

Penerapan Teknologi Full Stack dalam Pengembangan Sistem Formulir Digital Rumah Sakit Menggunakan Metode Waterfall Development

Tia Ifania Nugrahaningtyas¹, Herbert Siregar^{1,*}

¹ Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Bandung, Indonesia

Jalan Dr. Setiabudhi No. 229, Bandung, Jawa Barat, Indonesia

Email: ¹tiaifanian@upi.edu, ^{2,*}herbert@upi.edu,

(* : coressponding author)

Abstrak— Sektor layanan kesehatan saat ini menghadapi tantangan besar dalam pengelolaan data pasien, di mana pencatatan medis secara manual sering kali menyebabkan inefisiensi, resiko kehilangan data, dan lambatnya akses informasi oleh tenaga medis. Masalah utama yang dihadapi oleh banyak rumah sakit adalah belum optimalnya integrasi antara pengumpulan data pasien dengan sistem basis data terpusat, sehingga menghambat proses pelayanan berkelanjutan. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem formulir digital yang terintegrasi di lingkungan rumah sakit guna memfasilitasi pencatatan data medis secara akurat, cepat, dan aman. Solusi yang ditawarkan adalah penerapan teknologi full stack menggunakan *framework* CodeIgniter, bahasa pemrograman PHP, dan DBeaver untuk pengelolaan basis data. Penelitian ini menggunakan metode *Waterfall Development* yang selaras dengan alur kerja tim pengembang. Proses pengerjaan terbagi dalam tahapan iteratif, mulai dari penugasan berdasarkan rancangan antarmuka dari *Project Manager*, pengolahan data dinamis, hingga kolaborasi kode melalui platform GitHub. Setiap pembaruan sistem melalui mekanisme *Pull Request* dan diuji secara teliti oleh tim *Quality Assurance* (PDI) agar memastikan fungsionalitas dan mencegah error. Hasil yang dicapai menunjukkan bahwa pendekatan *full stack* dengan metode *Waterfall* berhasil menghasilkan purwarupa sistem formulir digital yang responsif, meminimalisir *bug* melalui ulasan kode yang ketat, dan mempercepat proses adaptasi perangkat lunak. Kontribusi penelitian ini adalah menyediakan model arsitektur pengembangan sistem informasi kesehatan yang skalabel, yang dapat menjadi referensi bagi institusi medis dalam melakukan transformasi digital pelayanan kesehatan.

Kata Kunci: *Full-stack; Waterfall; CodeIgniter; DBeaver; Kesehatan.*

Abstract— The healthcare sector currently faces significant challenges in patient data management, where manual medical recording often leads to inefficiency, the risk of data loss, and slow information access by medical personnel. The main problem faced by many hospitals is the suboptimal integration between patient data collection and centralized database systems, which hinders continuous care delivery. Therefore, this study aims to design and develop an integrated digital form system in a hospital environment to facilitate accurate, fast, and secure medical data recording. The proposed solution is the application of full-stack technology using the CodeIgniter framework, PHP programming language, and DBeaver for database management. This research employs the Waterfall Development method, which aligns with collaborative team workflows. The development process is divided into iterative stages, starting from task assignments based on interface mockups from the Project Manager, dynamic data processing, to code collaboration via GitHub. Every system update goes through a Pull Request mechanism and is comprehensively tested by the Quality Assurance (PDI) team to ensure functionality and prevent errors. The achieved results indicate that the full-stack approach combined with the Waterfall method successfully produces a responsive digital form prototype, minimizes bugs through rigorous code reviews, and accelerates software adaptation. The contribution of this research is providing a scalable architectural model for health information systems, which can serve as a reference for medical institutions in executing digital transformation in healthcare services.

Keywords: *Full-Stack; Waterfall; CodeIgniter; DBeaver; Healthcare.*

1. PENDAHULUAN

Perkembangan teknologi informasi di era Revolusi Industri 4.0 telah membawa dampak signifikan pada berbagai sektor, termasuk sektor kesehatan. Digitalisasi layanan kesehatan kini bukan lagi sekadar opsi tambahan, melainkan sebuah kebutuhan strategis untuk menjamin keberlanjutan, efisiensi, dan kualitas pelayanan medis [1]. Rumah sakit sebagai institusi pelayanan kesehatan yang kompleks dituntut untuk mampu mengelola data pasien dalam volume besar dengan tingkat akurasi yang tinggi. Transformasi dari sistem konvensional berbasis kertas (*paper based*) menuju sistem elektronik (*paperless*) menjadi langkah krusial dalam memodernisasi manajemen rumah sakit. Hal ini sejalan dengan upaya global untuk menciptakan *Smart Hospital*, di mana integrasi data dan otomatisasi proses menjadi tulang punggung operasional sehari-hari [2].

Salah satu komponen paling penting dalam administrasi rumah sakit adalah pengelolaan formulir medis. Formulir ini mencakup berbagai data penting, mulai dari pendaftaran pasien, riwayat penyakit, persetujuan tindakan medis (*informed consent*), hingga resep obat. Dalam praktik konvensional, penggunaan formulir fisik sering kali menimbulkan berbagai kendala teknis dan administratif. Studi menunjukkan bahwa pencatatan manual rentan terhadap kesalahan manusia (*human error*), seperti tulisan yang tidak terbaca, data yang hilang, atau duplikasi informasi yang tidak perlu [3]. Selain itu, akumulasi formulir fisik menuntut penyediaan ruang penyimpanan (gudang arsip) yang luas serta biaya pemeliharaan yang tidak sedikit. Kendala yang lebih serius muncul ketika data tersebut dibutuhkan secara mendesak; proses pencarian kembali dokumen fisik memakan waktu yang lama sehingga dapat memperlambat pengambilan keputusan medis [4].

