM) Agen ini mengelola semua aspek keuangan pasien, yang sering disebut *Revenue Cycle Management* (RCM).

- **Tugas Utama:**
 - Membuat tagihan atau faktur untuk layanan yang diterima.
 - Membantu proses pengajuan klaim asuransi.
 - Menjawab pertanyaan pasien terkait rincian tagihan.
 - Memberikan perkiraan biaya untuk prosedur medis.
- **Tools:**
 - **Generate Document:** Untuk membuat faktur, laporan keuangan, atau rincian biaya.

Arsitektur yang terstruktur ini menjadi dasar bagi fitur-fitur unggulan yang dapat dirasakan langsung oleh pengguna.

4. Fitur Unggulan dan Pengalaman Pengguna (UX)

4.1. Transparansi dan Keterlacakan Proses AI

Di lingkungan medis, kepercayaan terhadap sistem teknologi adalah hal mutlak. Menyadari hal ini, MHO dirancang dengan prinsip transparansi sebagai inti dari pengalaman pengguna (UX). Alih-alih beroperasi sebagai "kotak hitam" (*black box*), setiap tindakan yang diambil oleh sistem AI dibuat terlihat dan dapat dilacak. Fitur-fitur MHO secara khusus dirancang untuk memberikan visibilitas penuh kepada pengguna, memungkinkan mereka memahami "proses berpikir" sistem dan memverifikasi setiap langkahnya.

4.2. Analisis Fitur-Fitur Kunci

Smart Orchestration UI Antarmuka ini secara visual merepresentasikan alur kerja agen. Ketika pengguna mengajukan permintaan, UI akan menyorot agen mana yang sedang aktif memproses tugas. Visualisasi *real-time* ini mendemistifikasi cara kerja AI dan secara signifikan meningkatkan kepercayaan pengguna, karena mereka dapat melihat dengan jelas bagaimana sistem mendelegasikan dan menyelesaikan tugas.

Audit Log (Immutable) Pada bagian *sidebar* aplikasi, MHO menampilkan jejak audit (*audit trail*) dari setiap interaksi yang bersifat *immutable* (tidak dapat diubah). Log ini mencatat semua aktivitas secara kronologis, seperti "Permintaan diterima oleh Orchestrator" hingga "Dokumen pendaftaran berhasil dibuat." Fitur ini krusial untuk akuntabilitas medis dan transparansi operasional, menyediakan catatan permanen yang dapat diverifikasi kapan saja.

Document Generation Sistem MHO mampu mengubah data terstruktur dalam format JSON menjadi dokumen visual yang profesional, seperti resep atau faktur. Setiap dokumen yang dihasilkan secara otomatis menyertakan **Compliance Note**, yang berfungsi sebagai penanda bahwa dokumen tersebut dibuat oleh sistem sesuai dengan protokol yang ditentukan, memberikan jaminan kepatuhan.

Grounding (Verifikasi Sumber) Ketika seorang agen menggunakan **Google Search** untuk mencari informasi eksternal, sistem tidak hanya menyajikan jawabannya, tetapi juga secara otomatis menampilkan tautan ke sumber asli informasi tersebut. Fitur *grounding* ini sangat penting untuk menjawab tantangan keandalan dan validitas AI. Dengan memungkinkan verifikasi langsung oleh manusia (*human-in-the-loop*), MHO memastikan tenaga medis dapat dengan mudah memeriksa keakuratan informasi sebelum mengambil keputusan klinis.

Selain fungsionalitas, pilar utama lain dari desain MHO adalah memastikan semua proses berjalan aman dan patuh terhadap regulasi.

5. Analisis Keamanan dan Kepatuhan Regulasi

5.1. Membangun Sistem yang Aman dan Patuh Sejak Awal

Dalam pengembangan MHO, keamanan dan kepatuhan regulasi seperti HIPAA (dan secara implisit, peraturan data kesehatan di Indonesia) bukanlah fitur tambahan, melainkan merupakan fondasi dari seluruh arsitektur. Pendekatan *security-by-design* ini memastikan bahwa setiap komponen dan alur data telah dirancang sejak awal untuk melindungi informasi kesehatan pribadi (*Protected Health Information - PHI*).

5.2. Implementasi "HIPAA Safeguards"

Untuk menjamin keamanan data, MHO mengimplementasikan beberapa mekanisme perlindungan teknis yang terinspirasi dari prinsip-prinsip HIPAA.

