

# RANCANGAN PROJECT - SUARA: PLATFORM ASPIRASI DAN VOTING KEBIJAKAN BAGI WARGA

|                |   |
|----------------|---|
| Mata Kuliah    | Pemrograman Mobile  |
| Dosen Pengampu | Muhayat, M.IT   |
| Nama Mahasiswa | M. Kaspul Anwar   |
| NIM            | 230104040212  |
| Praktikum      | Tugas Akhir   |
| Kelas          | TI 23A  |
| Tanggal        | 03/12/2025  |
| Repo Github    | <a href="https://github.com/mkaspulanwar/SUARA-230104040212-TUGASAKHIR">https://github.com/mkaspulanwar/SUARA-230104040212-TUGASAKHIR</a> |

## 1. ANALISIS MASALAH & TUJUAN

### 1.1. Latar Belakang

Partisipasi publik dalam proses perumusan dan persetujuan kebijakan di Indonesia seringkali terhambat oleh beberapa masalah struktural:

- **Anonimitas dan Akuntabilitas Rendah:** Platform aspirasi yang ada seringkali memperbolehkan komentar atau suara anonim, yang berujung pada noise diskusi, pesan yang tidak konstruktif, dan sulitnya pelacakan identitas.
- **Keterlibatan Warga Negara yang Terbatas:** Mekanisme partisipasi yang ada belum terintegrasi secara nasional dan kurang mudah diakses oleh seluruh lapisan masyarakat secara langsung.
- **Validasi Data Aspirasi:** Sulitnya bagi pembuat kebijakan untuk memvalidasi apakah suara yang terkumpul mewakili warga negara yang sah, mempersulit pengumpulan data yang valid dan dapat dipertanggungjawabkan untuk dasar pengambilan keputusan.
- **Disorganisasi Isu:** Aspirasi yang disampaikan seringkali tidak terstruktur atau terkelompok berdasarkan sektor atau kementerian yang bertanggung jawab, menyebabkan inefisiensi dalam tindak lanjut.

### 1.2. Tujuan Projek

Tujuan utama dari pembangunan platform SUARA adalah:

- Menciptakan kanal resmi dan terintegrasi secara nasional untuk partisipasi warga negara dalam proses kebijakan publik.
- Memastikan setiap suara adalah sah dan akuntabel melalui mekanisme verifikasi identitas (KTP) wajib saat pendaftaran.
- Meningkatkan transparansi dan kualitas kebijakan yang dihasilkan dengan menyajikan data aspirasi dan hasil voting yang valid dan terstruktur berdasarkan Kementerian terkait.

- Mendorong diskusi yang konstruktif dengan menghilangkan fitur anonimitas dan mewajibkan nama asli dalam setiap interaksi.

## 2. DESKRIPSI APLIKASI

SUARA (Sistem untuk Aspirasi Rakyat) adalah aplikasi mobile native Android yang didesain sebagai e-democracy platform berskala nasional. Platform ini berfungsi sebagai jembatan langsung dan aman antara seluruh warga negara Indonesia dan pemerintah pusat. Aplikasi ini menyajikan dua fungsi inti:

- **Aspirasi Rakyat:** Tempat warga menyuarakan isu atau usulan, dilengkapi dengan fitur komentar yang transparan (menampilkan nama asli).
- **Voting Kebijakan:** Mekanisme partisipasi langsung dalam penentuan sikap (Setuju/Tidak Setuju) terhadap isu-isu kebijakan resmi yang diusulkan oleh Kementerian.

Dengan fokus pada akuntabilitas dan skala nasional, SUARA dirancang agar setiap warga merasa terlibat, memiliki data yang terstruktur, dan setiap interaksi memiliki bobot pertanggungjawaban.

## 3. TOOLS & STACK

Proyek ini akan dibangun sebagai aplikasi full-stack menggunakan teknologi berbasis Kotlin untuk efisiensi dan multiplatform capability di masa depan.

### Frontend Mobile (Android):

- **Bahasa:** Kotlin
- **UI Toolkit:** Jetpack Compose (Modern, Declarative UI)
- **Desain:** Material 3 (Estetika UI/UX terbaru Google)
- **IDE:** Android Studio

### Backend & API (Full-Stack Kotlin):

- **Framework:** Ktor (Ringan, Asynchronous, Native Kotlin Web Server)
- **ORM/Database:** Exposed (Idiomatic Kotlin ORM) dengan Database PostgreSQL/MySQL
- **Alternatif Cepat (Scalability):** Firebase (Authentication, Firestore, Cloud Functions) dapat digunakan sebagai alternatif jika kecepatan scaling awal sangat diutamakan, namun Ktor dengan Exposed akan memberikan kontrol penuh terhadap data dan logika server yang lebih baik untuk jangka panjang.

## 4. ARSITEKTUR & STRUKTUR PROJECT

Kami mengadopsi pola arsitektur yang modern, terstruktur, dan testable untuk menjamin kualitas kode.

