

# Analisis Laporan Analisis dan Desain: Aplikasi "Pintar Rumah Sakit" Berbasis Sistem Agen AI

---

Disusun oleh = Desta Bagaskara/12030124120055/SIA/Kelas E

Dosen Pembimbing = Dr. Totok Dewayanto, S.E., M.Si., Akt.

---

## 1.0 Pendahuluan: Modernisasi Manajemen Rumah Sakit melalui Kecerdasan Buatan

Operasional rumah sakit modern dihadapkan pada tantangan fundamental terkait efisiensi alur kerja dan integritas data. Sistem informasi warisan sering kali menciptakan silo data dan hambatan proses yang sekuensial, di mana setiap proses menuntut kecepatan, ketepatan, dan kerahasiaan tingkat tinggi. Dalam konteks ini, sistem agen berbasis Kecerdasan Buatan (AI) hadir sebagai solusi strategis untuk mentransformasi tantangan tersebut menjadi keunggulan operasional. Dengan mengalokasikan tugas-tugas spesifik kepada agen AI yang terlatih, arsitektur ini secara efektif membongkar silo tersebut, memungkinkan pemrosesan permintaan secara paralel dan mengurangi beban kerja administratif secara signifikan.

Laporan ini bertujuan untuk menyajikan analisis komprehensif mengenai arsitektur, fungsionalitas, dan panduan implementasi konseptual dari aplikasi "Pintar Rumah Sakit". Aplikasi ini dirancang sebagai sebuah sistem agen AI terpadu yang mampu mengelola berbagai aspek operasional rumah sakit secara cerdas dan efisien.

Struktur laporan ini akan memandu pembaca melalui analisis mendalam mengenai arsitektur sistem inti, dekomposisi fungsional setiap agen, panduan praktis untuk implementasi, dan analisis potensi pengembangan di masa depan. Melalui pembahasan ini, kita akan melihat bagaimana sebuah konsep arsitektur AI dapat diwujudkan menjadi solusi praktis untuk industri kesehatan, yang berpuncak pada pembahasan arsitektur sistem yang menjadi fondasinya.

## 2.0 Arsitektur Sistem Inti: Model Delegasi Berbasis Agen

Fondasi dari setiap sistem informasi yang andal dan dapat diskalakan adalah arsitektur yang solid. Untuk aplikasi "Pintar Rumah Sakit", model yang diusulkan adalah sistem agen terpadu yang beroperasi berdasarkan prinsip delegasi tugas yang cerdas. Desain ini dirancang untuk mencapai efisiensi maksimum dengan memastikan bahwa setiap permintaan atau tugas dialokasikan secara otomatis ke agen spesialis yang paling kompeten.

Konsep utama di balik arsitektur ini adalah pembagian kerja yang tegas. Daripada menggunakan satu agen monolitik, sistem ini berfungsi sebagai orkestrator alur kerja yang mendistribusikan tugas ke beberapa agen yang lebih kecil dan terfokus. Pendekatan ini secara signifikan meningkatkan kecepatan respons dan akurasi data, karena setiap agen memiliki pengetahuan dan alat yang dioptimalkan untuk domain tugasnya sendiri.

Komponen sentral yang berfungsi sebagai lapisan perutean cerdas (*intelligent routing layer*) adalah "**Penavigasi Pintar Rumah Sakit**" (*Hospital System Navigator*). Perannya bukan untuk menyelesaikan masalah, melainkan untuk memahami maksud inti (*core intent*) dari permintaan pengguna dan kemudian meneruskan permintaan tersebut ke sub-agen yang tepat. Terdapat satu instruksi kritis yang menjadi inti dari arsitektur ini: *agen utama tidak boleh*

*menjawab permintaan secara langsung, melainkan harus selalu mendelegasikan tugas.* Aturan fundamental ini memberlakukan *separation of concerns* yang ketat, mencegah agen navigator menjadi *monolithic bottleneck* dan memastikan setiap tugas dieksekusi oleh spesialis sejati, yang pada gilirannya memaksimalkan akurasi, keamanan, dan kemudahan pemeliharaan sistem.

Dengan fondasi arsitektur ini, sistem "Pintar Rumah Sakit" mampu beroperasi secara efisien. Bagian selanjutnya akan merinci setiap komponen, dimulai dari agen utama itu sendiri.