Lazy Initialization Mekanisme ini memastikan bahwa kunci API dan koneksi sensitif hanya diinisialisasi saat benar-benar dibutuhkan. Manfaat utamanya adalah mengurangi risiko kebocoran kredensial saat aplikasi dimuat, karena data sensitif hanya ada di memori untuk waktu yang sangat singkat.

Role Masking Fitur ini mengisolasi data dan fungsionalitas antar agen. Dengan membatasi akses setiap agen hanya pada data yang relevan dengan tugasnya, MHO menerapkan **Prinsip Hak Akses Minimal (*Principle of Least Privilege - PoLP*)** pada tingkat arsitektural. Ini secara efektif mencegah pergerakan lateral (*lateral movement*) bahkan jika satu agen berhasil disusupi, sehingga secara signifikan mengurangi risiko paparan data.

Graceful Degradation (Penanganan Error) Sistem MHO dilengkapi dengan mekanisme *fallback* yang aman untuk menangani kegagalan API atau jaringan. Ini penting tidak hanya untuk menjaga pengalaman pengguna, tetapi juga untuk mencegah **kegagalan berjenjang (*failure cascades*)** yang katastrofik—sebuah kerentanan umum dalam sistem monolitik yang terintegrasi secara ketat—dan mencegah kebocoran data yang tidak disengaja saat terjadi kondisi *error*.

5.3. Kesiapan Interoperabilitas

Peraturan Menteri Kesehatan (Permenkes) No. 82 Tahun 2013 Pasal 5 secara eksplisit mewajibkan SIMRS memiliki kemampuan interoperabilitas. Arsitektur MHO yang berbasis agen dan API secara inheren dirancang untuk memenuhi mandat ini secara superior dibandingkan sistem monolitik lawas yang sulit diintegrasikan. Sebagaimana dicatat oleh pakar industri, standar modern seperti FHIR (*Fast Healthcare Interoperability Resources*) bersifat "**API-driven**" dan dirancang untuk "**modern digital health apps.**" Filosofi ini tertanam dalam desain MHO. Dengan demikian, kepatuhan terhadap Pasal 5 menjadi hasil alami dari desain arsitekturnya, bukan sebuah fitur tambahan (*bolt-on feature*). Hal ini memastikan MHO siap untuk diintegrasikan dengan sistem pemerintah lainnya (misalnya, SIRS, INACBG's) sebagai bagian dari ekosistem kesehatan nasional.

6. Analisis Komparatif: Posisi MHO dalam Ekosistem HIMS/SIMRS

6.1. Lanskap Solusi Manajemen Rumah Sakit Saat Ini

Pasar *Hospital Information Management Systems (HIMS)* atau *Enterprise Resource Planning (ERP)* untuk rumah sakit saat ini sangat beragam, mulai dari sistem *open-source* seperti ERPNext hingga platform *cloud* terintegrasi seperti Microsoft Cloud for Healthcare. Beberapa vendor HIMS konvensional terkemuka, seperti MEDITECH Expanse, juga telah mulai mengadopsi AI untuk meningkatkan fungsionalitas sistem mereka. Lanskap ini menunjukkan pergeseran industri menuju sistem yang lebih cerdas dan terintegrasi.

6.2. Tabel Perbandingan MHO vs. HIMS Konvensional

Untuk memahami posisi unik MHO, berikut adalah perbandingan antara pendekatan MHO dengan HIMS/SIMRS konvensional:

Aspek	HIMS/SIMRS Konvensional	Sistem MHO (Agentic AI)
Inti Pemrosesan	Berbasis alur kerja (<i>workflow</i>) yang telah ditentukan atau modul perangkat lunak statis.	Berbasis <i>Generative AI</i> dengan alur kerja agen yang dinamis dan adaptif.
Arsitektur	Seringkali monolitik atau berbasis modul yang terintegrasi secara ketat.	Arsitektur mikro-servis berbasis AI (<i>Hub-and-Spoke</i>) yang fleksibel dan terisolasi.
Interaksi Pengguna	Melalui formulir dan antarmuka grafis standar.	Melalui input bahasa alami (<i>natural language</i>) yang diproses oleh Orchestrator.
Transparansi & Audit	Log audit seringkali bersifat teknis dan sulit diakses oleh non-spesialis.	Jejak audit <i>real-time</i> yang <i>immutable</i> dan mudah dibaca, ditampilkan langsung di UI.
Aksesibilitas	Umumnya aplikasi desktop atau web-based tradisional.	<i>Progressive Web App (PWA)</i> yang <i>mobile-first</i> dan dapat diinstal di perangkat apa pun.