Untuk mengatasi permasalahan tersebut, banyak rumah sakit mulai beralih ke Sistem Informasi Manajemen Rumah Sakit (SIMRS). Namun, tantangan baru muncul dalam aspek fleksibilitas perangkat lunak. Formulir medis memiliki karakteristik yang unik dan dinamis; format isian dapat berubah sewaktu-waktu mengikuti regulasi kesehatan

atau kebutuhan spesifik poli [5]. Sistem basis data relasional (RDBMS) tradisional sering kali dianggap kaku terhadap perubahan struktur data. Namun, kebutuhan akan integritas data medis yang sangat ketat menuntut penggunaan RDBMS yang teruji seperti PostgreSQL. Untuk mengatasi kendala fleksibilitas, PostgreSQL modern menawarkan dukungan tipe data JSONB yang memungkinkan penyimpanan format dokumen dinamis di dalam skema relasional. Hal ini menjadikannya solusi ideal untuk menampung data formulir medis yang bervariasi tanpa mengorbankan kepatuhan terhadap prinsip ACID (*Atomicity, Consistency, Isolation, Durability*) yang sangat krusial dalam rekam medis [6].

Dalam konteks pengembangan aplikasi modern, penerapan teknologi *Full Stack* menjadi solusi yang semakin populer. Pendekatan *Full Stack* memungkinkan pengembangan yang terintegrasi antara sisi antarmuka pengguna (*front-end*) dan sisi server (*back-end*). Fleksibilitas ini memungkinkan rumah sakit untuk menambah atau mengurangi *field* data pada formulir digital tanpa harus merombak struktur basis data secara keseluruhan, yang merupakan keunggulan signifikan dibandingkan sistem konvensional. Di sisi lain, kerangka kerja (*framework*) pengembangan juga memegang peranan penting dalam performa aplikasi. CodeIgniter, sebagai salah satu *framework* PHP yang populer, dikenal memiliki performa yang ringan (*lightweight*), arsitektur *Model-View-Controller* (MVC) yang rapi, serta keamanan yang baik [7]. Kombinasi antara CodeIgniter sebagai pengelola logika aplikasi dan DBBeaver sebagai penyimpan data menciptakan arsitektur sistem yang handal, cepat, dan mudah dikembangkan (*scalable*). Namun, digitalisasi formulir tidak hanya berhenti pada penyimpanan data. Terdapat aspek legalitas dan administratif yang mengharuskan data digital tersebut dapat direpresentasikan kembali dalam format dokumen fisik yang valid. Oleh karena itu, kemampuan sistem untuk mengonversi data dinamis menjadi dokumen statis (seperti format PDF) yang tata letaknya persis menyerupai formulir fisik menjadi fitur yang krusial.

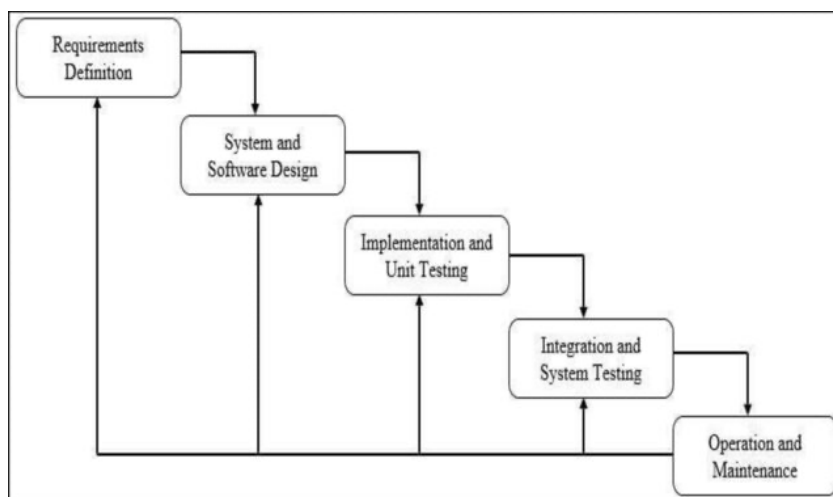
Meskipun banyak penelitian sebelumnya telah membahas digitalisasi kesehatan, seperti penelitian oleh Pongtambing et al. (2024) yang membahas literasi kesehatan digital [1], serta Kriswibowo et al. (2025) tentang digitalisasi layanan kesehatan [8], masih sedikit literatur yang secara spesifik membahas implementasi teknis penggabungan *framework* CodeIgniter dan PostgreSQL khusus untuk modul formulir digital yang membutuhkan presisi format cetak (PDF). Kebanyakan sistem yang ada hanya berfokus pada *input* data, namun mengabaikan pentingnya *output* dokumen yang sesuai standar akreditasi rumah sakit. Kesenjangan (*gap*) inilah yang menjadi landasan dilakukannya penelitian ini.

Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem formulir digital rumah sakit dengan menerapkan teknologi *Full Stack*. Fokus utama pengembangan diarahkan pada dua aspek teknis: (1) merancang mekanisme integrasi antara CodeIgniter dan PostgreSQL untuk menangani penyimpanan data formulir yang dinamis; serta (2) mengembangkan algoritma generasi dokumen otomatis yang mampu memetakan data dari basis data ke dalam format PDF dengan presisi tinggi. Dengan adanya sistem ini, diharapkan proses administrasi medis dapat berjalan lebih efisien, risiko kehilangan data dapat diminimalisir, dan kebutuhan akan dokumentasi fisik yang valid tetap terpenuhi. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi praktis bagi pengembang sistem informasi kesehatan dan menjadi referensi akademis mengenai arsitektur aplikasi medis.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Penelitian ini menerapkan kerangka kerja Waterfall (Air Terjun) sebagai metode pengembangan perangkat lunak. Penggunaan metode ini didasarkan pada teori bahwa model Waterfall merupakan paradigma rekayasa perangkat lunak yang menggunakan pendekatan sekuensial linier, di mana setiap fase harus diselesaikan secara penuh sebelum beralih ke fase berikutnya [9]. Secara teoritis, model ini memberikan struktur yang kaku namun terukur, yang terdiri dari tahapan analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan. Penjelasan mengenai alur kerja yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 1 di bawah ini.



