# **Framework Business Model & Flowchart Feature -**[votim](http://vot.ai)

Platform SaaS (Software as a Service) dengan tema voting dan demokrasi, yang berfokus pada kemudahan dan kecepatan analisis dengan strategi **"Zero-to-Insight"**. USP utama adalah menggunakan AI (Gemini) untuk menganalisis dan menyajikan insight secara instan.

# **1. Project Planning Stage**

## **1.1 Needs Identification & Analysis**

### **Define Project Objectives**

Membangun platform SaaS (Software as a Service) dengan tema voting dan demokrasi, yang berfokus pada **kemudahan dan kecepatan analisis** (strategi "Zero-to-Insight"). USP utama adalah menggunakan AI (Gemini) untuk menganalisis dan menyajikan insight secara instan, menggantikan kebutuhan akan analis data manual.

### **Market Research**

Celah pasar teridentifikasi. Platform besar (Google Forms, SurveyMonkey) menyediakan tools (alat) yang kuat namun kompleks, yang membutuhkan keahlian (analis data) untuk mengolah data mentahnya. Platform kita menargetkan segmen yang membutuhkan insight (jawaban) instan tanpa keahlian teknis.

### **Define Target Users**

**Target Utama:**

* Manajer Departemen (misal: HR yang butuh feedback karyawan)
* Pemilik UMKM (yang ingin riset pasar cepat)
* Ketua Komunitas atau Organisasi
* **Target utama BUKAN analis data profesional**

**Persona 1: Manajer Departemen & Pemilik UMKM**

* **Manajer Departemen (HR/Marketing):** Membutuhkan feedback cepat dan matang untuk rapat
* **Pemilik UMKM:** Ingin riset pasar cepat tanpa biaya agensi mahal

**Persona 2: Ketua Komunitas/Organisasi**

* Perlu mengambil keputusan berdasarkan suara anggota yang tervalidasi

### **Functional Requirements Analysis**

**Fitur 1 (Voting / Polling):**

* Admin membuat voting/polling
* Membagikan link
* Menutup polling
* Melihat hasil (persentase, jumlah)

**Fitur 2 (Public Opinion):**

* Admin membuat form opini (teks bebas)
* Membagikan link
* Dasbor admin menampilkan analisis AI (sentimen, word cloud, ringkasan keyword) secara otomatis

**Fitur 3 (Room Forum / Discussion):**

* Admin membuat ruang diskusi terjadwal (waktu buka/tutup)
* Moderasi penuh (regex ban)
* AI Co-Moderator
* Opsi simpan transkrip

**Authentication:**

* Login standar (email/password) DAN Google OAuth (Otentikasi Sosial)
* Sistem **Wajib** menggunakan "Display Name" unik untuk semua interaksi publik (terutama Forum) untuk melindungi privasi user yang terpisah dari email

**Dashboard:**

* Menggunakan layout Sidebar Kiri dengan navigasi utama:
  + Insight Studio
  + Voting / Polling
  + Public Opinion
  + Room Forum / Discussion
  + Settings

**Profile Management:**

* Berada di bawah tab Settings
* Mengizinkan user mengubah "Display Name", password, dan mengelola status langganan & penagihan

**Customization:**

* Pro Plan akan mendapatkan fitur "Hapus Branding (White-label)"

**Analytics:**

* Ini adalah core USP. Dasbor harus menyediakan "Dasbor Analisis Sekali Klik" yang berisi:
  + Analisis Sentimen (Positif, Negatif, Netral)
  + Word Cloud interaktif (sudah bersih dari stopword)
  + Ringkasan poin utama (dihasilkan oleh AI)
  + Opsi untuk mengunduh laporan dalam format PDF/PNG

### **Non-Functional Requirements Analysis**

**Performance (Kritis):**

Pemrosesan AI tidak boleh menghalangi user experience. Semua analisis AI harus dijalankan secara asynchronous (menggunakan Queues) di latar belakang. Dasbor analitik harus dimuat cepat (data di-cache atau di-pre-compute).

**Implementation:** Semua analisis AI harus asynchronous menggunakan Laravel Queues. Untuk Zero Capital, driver queue akan menggunakan database. Visualisasi data (Chart.js) akan di-render di sisi klien untuk mengurangi beban server.

**Security (Kritis):**

Selain moderasi Fitur 3, kebutuhan keamanan diperluas untuk mencakup:

* **Pencegahan Spam Bot:** Google reCAPTCHA v3 wajib di semua form publik
* **Pencegahan Brute Force:** Rate Limiting bawaan Laravel di rute login dan pengiriman form
* **Pencegahan IDOR:** Laravel Policies wajib untuk memvalidasi kepemilikan data (auth()->id())
* **Privasi Forum:** Wajib menggunakan "Display Name" (dari Google OAuth atau manual), **DILARANG** menampilkan email mentah

**Scalability:**

Ini adalah tantangan utama. Skalabilitas server dan biaya kini dimitigasi secara langsung oleh **3 Pilar Bisnis & Teknis Inti:**

* **Pilar #2 (Auto-Close Link):** Membatasi link free user hingga 3 jam untuk mencegah beban traffic jangka panjang
* **Pilar #3 (Auto-Purge Data):** Menghapus data free user setelah 7 hari untuk menjaga database tetap ramping dan murah
* **Biaya API AI:** Dikontrol via Priority Queues (high untuk Pro, low untuk Free) dan model pricing berbasis usage

**Usability:**

USP utama. Platform harus "simpel, ringkas, padat". Menjual kemudahan dan kecepatan, bukan kelengkapan fitur.

**Compatibility:**

Aplikasi web modern yang kompatibel dengan browser utama (Chrome, Firefox, Safari, Edge) di desktop dan seluler (Desain Responsif).

