

IMPLEMENTASI REMULT DALAM PENGEMBANGAN SISTEM LOGBOOK KOMPETENSI LSP TAMBANG MINERBA INDONESIA SESUAI TECHNICAL PROCESSES ISO/IEC/IEEE 12207:2017

Oleh :

Muhammad Rizkiansyah¹⁾, Maimun²⁾, Arief Ichwani³⁾, Theodora Maria Putri Komul⁴⁾

^{1,2,3,4} Universitas Esa Unggul

email: muhammadrizkiansyah60@gmail.com

Informasi Artikel

Riwayat Artikel :

Submit, 1 Agustus 2026

Revisi, 27 September 2026

Diterima, 29 September 2026

Publish, 30 September 2026

Kata Kunci :

Remult,

Logbook Kompetensi,

ISO/IEC/IEEE 12207:2017,

LSP,

eXtreme Programming,

SvelteKit.



ABSTRAK

Pemeliharaan kompetensi pemegang sertifikat (surveillance) pada Lembaga Sertifikasi Profesi Tambang Minerba Indonesia (LSP-TMI) membutuhkan pendokumentasian rekam jejak aktivitas harian atau logbook yang terstruktur. Pengelolaan data logbook secara konvensional rentan terhadap inefisiensi, risiko redundansi, serta kesulitan dalam proses pemantauan oleh asesor. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun Sistem Logbook Asesi berbasis web guna menjadi sarana dalam pemeliharaan kompetensi yang sudah dimiliki oleh pemegang sertifikat. Pengembangan sistem ini mengadopsi standar siklus hidup perangkat lunak dengan membatasi ruang lingkup implementasi hanya pada kelompok Technical Processes berdasarkan ISO/IEC/IEEE 12207:2017. Arsitektur perangkat lunak dibangun menggunakan bahasa pemrograman TypeScript dengan framework SvelteKit untuk penanganan server-side rendering (SSR) dan antarmuka pengguna. Pengelolaan lapisan database relasional MariaDB dikonfigurasi melalui library Remult guna mendefinisikan entitas secara deklaratif, sementara penyimpanan berkas bukti fisik diimplementasikan menggunakan layanan object storage Cloudflare R2. Deployment sistem dilakukan pada staging environment di dalam Virtual Private Server (VPS) agar pengujian terisolasi dari server produksi utama.

This is an open access article under the [CC BY-SA](#) license



Corresponding Author:

Nama: Muhammad Rizkiansyah

Afiliasi: Universitas Esa Unggul

Email: muhammadrizkiansyah60@gmail.com

1. PENDAHULUAN

Lembaga Sertifikasi Profesi (LSP) Tambang Minerba Indonesia, sebagai lembaga yang telah mendapat lisensi dan diakreditasi oleh Badan Nasional Sertifikasi Profesi (BNSP), bertugas menyelenggarakan asesmen dan sertifikasi kompetensi profesi khususnya di bidang pertambangan, yang sudah memiliki sistem informasi untuk manajemen asesmen dan pengisian dokumen peserta uji kompetensi secara digital. Namun, hingga saat ini, proses pencatatan kegiatan harian pasca-sertifikasi sesuai dengan skema kompetensi yang dikenal sebagai logbook kompetensi masih dilakukan secara manual melalui buku fisik atau mengetik dokumen logbook

berdasarkan templat Microsoft Word, kemudian dikumpulkan kepada asesor kompetensi dalam bentuk PDF atau berkas yang dicetak secara fisik. Praktik ini menyebabkan berbagai masalah operasional, antara lain: ketidakkonsistenan format data, keterlambatan pelaporan, kerentanan terhadap kehilangan dokumen, serta kesulitan dalam verifikasi oleh asesor secara terintegrasi. Hal ini tidak hanya menghambat efisiensi administratif, tetapi juga berpotensi mengurangi validitas dan integritas proses sertifikasi itu sendiri.

Dalam konteks pengembangan sistem informasi, pendekatan yang digunakan selama ini, baik oleh LSP Tambang Minerba maupun lembaga serupa seperti LSP Indohusada (Sri Rumini &

Sulistiyani, 2025) atau LSP Polimarin Semarang (Deviana dkk., 2023), cenderung mengandalkan metodologi pengembangan tradisional seperti waterfall atau Design Science Research (DSR) tanpa eksplisit merujuk pada standar internasional pengembangan perangkat lunak. Padahal, untuk memastikan kualitas, keandalan, dan keberlanjutan sistem, penerapan standar rekayasa perangkat lunak yang diakui secara global menjadi sangat penting. Salah satu standar tersebut adalah ISO/IEC/IEEE 12207:2017, yang memberikan kerangka lengkap untuk siklus hidup perangkat lunak, mencakup proses akuisisi, pengembangan, operasi, pemeliharaan, dan pembuangan sistem.

ISO/IEC/IEEE 12207:2017 menekankan pentingnya pengelolaan proses yang terstruktur, dokumentasi teknis yang memadai, verifikasi dan validasi berkala, serta manajemen konfigurasi dan kualitas sepanjang siklus hidup perangkat lunak ("ISO/IEC/IEEE International Standard – Systems and software engineering – Software life cycle processes," 2017). Penerapan standar ini tidak hanya meningkatkan kualitas produk akhir, tetapi juga memperkuat akuntabilitas dan transparansi dalam pengembangan sistem, faktor yang sangat relevan dalam konteks lembaga sertifikasi profesi yang diawasi oleh regulator nasional.

Di sisi teknologi, perkembangan full-stack type-safe framework seperti Remult menawarkan solusi modern untuk pengembangan sistem berbasis web yang efisien, aman, dan terintegrasi secara end-to-end. Remult memungkinkan pengembang menggunakan satu definisi entitas berbentuk class untuk menghasilkan REST API secara otomatis, mekanisme kendali akses berdasarkan role (role-based access control / RBAC), serta sinkronisasi real-time antara klien dan server semua tanpa menulis kode boilerplate secara berulang (Remult, 2025). Pendekatan Single Source of Truth (SSOT) yang diusung oleh Remult selaras dengan prinsip ISO/IEC/IEEE 12207:2017 dalam hal manajemen data dan konsistensi arsitektur sistem.

