



Ramadhani Nur Sarjito

Fullstack Developer

✉ rama09n@gmail.com
🐙 github.com/rdsarjito
🌐 linkedin.com/in/ramadhani-nur-sarjito
🔗 ramadhaninursarjito.dev
📍 Jakarta Selatan, Indonesia

Summary

I am a dedicated Full Stack Developer with a strong foundation in web and mobile development. I enjoy building scalable applications from scratch, solving complex challenges, and improving user experience through clean code and thoughtful architecture. Skilled in modern technologies such as TypeScript, React, Next.js, Vue/Nuxt, Node.js, Bun.js, Laravel, PostgreSQL, MySQL, and MongoDB, I thrive in creating impactful solutions across both frontend and backend.

Education

Dicoding Indonesia Feb - Jun 2020 | Jakarta

Front-End Web Development

Student

RevoU 16 Feb - 30 Jun 2024 | Jakarta

Software Development

Fullstack Division Team Leader

Universitas Mercu Buana 2021 - 2025 | Jakarta

Bachelor of Informatics Engineering

GPA of 3.88/4.00

Work Experience

Full Stack Web Developer - Internship Feb – Mei 2024 | Jakarta

PT. Traspac Makmur Sejahtera

- Designed visually appealing website interfaces aligned with brand guidelines, producing attractive and responsive user interfaces.
- Create a website for inputting military data using Vue.JS and Laravel.

Fullstack Web Developer - Freelance Feb – Sep 2025 | Jakarta

PT. Bank Rakyat Indonesia (Persero) Tbk

- Creating the architectural considerations of building a full-stack app with a monorepo using Next.js (App Router, TypeScript, Tailwind, shadcn/ui, Zustand) on the front end and Elysia (Bun/Node) on the backend with REST APIs and Socket.IO.
- Connect a front-end Next.js app to an Elysia backend via REST endpoints and real-time Socket.IO channels.
- Communicate data from a MySQL database to the React application
- Understand how to route user requests on the front end with Next.js App Router and on the backend with Elysia route handlers/controllers.
- Build reusable user inputs with the shared UI system.
- Enhance authentication flows with stateless JWT authentication, including middleware enforcement on the backend and protected routes/hooks on the frontend.
- Separate production and development resources using environment variables for API base URLs, database credentials, and JWT_SECRET, with Bearer token handling for API access.
- Enable real-time features and live updates using Socket.IO (socketServer.ts on the server and useSocket.ts on the client).

Fullstack Web Developer - Contract Okt – Des 2025 | Jakarta

PT. SUCOFINDO

- Developed SIOS (Sistem Informasi Operasi Serco) using Laravel 10 and PHP 8.1 to manage ISO and SNI certification workflows with end-to-end application processing from customer submission to certificate issuance.
- Built multi-stage workflow management system for certification applications (Draft → Deal → Loss) and quotation approval processes (Draft → Kortek → Review → Deal) with automated status transitions and Laravel Queue Jobs for email

notifications.

- Integrated with Touchpoint customer portal system to synchronize certification application data and developed REST API integrations with ERP system (SCI API) for real-time customer data synchronization, order management, and SSO authentication.
- Developed data synchronization service to map and transform complex nested data structures from Touchpoint database to SIOS internal structure, handling customer, location, product, and certification schema mappings.
- Built certificate management system with automated certificate generation using DomPDF, QR code generation for verification, Excel export functionality using Maatwebsite Excel, and comprehensive reporting system with filtering capabilities.
- Implemented role-based access control (RBAC) with dynamic module management, multi-language support system (Indonesian, English, French, Philippine), and responsive admin dashboard with customizable data tables.
- Developed SILK (Sistem Informasi Legalitas Kayu) using Laravel 12 and PHP 8.2 for timber and forestry product certification compliance with LVLK standards, managing SLK, Non-SLK, and PPDVL certificates.
- Implemented SOAP API integration with VLegal system (Ministry of Environment and Forestry) for certificate status verification, data submission, compliance checking, and developed Web Service Activity module for tracking all API interactions.
- Developed blanko (certificate blank) management system with request workflow, activation/deactivation processes, mutation tracking (KB, KG, KO), and transaction management system for blanko requests with file upload/download capabilities.
- Built modern responsive UI using Tailwind CSS 4.0 and Vite build system, implemented reporting system with VLegal data integration including income reports, inspection reports, and PEB recapitulation, and created PDF generation system using Spatie Laravel PDF.