### **Dependencies**

**Internal:**

* Sistem Antrian (Queues) internal Laravel (driver database)
* Laravel Scheduler (Cron Job) untuk menjalankan Auto-Purge
* Komponen Livewire untuk interaktivitas frontend
* Library Chart.js untuk visualisasi data
* Multi-Worker Architecture
* Connection Pooling (PgBouncer)

*"Untuk skalabilitas tinggi, sistem menggunakan arsitektur multi-worker dengan strategi fault tolerance (lihat Survival Mode di Addendum)."*

**External:**

* API AI (Google Gemini)
* Payment Gateway (untuk Pro Plan, misal: Stripe/Paddle)
* Servis WebSocket (misal: Pusher, atau Laravel Reverb) untuk Fitur 3

### **Assumptions**

* Asumsi user (target pasar) lebih menghargai kemudahan & insight instan daripada kustomisasi mendalam & data mentah
* Asumsi model pricing (berbasis usage) dapat menutupi biaya variabel API AI
* Asumsi free tier dari hosting dan database cukup untuk menopang free plan kita (model zero capital)

### **Risks & Mitigations**

| **Risk** | **Mitigation** |
| --- | --- |
| Biaya API AI boros dan tidak terprediksi | Model pricing berbasis usage DAN implementasi Priority Queues (high/low) untuk mengontrol alur pemrosesan |
| Performa lambat saat analisis AI | Wajib menggunakan Laravel Queues asinkron dengan driver database untuk MVP |
| Gagal bersaing dengan platform besar yang fiturnya lebih lengkap | Tidak bersaing head-to-head. Fokus pada niche "Zero-to-Insight" dan target pasar manajer (bukan analis) |
| Moderasi "Open Conversation" (Fitur 3) gagal dan menjadi toksik | Mengandalkan kombinasi AI Co-Moderator (deteksi otomatis) + Admin sebagai "hakim" dengan tools moderasi yang kuat (regex ban, manual kick/ban) |

## **1.2 Scope & Timeline Determination**

### **Define the Project Scope**

**In Scope:**

* Public Website (Home, Features, Pricing, Contact)
* Sistem Autentikasi User
* Dashboard Admin dengan 3 Fitur Utama
* Integrasi AI (Gemini) untuk Fitur 2 dan 3
* Sistem Pembayaran (Pro Plan)

**Out of Scope (for MVP):**

**Saran:** Untuk Proof of Concept (PoC) awal, kita bisa fokus hanya pada **Fitur 2 (Analisis Opini AI) + Dasbor**. Ini adalah USP terkuat kita. Jika ini terbukti berhasil, Fitur 1 (Polling) dan Fitur 3 (Open Conversation) dapat menyusul.

### **Create a Milestone Schedule**

* **Week 1:** Setup Proyek (Laravel, DB, Git), Desain UI/UX (Figma), implementasi Autentikasi (OAuth) & Halaman Publik
* **Week 2:** Bangun CRUD 11 Subbab Fitur & Alur Link Publik (termasuk reCAPTCHA & Rate Limiting)
* **Week 3:** Bangun Alur AI (Queue database, Job ProcessOpinionJob) & Insight Studio (Chart.js, Ekspor PDF/CSV)
* **Week 4:** Implementasi 3 Pilar Bisnis (Auto-Close, Auto-Purge), integrasi Payment Gateway, testing akhir, dan deployment ke hosting (Railway/Koyeb)

### **Prepare Time Estimates**

The total project time is estimated to be approximately **one month**, assuming a full-time commitment.

## **1.3 Tools & Resources Preparation (Free)**

* **Project management:** Trello atau Notion (Free tier)
* **UI/UX design:** Figma (Free tier)
* **Version control:** Git / GitHub (Free public/private repo)
* **Text editor/IDE:** VS Code (Gratis)

## **1.4 Feature & User Limitations**

### **Basic Concepts Used**

**Freemium Model:**

* **Free Plan:** Berfungsi sebagai data collection tool. User bisa membuat voting/polling, public opinion dan forum/discussion (dengan batasan), tapi hanya melihat data mentah/hasil persentase dasar (csv)
* **Pro Plan:** Berfungsi sebagai insight tool. Membuka semua fitur AI (analisis sentimen, word cloud, ringkasan) pada Report Composer, fitur Real-time moderation (forum/discussion), dan batasan (limit) yang lebih tinggi

**Batasan Fitur Inti (Pendorong Upgrade):**

### **3 Pilar Bisnis & Teknis**

| **Pilar** | **Free Plan** | **Pro Plan** |
| --- | --- | --- |
| **Pilar #1: Waktu Link (Auto-Close)** | Link voting/opini/forum otomatis kadaluarsa setelah **3 jam. Fungsional:** Semua Admin dapat mengaktifkan tombol "Manual Close" disaat Link masih berjalan. | Admin dapat mengatur waktu kadaluarsa kustom (misal: 30+ hari). **Fungsional:** Semua Admin dapat mengaktifkan tombol "Manual Close" disaat Link masih berjalan. |
| **Pilar #2: Penyimpanan (Auto-Purge)** | Semua data hasil (voting,opini,forum) otomatis dihapus permanen setelah **7 hari** | Retensi data penuh (minimal 1 tahun) |
| **Pilar #3: Prioritas (Kecepatan AI)** | Analisis AI masuk ke **Antrian Prioritas Rendah** (hasil bisa tertunda saat jam sibuk) | Analisis AI masuk ke **Antrian Prioritas Tinggi** (hasil diproses instan) |

### *"Untuk kontrol biaya teknis yang lebih granular, sistem 3 pilar dilengkapi dengan mekanisme AI Credit berbasis token dan Aggregator Engine (lihat detail teknis di Bab 7.6)."*

| **Fitur** | **Free Plan** | **Pro Plan** |
| --- | --- | --- |
| **Limit Request** | Maks 3 Batch/Hari | 50 Batch/Hari |
| **Kecepatan** | Batch Lambat (Tunggu 20 item)/5 menit | Batch Cepat (Tiap 5 item/Instan) |
| **Report Composer** | 1 Laporan/Hari | Unlimited |
| **Co-Moderator** | Regex Filter (Tanpa AI) | AI Aktif (Strict Mode) |