### 4.1. Arsitektur Frontend (Android)

- **Pola:** MVVM (Model-View-ViewModel). Memastikan pemisahan tanggung jawab yang jelas. [Image of MVVM architecture diagram]
- **View ( Jetpack Compose):** Hanya bertugas menampilkan UI dan memicu event. View mengamati State dari ViewModel.
- **ViewModel:** Berisi logika presentasi, mengelola State UI, dan berkomunikasi dengan Repository.
- **Repository:** Bertanggung jawab untuk mengambil dan mengelola data (dari API Ktor atau Local Cache).
- **Manajemen Asynchronous:** Menggunakan Kotlin Coroutines dan Flow untuk operasi jaringan dan manajemen state yang reactive dan efisien.

### 4.2. Arsitektur Backend (Ktor)

Ktor akan menggunakan Arsitektur Berlapis (Layered Architecture) untuk modularitas dan kemudahan maintenance.

- **Routes/Controller:** Lapisan paling luar, bertanggung jawab menerima request dan mengirim response.
- **Services/Business Logic:** Lapisan tengah, berisi semua logika utama aplikasi (e.g., validasi voting, perhitungan hasil, otorisasi).
- **Repository/DAO:** Lapisan terdalam, berinteraksi langsung dengan database (melalui Exposed/Firebase) untuk operasi CRUD.
- **Keamanan:** Menggunakan JWT ( JSON Web Tokens) untuk otentikasi API yang aman.

## 5. LIST FITUR WAJIB

| Modul                | Fitur Wajib  | Alasan   |
|----------------------|--|--|
| Autentikasi & Profil | A1. Pendaftaran/Login dengan Verifikasi (Email/Nomor HP) | Memastikan keamanan dan identitas unik setiap pemilih. |
|                      | A2. Pembuatan Profil (Nama, Lokasi/Wilayah)              | Mengaitkan aspirasi/suara dengan wilayah spesifik.     |
| Aspirasi & Diskusi   | B1. Posting Aspirasi/Usulan (Teks & Gambar)              | Memungkinkan warga menyuarakan isu secara detail.      |

|                               |  |   |
|-------------------------------|--|---|
|                               | B2. Komentar & Balasan   | Mendorong diskusi konstruktif dan interaksi komunitas.                                    |
|                               | B3. Filter & Pencarian (Berdasarkan Wilayah, Tag, Trending)      | Memudahkan pengguna menemukan isu relevan dengan cepat.                                   |
| <b>Voting &amp; Kebijakan</b> | C1. List Kebijakan/Isu untuk Voting                              | Menyajikan isu-isu resmi yang perlu keputusan.  |
|                               | C2. Mekanisme Voting Sederhana (Setuju/Tidak Setuju)             | Implementasi inti platform: partisipasi pengambilan keputusan. Harus sekali vote per isu. |
| <b>Notifikasi</b>             | D1. Notifikasi In-App (Contoh: Kebijakan Baru, Balasan Komentar) | Menjaga engagement pengguna.  |

## 6. Ringkasan Fitur MVP (Minimum Viable Product) Aplikasi Partisipatif

MVP aplikasi digital partisipatif ini dirancang untuk mencapai validasi inti, yaitu menyediakan platform bagi pengguna untuk mengakses informasi kebijakan dan menyalurkan partisipasi mereka secara sederhana dan terstruktur. Fokus utamanya adalah pada modul Autentikasi, Kebijakan, dan Partisipasi (Voting dan Aspirasi) yang berfungsi penuh.

### Fokus Inti MVP

MVP ini terdiri dari tiga (3) modul fungsional utama yang harus tersedia:

#### 6.1. Modul Autentikasi Dasar

Modul ini memastikan bahwa setiap pengguna teridentifikasi dan dapat mengamankan sesi mereka.

- **Registrasi dan Login:** Pengguna dapat mendaftar dan masuk ke aplikasi menggunakan kredensial dasar.
- **Keamanan Sesi:** Penggunaan JSON Web Tokens (JWT) untuk mengelola sesi secara tanpa status (*stateless*).

#### 6.2. Modul Kebijakan (Informasi Publik)

Modul ini berfungsi sebagai sumber utama informasi kebijakan yang sedang dibahas atau diberlakukan.

- **Daftar Kebijakan:** Menampilkan daftar kebijakan yang aktif (judul, deskripsi singkat, status).
- **Detail Kebijakan:** Menyajikan informasi lengkap tentang kebijakan tertentu, termasuk latar belakang, tujuan, dan periode voting yang aktif.

### 6.3. Modul Partisipasi Pengguna

Modul inilah yang memungkinkan interaksi dua arah antara pengguna dan kebijakan.