Namun, penerapan Remult dalam pengembangan sistem informasi di lingkungan LSP, khususnya LSP yang bergerak di sektor sertifikasi profesi pertambangan mineral dan batubara, belum banyak dieksplorasi. Sebagian besar studi sebelumnya masih menggunakan stack teknologi konvensional seperti PHP-Laravel (Fasih & Nisa, 2023) atau CodeIgniter (Rhamanti, 2024), yang meskipun fungsional, kurang mendukung prinsip type-safety dan integrasi full-stack yang ketat. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan menggabungkan teknologi mutakhir (dalam hal ini adalah TypeScript, Remult dan Sveltekit), metodologi pengembangan yang adaptif (eXtreme Programming / XP), dan kepatuhan terhadap standar internasional di dalam pengembangan sistem ("ISO/IEC/IEEE International

Standard – Systems and software engineering – Software life cycle processes," 2017).

Berdasarkan konteks tersebut, penelitian ini bertujuan mengembangkan Sistem Logbook Kompetensi Pengawas Operasional berbasis web menggunakan framework Remult, yang tidak hanya meningkatkan akurasi, efisiensi, dan transparansi dalam manajemen kompetensi di LSP Tambang Minerba Indonesia, tetapi juga memenuhi prinsip-prinsip rekayasa perangkat lunak yang ditetapkan dalam ISO/IEC/IEEE 12207:2017. Dengan demikian, sistem ini diharapkan menjadi model referensi bagi lembaga sertifikasi profesi lain dalam menerapkan digitalisasi yang berstandar, berkelanjutan, dan berbasis bukti.

2. METODE PENELITIAN

Objek penelitian ini berfokus pada infrastruktur digital dan tata kelola pencatatan logbook harian kinerja serta pengembangan kapasitas diri bagi pemegang sertifikat kompetensi di bidang industri pertambangan mineral dan batu bara. Pengembangan difokuskan pada transformasi form manual menjadi arsitektur aplikasi berbasis web modern menggunakan framework SvelteKit dan Remult, yang mengimplementasikan prinsip data terpusat (Single Source of Truth / SSOT) guna memastikan reliabilitas database logbook serta validasi API yang konsisten. Untuk mendukung pengembangan tersebut, pengumpulan data dilakukan melalui tiga teknik utama, yaitu wawancara teknis (technical interview) dengan Manajer IT terkait spesifikasi infrastruktur server dan database, observasi langsung berupa pembedahan dokumen templat logbook di LSP Tambang Minerba Indonesia untuk mengekstrak entitas data mentah (seperti field Waktu, Nama Program, Lokasi, Status Peserta/Narasumber, dan Tanda Tangan Integritas) menjadi entity class TypeScript pada Remult, serta studi pustaka mendalam terhadap jurnal, artikel ilmiah, dokumentasi resmi full-stack type-safe framework, dan standar operasional ISO/IEC/IEEE 12207:2017 guna mengidentifikasi celah penelitian yang dapat diisi.

Metode pengembangan sistem yang digunakan dalam penelitian ini mengintegrasikan kerangka kerja Extreme Programming (XP) dengan Technical Processes dari standar ISO/IEC/IEEE 12207:2017. Fase pertama, yaitu Planning (Perencanaan), berfokus pada analisis komprehensif mengenai fungsi dan fitur inti aplikasi melalui tiga proses teknis ISO/IEC/IEEE 12207:2017: Business or Mission Analysis Process untuk menganalisis urgensi transformasi logbook manual menjadi sistem digital real-time, Stakeholder Needs and Requirements Definition Process untuk menggali kebutuhan non-fungsional dan teknis berdasarkan wawancara bersama Manajer IT serta regulasi yang berlaku, dan System/Software Requirements Definition Process untuk menetapkan spesifikasi

rekayasa perangkat lunak secara formal guna membatasi ruang lingkup implementasi pada lingkungan production.

Fase kedua, yaitu Design (Perancangan), memformulasikan antarmuka aplikasi dan fondasi arsitektur routing server melalui tiga proses teknis ISO/IEC/IEEE 12207:2017: Architecture Definition Process untuk menentukan konsep, elemen, dan relasi struktural software yang optimal sebagai landasan baku perancangan, Design Definition Process untuk mentransformasikan spesifikasi kebutuhan menjadi deskripsi teknis terperinci mencakup perancangan database relasional dan user interface, serta System Analysis Process yang mengevaluasi spesifikasi arsitektur melalui Technical Risk Assessment dan Trade-off Analysis untuk membuktikan kelayakan desain sebelum memasuki fase konstruksi kode.

Fase ketiga, yaitu Coding (Pengkodean), menerjemahkan spesifikasi desain ke dalam baris kode berstandar Clean Code melalui dua proses teknis ISO/IEC/IEEE 12207:2017: Implementation Process yang mengonversi hasil perancangan menjadi skema database, logika backend, dan komponen frontend, serta Integration Process yang menggabungkan subsistem perangkat lunak menjadi satu kesatuan fungsional menggunakan Version Control System, mekanisme Deployment, dan Cloud Storage terisolasi untuk penyimpanan berkas media bukti fisik asesi. Selanjutnya, fase keempat, yaitu Testing (Pengujian), membuktikan kualitas rekayasa melalui Verification Process dengan metode Black-box Testing untuk menguji logika input-output sistem secara internal, serta Validation Process berupa User Acceptance Testing (UAT) bersama 5 orang asesi, 1 admin, dan 1 asesor untuk memastikan produk sesuai dengan tujuan bisnis awal.