Gambar 1. Metode Waterfall

Pemilihan model Waterfall pada Gambar 1 di atas didasarkan pada karakteristik sistem formulir digital rumah sakit yang memerlukan spesifikasi kebutuhan yang kaku, terstruktur, dan terdokumentasi dengan baik sejak fase awal. Menurut literatur, metode ini sangat efektif untuk meminimalisir kesalahan medis (medical error) akibat malfungsi sistem, karena arsitektur dan batasan sistem telah dikunci sejak awal sebelum penulisan kode dimulai [10][11]. Pendekatan ini memastikan progres pengembangan mengalir secara logis dan meminimalisir risiko perubahan mendadak di tengah proses yang dapat mengganggu integritas data medis.

2.2 Tahapan Pengembangan Sistem

Pengembangan sistem formulir digital ini mengikuti alur sistematis dari metode Waterfall yang telah ditunjukkan pada Gambar 1, di mana setiap solusi teknis diterapkan sesuai urutan fase berikut:

a. Analisis Kebutuhan (*Requirements Analysis*)

Tahap awal difokuskan pada pengumpulan seluruh kebutuhan sistem secara menyeluruh. Secara teoritis, analisis kebutuhan bertujuan untuk memitigasi risiko ketidaksesuaian fitur dengan kebutuhan pengguna di akhir proyek [12]. Informasi dan spesifikasi teknis dihimpun dari *Project Manager* (PM) sebagai representasi rumah sakit. Pada fase ini, seluruh *mockup* antarmuka, daftar struktur data, dan alur formulir medis didefinisikan secara final, termasuk mekanisme konversi output ke format PDF.

b. Perancangan Sistem (*System Design*)

Berdasarkan spesifikasi dari PM, dilakukan pembuatan cetak biru (blueprint) sistem. Kajian pustaka menyebutkan bahwa perancangan sistem yang matang akan memudahkan skalabilitas aplikasi di masa depan [13]. Arsitektur aplikasi dirancang menggunakan pola Model-View-Controller (MVC) untuk memisahkan logika bisnis dari antarmuka. Perancangan basis data menggunakan PostgreSQL dengan fitur JSONB untuk mengakomodasi tipe isian formulir yang dinamis tanpa melanggar skema relasional utama.

c. Implementasi (*Implementation*)

Fase ini merupakan penerjemahan desain ke dalam baris kode program secara Full Stack. Mengikuti desain yang sudah dikunci, pengembang membangun *Front-end* (HTML5, CSS3) dan *Back-end* (PHP, CodeIgniter). Implementasi berbasis framework seperti CodeIgniter diakui dalam berbagai literatur dapat mempercepat proses pengembangan karena ketersediaan pustaka (*library*) yang lengkap [14]. Tim juga menggunakan *Version Control System* (GitHub) dengan mekanisme *Pull Request* untuk menjaga integritas repositori [15], [16].

d. Pengujian (*Testing*)

Setelah seluruh baris kode selesai, sistem memasuki fase pengujian oleh tim *Quality Assurance*. Pengujian dilakukan menggunakan metode Black Box Testing, yaitu teknik pengujian perangkat lunak yang berfokus pada fungsionalitas aplikasi tanpa harus memeriksa struktur kode internalnya [17]. Tim mengevaluasi kesesuaian antarmuka, validasi formulir, dan presisi algoritma PDF. Jika ditemukan anomali, sistem dikembalikan ke pengembang untuk direvisi hingga disetujui [18].

e. Pemeliharaan (*Maintenance*)

Sistem yang lolos uji diimplementasikan ke jaringan lokal rumah sakit. Fase ini berfokus pada penanganan kendala operasional dan pemantauan performa server. Sebagaimana dikemukakan dalam teori pemeliharaan perangkat lunak, tahap ini krusial untuk menjaga agar sistem tetap relevan dan berjalan optimal dalam lingkungan produksi yang dinamis [19].

2.3 Teknologi Pengembangan

Teknologi utama yang diintegrasikan dalam penelitian ini berpusat pada ekosistem *open source*. Bahasa pemrograman PHP bertindak sebagai *server-side scripting* yang memproses setiap permintaan klien secara tertutup sebelum dirender menjadi HTML, memastikan kerahasiaan data medis terjaga [20]. Integrasi CodeIgniter sebagai *framework* utama memberikan arsitektur yang ringan sekaligus menyediakan lapisan keamanan inheren, seperti proteksi *Cross-Site Request Forgery* (CSRF) dan penyaringan *Cross-Site Scripting* (XSS) yang wajib ada pada sistem kesehatan [21]. Pada sisi pengelolaan data, sistem ini menggunakan PostgreSQL yang diakses melalui perangkat lunak DBeaver. Pemilihan PostgreSQL didasarkan pada kemampuannya menjaga integritas data relasional yang ketat (ACID) sekaligus memberikan fleksibilitas pengelolaan struktur data dinamis, menjadikan arsitektur sistem ini sangat andal untuk lingkungan medis [22][23].

3. HASIL DAN PEMBAHASAN

3.1 Lingkungan Implementasi

Pengembangan sistem formulir digital rumah sakit ini dilakukan dalam sebuah ekosistem pengembangan terintegrasi yang dirancang untuk meminimalkan hambatan teknis selama siklus hidup pengembangan perangkat lunak (*Software Development Life Cycle*). Lingkungan implementasi mencakup sinergi antara infrastruktur perangkat keras dan konfigurasi perangkat lunak yang mumpuni guna memastikan stabilitas performa saat menangani beban kerja tinggi, khususnya pada proses penampilan dokumen PDF dan manipulasi basis data rekam medis.

