

# Dwiky Hardiansah

Mobile Application Developer (Flutter)

Jakarta, 10220 | [LinkedIn](#) | [GitHub](#) | +6287885486650 | [dwikyhardiansah93@gmail.com](mailto:dwikyhardiansah93@gmail.com)

## SUMMARY

---

Mobile Application Developer with 5+ years of experience building fintech and consumer apps with Flutter and native Android. I work closely with product and design to define requirements, and I use analytics to decide what to build and fix. My work spans corporate finance, payments/PPOB, and delivery apps. I also maintain 7 published Flutter/Dart packages on pub.dev, including a GraphQL client I forked from Artemis and heavily modified that now runs in production.

## EXPERIENCE

---

### Mobile Application Developer PT Monit Teknologi Indonesia (Fulltime) | Mar 2022 - Present

- Led the first MVP phase of Monit's hybrid iOS/Android app for corporate cards, expense tracking, and reimbursement, working with the team to define requirements and scope the MVP. Shipped in 9 months with few post-release bugs as a core contributor across the company's two mobile apps.
- Proposed and led a ~6-month migration from GetX to BLoC, setting the state-management standard the team still uses.
- Forked the unmaintained Artemis GraphQL library and heavily modified it into `dartpollo`, an open-source client and code generator I published and maintain, so the data layer is kept in-house and easier for the team to work with.
- Profiled and optimized data processing and network handling with Dart DevTools, and refactored the network library to centralize error handling and retries.
- Reviewed pull requests and left review comments across the org, helping teammates and keeping code quality consistent.
- Added secure biometric authentication (FaceID/TouchID/Fingerprint) to make logging in faster and cut related support requests.

### Mobile Application Developer PT Weyland Indonesia Perkasa (Fulltime) | May 2019 - Feb 2022

- Built and maintained the AtozPay (fintech wallet) and AtozGo (delivery) native Android apps (Java/Kotlin), including a complex payment transaction flow. Delivered secure transactions, authentication, push notifications, and location services.
- Led the AtozGo/AtozPay move from native Android to a single Flutter codebase as the main contributor, removing duplicated work across platforms. Used BLoC, Firebase, and Google Maps, and completed a full Dart null-safety migration while keeping native-level performance.
- Integrated Firebase (Firestore, Cloud Messaging, Crashlytics, Analytics) into AtozGo/AtozPay and used crash and usage data to prioritize fixes and improvements.
- Built the bill-payment and top-up features, including REST integrations for PPOB flows (train-ticket booking, electricity/TV bill inquiry) and Finnet payment-gateway balance transfers, with optimized parsing, caching, and data security.
- Fixed slow loading and UI lag in AtozGo/AtozPay using Dart DevTools, applying caching and code optimization.

## PROJECT EXPERIENCE

---

- **Monit (Platform Keuangan):** Core developer of Monit's business finance platform for managing company cash, vendor payments, expense control, and corporate cards in one app. Built the QRIS payment flow, new card-issuance flow, and biometric/on-device authentication (FaceID/TouchID/Fingerprint) on a BLoC architecture, with a network layer that centralizes error handling and retries.
- **Monit Card (Monit V2):** Contributed from the start to Monit's employee card-control app, letting cardholders reveal card numbers, monitor card transactions, pay via QRIS, request new cards or balance top-ups, and freeze and configure their cards, plus a task flow to review and approve or reject requests with push-notification status updates. Added security monitoring with Talsec/ freerasp RASP — anti-tamper/app-integrity, jailbreak and root detection, and hooking/emulator checks — with remote-flag-gated enforcement, a broken-trust dialog, and Sentry threat reporting. Worked in a ticket-driven (ClickUp) Agile workflow with conventional commits and automated weekly changelogs.
- **AtozPay (Fintech Wallet & PPOB):** Main contributor to the Flutter rewrite of AtozPay (PT Weyland), a payments app for paying bills and topping up. Built PPOB flows (train-ticket booking, electricity/TV bill inquiry) and Finnet payment-gateway balance transfers, plus a full Dart null-safety migration and Firebase Cloud Messaging integration.
- **AtozGo (On-Demand Delivery):** Contributed to AtozGo (PT Weyland), the delivery app alongside AtozPay, across both its native Android app and the Flutter migration. Built merchant-side order-management flows (new-order and ongoing-order lists) and the deep-link app-switch between AtozGo and AtozPay.
- **Shafood (Food Delivery):** Built a food-ordering app end-to-end (UI/UX to deployment) in Flutter (high school internship). Implemented registration, restaurant selection, ordering, and delivery tracking with Firebase and Google Maps.
- **Hijrah Coach (E-Learning):** Built the front-end (Kotlin) for an e-learning app (PT Weyland) covering course discovery and enrollment: registration/login, course selection, and Firebase integration.
- **MHDC (Clinic Booking & Loyalty):** Solo-built a clinic booking and loyalty app end-to-end: appointment booking across medical specialties, a points/vouchers rewards system, QR/barcode support, balance top-up, an insurance module, and Firebase push notifications.
- **BNI Agen46 (POS App):** Built a point-of-sale app (Kotlin) for Verifone x990 agents (freelance) with NFC card top-up and transaction history, integrating the Verifone SDK plus REST and GraphQL data. NFC hardware integration was the main challenge.