Fullstack Web Developer - Contract

Jan – Jun 2026 | Jakarta

PT Kereta Cepat Indonesia China

- Developed Frequent Whoosher Card (FWC) system using Elysia.js (Bun runtime) and Next.js with TypeScript to manage transit card lifecycle for high-speed railway services, covering card generation, distribution, sales, usage tracking, and expiration across multiple station points.
- Built comprehensive card inventory management system with stock movement workflows (Stock-In → Distribution → Transfer), inter-station card transfers with BAST and Nota Dinas document generation, stock opname reconciliation (physical vs system count), and automated low-stock alert notifications via Telegram Bot and Firebase Cloud Messaging.
- Developed card purchase transaction module with EDC payment integration, bulk purchase support with quantity-based discount rules, payment method/channel management, purchase cancellation/refund workflow, and automated invoice generation using PDFKit with barcode printing (bwip-js).
- Implemented card redemption (usage) system for train ticket issuance with passenger identity capture, quota tracking per card, usage log history, cancel redeem workflow, and task delegation mechanism allowing secondary operators to process transactions on behalf of primary users.
- Built ticket sales matching and reconciliation engine to auto-match external ticket sales reports (from KAI system) against internal redeem records, with overredeem detection, expired card usage flagging, invalid route/seat class validation, and settlement workflow for finalizing matched transactions.
- Developed card swap request workflow system (Request → Approve → Execute → Reject) for card replacements across stations, with full audit trail tracking and cross-station swap support between source and target stations.
- Implemented role-based access control (RBAC) with dynamic permission-menu mapping, JWT authentication with password reset tokens, device registration and locking per station, Cloudflare Turnstile bot protection, and Firebase App Check integration for mobile security.
- Built shift management system with configurable shift schedules (start/end times, break sessions), shift logging per device, and user delegation system for operational handover between main and secondary operators during shift transitions.
- Developed member registration module with KTP (Indonesian ID card) OCR detection using Python scripts, identity validation, and member-card association management supporting multiple program types (FWC and Voucher programs).
- Built comprehensive reporting system with ticket sales reports, quantity reports, stock summary dashboards, Excel export functionality (xlsx), activity audit logs, and data visualization for operational metrics across all stations.
- Implemented CI/CD pipeline using GitHub Actions for automated deployment to development and production environments, covering backend, frontend, and email microservice with separate workflow configurations.
- Developed dedicated email microservice using Express.js and Nodemailer with SMTP integration, featuring Handlebars HTML email templating for purchase confirmation and voucher bulk purchase notifications, file attachment support, email delivery logging with status tracking (SENT/FAILED), resend capability for updated member emails, and comprehensive email log module with filtering by member, purchase, date range, and delivery status.

Project

PAKSA (Pengelola Kas Siswa):

- Explore a full-stack architecture based on CodeIgniter (MVC) with an AdminLTE/Bootstrap and jQuery UI.
- Connect the front-end layer with the PHP CodeIgniter backend.
- Communicate MySQL data to views via CodeIgniter Models.
- Understand user request routing via routes and. Build reusable input with CodeIgniter Form Validation.
- Enhance authentication flows using sessions and access protection based on login status.
- Separate development and production resources config.php, database.php, constants.php, managing base_url, encryption_key, and DB credentials.
- Provide dashboards and transaction history reports for monitoring and decision-making.

Brushing Teeth Reminder:

- Design the architecture of a cross-platform Flutter app.
- Connect Flutter UI to local notification services using flutter_local_notifications for daily reminders.
- Persist simple on-device data with shared_preferences for reminder time and toggles.
- Handle permissions and battery optimization ensuring reliable alerts with Android battery optimization handling.
- Route user flows with Flutter navigation using Navigator and named routes for home, settings, and scanner.
- Build reusable input components including buttons, time pickers, and reminder controls.
- Add optional QR scanning with qr_code_scanner for custom habit flows.
- Separate development and production configs adjusting icons, permissions, and notification behavior per environment.

Covid-19:

- Design a modular front-end architecture with Webpack, where the code is organized into components/, data/, utils/, and presentations/ to ensure readability and scalability.
- Connect the UI to a COVID-19 data source by fetching data through data-covid and distributing it to the presentation components.
- Coordinate components via an event bus using a simple pub/sub to reduce tight coupling between components.
- Build reusable country search inputs through covid-country-search.js and covid-country-input-search.js, which enable country search and filtering.
- Compose list and summary views with covid-country-list.js, covid-country-item.js, and covid-box-summary.js, which separate the item, list, and summary layers.
- Improve numeric readability by using format.js to format case, death, and recovery counts.
- Provide component-driven navigation without a router through nav-bar-covid.js, which handles lightweight navigation without React Router.
- Split development and production builds with webpack.dev.js, webpack.prod.js, and webpack.common.js, which configure bundling, assets, and optimizations for each environment.

D-Budget:

- Create architectural considerations for a full-stack app by establishing a clear separation between a React frontend and a Node.js/Express backend, while maintaining a shared contract through REST APIs and MongoDB schemas.
- Connect a Create React App frontend to a Node/Express backend by dispatching frontend actions to backend routes under routes/ using fetch or axios, and managing state with Redux.
- Communicate data from MongoDB to React through Mongoose models that persist data, expose it via Express controllers, and make it consumable by React components and containers.
- Enhance authentication with Google OAuth by implementing a Google sign-in flow via LoginGoogle.js, handling backend tokens and managing user logic.
- Separate production and development resources with secure key handling through environment-aware configuration, ensuring OAuth secrets and database credentials remain isolated across development and production.

Emaily:

- Creating the architectural considerations of building a full stack app.
- Connect a front-end Create-React-App server to a NodeJS and Express backend.
- Communicate data from my Mongo database to my React application.
- Understand how to route user requests on the front end with React Router and on the backend with Express
- Build reusable user inputs with Redux Form, complete with navigation.
- Enhance authentication flows in my app with Google OAuth authentication.
- Separate production and development resources with advanced API key handling techniques.

Language

- Intermediate in English
- Native in Indonesia