3.1.1 Infrastruktur Perangkat Keras

Pada sisi perangkat keras, sistem dikembangkan menggunakan unit komputasi dengan spesifikasi tinggi guna mendukung aktivitas *multitasking* dan simulasi server lokal secara optimal. Spesifikasi perangkat keras yang digunakan meliputi:

- a. Prosesor: *Intel Core i7* Generasi ke-11. Pemilihan prosesor ini didasarkan pada kebutuhan daya pemrosesan multi-core untuk mempercepat proses kompilasi kode, penanganan antrean basis data, dan eksekusi skrip sisi server, terutama saat melakukan *rendering* dokumen dari format HTML ke dalam bentuk PDF yang membutuhkan komputasi CPU yang intensif.
- b. Memori (*RAM*): 16GB. Kapasitas memori ini dialokasikan untuk memungkinkan berjalannya berbagai layanan secara simultan, seperti web server lokal (*Apache/Nginx*), database *engine* PostgreSQL, perangkat lunak desain, dan *Integrated Development Environment* (*IDE*) tanpa mengalami degradasi performa atau *bottleneck* memori.
- c. Penyimpanan: *Solid State Drive* (*SSD*) berkapasitas 512GB. Kecepatan *input/output* (*I/O*) data yang ditawarkan oleh *SSD* sangat krusial saat melakukan *query* besar terhadap basis data rekam medis atau saat memuat pustaka (*library*) pihak ketiga yang digunakan dalam kerangka kerja aplikasi.

3.1.2 Arsitektur Perangkat Lunak dan Teknologi Full-stack

Pemilihan teknologi full-stack didasarkan pada kebutuhan akan fleksibilitas dan skalabilitas sistem dalam jangka panjang. Lingkungan perangkat lunak yang diadopsi dalam penelitian ini meliputi:

- a. Sistem Operasi: Windows 11 digunakan sebagai platform pengembangan yang stabil dan memiliki kompatibilitas tinggi terhadap berbagai perangkat pengembangan *open-source*.
- b. *Integrated Development Environment* (*IDE*): *Visual Studio Code*, yang dilengkapi dengan berbagai ekstensi untuk *debugging* dan manajemen *source code*. *Visual Studio Code* dipilih karena sifatnya yang ringan namun sangat dapat dikembangkan melalui ekstensi pendukung untuk proses *debugging*, penyorotan sintaks (*syntax highlighting*), dan manajemen *source code*.
- c. Bahasa Pemrograman dan Framework: Pengembangan *back-end* mengandalkan bahasa pemrograman PHP yang diintegrasikan dengan *framework* CodeIgniter. Pemilihan CodeIgniter didasari oleh efisiensinya dalam menerapkan pola arsitektur Model-View-Controller (*MVC*). Pola ini secara tegas memisahkan logika antarmuka pengguna (*View*), pengatur aliran interaksi data (*Controller*), dan logika pemrosesan ke basis data (*Model*), sehingga memudahkan pemeliharaan kode di kemudian hari.
- d. Manajemen Basis Data: PostgreSQL digunakan sebagai database *engine* utama, diakses melalui perangkat DBEaver sebagai *Universal Database Tool*. DBEaver mempermudah perancangan skema relasional tabel dan pengelolaan tipe data dinamis melalui antarmuka visual yang komprehensif.
- e. *Version Control System* (*VCS*): Git dan GitHub digunakan untuk melacak setiap perubahan kode secara kronologis. Hal ini sangat penting untuk mencegah terjadinya penumpukan error dan memudahkan penelusuran kembali ke versi kode sebelumnya (*rollback*) jika ditemukan malfungsi pada saat proses *deployment*.

3.2 Hasil Penerapan Metode Waterfall

Sesuai dengan kerangka kerja metodologi yang telah ditetapkan, pengembangan sistem ini dieksekusi menggunakan model sekuensial Waterfall. Penerapan tahapan-tahapan tersebut menghasilkan purwarupa yang terstruktur dengan alur logika yang dapat dipertanggungjawabkan.

3.2.1 Penerapan Tahap Analisis Kebutuhan

Pada fase pertama pengembangan, aktivitas secara komprehensif difokuskan pada pemetaan kebutuhan fungsional dari formulir medis yang berlaku di lingkungan rumah sakit. Berdasarkan hasil analisis dan penugasan dari Manajemen Proyek, sistem dituntut untuk mengakomodasi digitalisasi puluhan dokumen rekam medis krusial yang saling berkaitan. Secara spesifik, teridentifikasi sebanyak 35 formulir operasional utama yang harus diubah ke dalam format digital. Untuk memudahkan analisis struktur data, seluruh formulir tersebut diklasifikasikan ke dalam enam kategori layanan utama, yaitu:

- a. Kategori Gawat Darurat (*IGD*): Mencakup formulir yang membutuhkan pengisian cepat dengan format data observasi singkat, yaitu Catatan Medis Instalasi Gawat Darurat, Formulir Medis Asuhan Keperawatan *IGD*, dan Asesmen Awal Medis Instalasi Gawat Darurat.
- b. Kategori Rawat Jalan dan Khusus: Mencakup formulir untuk pemeriksaan poliklinik dengan parameter spesifik, meliputi Resume Pengobatan Rawat Jalan, Pengkajian Awal Medis Gigi, dan Pengkajian Awal Pasien Rawat Jalan Poli Sunat.
- c. Kategori Rawat Inap dan Perawatan Kritis: Kategori ini memiliki kompleksitas input tertinggi karena melibatkan data *monitoring* kontinu. Formulir yang masuk di dalamnya adalah Pengkajian Awal Medis dan Keperawatan Rawat Inap, Pengkajian Awal Medis Rawat Inap (Umum), Kriteria Pasien Masuk dan Keluar Ruang ICU, Kriteria Pasien Masuk dan Keluar Ruang PICU, Kriteria Pasien Masuk dan Keluar Ruang NICU, Form Medis Daftar Kurva Gula Darah Harian, *Monitoring* Pemberian Transfusi Darah, dan Observasi Pengkajian Nyeri Ulang.
- d. Kategori Tindakan Bedah dan Anestesi: Membutuhkan tingkat validasi ganda untuk mencegah malapraktik, yang terdiri dari Asesmen Praanestesi Pra Sedasi, Kajian Pra Anestesi dan Tindakan Bedah, Formulir Laporan Operasi, Status Post Anestesi Baru, dan *Monitoring* Pasca Anestesi di Ruang Pemulihan.