## OPEN SOURCE

---

- **dartpollo:** Dart GraphQL client and code generator (`build_runner`), forked from [Artemis](#) and heavily modified, published to pub.dev and running in production at PT Monit.
- **remote\_in\_app\_messaging:** Host-agnostic, JSON-driven in-app messaging (modals, bottom sheets, audience targeting, sticky A/B variants, scheduling), published to pub.dev.

## SKILLS

---

### Mobile Development:

- **Flutter (Dart):** Proficient in building cross-platform mobile applications; author of 7 published Flutter/Dart packages on pub.dev.
- **Native Android (Kotlin/Java):** Experienced in developing native Android apps and platform plugins/channels (NFC, image cropping, Verifone SDK).
- **GraphQL:** Maintain a production Dart GraphQL client & code generator (`dartpollo`, forked from Artemis and heavily modified); experienced integrating GraphQL for efficient data fetching.

- **REST APIs:** Proficient in integrating REST APIs, including complex PPOB bill-payment and Finnet payment-gateway flows.
- **Firebase:** Firestore, Cloud Messaging (FCM), Crashlytics, Analytics, and Remote Config-style remote-driven UX.
- **Local Database & Persistence:** On-device storage and caching with **Hive** (NoSQL key-value; ran the production `hive_flutter` → `hive_ce_flutter` migration) and **SQLite** (schema design, indexing, and additive backward-compatible migrations).
- **Security & Cryptography:** Two-way cryptography with Pointycastle: AES/RSA/Salsa20 and signing (authored `encryptor_plus`); secure biometric authentication (FaceID/TouchID/Fingerprint).
- **Mobile App Hardening (RASP):** Runtime application self-protection with Talsec/ `freerasp` — anti-tamper/app-integrity checks, jailbreak & root (privileged-access) detection, hooking/Frida and emulator detection, with remote-flag-gated enforcement and Sentry threat reporting.
- **Maps & Location:** Google Maps integration and location services.
- **UI & Performance:** Custom widgets and reusable UI packages; performance profiling and optimization with Dart DevTools.
- **CI/CD & Release Management:** Automate mobile build/test/release pipelines with **GitHub Actions** and **Codemagic** — code signing, OIDC-trusted publishing, semantic versioning, pub.dev publishing of 7 packages, Play/App Store distribution, and conventional commits with automated changelogs.
- **Testing:** Unit, widget, and integration testing with `flutter_test`; BLoC testing via `bloc_test`/MockBloc and mocking with `mockito`; defined the automated testing strategy for the RFC-led GetX→BLoC migration (RFC-001), and engaged upstream with Dart's `build/mockito` test tooling (`dart-lang/build#4489`).

#### Architectures & Patterns:

- **State Management & Architecture:** Proficient in **BLoC**, **Clean Architecture**, and **MVVM**; led a ~6-month, **RFC-led architectural migration** from GetX to BLoC (RFC-001) including testing strategy and risk assessment.
- **API & Data Patterns:** GraphQL introspection & code generation (author of `dartpollo`); complex **PPOB, QRIS, & payment-gateway flows** (Finnet, train-ticketing, bill inquiry); **REST API multi-provider integrations**.
- **Specialized Patterns:** Event-driven / pub-sub decoupling (author of `flutter_broadcast_receiver`); **JSON-driven remote UI/UX** (audience targeting, A/B testing, frequency capping); **KYC (Know-Your-Customer)** compliance flows; **App-to-app deep linking**.
- **Migration & Modernization:** Driven large-scale **Dart null-safety migrations**; automated **code generation & build tooling** (`build_runner`, `Melos monorepos`).

#### Tools:

- **Development & Analysis:** Git (GitHub/GitLab), Dart DevTools, IntelliJ IDEA, **Melos** (monorepo management), **Build Runner**, `gql_code_builder`.
- **Local Data & Storage:** **Hive** / `hive_ce`, **SQLite**.
- **Backend & DevOps:** **Firebase Console**, **GitHub Actions**, **Codemagic**.
- **Security & Monitoring:** **Talsec** / `freerasp`, **Sentry**.
- **Workflow & Productivity:** **ClickUp** (ticket-driven Agile/Scrum), **Verifone SDK** (NFC hardware integration).

#### Methodologies:

- Agile Software Development; working with product and design teams, shaping requirements, and using data to prioritize.