

Deliverable 5.2: Desain Arsitektur Proyek Aplikasi Manajemen RT RW

1. Rasionalitas Pemilihan Arsitektur

Arsitektur yang dipilih adalah **Monolitik Modular (3-Tier)**. Pilihan ini adalah yang paling sesuai untuk proyek dengan lingkup kecil hingga menengah (*scope* FR-1.x dan FR-2.x) dan tim terbatas, sangat ideal untuk memenuhi *deadline Pembelajaran Berbasis Proyek (PjBL)*.

Kriteria	Arsitektur Monolitik Modular	Keunggulan untuk Proyek Manajemen Data Warga
Kecepatan Pengembangan	Sangat Tinggi	Memungkinkan tim fokus pada fitur inti dan <i>coding</i> yang cepat.
Organisasi Kode	Modular	Kode dibagi berdasarkan fungsi logis (Modul Otentikasi, Modul Data Warga), menjaga kerapian untuk penilaian CMMI.
Pencatatan Audit (C.7)	Ideal	Memudahkan implementasi Pencatatan Riwayat Perubahan Data (Audit Log) di satu Basis Data sentral.

2. Lapisan Implementasi (Tech Stack Resmi)

Lapisan teknologi berikut sudah **ditetapkan** dan akan menjadi dasar penulisan semua kode Anda mulai dari **Minggu 8**.

Lapisan	Teknologi/Framework	Bahasa Kunci	Tujuan di Proyek RT/RW

A. Lapisan Tampilan Depan (Frontend)	Flutter	Dart	Antarmuka pengguna yang dinamis dan <i>responsive</i> (tampilan Data Warga, Form Input).
B. Lapisan Aplikasi (Backend)	Node.js (Express.js)	JavaScript/TypeScript	Memproses logika bisnis (validasi data, pembatasan akses, CRUD), dan mengelola <i>Token</i> autentikasi.
C. Lapisan Data (Database)	PostgreSQL	SQL	Penyimpanan Data Relasional yang stabil untuk tabel warga, pengguna, dan <code>audit_log</code> .

3. Struktur Arsitektur Monolitik (3-Tier)

Aplikasi akan mengikuti struktur 3-Tier standar (seperti yang ditunjukkan dalam referensi MiniBank):

1. **Presentation Layer (Frontend):** Menangani interaksi pengguna.
2. **Application Layer (Backend):** Menerima permintaan, menjalankan logika bisnis (termasuk validasi dan otorisasi), dan berinteraksi dengan basis data.
3. **Data Layer (Database):** Menyimpan dan mengambil data, termasuk tabel `audit_log` untuk kepatuhan CMMI.

4. Implementasi Kunci & Kepatuhan CMMI

- **Audit Log (C.7):** Setiap aksi perubahan pada data warga akan memicu pencatatan otomatis di **Tabel `audit_log`** dalam PostgreSQL.
- **Hak Akses:** Logika di Backend (Node.js) akan membatasi **Admin RT** agar hanya dapat melihat, menambah, dan mengubah data Warga di wilayah RT-nya saja.