### **Batasan Tambahan**

| **Fitur** | **Free Plan** | **Pro Plan** |
| --- | --- | --- |
| **Download Laporan** | Hanya CSV (data mentah) | Laporan PDF/PNG yang sudah jadi |
| **Forum Diskusi (V2)** | Dibatasi (misal: 1 room / 10 user) | Skala penuh |
| **AI Co-Moderator (V2)** | Tidak tersedia (Regex Filter) | Tersedia (biaya API real-time tinggi) |

### **Feature-Problem Fit**

* **Free Plan:** Mengatasi masalah pengumpulan data sementara/instan
* **Pro Plan:** Mengatasi masalah analisis data permanen/jangka panjang dan berskala besar

### **Analytics**

* **Free:** Hanya menampilkan data mentah atau jumlah voting
* **Pro:** Membuka "Dasbor Analisis Sekali Klik" (Sentimen, Word Cloud, Ringkasan AI)

### **Limitations on Certain Features**

**Analytics:**

* Free user hanya bisa mengunduh **CSV (Data Mentah)**
* Pro user bisa mengunduh **Laporan PDF/PNG** yang sudah matang (visualisasi + ringkasan AI)

**Authentication:**

* Semua user (termasuk Free) wajib mendaftar dan memiliki akun terverifikasi untuk membuat koleksi data. Ini untuk mencegah abuse

### **What Users Can and Cannot Do**

**Can (Free):**

* Menggunakan semua fitur inti (Polling, Opini AI, Forum) untuk kebutuhan jangka pendek/skala kecil

**Cannot (Free):**

* Menyimpan data lebih dari 7 hari
* Menjalankan polling lebih dari 3 jam
* Mendapat jaminan kecepatan analisis AI saat jam sibuk

### **Restrictions**

**Free Plan:**

* 3 Folder Collection aktif, Setiap Collection mampu menampung maksimal 2 activity subbab (total maksimal 6 activity)
* Maksimal 100 responden per aktivitas (subbab)
* 1 room Forum aktif (maks 10 user)

# **2. Design Phase**

## **2.1 System & Architecture Design**

### **Determine the Application Architecture**

Laravel (Full Stack / Monolith Modular), dengan arsitektur **Queue-Heavy** (sangat bergantung pada antrian) menggunakan driver database untuk Zero Capital memproses semua job AI secara asynchronous.

### **Database Design**

Perlu dipertimbangkan pemisahan antara data transaksional (opini mentah yang masuk) dan data analitikal (hasil chart/cache) untuk memastikan kecepatan load dasbor admin.

### **API Design**

Akan ada API internal (untuk frontend jika menggunakan Vue/React) dan komunikasi API eksternal yang intensif ke Google Gemini API.

### **System Diagram**

Diagram akan menunjukkan:

1. **Monolith Modular Laravel** yang menerima request HTTP
2. Request berat (seperti Analisis AI) dialihkan ke **Queue** (driver database) dan diproses oleh Worker terpisah
3. Fitur real-time (Forum) akan berkomunikasi dua arah dengan layanan eksternal **Pusher**
4. Semua data disimpan di **Database PostgreSQL**
5. **Cron Job (Scheduler)** akan berjalan secara periodik untuk memicu Auto-Purge. **Implementasi:** Laravel Scheduler (Cron Job) akan menjalankan perintah purge:free-user-data setiap hari pada pukul 02:00.
6. Multi-Worker Architecture
7. Connection Pooling (PgBouncer)

## **2.2 Tech Stack Selection**

| **Component** | **Technology** |
| --- | --- |
| **Frontend** | To Be Determined (Laravel Blade+Livewire untuk kecepatan development, atau Inertia+Vue/React untuk modernitas) |
| **Visualisasi** | Chart.js |
| **Backend** | Laravel 11+ (PHP 8.2+) |
| **Database** | Postgres (via Neon/Railway) atau MySQL (via PlanetScale) untuk mendukung strategi zero capital |
| **Hosting/Deployment** | [Railway.app](http://railway.app) atau Koyeb (untuk mendukung zero capital) |
| **Real-time** | Pusher (dikelola via Laravel Reverb) |

## **2.3 Business Model Comparison**

| **Column** | **Free Plan (Feature-Problem Fit)** | **Pro Plan (Solution-Segment Fit)** |
| --- | --- | --- |
| **User Value Proposition** | "Alat pengumpulan data" "Jawaban instan / Analis data junior virtual" | "Alat pengumpulan data" "Jawaban instan / Analis data junior virtual" |
| **Core Features** | Voting dasar, Form opini dasar (data mentah) Analisis AI (Fitur 2), Open Conversation (Fitur 3), Limit tinggi | Voting dasar, Form opini dasar (data mentah) Analisis AI (Fitur 2), Open Conversation (Fitur 3), Limit tinggi |
| **Your Cost** | $0 (Mengandalkan Free Tiers) Variabel (Biaya API AI per panggilan) | $0 (Mengandalkan Free Tiers) Variabel (Biaya API AI per panggilan) |
| **Monetization** | Umpan (hook) untuk upgrade Langganan bulanan/tahunan (berbasis usage) | Umpan (hook) untuk upgrade Langganan bulanan/tahunan (berbasis usage) |
| **PMF Level** | Mengatasi masalah pengumpulan data. Mengatasi masalah analisis data (USP utama). | Mengatasi masalah pengumpulan data. Mengatasi masalah analisis data (USP utama). |

# **3. User Journey**

## **User Journey: The Free Plan**

### **1. Discovery**

* User menemukan platform melalui search engine, sosial media, atau referral
* **Action:** Mengunjungi website publik