Fase terakhir, yaitu Release dan Software Increment, mengeksekusi peluncuran (deployment) serta pembaruan berkesinambungan melalui empat proses teknis ISO/IEC/IEEE 12207:2017: Transition Process untuk migrasi aplikasi dari lingkungan localhost ke server staging, Operation Process untuk mengoperasikan sistem di lingkungan target serta memantau kinerja komputasi dan interaksi pengguna secara runtime, Maintenance Process untuk penyusunan keandalan jangka panjang meliputi bug-patching, pembaruan dependency Node.js dan SvelteKit, serta pemeliharaan tabel MariaDB, serta Disposal Process berupa prosedur pengarsipan dan pemusnahan (decommissioning) file system maupun database secara aman guna menjaga kerahasiaan data pemegang sertifikat apabila aplikasi dimigrasikan atau dinonaktifkan secara permanen.

3. HASIL DAN PEMBAHASAN

Tahap perencanaan (planning) merupakan fase awal yang sangat krusial dalam metodologi eXtreme Programming (XP) untuk mendefinisikan ruang lingkup, kebutuhan fungsional, serta batasan-batasan sistem logbook kompetensi pada Lembaga Sertifikasi

Profesi Tambang Minerba Indonesia (LSP TMI). Penyelarasan fase perencanaan ini dengan standar internasional ISO/IEC/IEEE 12207:2017 diwujudkan secara sistematis melalui tiga sub-proses teknis utama, yaitu Business or Mission Analysis Process, Stakeholder Needs and Requirements Definition Process, dan System/Software Requirements Definition Process. Melalui Business or Mission Analysis Process, permasalahan operasional nyata berupa pengelolaan dokumen logbook manual yang terdesentralisasi—yang memicu risiko hilangnya bukti fisik unjuk kerja, keterlambatan verifikasi kepatuhan, serta ketidakakuratan data pemeliharaan sertifikat—dipetakan dan dievaluasi kelayakannya menggunakan analisis SWOT. Hasil analisis menunjukkan bahwa meskipun LSP TMI telah memiliki infrastruktur portal web yang stabil dan responsif selama dua tahun (Strengths), sistem eksisting masih memiliki kelemahan (Weaknesses) berupa ketiadaan portal khusus pencatatan harian dan sarana surveillance bagi asesor. Kondisi ini membuka peluang (Opportunities) untuk merancang logbook surveilans yang terintegrasi dengan skema kompetensi asesi, sekaligus memitigasi ancaman (Threats) seperti risiko serangan SQL Injection pada endpoint API publik dan potensi kehilangan data akibat penulisan manual.

Proses dilanjutkan dengan Stakeholder Needs and Requirements Definition Process untuk menyelaraskan kapabilitas teknis sistem dengan ekspektasi fungsional dari empat entitas utama melalui wawancara terstruktur dan observasi teknis bersama Manajer IT LSP TMI. Kebutuhan tersebut mencakup Regulator (BNSP) yang memandatkan adanya sistem surveillance sesuai Peraturan BNSP Nomor 2/BNSP/VIII/2017, Asesi yang memerlukan media terpusat untuk mendokumentasikan kegiatan keprofesian harian, Admin yang membutuhkan fasilitas pengelolaan data sertifikat serta instrumen pemantauan bagi asesor, hingga Asesor yang memerlukan sarana pembuktian valid dalam memutuskan kelayakan perpanjangan atau peningkatan level kompetensi asesi. Selanjutnya, System/Software Requirements Definition Process menetapkan spesifikasi teknis non-fungsional, batasan sistem, dan lingkungan operasional production. Kebutuhan non-fungsional menekankan efisiensi load time, kompatibilitas lintas browser, serta keamanan data berbasis token terenkripsi; sementara batasan sistem menetapkan batas ukuran berkas bukti fisik maksimal 5 MB dengan format PDF, PNG, JPG, atau JPEG. Lingkungan operasional dikonfigurasi pada Virtual Private Server (VPS) berarsitektur RAM 8 GB dan NVMe 100 GB, di mana berkas media dialihkan ke Cloudflare R2 object storage, serta menggunakan Ubuntu Server 24.04 LTS, Docker Engine yang diorkestrasi via Coolify, runtime Node.js v22.12, DBMS MariaDB v11, dan integrasi autentikasi REST API terpisah.

Memasuki tahap perancangan (design), spesifikasi kebutuhan ditransformasikan menjadi model arsitektur sistem, struktur basis data relasional, dan desain antarmuka pengguna di bawah kendali dua aktivitas teknis ISO/IEC/IEEE 12207:2017, yaitu Architecture Definition Process dan Design Definition Process. Sesuai dengan pedoman deskripsi arsitektur ISO/IEC/IEEE 42010:2011, arsitektur sistem diposisikan sebagai konsep dan properti fundamental yang memodelkan struktur komponen, alur data, dan perspektif infrastruktur secara komprehensif. Struktur komponen memisahkan interface reaktif di sisi frontend dengan komputasi backend melalui integrasi isomorphic ORM Remult berbasis TypeScript sebagai Single Source of Truth (SSOT) yang menerapkan Role-Based Access Control (RBAC). Alur pertukaran data (Data Flow) antarentitas (Asesi, Asesor, dan Admin) dimodelkan menggunakan Data Flow Diagram (DFD) dan enkapsulasi JSON stateless HTTP request, yang mencakup 10 alur utama meliputi autentikasi token via cookies, pengunggahan bukti fisik ke Cloudflare R2 via asynchronous stream, pengelolaan data logbook via Remult middleware dan MariaDB, evaluasi logbook oleh asesor, serta pengelolaan data master oleh admin. Perspektif infrastruktur (Infrastructure View) mengadopsi abstraksi PaaS self-hosted Coolify pada VPS Ubuntu, yang mengotomatisasi buildpack Docker Image menggunakan Nixpacks, mengelola lalu lintas HTTPS dan SSL Let's Encrypt melalui Traefik reverse proxy, mengisolasi basis data MariaDB dalam Docker Bridge Network, serta mengintegrasikan otorisasi lintas subdomain.