6.3. Diferensiator dan Keunggulan Kompetitif MHO

Berdasarkan perbandingan di atas, keunggulan kompetitif utama dari Sistem MHO dapat disintesis sebagai berikut:

- Model Interaksi Fundamental:** Sementara solusi seperti MEDITECH Expanse mengintegrasikan AI ke dalam alur kerja EHR yang ada, MHO merumuskan ulang interaksi fundamentalnya melalui bahasa alami. Ini merupakan sebuah lompatan konseptual yang membedakannya dari sekadar penambahan fitur cerdas pada paradigma lama.

- **Fleksibilitas Alur Kerja Agen:** Pendekatan *agentic workflow* memungkinkan pemrosesan tugas yang dinamis dan adaptif berdasarkan konteks permintaan pengguna, tidak seperti HIMS konvensional dengan alur kerja yang kaku.
- **Transparansi Radikal:** Fitur seperti *Smart Orchestration UI* dan *Immutable Audit Log* memberikan tingkat transparansi operasional yang belum pernah ada sebelumnya pada sistem rumah sakit, membangun kepercayaan dan akuntabilitas.
- **Arsitektur Modern dan Mudah Diakses:** Sebagai PWA, MHO menawarkan aksesibilitas superior dan biaya pemeliharaan yang lebih rendah dibandingkan aplikasi desktop atau web lawas.

7. Kesimpulan dan Arah Pengembangan Masa Depan

7.1. Ringkasan Kontribusi Proyek

Proyek MHO telah berhasil menyajikan lebih dari sekadar aplikasi; ia merupakan **demonstrasi kelayakan sebuah paradigma baru** dalam arsitektur sistem informasi kesehatan. Sebagai sebuah **prototipe arsitektur** SIMRS cerdas, MHO membuktikan bahwa konsep "*Agentic Workflow*" berbasis *Generative AI* dapat secara efektif menjawab tantangan sistem tradisional sekaligus memenuhi mandat regulasi di Indonesia. Sistem ini mengatasi masalah teknis, adopsi, dan keandalan yang melekat pada implementasi AI di sektor kesehatan, dengan menawarkan model yang transparan, aman, dan dapat diskalakan.

7.2. Arah Pengembangan di Masa Depan

Potensi MHO untuk terus berkembang sangat besar. Berdasarkan tren teknologi di industri kesehatan, beberapa arah pengembangan logis di masa depan meliputi:

- **Integrasi Telemedicine:** Memperluas fungsionalitas untuk memungkinkan konsultasi virtual antara pasien dan dokter langsung melalui platform MHO.
- **Analitik Prediktif:** Menggunakan data operasional untuk membangun model yang dapat memprediksi tingkat hunian tempat tidur (*patient admission rates*) dan mengantisipasi kebutuhan inventaris secara lebih akurat, memungkinkan manajemen sumber daya yang proaktif.
- **Penguatan Keamanan dengan Blockchain:** Mengeksplorasi penggunaan teknologi *blockchain* untuk menciptakan rekam medis elektronik (EHR) yang lebih aman dan tidak dapat diubah (*immutable*), sehingga meningkatkan integritas data pasien.
- **Perluasan Portal Pasien:** Mengembangkan fitur portal yang lebih komprehensif, memungkinkan pasien mengakses riwayat medis dan informasi tagihan mereka secara mandiri dan aman.

7.3. Penutup

Sistem MHO menawarkan sebuah visi masa depan untuk manajemen rumah sakit yang lebih cerdas, adaptif, dan berpusat pada pengguna. Dengan menyeimbangkan inovasi teknologi,

efisiensi operasional, dan kepatuhan regulasi, MHO berpotensi menjadi model acuan bagi transformasi digital di sektor kesehatan Indonesia, membuktikan bahwa teknologi dapat menjadi mitra strategis dalam mewujudkan pelayanan kesehatan yang lebih baik.

Folder Github dan Modul Aplikasi dapat diakses di link Repository GitHub berikut:

<https://github.com/dandywork28-prog/Sistem-Rumah-Sakit-Dandy>