### **2. Onboarding**

* User melihat halaman Home dengan value proposition yang jelas
* User memutuskan untuk mendaftar
* **Action:** Klik "Daftar Gratis" atau "Sign in with Google"

### **3. Free Dashboard Access**

* User berhasil login dan melihat dashboard dengan Template Gallery
* User melihat batasan Free Plan (3 jam link, 7 hari data retention)
* **Action:** Mulai membuat koleksi data pertama

### **4. Create & Customize**

* User memilih template dari gallery (misal: "Open Opinion")
* User mengkustomisasi pertanyaan
* User mendapatkan link publik untuk dibagikan
* **Action:** Bagikan link ke audiens

### **5. Share & View**

* Audiens mengisi form melalui link publik
* User melihat data masuk secara real-time
* User melihat visualisasi dasar (untuk Free: hanya data mentah)
* **Action:** Menunggu hasil analisis atau memutuskan untuk upgrade

## **User Journey: The Pro Plan Upgrade (Manual Process)**

### **6. Upgrade Attempt**

* User melihat batasan Free Plan dan tertarik fitur Pro
* User melihat halaman Pricing
* **Action:** Klik "Upgrade to Pro"

### **7. Checkout**

* User diarahkan ke halaman checkout (Stripe/Paddle)
* User melakukan pembayaran
* **Action:** Konfirmasi pembayaran

### **8. Manual Activation**

* Sistem menerima webhook dari payment gateway
* Status akun user diupdate menjadi "Pro"
* **Action:** Sistem mengirim email konfirmasi upgrade

### **9. Pro Dashboard Access**

* User login kembali dan melihat fitur Pro sudah aktif
* User melihat badge "Pro" di dashboard
* **Action:** Mulai menggunakan fitur Pro

### **10. Pro Features in Use**

* User membuat koleksi data dengan custom expiration time
* User mendapatkan analisis AI dengan prioritas tinggi
* User mengunduh laporan PDF yang sudah matang
* **Action:** Memanfaatkan semua fitur Pro untuk produktivitas maksimal

# **4. Design Plan: Information Architecture**

The new design will be split into two primary experiences: the **Public Website** and the **User Dashboard**.

Desain: Deep Glassmorphism (Kartu transparan, *border* tipis, *background-blur*).

## **1. The Public-Facing Website**

### **Home/Overview**

* Headline "Ubah Opini Menjadi Aksi" dengan visual dashboard yang glowing
* Value proposition yang jelas
* CTA utama: "Daftar Gratis" atau "Masuk ke Dashboard"
* Halaman: Home (Menjelaskan, Mencontohkan, Menjawab)

### **Features/How to Work**

* Visualisasi alur 3 langkah:
  1. Kumpulkan Data
  2. Proses AI/Analisis
  3. Unduh Laporan PDF

### **Plans/Pricing**

* Perbandingan transparan Free vs. Pro
* Menyoroti 3 Pilar Bisnis

### **Contact**

* Form standar untuk pertanyaan atau feedback

## **2. The User Dashboard**

### **Dashboard Layout**

**Sidebar Kiri dengan navigasi utama:**

* Insight Studio
* Voting / Polling
* Public Opinion
* Room Forum / Discussion
* Settings

### **Insight Studio (Dashboard Utama)**

Ini adalah dashboard utama kita. Ini bukan list data mentah, ini adalah **"Pabrik Laporan/Report Builder" - Pusat komando untuk membuat, menggabungkan, dan mengekspor laporan analitis (lihat detail di Bab 7.3).**

**Alur Kerja 3 Langkah:**

1. **Pilih Sumber:** "Ambil data dari 'Voting Karyawan Q3'"
2. **Pilih Lensa:** "Terapkan template 'Executive Summary'" (AI akan meringkas + Chart.js akan memvisualisasikan)
3. **Ekspor:** Unduh laporan instan dalam format PDF, CSV, atau PNG
4. Implementasi PDF: Library laravel-dompdf.
5. Implementasi CSV: Library laravel-excel.

### **Voting & Opini (List Management)**

* Manajemen CRUD untuk membuat koleksi data.
* Konfigurasi Opini: Batasan karakter (default 500) dan validasi regex opsional.

### **Forum Diskusi**

* Layout: 2 kolom (Kiri: Konfigurasi, Kanan: Preview Chat).
* Konfigurasi: Jadwal Buka/Tutup, Kapasitas, Opsi Simpan Transkrip, Filter Regex (dengan *preset* SARA).
* Fungsionalitas Chat: Teks dan Emoji standar diizinkan. **Upload media (gambar/video) dilarang**.

### **Autentikasi & Privasi**

* Metode: Login standar (email/password) DAN **Google OAuth (Otentikasi Sosial)**.
* Privasi Forum: Sistem **WAJIB** menggunakan **"Display Name"** (diambil dari Google OAuth atau di-set manual). **DILARANG KERAS** menampilkan email mentah di *chat room*.