### 3.0 Dekomposisi Fungsional: Analisis Agen Utama dan Sub-Agen Spesialis

Kekuatan sebenarnya dari sistem "Pintar Rumah Sakit" terletak pada dekomposisi fungsionalnya menjadi agen-agen spesialis yang bekerja secara sinergis. Dengan memecah fungsionalitas kompleks menjadi unit-unit yang lebih mudah dikelola, sistem ini mencapai tingkat akurasi dan efisiensi yang tinggi. Bagian ini akan menganalisis peran, tanggung jawab, dan kapabilitas setiap agen baik agen utama maupun sub-agen spesialis secara mendalam.

#### 3.1 Agen Utama: Penavigasi Pintar Rumah Sakit

Agen ini adalah titik masuk tunggal untuk semua interaksi pengguna dan bertindak sebagai koordinator utama dalam sistem.

- **Nama Sistem:** Penavigasi Pintar Rumah Sakit
- **Deskripsi:** Merupakan agen AI komprehensif untuk sistem rumah sakit yang mampu menavigasi informasi pasien, janji temu, rekam medis, dan penagihan dengan mendelegasikannya ke sub-agen spesialis.
- **Peran Utama:** Bertindak sebagai navigator pusat yang bertugas mengidentifikasi *core intent* dari setiap permintaan pengguna. Setelah maksud teridentifikasi, tugas utamanya adalah mengalokasikan seluruh konteks permintaan tersebut ke sub-agen yang paling relevan.
- **Instruksi Kritis:** Terdapat aturan fundamental yang tidak dapat dilanggar: agen ini **TIDAK PERNAH** menjawab permintaan pengguna secara langsung dan **SELALU** mendelegasikan tugas ke salah satu sub-agen spesialisnya.

#### 3.2 Empat Sub-Agen Spesialis

Setelah agen utama mengidentifikasi maksud pengguna, tugas dieksekusi oleh salah satu dari empat sub-agen berikut. Setiap sub-agen dirancang dengan fungsi, alat, dan ekspektasi output yang spesifik.

Nama Sub-Agen	Fungsi dan Tugas Utama	Alat Kunci yang Digunakan	Hasil yang Diharapkan (Output)
Agen Informasi Pasien	Mengelola pendaftaran pasien baru, memperbarui detail pribadi, mengambil informasi umum, dan memeriksa status pasien.	Generate Document, Google Search	Memberikan informasi yang diminta atau konfirmasi pembaruan data. Jika diminta, menghasilkan dokumen formulir pendaftaran yang relevan.

<b>Penjadwal Janji Temu</b>	Menangani semua tugas terkait janji temu, termasuk menjadwalkan, menjadwalkan ulang, dan membatalkan. Mengonfirmasi detail penting seperti dokter, tanggal, dan waktu.	Google Search	Status janji temu yang terkonfirmasi (dijadwalkan, dijadwal ulang, atau dibatalkan) dengan rincian eksplisit (dokter, tanggal, waktu, nama pasien). Jika informasi kurang, agen akan meminta data tambahan.
<b>Agen Rekam Medis</b>	Memproses permintaan untuk mengambil rekam medis, hasil tes, diagnosis, dan riwayat perawatan pasien. <b>Menjaga kerahasiaan adalah prioritas utama.</b>	Generate Document	Dokumen rekam medis yang aman, akurat, dan lengkap dalam format terstruktur yang diminta (misalnya, PDF, DOCX), dengan kerahasiaan pasien yang terjaga setiap saat.
<b>Agen Penagihan dan Asuransi</b>	Menjawab pertanyaan terkait faktur penagihan, cakupan asuransi, dan opsi pembayaran. Memberikan penjelasan yang mudah dipahami oleh pasien.	Google Search, Generate Document	Respons komprehensif dan empatik yang menjelaskan tagihan secara jelas, mengklarifikasi manfaat asuransi, dan memberikan informasi mengenai opsi pembayaran atau bantuan keuangan.

Setelah memahami arsitektur dan peran masing-masing komponen, langkah berikutnya adalah mempelajari bagaimana sistem konseptual ini dapat dibangun dan digunakan dalam aplikasi nyata.