Pada Design Definition Process, spesifikasi teknis diklasifikasikan ke dalam pemodelan perilaku fungsional menggunakan Use Case Diagram serta perancangan logika alur kerja dinamis menggunakan Activity Diagram berbasis Unified Modeling Language (UML). Pemodelan Use Case menetapkan batas interaksi tiga aktor utama (Asesi, Asesor, dan Admin) di mana seluruh operasi esensial terikat secara inheren pada mekanisme autentikasi terpusat melalui relasi include menuju use case Login. Entitas Asesi berwenang meninjau sertifikat dan mengelola logbook yang didekomposisi melalui relasi extend (tambah, ubah, hapus, dan persetujuan integritas); Asesor berwenang meninjau detail asesmen dan memverifikasi logbook (setuju/tolak); sedangkan Admin memegang kontrol master atas daftar asesi, verifikasi logbook, serta manipulasi data sertifikat kompetensi. Untuk menjamin ketepatan eksekusi komputasional sebelum diimplementasikan ke antarmuka visual, dirancang empat activity diagram utama yang membagi tanggung jawab antara entitas pengguna (User) dan Sistem menggunakan pemodelan swimlane. Activity Diagram Login memodelkan alur autentikasi dua tingkat decision node yang memeriksa kelengkapan form input dan

verifikasi kredensial database sebelum menerbitkan token otorisasi via session management.

Selanjutnya, tiga activity diagram operasional dirancang secara terperinci untuk mengendalikan alur kerja manipulasi data, persetujuan komitmen, dan pencatatan harian. Activity Diagram Manajemen Data Sertifikat menguraikan operasi CRUD oleh Admin, di mana perintah penambahan dan pembaruan data melewati validasi integritas form, sedangkan perintah penghapusan dilindungi oleh confirmation dialog preventif sebelum kueri eksekusi dijalankan dan diakhiri dengan toast notification. Activity Diagram Persetujuan Integritas memodelkan mekanisme komitmen Asesi, di mana sistem mengevaluasi state integrity_agreed di database; jika belum disetujui, Asesi diwajibkan menyetujui form komitmen melalui decision node agar rute akses menuju tab logbook kinerja terbuka. Terakhir, Activity Diagram Tambah dan Edit Data Logbook memodelkan pengelolaan kegiatan harian oleh Asesi yang diawali dengan verifikasi hak akses tingkat database. Pengisian form dan pengunggahan berkas bukti fisik diproses secara sekuensial melalui endpoint /api/upload dan dikirim sebagai payload JSON, di mana validasi server-side memastikan kepatuhan kriteria sebelum kueri Create atau Update dieksekusi di database MariaDB. Seluruh rangkaian perancangan sistem ini memberikan kerangka kerja rekayasa perangkat lunak yang terstruktur, aman, dan dapat dipertanggungjawabkan secara ilmiah.

Prosedur operasional evaluasi data logbook asesi dimodelkan secara sistematis melalui Activity Diagram Verifikasi Logbook yang memisahkan domain tanggung jawab antara Admin/Asesor dan Sistem menggunakan pemodelan swimlane. Alur komputasional diawali dari initial node pada partisipasi pengguna menuju decision node pertama untuk memeriksa tingkat otorisasi akun; peran Admin secara otomatis diarahkan ke endpoint /participants untuk memuat halaman daftar asesi, sedangkan peran Asesor dialihkan ke endpoint /assessments untuk memuat jadwal asesmen. Kedua jalur navigasi tersebut berkonvergensi pada instruksi membuka detail sertifikat atau rekaman logbook peserta, yang direspons oleh Sistem dengan fetching data dari database MariaDB serta merender riwayat kegiatan dan dokumen lampiran secara visual. Setelah peninjauan komprehensif dilakukan, keputusan persetujuan (approve) atau penolakan (reject) ditransmisikan kembali ke Sistem untuk dievaluasi melalui decision node kedua guna memperbarui atribut status rekaman secara fisik menjadi approved atau rejected. Siklus penilaian ini ditutup dengan prosedur pengamanan berupa penguncian hak akses edit pada dokumen yang telah dinilai, diikuti pemunculan toast notification dan penyegaran tabel secara reaktif hingga mencapai end state node.

Spesifikasi interaksi teknis pada tingkat mikro dimodelkan secara terperinci menggunakan Sequence Diagram untuk menggambarkan pertukaran pesan

(message passing) berbasis urutan waktu (time-ordered interaction) antar-komponen perangkat lunak. Ekosistem Sistem Logbook Asesi LSP TMI mengintegrasikan komunikasi dinamis antara antarmuka client-side berbasis Svelte, controller logika bisnis Remult, External Auth API, MariaDB, dan Cloudflare R2 object storage. Pada Sequence Diagram Login, pengiriman kredensial melalui rute /login diproses oleh backend SvelteKit yang bertindak sebagai proxy menuju Auth API eksternal pada rute /api/login; jika valid, token otorisasi disematkan ke dalam cookie sesi (authToken) dan memicu instruksi pengalihan redirect 303 ke dashboard, sementara kegagalan validasi atau batasan otorisasi hak akses (Role-Based Access Control) akan menggugurkan pembuatan sesi dan merender pesan error secara reaktif. Pada modul Detail Sertifikat, Sequence Diagram Persetujuan Integritas memodelkan pemuatan data di mana state integrity_agreed: false menahan rute utama dan menayangkan form Pakta Integritas (IntegrityCard); persetujuan asesi memicu pemanggilan metode repo(Certificate).save() untuk memperbarui atribut di MariaDB menjadi true, yang secara otomatis menyembunyikan form komitmen dan membuka akses penuh menuju tab logbook.