# **5. Flowchart Process: User Views**

## **1. Public Website View**

### **Arrival**

* User mengunjungi landing page
* Melihat hero section dengan value proposition

### **Exploration**

* User scroll untuk melihat features
* User membaca pricing comparison
* User melihat testimonial atau case study (jika ada)

### **Action**

* User klik "Daftar Gratis" atau "Login"
* User diarahkan ke halaman authentication

## **2. User Registration & Login**

### **Form Submission**

* User memilih metode: Email/Password atau Google OAuth
* User mengisi form atau authorize Google

### **Authentication**

* Sistem memvalidasi kredensial
* Sistem membuat session user

### **Redirection**

* User diarahkan ke Dashboard
* User melihat welcome message atau onboarding tour

## **3. User Dashboard View (The Core Experience)**

### **Default View**

* User melihat Template Gallery di halaman utama
* User melihat statistik koleksi data yang sudah dibuat
* User melihat notifikasi atau tips penggunaan

### **Navigation**

* User dapat berpindah antar menu melalui sidebar
* User dapat mengakses Settings untuk profile management
* User dapat melihat status plan (Free/Pro)

## **4. Alur Kerja Kunci (Workflows)**

### **4.1 Alur Backend Queue (Async AI)**

1. *User* mengirim opini.
2. PublicController menerima, validasi, dan menyimpan data mentah ke opinion\_results.
3. PublicController **HANYA** melempar *job*: ProcessOpinionJob::dispatch($result)->onQueue($priority).
4. Variabel $priority ditentukan oleh paket *user*: high (Pro) atau low (Free).
5. *Worker* (php artisan queue:work --queue=high,low) di server akan selalu memproses antrian high terlebih dahulu.
6. ProcessOpinionJob memanggil API Gemini, menerima JSON (sentimen/keyword), lalu memperbarui baris di opinion\_results.

### **4.2 Alur WebSocket (AI Co-Moderator)**

1. User A mengirim pesan ke *room chat*.
2. Laravel menerima pesan, menyimpan ke DB, DAN langsung *broadcast* ke Pusher (User B, C, D melihat pesan).
3. *Bersamaan*, Laravel melempar ModerateMessageJob::dispatch($message).
4. *Worker* memanggil AI Gemini ("Apakah ini SARA? Ya/Tidak").
5. *Job* menerima balasan "Ya".
6. *Job* menyiarkan *event* WebSocket **kedua** (message-flagged) yang HANYA didengar oleh *dashboard* Admin.
7. *Dashboard* Admin memberi sorotan merah pada pesan tersebut. (AI tidak memblokir, AI memberi *flag*).

## **5. Kebutuhan Non-Fungsional (Keamanan & Performa)**

### **5.1 Keamanan**

1. **Pencegahan Spam Bot:** **Google reCAPTCHA v3** Wajib diimplementasikan di *semua* form publik (voting, opini, forum).
2. **Pencegahan Brute Force:** **Rate Limiting** bawaan Laravel Wajib diaktifkan di rute login dan pengiriman form.
3. **Pencegahan IDOR (Insecure Direct Object Reference):** **Laravel Policies** Wajib digunakan. Setiap *request* di *dashboard* (misal: VotingController@update) harus memvalidasi bahwa data yang diakses milik auth()->id().

### **5.2 Performa & Skalabilitas**

* Performa *frontend* dijaga dengan Livewire (menghindari *full page reload*).
* Performa *backend* dijaga dengan memindahkan semua proses berat (AI, moderasi) ke *Queue Worker* asinkron.
* Skalabilitas biaya dijaga oleh **Pilar #2 (Auto-Close)** dan **Pilar #3 (Auto-Purge)**.

# **6. Kerangka Ide Pitch Deck: "**[votim](http://vot.id)**" (Versi 2.0)**

## **Slide 1: Judul & Konsep Inti**

* **Judul:** [votim](http://vot.id) (atau Nama Proyek Anda)
* **Tagline:** Ubah Ribuan Opini Menjadi Satu Laporan Strategis
* **Konsep:** Platform SaaS "Zero-to-Insight" yang mengubah noise (umpan balik mentah) menjadi signal (laporan analisis matang) menggunakan AI

## **Slide 2: Masalah (The Problem)**

* Organisasi (terutama non-teknis) butuh akan feedback (umpan balik)
* **Masalah 1:** Alat yang ada (Google Forms, dll.) hanya mengumpulkan data. Admin masih harus menghabiskan waktu berjam-jam untuk membaca, menganalisis, dan membuat laporan secara manual di Excel/Spreadsheet
* **Masalah 2:** Diskusi terbuka (Discord, WA, Live Chat) terlalu "ramai" (noise), penuh spam, dan sulit dikontrol untuk diskusi serius (FGD)
* **Inti Masalah:** Ada gap besar antara "Mengumpulkan Data" dan "Mendapatkan Insight"

## **Slide 3: Solusi Kita (The Solution)**

* Kami menyediakan platform integrasi **3-in-1** yang tidak hanya mengumpulkan, tapi juga **Menganalisis** (via AI) dan **Memfasilitasi** (via Forum Steril)
* Platform kami bertindak sebagai **"Analis Data Junior Virtual"** dan **"Moderator Profesional"** Anda
* Kami menjual kemudahan, kecepatan, dan hasil instan, bukan kerumitan tools

## **Slide 4: Target Pasar (The User)**

### **Siapa Target Kita:**

* Manajer Departemen (HR, Marketing) yang butuh feedback cepat
* Pemilik UMKM yang ingin riset pasar tanpa biaya mahal
* Ketua Komunitas & Organisasi

### **Target Kita BUKAN:**

* Analis data profesional (mereka lebih suka data mentah yang rumit)

## **Slide 5: Alur Website Publik (Etalase Kita)**

**Desain "Deep Glassmorphism"**

Kerangka Website publik kita sederhana dan langsung pada intinya, berfungsi sebagai corong (funnel) utama:

* **Tujuan:** Menjelaskan, Mencontohkan, Menjawab
* **Home/Overview:** Headline "Ubah Opini Menjadi Aksi" dengan visual dashboard yang glowing
* **Features/How to work:** Visualisasi alur 3 langkah: 1) Kumpulkan Data → 2) Proses AI/Analisis → 3) Unduh Laporan PDF
* **Plans/Pricing:** Perbandingan transparan Free vs. Pro, menyoroti 3 Pilar Bisnis
* **Contact:** Form standar
* **Call to Action (CTA) Utama:** "Daftar Gratis" atau "Masuk ke Dashboard"