## 4.0 Panduan Implementasi dan Penggunaan Aplikasi

Mewujudkan arsitektur konseptual "Pintar Rumah Sakit" menjadi aplikasi fungsional dapat dicapai menggunakan platform modern seperti Google AI Studio. Implementasi ini berpusat pada perancangan *prompt* yang presisi untuk setiap agen. Instruksi ini bukan sekadar deskripsi, melainkan *prompt* literal yang dimasukkan ke dalam model AI untuk memprogram perilakunya, mendefinisikan peran, kapabilitas, dan batasannya.

### 4.1 Proses Pembuatan Aplikasi di Google AI Studio

Setiap agen, baik utama maupun sub-agen, dikonfigurasi sebagai entitas terpisah di dalam platform. Prosesnya melibatkan pendefinisian Name, Description, dan Instructions, di mana Role dan Output Expectations ditetapkan secara eksplisit.

#### Instruksi untuk Agen Utama ("Hospital System Navigator")

Prompt ini dirancang untuk memastikan agen utama fokus hanya pada analisis dan delegasi, sesuai dengan arsitektur yang telah ditetapkan.

**Role** You are an expert Hospital System Navigator. Your primary role is to act as a central navigator for all hospital system inquiries. Carefully analyze the user's request to identify the core intent, and delegate the task to the most appropriate sub-agent: Patient Information Agent for patient registration or details, Appointment Scheduler for scheduling or modifying

appointments, Medical Records Agent for retrieving health records, or Billing And Insurance Agent for financial queries.

**Output Expectations** The agent's output should be a clear delegation of the user's request to the most appropriate sub-agent, ensuring all necessary context is passed.

1. Accurately identify the core intent of the user's request.
2. Select the single most relevant sub-agent from Patient Information Agent, Appointment Scheduler, Medical Records Agent, or Billing And Insurance Agent.
3. Pass the entire context of the user's request to the chosen sub-agent.
4. Do not attempt to answer the user's request directly; always delegate to a sub-agent.

### **Instruksi untuk Agen Rekam Medis**

Prompt ini memprogram agen untuk menangani data sensitif dengan prioritas utama pada keamanan dan kerahasiaan.

**Role** You are an expert Medical Records Agent. Your role is to process requests for patient medical records, including test results, diagnoses, treatment history, and other health information. You must ensure all information is handled securely and confidentially. You can use Generate Document to provide records in a structured format.

**Output Expectations** The primary goal is to securely and confidentially provide requested patient medical records in a structured format... Confidentiality of patient information must be maintained at all times.

### **Instruksi untuk Penjadwal Janji Temu**

Prompt ini memastikan agen dapat mengelola alur kerja penjadwalan secara lengkap, termasuk menangani informasi yang tidak lengkap.

**Role** You are an expert Appointment Scheduler. Your role is to manage all appointment-related tasks, including scheduling new appointments, modifying existing ones (rescheduling), or canceling appointments. You will ensure to confirm details like doctor, date, time, and patient preferences.

**Output Expectations** The agent's final output must be a clear and confirmed status of the requested appointment task (scheduled, rescheduled, or canceled).

## **4.2 Skenario Penggunaan: Alur Interaksi Pengguna**

Dari sudut pandang pengguna, interaksi dengan aplikasi ini terasa mulus. Di balik layar, arsitektur delegasi bekerja untuk memastikan permintaan ditangani oleh spesialis yang tepat.

Mari kita gunakan studi kasus berdasarkan kueri: *"Saya ingin mendaftar sebagai pasien baru di rumah sakit."*

1. **Input Pengguna:** Pasien memasukkan permintaan pendaftaran melalui antarmuka aplikasi.
2. **Analisis oleh Agen Utama:** "Penavigasi Pintar Rumah Sakit" menerima permintaan ini. Agen tersebut tidak mencoba mendaftarkan pasien, melainkan menganalisis permintaan dan mengidentifikasi *inti maksud* sebagai "pendaftaran pasien".
3. **Delegasi Tugas:** Berdasarkan identifikasi maksud, Agen Utama secara otomatis meneruskan seluruh konteks permintaan ke sub-agent yang paling relevan: **Agen Informasi Pasien**.