Pengelolaan data kegiatan harian dimodelkan dalam Sequence Diagram Pengelolaan Logbook yang mencakup operasi Create, Read, Update, dan Delete (CRUD) secara sekuensial. Pembacaan data diawali dengan pemanggilan fungsi asinkron subscribe liveQuery pada entitas LogbookRecord untuk membuka koneksi aliran data real-time berbasis kueri SQL SELECT. Operasi penambahan (Create) memproses pengunggahan berkas bukti fisik melalui rute /api/upload menuju Cloudflare R2, disusul eksekusi repo.insert() yang mentranslasikan payload menjadi kueri INSERT serta merespons dengan status HTTP 201 Created. Operasi pembaruan (Update) memuat data historis untuk dimodifikasi dan diperbarui di database via repo.update() dengan status HTTP 200 OK, sedangkan operasi penghapusan (Delete) didahului oleh confirmation dialog preventif sebelum metode repo.delete() mengeksekusi kueri DELETE secara definitif pada database. Pada sisi evaluasi, Sequence Diagram Verifikasi Logbook memodelkan tindakan Asesor/Admin yang memicu kueri SQL modifikasi untuk mengubah status rekaman dan menyegarkan antarmuka, sementara Sequence Diagram Manajemen Sertifikat menguraikan kendali Admin dalam mengeksekusi operasi insert() (HTTP 201 Created), update() (HTTP 200 OK), dan delete() (HTTP 200 OK / 204 No Content) pada master data sertifikat.

Struktur fisik dan relasi penyimpanan data dirancang secara sistematis melalui Class Diagram dan kamus data terstruktur untuk menjamin integritas relasional serta mencegah redundansi pada database MariaDB. Ekosistem basis data ini terdiri dari

sepuluh entitas utama yang saling terhubung: users sebagai entitas identitas terpusat berskema UUID; roles yang menyimpan konfigurasi flags hak akses granular (seperti assessment_inspection dan certificate_management); participants dan assessors yang terikat secara One-to-One dengan users; schemes dan assessment_places sebagai data referensi skema dan TUK; serta assessments yang menghubungkan sesi ujian dengan asesor dan skema secara Many-to-One. Selanjutnya, entitas certificates memegang informasi registrasi BNSP beserta flag integrity_agreed, yang memiliki relasi One-to-Many menuju entitas logbook_records. Tabel logbook_records mengapsulasi data kegiatan harian (kinerja atau pengembangan diri) lengkap dengan atribut role, lokasi, pimpinan tim, status verifikasi (created, approved, rejected), serta array URL attachments yang menunjuk ke object storage Cloudflare R2.

Perancangan antarmuka pengguna (User Interface) ditransformasikan menjadi spesifikasi visual yang terstruktur mengadopsi pendekatan design system berbasis framework Svelte 5, Tailwind CSS, dan komponen Shadcn-Svelte. Bahasa visual aplikasi menerapkan gaya modern minimalis yang responsif lintas perangkat, dengan standar tipografi menggunakan font Inter serta skema warna primer hijau emerald sesuai identitas korporat LSP TMI. Halaman Login menyediakan gerbang masuk terenkripsi bagi Asesi, Admin, dan Asesor; Halaman Utama Asesor dan Detail Asesmen menyajikan kartu informasi sertifikasi beserta tabel peserta ujian; sedangkan Halaman Daftar Asesi dan Riwayat Sertifikat memberikan kontrol navigasi master bagi Admin untuk mengelola data keprofesian. Terakhir, Halaman Detail Sertifikat dan Pengisian Logbook menyediakan antarmuka khusus bagi Asesi untuk mencatat aktivitas kinerja maupun pengembangan diri (sebagai narasumber atau peserta), melampirkan berkas bukti fisik, serta memantau status verifikasi secara real-time setelah Pakta Integritas disetujui.

Berdasarkan klausul Technical Processes pada standar ISO/IEC/IEEE 12207:2017, tahapan System Analysis Process bertujuan mengevaluasi spesifikasi arsitektur secara teknis guna menyediakan basis data analitis dan justifikasi rekayasa (engineering justification) sebelum memasuki fase implementasi. Kebutuhan analisis diidentifikasi berdasarkan parameter risiko operasional sistem sertifikasi digital, khususnya terkait lonjakan beban penyimpanan, isolasi lingkungan server, dan integritas tipe data. Evaluasi divalidasi menggunakan pengujian lingkungan komputasi terisolasi melalui dua pendekatan utama: Technical Risk Assessment dan Trade-off Analysis. Penilaian risiko teknis mengidentifikasi potensi kegagalan seperti bottleneck I/O akibat unggahan masif, bentrokan dependency, dan ketidaksesuaian payload API, yang masing-masing dimitigasi melalui penggunaan asynchronous stream handler ke Object Storage, pembungkusan

lingkungan eksekusi ke dalam container terisolasi, serta penegakan type safety menggunakan shared code entity bridge. Hasil analisis trade-off mengonfirmasi bahwa konsekuensi teknis seperti ketergantungan latensi pihak ketiga pada Cloudflare R2, alokasi RAM ekstra untuk Docker, dan keterbatasan fleksibilitas query SQL kustom pada Remult berada dalam batas toleransi yang sepadan dengan efisiensi komputasi yang diperoleh. Keterlacakan hasil analisis ini menjamin struktur data terintegrasi secara otomatis antara backend dan frontend, sekaligus mengeliminasi anomali relasional saat validasi logbook sehingga menjadi basis konklusi resmi untuk menyetujui transisi proyek dari fase perancangan menuju tahap penulisan kode.

Memasuki fase Coding melalui Implementation Process sesuai standar ISO/IEC/IEEE 12207:2017, seluruh spesifikasi desain arsitektur dan model data diterjemahkan secara sistematis menjadi program fungsional pada lapisan database, backend, dan user interface. Pada lapisan database dan backend, setiap entitas seperti User, Participant, Scheme, Assessment, Certificate, dan LogbookRecord dimodelkan ke dalam kelas TypeScript menggunakan ORM Remult yang terhubung dengan MariaDB. Remult bertindak sebagai abstraksi untuk mengatur foreign key, validasi server-side, penerapan Row-Level Security (RLS) via apiPrefilter guna membatasi hak akses data secara granular, serta pemanfaatan valueConverter kustom untuk serialisasi data JSON. Konfigurasi backend dieksekusi di lingkungan SvelteKit pada rute /api/[...remult]/+server.ts dengan memanfaatkan Knex sebagai data provider untuk mentransformasikan query ORM ke MariaDB, disertai mekanisme autentikasi getUser yang memverifikasi token cookie ke API utama.