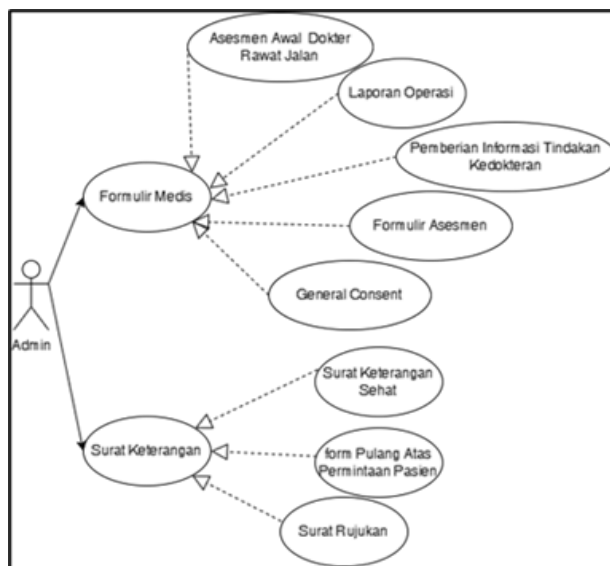
- e. Kategori Persetujuan Tindakan (*Consent*) dan Legalitas: Formulir pada kategori ini wajib mengakomodasi tanda tangan digital dan persetujuan bermeterai, meliputi *General Consent*, *Informed Consent*, Form Penolakan Tindakan Medis, Pulang Atas Permintaan Pasien / Penghentian Pengobatan / Perawatan, Formulir Persetujuan Rujukan, dan Surat Rujukan.
- f. Kategori Administrasi, Komunikasi, dan Evaluasi Lanjutan: Melibatkan alur data antar unit dan pendataan administratif akhir, meliputi Pengkajian Awal Pasien, Formulir Identitas Pasien, Rekonsiliasi Obat, Tanda Terima Leaflet Hak & Kewajiban, Formulir Komunikasi Efektif SBAR, Formulir Monitoring Restrain, Surat Keterangan Sehat, Kajian Pasien Akhir Hayat, Kajian Ulang Pasien Akhir Hayat, serta Kuesioner Kepuasan Pasien.

Hasil analisis mendalam terhadap ke-35 formulir di atas menunjukkan bahwa setiap dokumen memiliki format isian yang sangat variatif, mulai dari teks bebas (*free text*), angka desimal, skala pengukuran (*radio button/checkbox*), hingga kanvas tanda tangan digital. Tantangan utama yang diselesaikan pada tahap ini adalah merancang sebuah arsitektur sistem yang mampu mengakomodasi semua bentuk tipe data tersebut tanpa harus membuat 35 tabel basis data (*database tables*) yang terpisah-pisah, yang mana akan sangat membebani kinerja server.

Dengan demikian, hasil dari tahap analisis ini menetapkan batasan ruang lingkup bahwa aplikasi harus menggunakan pendekatan penyimpanan basis data dinamis, mampu mengonversi dokumen fisik ke dalam bentuk web *form* (HTML) yang ergonomis bagi perawat, dan yang paling krusial, sistem harus memetakan semua variabel tersebut kembali menjadi fail unduhan PDF dengan tata letak (*layout*) statis yang identik seratus persen dengan dokumen cetak standar akreditasi rumah sakit.

3.2.2 Penerapan Tahap Perancangan Sistem

Fase perancangan menerjemahkan kebutuhan fungsional ke dalam rancangan teknis (*blueprint*). Salah satu representasi utama dalam tahap ini adalah identifikasi interaksi pengguna terhadap sistem yang dimodelkan melalui Use Case Diagram. Implementasi dari interaksi tersebut dapat dilihat pada Gambar 2 di bawah ini, yang mendeskripsikan hubungan antara aktor dengan fungsionalitas utama sistem.



Gambar 2. Use case diagram

Berdasarkan Gambar 2 di atas, dapat dijelaskan bahwa sistem mengakomodasi berbagai kebutuhan formulir medis digital mulai dari Asesmen Awal, Laporan Operasi, hingga Surat Rujukan. Aktor (tenaga medis atau administrator) diharuskan melakukan autentikasi sistem sebelum diberikan hak akses untuk mengelola data. Alur interaksi utama yang terpampang pada Gambar 2 meliputi pengisian data pada formulir digital, verifikasi kelengkapan data, penyimpanan, hingga eksekusi perintah pencetakan. Pembatasan hak akses ini dirancang sebagai bentuk kepatuhan terhadap regulasi privasi rekam medis pasien agar setiap aktivitas pengisian formulir, seperti *General Consent* atau Surat Keterangan Sehat, tetap terjaga integritasnya.