## **Slide 6: Model Bisnis (Freemium "Zero Capital")**

| **Fitur** | **Free Plan (The "Taste")** | **Pro Plan (The "Tool")** | **Keterangan "Markup"** |
| --- | --- | --- | --- |
| **Pilar: Waktu Link** | Auto-Close 3 Jam | Kustom (hingga 30+ hari) | Anda membayar untuk **Waktu** |
| **Pilar: Retensi Data** | Auto-Purge 7 Hari | Penyimpanan Penuh (min. 1 tahun) | Anda membayar untuk **Memori** |
| **Pilar: Kecepatan AI** | Antrian Prioritas Rendah (Bisa lambat) | Antrian Prioritas Tinggi (Instan) | Anda membayar untuk **Kecepatan** |
| **Koleksi Data** | 3 Folder Collection aktif, Setiap Collection mampu menampung maksimal 2 activity subbab (total maksimal 6 activity) | Tidak terbatas | Standar SaaS |
| **Responden** | Maksimal 100 per aktivitas (subbab) | Tidak terbatas (atau sangat tinggi) | Standar SaaS |
| **Download Laporan** | Hanya CSV (Data Mentah) | PDF, PNG, CSV (Laporan Jadi) | Markup nilai tambah (Hasil Olahan vs. Mentah) |
| **Forum Diskusi** | 1 room aktif (Maks 10 user) | Room tidak terbatas (Maks 100+ user) | Anda membayar untuk **Skala** |
| **AI Co-Moderator** | Tidak Tersedia, Regex Filter | Tersedia | Ini fitur premium murni karena biaya API real-time-nya mahal |
| **Kustomisasi** | Branding [votim](http://vot.id) | Hapus Branding (White-label) | Standar SaaS |
| **Dukungan** | Komunitas / Email | Prioritas Email / Chat | Standar SaaS |

## **Slide 7:** Insight Studio **- Dashboard Admin**

Ini adalah dashboard utama kita (Insight Studio). Ini bukan list data mentah, ini adalah '**Pabrik Laporan/Report Builder**' yang memungkinkan **Report Composer** (menggabungkan beberapa data).

**Alur Kerja 3 Langkah:**

1. **Pilih Sumber:** "Ambil data dari 'Voting Karyawan Q3'"
2. **Pilih Lensa:** "Terapkan template 'Executive Summary'" (AI akan meringkas + Chart.js akan memvisualisasikan)
3. **Ekspor:** Unduh laporan instan dalam format **PDF, CSV, atau PNG**

## **Slide 8: Detail Fitur Pengumpul Data #1: Simple Polling**

### **Tujuan**

Mengumpulkan data kuantitatif untuk Insight Studio.

### **Fungsi**

Admin membuat polling (pilihan ganda, skala 1-5), mendapat link.

### **Aturan Bisnis**

Tunduk pada Pilar #2 (Auto-Close 3 Jam) dan Pilar #3 (Auto-Purge 7 Hari) untuk free user.

### **Alur Admin:**

1. Buat pertanyaan
2. Buat opsi pilihan ganda (Voting)
3. Bagikan link unik

### **Alur User (Audiens)**

Buka link → Pilih Opsi → Selesai

### **Hasil Dashboard Admin**

Tampilan real-time berapa persen suara dan jumlah total voting. Admin bisa menutup polling kapan saja.

### **Peran AI**

Tidak ada. Ini adalah fitur dasar yang "wajib ada".

## **Slide 9: Detail Fitur Pengumpul Data #2: Opinion Analysis (USP Utama)**

### **Tujuan**

Mengumpulkan data kualitatif (teks) menjadi kuantitas yang ringkas untuk Insight Studio.

### **Fungsi**

Admin membuat form opini (teks bebas, limit 500 karakter, regex kustom).

### **Peran AI (Asinkron)**

Saat data masuk, job antrian low/high (Pilar #3) memanggil Gemini untuk menghasilkan **Analisis Sentimen** dan **Ekstraksi Keyword** secara otomatis.

### **Hasil**

Data ini langsung siap dipakai di Insight Studio.

### **Alur Admin:**

1. Buat pertanyaan opini (teks bebas)
2. Atur opsi (misal: limit karakter)
3. Bagikan link unik

### **Alur User (Audiens)**

Buka link → Menulis opini mereka di textarea → Kirim

### **Hasil Dashboard Admin (The Magic)**

Dasbor "Sekali Klik" yang otomatis dihasilkan AI:

1. **Analisis Sentimen:** Pie chart (misal: 70% Positif, 20% Negatif, 10% Netral)
2. **Word Cloud:** Visualisasi kata kunci yang paling sering muncul (sudah bersih dari kata "yang", "di", "dari", dll)
3. **Ringkasan AI:** 5 poin utama atau keluhan teratas yang diringkas oleh Gemini dari ratusan opini
4. Tombol **"Download Laporan PDF"**

### **Peran AI**

Bekerja di latar belakang (via Queues) sebagai **Analis Data Instan**.

## **Slide 10: Detail Fitur Pengumpul Data #3: Open Conversation (Ruang Rapat Steril)**

### **Tujuan**

Mengumpulkan insight kualitatif real-time (FGD Virtual). Fasilitasi diskusi fokus, serius, dan aman (Anti-Noise).

### **Fungsi**

Admin menjadwalkan room (kapasitas, waktu), mendapat link.

### **Privasi & Fokus:**

* User login via Google OAuth atau Display Name (BUKAN email)
* Hanya Teks & Emoji. Media (Gambar/Video) **Dilarang Keras**

### **Peran AI (Co-Moderator)**

AI memberi flag (tanda) real-time ke Admin jika ada pesan SARA/Toksik, tanpa memblokir pesan (Admin tetap hakim).