Pengembangan User Interface (UI) diimplementasikan secara modular memanfaatkan kapabilitas reaktif SvelteKit dan Svelte 5 untuk membangun antarmuka yang dinamis, aman, dan responsif. Tata letak aplikasi diatur melalui Layout Utama (+layout.svelte) yang mengelola sinkronisasi konteks autentikasi global dan kerangka tata letak header, serta Layout Breadcrumb yang memantau penanda navigasi hierarkis secara reaktif menggunakan animasi transisi yang halus. Antarmuka mencakup berbagai halaman fungsional utama, dimulai dari Halaman Login yang mengintegrasikan fitur use:enhance untuk form submission progresif tanpa reload, Halaman Daftar Asesi dan Halaman Detail Asesi yang menampilkan profil peserta real-time beserta riwayat sertifikat, hingga Halaman Daftar Asesmen dan Halaman Detail Asesmen yang menyajikan kartu ringkasan serta tabel penugasan jadwal evaluasi kompetensi bagi asesor. Selain itu, Halaman Detail Sertifikat & Logbook mengimplementasikan kontrol integritas yang ketat; sistem mengunci area pengisian catatan pekerjaan hingga asesi menyetujui pakta integritas, serta secara

otomatis membekukan entri logbook yang telah berstatus approved atau rejected agar tidak dapat dimanipulasi secara manual.

Rangkaian alur pengembangan diakhiri dengan Integration Process yang mengorganisir dan menyatukan seluruh elemen komponen teknis menjadi satu kesatuan produk aplikasi final yang siap beroperasi di lingkungan produksi. Pengelolaan kode sumber disinkronisasikan menggunakan Version Control System (VCS) Git yang terhubung ke repositori GitHub, memungkinkan pelacakan riwayat perubahan secara presisi, memfasilitasi kolaborasi antar-pengembang, dan menyediakan pemicu deployment otomatis berbasis percabangan (branch). Untuk infrastruktur deployment, aplikasi dijalankan di atas Virtual Private Server (VPS) yang dikelola melalui control panel open-source Coolify, sehingga alur build dan pembaharuan aplikasi terintegrasi secara langsung dengan repositori Git tanpa memerlukan penyusunan workflow manual yang kompleks. Sementara itu, manajemen penyimpanan berkas unggahan pengguna ditangani secara terpisah menggunakan layanan cloud object storage Cloudflare R2 yang kompatibel dengan API AWS S3; integrasi ini dieksekusi melalui pustaka S3 SDK untuk Node.js, sehingga seluruh dokumen portofolio dapat diunggah langsung ke endpoint bucket eksternal tanpa membebani kapasitas penyimpanan lokal maupun kinerja I/O pada server VPS utama.

Pengembangan Sistem Informasi Logbook Kompetensi LSP-TMI mencapai hasil validasi yang sangat tinggi dengan persentase kelayakan User Acceptance Testing (UAT) sebesar 89,52%. Mengacu pada tabel interpretasi hasil penelitian Aliyah dkk. (2025), pencapaian ini berada pada kuartil teratas (interval 81%–100%) yang mengklasifikasikan sistem ke dalam kategori “Sangat Layak” atau “Sangat Baik” secara objektif, membuktikan bahwa integrasi antarmuka dan logika bisnis aplikasi telah memfasilitasi kebutuhan seluruh pengguna secara presisi. Transisi dari siklus iterasi Extreme Programming (XP) menuju operasional nyata diawali melalui Transition Process berbasis standar ISO/IEC/IEEE 12207:2017 untuk memindahkan aplikasi dari lingkungan development ke lingkungan produksi secara terstruktur. Infrastruktur diimplementasikan pada Virtual Private Server (VPS) terdedikasi (RAM 8GB, penyimpanan 100GB) menggunakan containerisasi Docker dan Platform as a Service (PaaS) Coolify untuk mencegah konflik dependensi serta menyediakan isolasi keamanan data sertifikasi. Pipeline otomatis dikonfigurasi melalui integrasi GitHub dan Coolify pada branch staging, di mana pembaruan kode memicu buildpack Nixpacks untuk memetakan runtime, mengompilasi citra kontainer, dan menginjeksikan variabel lingkungan secara aman. Setelah kontainer aktif, aplikasi dipublikasikan pada URL staging yang terhubung terpusat ke basis data MariaDB, API autentikasi utama LSP-TMI, serta

cloud storage Cloudflare R2, menandai kesiapan operasional sistem untuk pengujian pengguna.

Memasuki tahap pengoperasian berkesinambungan melalui Operation Process, sistem dijalankan secara terpusat dalam kontainer Docker yang dikelola via Coolify untuk menjamin ketersediaan layanan aplikasi SvelteKit dan backend Remult dari gangguan eksternal. Pemantauan performa dilakukan secara sistematis melalui fitur Metrics pada Coolify yang memperbarui data setiap 5 menit, serta pemantauan pemanfaatan sumber daya VPS secara menyeluruh melalui dasbor penyedia hosting. Untuk memantau stabilitas aplikasi dan mengidentifikasi anomali secara real-time, pustaka Sentry diintegrasikan pada berkas `hooks.client.ts` dan `hooks.server.ts`. Di sisi client, Sentry menangkap error browser melalui `constraint handleError`, sedangkan di sisi server, Sentry diinisialisasi menggunakan Data Source Name (DSN) dari variabel lingkungan untuk mencatat log operasional dan menangkap penanganan Internal Server Error (500) melalui fungsi `sentryHandle()`. Seluruh data analitik dan laporan error dihimpun secara terpusat ke dalam dasbor insights Sentry untuk memudahkan tim pengembang menganalisis serta menindaklanjuti isu teknis.