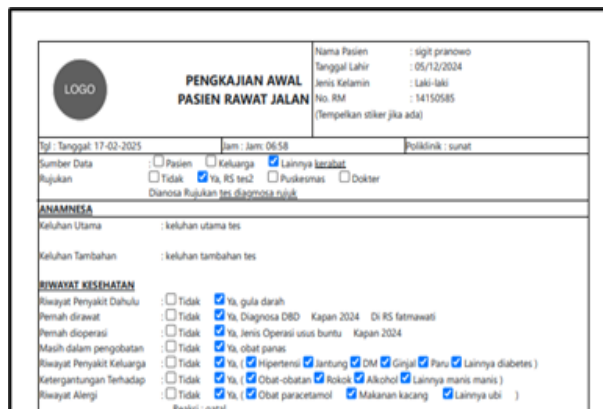
Selanjutnya, perancangan basis data dilakukan secara strategis. Untuk menghindari masalah struktur data formulir yang dinamis, perancangan mengimplementasikan penggunaan kolom berjenis JSONB (*JavaScript Object Notation - Binary*) pada PostgreSQL. Jika RDBMS tradisional memaksa setiap kolom isian menjadi entitas kolom dalam tabel, penggunaan algoritma struktur JSONB memungkinkan seluruh atribut formulir (*input array*) disimpan ke dalam satu kolom tunggal terenkripsi. Hal ini memberikan fleksibilitas tanpa mengorbankan integritas referensial dari sistem basis data.

3.2.3 Penerapan Tahap Implementasi dan Algoritma Sistem

Fase implementasi (*coding*) merupakan penerapan teknologi *Full Stack* secara komprehensif. Di sisi *Front-end*, antarmuka pengguna direkayasa menggunakan kombinasi HTML5, CSS3, dan JavaScript untuk menghasilkan form yang

interaktif, responsif, dan memberikan umpan balik validasi secara *real-time* kepada pengguna sebelum data dikirimkan ke server. Di sisi *Back-end*, algoritma utama yang menjadi pusat dari penelitian ini adalah Algoritma Pemrosesan Dinamis dan Konversi PDF. Proses atau algoritma sistem dalam menangani formulir berjalan melalui urutan langkah berikut:

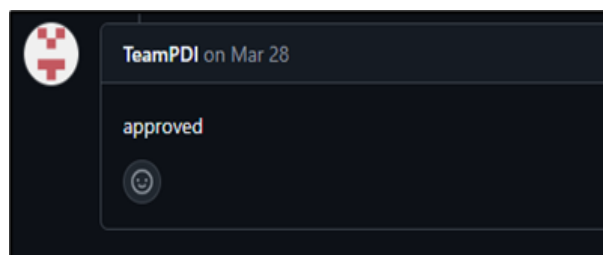
- Penerimaan Data (*Data Ingestion*): Saat petugas medis menekan tombol simpan, data yang dikirim (*payload*) akan ditangkap oleh *Controller* di CodeIgniter. Algoritma pembersihan bawaan CodeIgniter (XSS Filtering) secara otomatis memindai input dari potensi injeksi skrip berbahaya.
- Serialisasi Data (*Data Serialization*): Data *array* multidimensi dari formulir diubah (di-encode) ke dalam format JSON. Data terstruktur inilah yang kemudian dikirim ke Model untuk disimpan ke dalam kolom JSONB pada tabel transaksi formulir di PostgreSQL.
- Pengambilan Data (*Data Retrieval*): Saat pengguna meminta penayangan dokumen, *Controller* melakukan *query* ke basis data menggunakan *Query Builder* untuk mengambil baris data yang relevan berdasarkan ID unik pasien dan ID formulir.
- Pemetaan Antarmuka Cetak (*Template Mapping*): Basis data merespons dengan mengembalikan *string* JSON. Sistem kemudian menjalankan algoritma de-serialisasi untuk mengembalikannya menjadi *array* asosiatif di PHP. Data dari *array* tersebut disuntikkan secara dinamis ke dalam *View* khusus yang sudah berisi *template* tata letak formulir (margin, kop surat rumah sakit, dan ukuran kertas).
- Generasi PDF (*PDF Rendering*): *View* yang telah disuntik dengan data tersebut tidak ditampilkan ke browser konvensional, melainkan diteruskan ke pustaka *rendering* PDF (seperti mPDF atau DomPDF). Algoritma komputasi di sisi server akan mengonversi struktur elemen DOM HTML dan pengaturan style CSS menjadi dokumen PDF biner murni yang tidak dapat diubah (statis), siap untuk diunduh atau dicetak secara fisik. Untuk hasil generasi PDF yang telah dikembangkan dapat terlihat di gambar 3.



Gambar 3. Contoh cetakan formulir digital

3.2.4 Penerapan Tahap Pengujian (*Testing*)

Setelah implementasi selesai, sistem diuji secara ketat. Dari beberapa tahapan yang sudah dilakukan, Pengujian menjadi tahapan krusial dalam memastikan bahwa sistem tidak hanya berjalan secara teknis, tetapi juga memenuhi kebutuhan administratif dan estetika output akhir. Pengujian ini dipimpin oleh tim internal pengembang menggunakan metode *Black Box Testing*, di mana fokus pengujian tidak pada kode di baliknya, melainkan pada keakuratan input dan validitas output.



Gambar 4. Hasil Pengujian

Skenario pengujian mencakup penanganan data kosong, input karakter khusus (*special character*), batas maksimal ukuran *string* isian, dan simulasi pengisian data secara serentak. Fokus pengujian tertinggi ada pada pemeriksaan presisi dokumen PDF. Pengujian awal mencakup verifikasi kesesuaian tampilan dokumen hasil (*printout*) dengan referensi desain atau *mockup* yang telah disediakan sebelumnya. Setelah melalui pengujian mandiri di ranah pengembang, formulir ditinjau kembali oleh tim penguji untuk memastikan validitas data yang ditampilkan, kesesuaian format cetak, serta ketepatan elemen-elemen visual seperti margin, spasi antar baris, tata letak tabel, dan keberadaan label. Jika hasil *printout* telah sesuai dengan seluruh kriteria fungsional dan tata letak operasional rumah sakit, maka formulir dinyatakan lolos uji seperti yang dilampirkan pada Gambar 4.