### **Alur Admin (Setup):**

* **Jadwalkan Waktu:** Tentukan jam buka dan jam tutup otomatis
* **Atur Kapasitas:** Misal, "Maksimal 100 orang"
* **Aktifkan Opsi:** "Simpan Transkrip Percakapan" (Ya/Tidak)
* **Atur Filter:** Siapkan kata/karakter yang di-ban (via Regex)

### **Alur User (Audiens):**

* Masuk saat jadwal terbuka menggunakan email mereka
* Tidak ada avatar, tidak ada gambar, tidak ada media/GIF. Hanya teks murni untuk fokus pada argumen

### **Pengalaman Admin (Saat Live):**

* Admin bertindak sebagai "Hakim" atau moderator penuh
* Bisa menutup room secara manual kapan saja

### **Peran AI (Co-Moderator):**

* AI memonitor chat secara real-time
* Secara otomatis memberi flag atau peringatan kepada Admin jika ada user yang melanggar (SARA, hate speech, atau kata yang di-ban)

## **Slide 11: Arsitektur, Keamanan & Tantangan Inti**

### **Tech Stack**

* Laravel (Backend)
* Chart.js, Livewire (Frontend TBD)
* Database (Neon/PlanetScale)
* Hosting (Railway)

### **Infrastruktur**

* PostgreSQL (Neon) + Railway/Koyeb (Hosting) + Pusher (Websocket)
* Memaksimalkan free tier tanpa mengorbankan performa inti

### **Solusi Biaya**

* Kita mengontrol biaya AI dengan **Priority Queues**
* Kita mengontrol biaya storage dengan **Data Purging**

### **Keamanan**

* reCAPTCHA dan Rate Limiting di semua endpoint publik untuk memblokir bot spam

### **Tantangan & Solusi**

| **Tantangan** | **Solusi** |
| --- | --- |
| **Tantangan 1: Biaya AI** | Model pricing Pro Plan harus berbasis usage (kredit AI), bukan unlimited |
| **Tantangan 2: Performa (Beban AI)** | Wajib menggunakan Asynchronous Queues (Laravel) untuk semua pemrosesan AI agar tidak memblokir user |
| **Tantangan 3: Moderasi (Fitur 3)** | Kombinasi AI Co-Moderator (deteksi) + Admin (eksekusi) |

## **Slide 12: Kesimpulan USP (Why We Win)**

* Kita tidak bersaing dengan kelengkapan fitur Google
* Kita bersaing di **kecepatan mendapatkan insight ke laporan**
* Kita menjual **"Paket Lengkap Analisis Umpan Balik"** yang sudah matang, bukan tools mentah

# 7. Dokumen Pembaruan Teknis & Advance Requirements

**Status:** Finalisasi Konsep Fitur & Logika Visualisasi

**Versi:** 2.0 (Post-Brainstorming AI Strategy)

*"Setelah analisis mendalam, struktur fitur dikembangkan menjadi sistem hierarkis dengan Template Gallery untuk meningkatkan organisasi dan fleksibilitas. Fitur awal (Polling, Opini, Forum) kini menjadi 'BAB' utama, dengan 'Subbab' sebagai variasi spesifik."*

## **7.1. Struktur Hierarki Fitur S**ubbab **(Template Gallery)**

Finalisasi menu yang akan dipilih Admin saat membuat koleksi data baru. Menggunakan konsep **"Template Gallery"** (bukan form kosong).

### **7.1.1. Bab Voting & Polling (Kuantitatif)**

*Fokus: Angka pasti dan keputusan cepat.*

| **No** | **Subbab Fitur** | **Fungsi / Use Case** | **Intelligent Visual** |
| --- | --- | --- | --- |
| 1 | **Single Choice** | Voting standar (Pilkada, Ya/Tidak) | Pie/Donut Chart (Proporsi) + Tabel |
| 2 | **Multiple Choice** | Preferensi jamak (Hobi, Topping) | Vertical Bar Chart (Ranking Item) |
| 3 | **Rating Scale** | Kepuasan/Review (Bintang 1-5) | Gauge Meter (Rata-rata) + Bar Chart (Distribusi) |
| 4 | **Ranking (Drag-and-Drop)** | Mengurutkan prioritas/preferensi | Horizontal Bar Chart (Urutan Skor) |
| 5 | **Quick Feedback** | Like/Dislike instan atau Thumbs Up/Down | Scorecard (Ikon Jempol Hijau/Merah) |

### **7.1.2. Bab Public Opinion (Kualitatif)**

*Fokus: Teks, Sentimen, dan Kategorisasi Masalah.*

| **No** | **Subbab Fitur** | **Fungsi / Use Case** | **Intelligent Visual** |
| --- | --- | --- | --- |
| 1 | **Open Opinion** | Kotak saran bebas/Curhat (Teks 500 char) | Word Cloud (Keyword) + AI Summary |
| 2 | **Sentiment Check** | Cek ombak cepat (Teks pendek) | Sentiment Gauge (Positif/Negatif) + Stacked Bar |
| 3 | **Complaint System** | Formulir terstruktur (Input Judul + Pilih Kategori Masalah + Textbox Detail) | Treemap (Proporsi Kategori) + AI Summary |

### **7.1.3. Bab Forum & Discussion (Real-time)**

*Fokus: Interaksi dan Moderasi.*

| **No** | **Subbab Fitur** | **Fungsi / Use Case** | **Intelligent Visual** |
| --- | --- | --- | --- |
| 1 | **Town Hall (Moderated)** | Diskusi satu arah/Q&A terstruktur | Timeline Chart (Aktivitas) + Transkrip |
| 2 | **Open Forum** | Brainstorming bebas/Komunitas | Word Cloud (Topik Chat) + Leaderboard User |
| 3 | **Pinned Message** | Fitur Admin: Menyematkan topik/aturan di atas chat | Tidak ada visual khusus |

## **7.2. Standarisasi Visualisasi Data**

Finalisasi 9 jenis chart yang akan di-support oleh sistem (menggunakan Chart.js/ApexCharts) untuk menjaga kepraktisan namun tetap variatif.