Keberlanjutan sistem di lingkungan staging dijaga melalui Maintenance Process yang mencakup tindakan preventif, korektif, dan mekanisme pembaruan aman. Pemeliharaan preventif diimplementasikan tanpa mengubah kode sumber, melainkan memanfaatkan fitur backup otomatis mingguan dari penyedia VPS untuk menduplikasi seluruh keadaan server, sehingga pengembalian sistem (rollback) dapat dilakukan seketika jika terjadi kegagalan fatal. Pemeliharaan korektif dilakukan saat ditemukan kekeliruan atau kebutuhan penyempurnaan fitur; tim pengembang melakukan perbaikan lokal lalu mengirimkan (push) kode terbaru ke repositori staging. Proses penerapan pembaruan (deployment) ditangani otomatis oleh Coolify di latar belakang untuk menghindari hambatan operasional (zero downtime), lengkap dengan proteksi kegagalan otomatis yang menolak pembaruan dan mempertahankan versi stabil sebelumnya apabila kompilasi kode mengalami build error.

Rangkaian siklus hidup perangkat lunak ditutup dengan Disposal Process dan evaluasi metodologi untuk mengakhiri penggunaan sumber daya staging secara aman. Prosedur dekomisioning diawali dengan ekstraksi basis data MySQL/MariaDB staging menggunakan HeidiSQL menjadi berkas dump SQL terarsip, dilanjutkan pengasihan (backup) instansi Coolify ke bucket penyimpanan terpisah pada Cloudflare R2 via protokol S3. Selanjutnya, penghentian (termination) kontainer SvelteKit dan pembersihan volume persistent Docker dieksekusi melalui Danger Zone pada Coolify untuk membebaskan alokasi RAM,

ruang penyimpanan, serta konfigurasi jaringan/SSL pada VPS. Seluruh tahapan rekayasa ini divalidasi melalui matriks kesesuaian metodologi yang memetakan tahapan SDLC Extreme Programming (Perencanaan, Perancangan, Pengkodean, Pengujian, Implementasi) terhadap klausul Technical Processes ISO/IEC/IEEE 12207:2017 (mulai dari 6.4.1 hingga 6.4.12). Pemetaan ini membuktikan keterlacakan (traceability) penuh antara perencanaan kualitatif dengan bukti penerapan faktual—seperti black-box testing 30 skenario ber-metrik kelulusan 100%, tingkat penerimaan UAT 90%, hingga prosedur dekomisioning yang sistematis.

4. KESIMPULAN

Penelitian Tugas Akhir dengan judul “Implementasi Remult dalam Pengembangan Sistem Logbook Kompetensi Sesuai Technical Processes ISO/IEC/IEEE 12207:2017 Di LSP Tambang Minerba Indonesia” telah berhasil diselesaikan secara utuh untuk memenuhi salah satu syarat kelulusan Program Studi Teknik Informatika Universitas Esa Unggul. Berdasarkan hasil analisis, perancangan, implementasi, serta pengujian fungsional yang telah dilakukan pada bab-bab sebelumnya, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Aplikasi Logbook Asesi LSP TMI berhasil dikembangkan menggunakan Remult dengan mengacu pada Technical Processes dalam pedoman ISO/IEC/IEEE 12207:2017
2. Pemanfaatan library Remult dalam penelitian ini berhasil memenuhi verifikasi ditandai dengan Black-box testing yang 100% Lulus. Hal ini membuktikan bahwa Remult layak diandalkan dalam pengembangan software full-stack walaupun belum banyak digunakan.
3. Aplikasi berhasil di-deploy di server VPS dan berstatus online dengan eksekusi kode menggunakan runtime Node.js.
4. Aplikasi Logbook Asesi LSP TMI berhasil memenuhi kebutuhan stakeholders. Hal ini ditunjukkan oleh hasil UAT yang sebagian besarnya sangat setuju dengan hasil penerimaan sebesar 89,52%.
5. Penelitian ini berhasil merencanakan dan menerapkan Technical Processes berdasarkan ISO/IEC/IEEE 12207:2017 yang dipadukan dengan siklus hidup pengembangan sistem yaitu Extreme Programming (XP)

5. REFERENSI

- Aliyah, A., Hartono, N., & Muin, A. A. (2025). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch: Jurnal Sains Dan Teknologi Informasi*, 3(2), 42–58.
<https://doi.org/10.62951/switch.v3i1.330>
- Anis, Y. Y., Mulyani, S., Listiyono, H., & Ningtyas, S. A. (2022). *Sistem Informasi Log Book*