3.2.5 Penerapan Tahap Pemeliharaan (*Maintenance*)

Sistem yang telah dinyatakan lolos tahap pengujian kemudian diunggah ke peladen pusat untuk diimplementasikan di lingkungan layanan medis. Pada fase pemeliharaan awal, tim pengembang berfokus pada mitigasi kesalahan kecil yang baru muncul ketika sistem menangani aliran data (*traffic*) langsung dari pengguna nyata. Tahap pemeliharaan juga mencakup pencadangan (*backup*) data secara otomatis dan penyesuaian optimalisasi kueri PostgreSQL untuk menjamin agar pembuatan dokumen PDF tetap berlangsung di bawah target waktu kurang dari dua detik per dokumen, guna menjaga kualitas pengalaman pengguna (*user experience*) tenaga medis yang memiliki mobilitas tinggi.

3.3 Pembahasan Hasil Akhir

Hasil utama dari keseluruhan proses metodologis di atas adalah berdirinya sebuah sistem formulir digital rekam medis berbasis web yang tangguh, adaptif, dan memenuhi standar birokrasi institusi kesehatan. Selama masa pelaksanaan penelitian dan pengembangan, sebanyak 36 jenis formulir digital telah berhasil direkayasa dan diintegrasikan ke dalam ekosistem layanan rekam medis rumah sakit. Formulir-formulir yang berhasil didigitalkan tersebut memiliki cakupan fungsi yang luas, meliputi lembar data admisi pasien awal, *informed consent* (persetujuan tindakan medis yang memerlukan tanda tangan digital), rekam medis observasi harian rawat inap, hingga laporan akhir tindakan operatif dan penunjang diagnostik klinik.

Keberhasilan mencapai angka 36 formulir ini membuktikan bahwa pendekatan arsitektur Full-stack menggunakan PHP CodeIgniter dan manajemen skema JSONB pada PostgreSQL memberikan dampak efisiensi yang sangat signifikan. Pengembangan tidak lagi memerlukan pembuatan struktur tabel baru di basis data untuk setiap jenis formulir yang berbeda; pengembang cukup mendaftarkan *template* desain *Front-end* baru dan mengaitkannya dengan logika pemrosesan yang sudah terpusat. Hal ini secara drastis mempercepat proses pembuatan perangkat lunak dari hitungan minggu menjadi hanya beberapa hari untuk setiap modul formulir baru.

Lebih lanjut, algoritma generasi dokumen otomatis yang dibangun telah berhasil menjawab kendala utama administrasi medis, yaitu dualisme kebutuhan antara efisiensi digital (*paperless*) dan kewajiban ketersediaan arsip fisik untuk keperluan legalitas atau visitasi akreditasi. Dokumen PDF yang dihasilkan secara konsisten menampilkan struktur informasi dengan presisi tinggi, terbebas dari kesalahan pemotongan halaman, degradasi resolusi teks, maupun pergeseran tabel data.

Penerapan metode *Waterfall* pada proyek yang melibatkan banyak variabel isian ini terbukti merupakan langkah yang strategis. Spesifikasi kebutuhan cetak, ukuran kertas, pengaturan huruf, serta pembatasan peran akses yang dikunci dengan ketat sejak fase "Analisis Kebutuhan" meminimalisir risiko perombakan kode skala besar. Dokumen hasil akhir tidak hanya dapat diproduksi secara cepat, namun kualitas tampilannya berhasil mencapai standar visibilitas dan kepatuhan yang sama persis dengan formulir konvensional pracetak yang didistribusikan oleh rumah sakit, tanpa mengorbankan keamanan basis data secara keseluruhan.

4. KESIMPULAN

Penelitian ini berhasil membuktikan bahwa penerapan arsitektur Full-stack menggunakan framework CodeIgniter dan basis data PostgreSQL dengan pendekatan beruntuk *Waterfall* mampu menjadi solusi tepat dalam menjembatani kebutuhan digitalisasi rekam medis yang dinamis dengan regulasi pengarsipan cetak rumah sakit yang ketat. Melalui pemanfaatan fitur penyimpanan berformat JSONB dan algoritma rendering dokumen secara otomatis, sistem ini secara efektif telah mendigitalkan 36 jenis formulir medis krusial tanpa menimbulkan redundansi skema tabel. Inovasi penggunaan skema dinamis ini memungkinkan sistem untuk beradaptasi terhadap perubahan struktur formulir medis di masa depan tanpa harus melakukan perombakan besar pada arsitektur basis data, sekaligus memastikan presisi tata letak keluaran PDF tetap identik dengan standar akreditasi fisik yang berlaku. Keberhasilan pengujian fungsional menggunakan metode *Black Box Testing* serta evaluasi estetika dokumen pada puluhan modul tersebut menegaskan bahwa integrasi logika antarmuka dan manajemen basis data yang adaptif dapat mempercepat siklus rekayasa perangkat lunak kesehatan secara signifikan. Secara operasional, sistem ini mengurangi hambatan administratif dalam pengelolaan data pasien dan meminimalisir risiko kesalahan input manual yang sering terjadi pada sistem berbasis kertas. Sebagai implikasi teknis untuk pengembangan di masa depan, adopsi komponen antarmuka berbasis *reusable template* dan standarisasi dokumentasi sistem yang lebih komprehensif sangat direkomendasikan. Selain itu, pengembangan lebih lanjut dapat diarahkan pada integrasi fitur tanda tangan digital terenkripsi untuk meningkatkan aspek legalitas dokumen elektronik, guna memastikan kemudahan pemeliharaan dan skalabilitas sistem di tengah eskalasi kebutuhan fasilitas layanan kesehatan yang semakin kompleks dan menuntut efisiensi tinggi.

UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENCES

- [1] Y. S. Pongtambing, E. A. M. Sampetoding, R. Uksi, and E. S. Manapa, "Digitalisasi dan literasi kesehatan pada smart village,"

- Compromise J. Community Proffesional Serv. J.*, vol. 2, no. 1, pp. 11–18, 2024, doi: 10.57213/compromisejournal.v2i1.157.
- [2] S. N. Chotimah, “Implementasi sistem informasi kesehatan di fasilitas pelayanan kesehatan Indonesia: Literature review,” *J. Rekam Medis Manaj. Infomasi Kesehat.*, vol. 2, no. 1, pp. 8–13, 2022, doi: 10.53416/jurmik.v2i1.67.
- [3] A. Dhaja, L. K. Purba, M. F. K. Dewi, N. Fitriana, and R. A. Novitasari, “Patient risks from human error in medical record management,” vol. 1, no. 1, pp. 10–20, 2025.
- [4] M. Gusni and Y. Yunengsih, “Analisis Penerapan Rekam Medis Elektronik Terhadap Efisiensi Dan Kualitas Pendaftaran Rawat Jalan Di Rumah Sakit X,” vol. 18, no. 1978, pp. 3089–3100, 2024.
- [5] E. Apriliyanto, “Comparison of Measuring RDBMS Database Performance with NoSQL on Electronic Medical Record Systems (EMR),” no. 2, pp. 65–78, 2023.
- [6] M. Tebi, “Design and development of MedSync : A Web-Based electronic medical record system for healthcare management,” no. September, pp. 1–21, 2024.
- [7] E. Pratiwi, M. A. Hamzah, and Darmawati, “Rancang Bangun Sistem Informasi pada Klinik Kartini Farma Berbasis Website,” no. 2, pp. 31–38, 2024.
- [8] R. Kriswibowo, P. P. Rusina Widha Febriana, Johan Suryo Prayogo, S. Ningrum, A. T. Setyadi, F. K. Suhada, and M. A. Riskyansah, “Digitalisasi layanan kesehatan : Pelatihan IT untuk Kader posyandu Desa Simogirang dalam pencatatan data kesehatan,” 2025.
- [9] A. Saravanos and M. X. Curinga, “Simulating the software development lifecycle : The Waterfall Model,” 2023.
- [10] A.-M. Stefan, N.-R. Rusu, E. Ovreiu, and M. Ciuc, “Empowering healthcare : A comprehensive guide to implementing a robust medical information system — components , benefits , objectives , evaluation criteria , and seamless deployment strategies,” 2024.
- [11] O. V. Ajayi-kaffi and P. N. Buyurgan, “Is Agile Methodology better than Waterfall approach in enhancing effective communication in healthcare process improvementpProjects ?,” vol. 5, no. 11, pp. 3648–3651, 2024.
- [12] S. Hussain *et al.*, “Mitigating software vulnerabilities through secure software development with a policy-driven Waterfall Model,” vol. 2024, 2024, doi: 10.1155/2024/9962691.
- [13] R. Syarifuddin, Junaedy, and R. Syarifuddin, “Development of a Web-Based Accreditation System for study program management using the Waterfall Model,” vol. 1, no. 2, pp. 59–66, 2025, doi: 10.47776/nuai.v1i2.1663.
- [14] I. D. Ricci and F. A. Rahman, “Optimizing web development through strategic web,” no. May, pp. 1–6, 2025.
- [15] M. Wessel, J. Vargovich, M. A. Gerosa, and C. Treude, *Github Actions: The impact on the Pull Request Process*, vol. 123. 2023. doi: 10.1007/s10664-023-10369-w.
- [16] M. Watanabe, H. A. O. Li, Y. Kashiwa, B. Reid, H. Iida, and A. E. Hassan, “On the use of Agentic Coding : An empirical study of Pull Requests on on the use of Agentic Coding : An empirical study of Pull Requests on GitHub,” no. April, pp. 0–23, 2026, doi: 10.1145/3798166.
- [17] X. Jia, “The role and importance of software testing in software quality management,” vol. 1, no. 4, pp. 39–44, 2023.
- [18] Ç. Eren, K. Şahin, and E. Tüzün, “Analyzing bug life cycles to derive practical insights,” no. June 2023, 2026, doi: 10.1145/3593434.3593504.
- [19] Gilang Akbar Romadhoni, Susanto, and T. Handayani, “Rancang Bangun Sistem Informasi Pendaftaran Siswa Baru Berbasis Web dengan Metode Waterfall pada TK Pertiwi 14,” vol. 8, no. 1, pp. 121–137, 2026.
- [20] V. Costa, D. Alves, F. A. Bernardi, and R. P. C. L. Rijo, “Security approaches for electronic health data handling through the Semantic Web : A scoping review,” vol. 14, pp. 771–784, 2023, doi: 10.3233/SW-223088.
- [21] L. K. Linda Amelia Putri Andini, B. Karmanto, and M. I. Mohammad, “Evaluasi Usability Sistem Informasi Manajemen Rumah Sakit (SIMRS) dan Dampaknya Terhadap Efisiensi Pelayanan Pasien di RS Ciremai Menggunakan System Usability Scale (SUS),” vol. 8, no. 3, pp. 409–418, 2025.
- [22] M. Abbasi, M. V Bernardo, P. Váz, J. Silva, and P. Martins, “Adaptive and scalable database management with Machine Learning integration : A PostgreSQL case study,” pp. 1–25, 2024.
- [23] G. R. Enjam, “Optimizing PostgreSQL for high - volume insurance transactions & Secure backup and restore strategies for databases,” vol. 4, no. 1, pp. 104–111, 2023.