4. **Eksekusi oleh Sub-Agen:** "Agen Informasi Pasien" menerima tugas dan memprosesnya, mungkin dengan menggunakan alat Generate Document untuk membuat tautan ke formulir pendaftaran digital.
5. **Output ke Pengguna:** Pengguna menerima respons yang relevan dan dapat ditindaklanjuti langsung dari "Agen Informasi Pasien", seperti tautan ke formulir atau konfirmasi langkah selanjutnya.

Alur kerja yang efisien ini menunjukkan bagaimana arsitektur delegasi memastikan setiap tugas ditangani dengan benar. Selanjutnya, kita akan membahas bagaimana prototipe AI ini dapat dikembangkan menjadi aplikasi skala produksi.

## 5.0 Potensi Pengembangan: Dari Prototipe AI ke Aplikasi Produksi

Sistem agen yang dirancang di Google AI Studio pada dasarnya merupakan *backend* atau mesin logika yang cerdas. Untuk penggunaan di dunia nyata, mesin ini perlu dihubungkan dengan antarmuka pengguna (UI) yang ramah akses. Oleh karena itu, langkah logis berikutnya adalah pengembangan aplikasi *front-end* terpisah, seperti aplikasi web atau seluler, yang berinteraksi dengan sistem agen melalui API (*Application Programming Interface*).

Untuk penerapan skala produksi, sangat penting untuk mengadopsi alur kerja DevOps modern. Kami merekomendasikan pendekatan berikut:

- **Kontrol Versi:** Memanfaatkan **GitHub** untuk manajemen kode sumber *front-end*. Ini memastikan integritas kode, memfasilitasi pengembangan kolaboratif, dan memungkinkan pelacakan perubahan yang terstruktur.
- **Penerapan Berkelanjutan (Continuous Deployment):** Menggunakan layanan seperti **Netlify** untuk mengotomatiskan proses rilis. Setiap kali ada pembaruan kode di GitHub, Netlify dapat secara otomatis membangun dan menyebarkan versi terbaru aplikasi, meminimalkan waktu henti dan memastikan pengguna selalu memiliki akses ke fitur terkini.

Pendekatan ini secara arsitektural krusial; ia memisahkan 'otak' AI (*backend*) dari antarmuka pengguna (*front-end*). Keunggulan ini memungkinkan iterasi cepat pada logika AI tanpa mengganggu pengalaman pengguna sebuah faktor penentu di lanskap AI yang berevolusi pesat.

## 6.0 Kesimpulan

Laporan ini telah menguraikan desain arsitektur dan panduan implementasi untuk aplikasi "Pintar Rumah Sakit", sebuah sistem berbasis agen AI yang dirancang untuk merevolusi manajemen operasional di sektor kesehatan. Temuan utama menunjukkan bahwa kekuatan sistem ini terletak pada model delegasi cerdasnya, yang memastikan **efisiensi** melalui alokasi tugas yang tepat, **akurasi** melalui spesialisasi fungsional, dan **skalabilitas** untuk pengembangan di masa depan.

Dari perspektif Sistem Informasi Akuntansi (SIA), implementasi sistem ini memiliki dampak signifikan. Dengan mengintegrasikan Agen Penagihan dan Asuransi dengan Agen Rekam Medis, sistem ini menciptakan tautan langsung yang dapat diaudit antara layanan yang diberikan dan tagihan yang diterbitkan, secara dramatis mengurangi kesalahan rekonsiliasi dan memperkuat kepatuhan finansial. Kemampuan untuk mengotomatiskan proses ini secara

langsung meningkatkan akurasi dalam penagihan pasien dan klaim asuransi, serta mempercepat siklus pendapatan rumah sakit.

Sebagai penutup, penerapan teknologi agen AI seperti yang diuraikan dalam desain "Pintar Rumah Sakit" bukan lagi sekadar inovasi, melainkan sebuah langkah strategis yang krusial. Bagi institusi kesehatan yang ingin mencapai keunggulan operasional, meningkatkan kepuasan pasien, dan memastikan keberlanjutan finansial dalam lingkungan yang kompleks, adopsi solusi cerdas semacam ini adalah sebuah keharusan.

Link Aplikasi : <https://pintarrumahsakit.netlify.app/>

Link Repositories Github : <https://github.com/Dxsta/Pintar-Rumah-Sakit.git>