- **Mekanisme Voting (Dukungan Sederhana):** Pengguna dapat memberikan suara "Setuju" atau "Tidak Setuju" pada kebijakan yang sedang dalam masa voting. Setiap pengguna hanya dapat memberikan satu suara per kebijakan.
- **Pengiriman Aspirasi:** Pengguna dapat mengirimkan aspirasi atau masukan teks terkait dengan kategori tertentu (misalnya, kementerian atau bidang topik). Aspirasi ini dicatat secara privat oleh sistem untuk ditinjau lebih lanjut.

Dengan fitur-fitur ini, aplikasi sudah dapat menguji hipotesis dasar tentang bagaimana pengguna berinteraksi dengan kebijakan dan menyalurkan partisipasi mereka dalam lingkungan digital yang terstruktur.

## 7. RENCANA SPRINT 4 MINGGU (MINIMUM VIABLE PRODUCT - MVP)

Rencana ini membagi implementasi fitur-fitur inti MVP menjadi empat siklus mingguan (Sprint) dengan fokus pada pilar-pilar utama: (1) Setup & Autentikasi, (2) Informasi Kebijakan, (3) Mekanisme Partisipasi, dan (4) Finalisasi & Pengujian.

### Ringkasan Sprint

| Sprint   | Fokus Utama           | Output Utama (Goal)   | Modul yang Disentuh          |
|----------|-----------------------|---|------------------------------|
| Minggu 1 | Basis & Autentikasi   | Aplikasi dapat diakses dan pengguna dapat masuk ( <i>login</i> ).       | Autentikasi Dasar            |
| Minggu 2 | Modul Kebijakan       | Pengguna dapat melihat daftar dan detail kebijakan yang sedang aktif.   | Kebijakan (Informasi Publik) |
| Minggu 3 | Mekanisme Partisipasi | Pengguna dapat melakukan Voting dan mengirimkan Aspirasi.               | Partisipasi Pengguna         |
| Minggu 4 | Finalisasi & Testing  | Aplikasi stabil dan siap untuk uji coba pengguna awal ( <i>pilot</i> ). | Integrasi Penuh & QA         |

## Detail Rencana Sprint

### Minggu 1: Basis Data & Autentikasi

Fokus pada pondasi teknis dan modul yang paling krusial untuk akses pengguna.

- **Poin Kunci:**

- **Setup Lingkungan:** Inisialisasi proyek *backend* dan *frontend*, konfigurasi basis data (DB), dan integrasi Firebase/server.
- **Fitur Autentikasi:** Implementasi Registrasi dan Login pengguna dasar.
- **Keamanan Sesi:** Implementasi dan pengujian alur kerja JWT (*JSON Web Token*) untuk mengelola sesi pengguna.
- **UI/UX Dasar:** Rancang dan implementasikan tata letak dasar (header, footer, navigasi).

### Minggu 2: Modul Kebijakan (Informasi Publik)

Fokus pada menampilkan informasi inti yang menjadi subjek partisipasi.

- **Poin Kunci:**

- **Skema Data Kebijakan:** Finalisasi dan implementasi skema data kebijakan di DB.
- **API Kebijakan:** Buat endpoint API untuk mengambil data kebijakan.
- **Daftar Kebijakan:** Kembangkan halaman yang menampilkan daftar kebijakan aktif (Judul, deskripsi singkat, status).
- **Halaman Detail Kebijakan:** Kembangkan halaman untuk melihat detail lengkap kebijakan (latar belakang, tujuan, masa voting).

### Minggu 3: Mekanisme Partisipasi Pengguna

Fokus pada fungsionalitas inti aplikasi: interaksi pengguna.

- **Poin Kunci:**

- **Implementasi Voting Sederhana:** Buat endpoint API untuk menerima suara "Setuju/Tidak Setuju" (satu suara per pengguna).
- **Integrasi Voting UI:** Tambahkan tombol Voting ke Halaman Detail Kebijakan dan tampilkan hasil voting saat ini (jika ada).
- **Implementasi Pengiriman Aspirasi:** Buat form dan endpoint API untuk menyimpan aspirasi teks pengguna ke DB (pencatatan privat).
- **Validasi Partisipasi:** Pastikan pengguna yang tidak terautentikasi tidak dapat berpartisipasi (terapkan pengecekan JWT).

### Minggu 4: Finalisasi, Integrasi & Pengujian

Fokus pada pemolesan, pengamanan, dan kesiapan rilis.

- **Poin Kunci:**

- **Pengujian End-to-End:** Lakukan pengujian menyeluruh pada alur kerja utama (Login -> Lihat Kebijakan -> Voting/Aspirasi).
- **Penanganan Error:** Implementasi penanganan error yang ramah pengguna (misalnya, pesan error saat gagal login, saat mencoba voting dua kali).
- **Desain Responsif:** Pastikan tampilan aplikasi berfungsi dengan baik di perangkat seluler (*mobile-responsive*).
- **Deployment Awal:** Siapkan lingkungan *staging* atau *beta* untuk uji coba pengguna awal (UAT).
- **Dokumentasi:** Pembuatan dokumentasi teknis dasar untuk panduan rilis.