



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, Laravel, MySQL, and MongoDB, I thrive in creating impactful solutions across both frontend and backend.

Education

RevoU 16 Feb - 30 Jun 2024 | Jakarta
Software Development
Fullstack Division Team Leader

Dicoding Indonesia Feb - Jun 2020 | Jakarta
Front-End Web Development
Student

Universitas Mercu Buana 2021 - 2025 | Jakarta
Bachelor of Informatics Engineering
GPA of 3.88/4.00

Work Experience

Front End Web Developer - Internship Feb – April 2018 | Jakarta
PPPA Daarul Qur'an

- Designed visually appealing website interfaces aligned with brand guidelines, producing attractive and responsive user interfaces.
- Regularly updating and maintaining websites to keep them current, relevant, and functioning properly.
- Conducting testing to identify and fix any issues related to website performance, functionality, or compatibility.

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 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).

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