- Harian Peserta Magang Pada Dinas Kominfo Kota Semarang. *Bulletin of Information Technology (BIT)*, 3(2), 128–133. <https://doi.org/10.47065/bit.v3i2.285>
- Bestin, B., Addiansyah, M. G., Sofian, R., Ferdiansyah, F. R., & Wafa, M. N. (2023). Sistem Informasi Monitoring Peserta Magang Berbasis Web Pada Divisi Sales & Business Development Di Mayar Kota Bandung. *Technologia: Jurnal Ilmiah*, 14(2), 163–170. <https://doi.org/10.31602/tji.v14i2.10424>
- Bryantsev, S. (2025). THE SINGLE SOURCE OF TRUTH PARADIGM AS A TOOL FOR SUPPORTING SOFTWARE MAINTENANCE.
- Celander, E. K., Möllestål, A., & Kurti, A. (2024). A Comparative Analysis of Next.js, SvelteKit, and Astro for E-commerce Web Development. Linnaeus University.
- Deviana, J. A., Rahmawati, W. M., Setiowati, Y., & Santoso, D. S. (2023). Sistem Informasi Lembaga Sertifikasi Profesi Polimarin Semarang. *KERNEL: Jurnal Riset Inovasi Bidang Informatika Dan Pendidikan Informatika*, 4(1), 24–33. <https://doi.org/10.31284/j.kernel.2023.v4i1.4911>
- Fadilla, R. M., & Ariatmanto, D. (2025). AHP-TOPSIS AND ANOVA METHOD APPROACH IN SOFTWARE DEVELOPMENT CRITERIA SELECTION ACCORDING TO ISO 12207:2017. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 11(3), 397–404. <https://doi.org/10.33330/jurteks.v11i3.3698>
- Fasih, F. A., & Nisa, K. (2023). Sistem Informasi Logbook Berbasis Web Pada Stasiun Geofisika Kelas I Deli Serdang. *Jurnal Nasional Teknologi Komputer*, 3(4), 314–325. <https://doi.org/10.61306/jnastek.v3i4.117>
- Fenardi, O., & Lee, F. S. (2023). Aplikasi Akademik Berbasis Website Menggunakan Metode Extreme Programming Pada SMAN1 Belinyu. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(4), 440–447. <https://doi.org/10.47233/jteksis.v5i4.843>
- Friatna, A. A., Putra, N. S., Roekminiati, S., Pramono, S., & Sunarya, A. (2025). Inovasi Asesmen Digital: Pengabdian Masyarakat dalam Pengembangan Sistem Asesmen Jarak Jauh pada Lembaga Sertifikasi MSDM Universal Surabaya. *Aksi Kita: Jurnal Pengabdian Kepada Masyarakat*, 1(4), 664–671. <https://doi.org/10.63822/2ayppe03>
- Hermansyah, H. Y., & Maryam, M. (2023). IMPLEMENTASI TEKNOLOGI APPLICATION PROGRAMMING INTERFACE PADA PERANCANGAN APLIKASI ABSENSI PEGAWAI. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(3), 744–754. <https://doi.org/10.29100/jipi.v8i3.3890>
- ISO/IEC/IEEE International Standard—Systems and software engineering – Software life cycle processes. (2017). ISO/IEC/IEEE 12207:2017 First edition 2017-11, 1–157. <https://doi.org/10.1109/IEEESTD.2017.8100771>
- ISO/IEC/IEEE Systems and software engineering – Architecture description. (2011). ISO/IEC/IEEE 42010:2011 (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), 1–46. <https://doi.org/10.1109/IEEESTD.2011.6129467>
- Kurnia, A. A., & Sutedi, S. (2025). IMPLEMENTASI SISTEM INFORMASI LAYANAN LEMBAGA SERTIFIKASI PROFESI (LSP) IIB DARMAJAYA BERBASIS DIGITAL INNOVATION. *Jurnal Khatulistiwa Informatika*, 13(2), 113–122. <https://doi.org/10.31294/jki.v13i2.11390>
- Lamine, Y., & Cheng, J. (2022). Understanding and Supporting the Design Systems Practice (arXiv:2205.10713). arXiv. <https://doi.org/10.48550/arXiv.2205.10713>
- Montoya-Murillo, D. A., Galvan-Cruz, S., Mora, M., & Andrade, E. L. M. (2025). A Comprehensive Review of the Main Heavyweight and Lightweight SDLC for Big Data Analytics Systems (BDAS). *IEEE Access*, 13, 101328–101367. <https://doi.org/10.1109/ACCESS.2025.3577970>
- Muhammad, I. R. D., & Paputungan, I. V. (2024). Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 3(2), 79–87. <https://doi.org/10.20885/snati.v3i2.35>
- Nguyen, T. H. (2023). EXPLORING JAVASCRIPT AND ITS RELATED LANGUAGES, A Comparative Analysis with TypeScript, CoffeeScript, ActionScript, Dart, and Elm.
- Node.js Foundation. (2025). Node.js—About Node.js®. <https://nodejs.org/en/about> NPM. (2025). Npm Docs. <https://docs.npmjs.com/>
- PERATURAN BADAN NASIONAL SERTIFIKASI PROFESI NOMOR: 2/ BNSP/VIII/2017 TENTANG PEDOMAN PENGEMBANGAN DAN PEMELIHARAAN SKEMA SERTIFIKASI PROFESI (2017).
- Queiroz, M., Tallon, P., & Coltman, T. (2024). Data Value and the Search for a Single Source of Truth: What is it and Why Does it Matter? *Hawaii International Conference on System Sciences*. <https://doi.org/10.24251/HICSS.2024.795>
- Remult. (2025). <https://remult.dev/>

- Remult. (2026). Remult/remult [TypeScript]. <https://github.com/remult/remult> (Original work published 2017)
- Rhamanti, D. (2024). RANCANG BANGUN APLIKASI SMARTCERTIFY BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS: LSP MICROFINANCE INDONESIA) [Other, Sekolah Tinggi Teknologi Terpadu Nurul Fikri]. <https://repository.nurulfikri.ac.id/id/eprint/674>
- Romahdoni, M. R., Prikurnia, A. K., & Kurniawan, R.-. (2024). PENERAPAN SISTEM REGISTRASI UJI KOMPETENSI LSP PERDAGANGAN RITEL LAMPUNG. *Journal of Software Engineering and Technology.*, 4(1), 50. <https://doi.org/10.69769/seat.v4i1.177>
- Sri Rumini, M. S. A., & Sulistiyani, E. (2025). Pengembangan Sistem Informasi LSP Indohusada Berbasis Web Menggunakan Design Science Research (DSR). *Journal of Information System, Graphics, Hospitality and Technology*, 6(2), 92–101. <https://doi.org/10.37823/insight.v6i2.396>
- Wijaya, T. (2025). TEKNIK CODING HYBRID ANTARA TYPESCRIPT DAN JAVASCRIPT DALAM FRAMEWORK ANGULAR. *Information System Journal*, 8(01), Article 01. <https://doi.org/10.24076/infosjournal.2025v8i01.2048>
- Wulandari, W., Nofiyani, N., & Hasugian, H. (2023). USER ACCEPTANCE TESTING (UAT) PADA ELECTRONIC DATA PREPROCESSING GUNA MENGETAHUI KUALITAS SISTEM. *Jurnal Mahasiswa Ilmu Komputer*, 4(1), 20–27. <https://doi.org/10.24127/ilmukomputer.v4i1.3383>