### **7.2.1. Daftar Visualisasi Terpilih (The Chosen 9)**

1. **Pie/Donut Chart:** Proporsi sederhana (Single Choice)
2. **Vertical Bar Chart:** Perbandingan kategori (Multiple Choice)
3. **Horizontal Bar Chart:** Ranking/Urutan (Ranking)
4. **Gauge Meter:** Skor rata-rata/Meteran (Rating Scale, Sentiment Score)
5. **Word Cloud:** Frekuensi kata kunci (Open Opinion)
6. **Timeline Chart (Line/Area):** Aktivitas chat per waktu (Town Hall)
7. **Scorecard (Big Metric w/ Icon):** Angka besar/Ikon Jempol (Quick Feedback)
8. **Treemap:** Proporsi hierarki kategori masalah (Complaint System)
9. **Calendar Heatmap:** Intensitas respons berdasarkan tanggal

### **7.2.2. Logika "Intelligent Default"**

Sistem backend otomatis memilihkan visualisasi berdasarkan Subbab yang dipilih Admin:

**Contoh Logika:**

* **Input Admin:** Pilih "Rating Scale" → **Output Insight Studio:** Otomatis tampilkan **Bar Chart (Distribusi)** + **Gauge Meter (Rata-rata)**
* **Input Admin:** Pilih "Complaint System" → **Output Insight Studio:** Otomatis tampilkan **Treemap (Kategori)** + **AI Summary**

## 7.3. Logika "Report Composer" pada Insight Studio (Dashboard Utama)

Mekanisme pengolahan banyak data collection menjadi satu laporan.

### 7.3.1. Konsep "Report Composer"

Admin dapat memilih **lebih dari satu** data collection (misal: Data #1, #3, dan #5) untuk digabungkan menjadi satu laporan PDF.

### **7.3.2. Alur Kerja Composer**

Fitur untuk menggabungkan beberapa Subbab menjadi satu dokumen.

1. **Library View:** Admin melihat daftar semua koleksi datanya #1 (Voting), #3 (Komplain), #5 (Forum).
2. **Selection:** Admin mencentang beberapa koleksi (Checkbox)
3. **AI Super-Summary:** Backend mengirim data gabungan ke Gemini untuk meminta *satu paragraf kesimpulan* atau mencari korelasi (sebab-akibat) yang menghubungkan korelasi antar data tersebut
4. **Bundling:** Generate satu file PDF yang berisi narasi gabungan + detail masing-masing data. Sistem menyusun halaman PDF secara berurutan:
   * Cover
   * Summary Gabungan
   * Detail Data #1
   * Detail Data #3
   * Dan seterusnya

### **7.3.3. Manajemen Cache & Template**

* **Snapshot Cache:** Hasil hitungan (rata-rata/summary) disimpan sebagai JSON di Cache Server (agar halaman Insight Studio tidak me-load ulang AI).
* **Dual-Purging System:**
  1. Hapus Data Mentah (>7 Hari) -> Cron Job Harian.
  2. Hapus File Laporan PDF (>24 Jam) -> Cron Job Per Jam.
* **Desain Template:** "Laporan Jadi" (Executive Brief). Bukan chart kosong, tapi dokumen naratif dengan Header, KPI, Visualisasi, dan Footer.

### **7.3.4. Dinamisme View (Snapshot Cache)**

Untuk performa, hasil perhitungan (rata-rata, summary AI) disimpan sebagai **JSON Snapshot** di cache server, bukan dihitung ulang setiap kali halaman dibuka.

## **7.4. Manajemen Penyimpanan & Template**

Strategi teknis untuk menjaga prinsip "Zero Capital" namun tetap memberikan nilai premium.

### **7.4.1. Dual-Purging System (Sistem Pembersihan Ganda)**

1. **Cron Job #1 (Data Mentah):** Menghapus data input user di database (PostgreSQL) yang lebih tua dari **7 hari** (sesuai Pilar Bisnis)
2. **Cron Job #2 (File Laporan):** Menghapus file temporary (PDF/CSV yang baru di-generate) di folder storage yang lebih tua dari **24 jam**

### **7.4.2. Strategi Template "Laporan Jadi"**

**Format:** PDF Siap Cetak (Ready-to-Present)

**Desain:** Bukan grafik kosong, melainkan layout **"Executive Brief"** dengan struktur:

* **Header:** Judul & Metadata
* **Section 1:** AI Executive Summary (Narasi Teks)
* **Section 2:** Key Metrics (Angka Besar)
* **Section 3:** Visualisasi Utama (Chart)
* **Section 4:** Bukti Kualitatif (Quotes/Word Cloud)
* **Footer:** Branding [votim](http://vot.id)

## **7.5. Detail Advance Feature (Logika Khusus)**

Detail teknis untuk fitur spesifik yang baru ditambahkan.

### **7.5.1. Logika Complaint System**

**Input:**

* User wajib memilih "Kategori" (Dropdown) sebelum mengisi "Keluhan" (Textarea)

**Output:**

* Visualisasi **Treemap** dikelompokkan berdasarkan Kategori
* AI meringkas isi keluhan per kategori tersebut

### **7.5.2. Logika Quick Feedback**

**Input:**

* Tombol Biner (Like/Dislike atau Yes/No)

**Output:**

* **Scorecard** visual (Hijau/Merah) dengan persentase instan
* Tanpa analisis teks mendalam (hemat token AI)

### **7.5.3. Logika Pinned Message (Forum)**

**Fungsi:**

* Admin dapat menandai satu pesan sebagai "Sticky" di bagian atas UI Chat

**Storage:**

* Disimpan di tabel room\_settings atau kolom is\_pinned di tabel pesan
* Di-load pertama kali saat WebSocket connect

## **7.6. Strategi Implementasi AI (The Aggregator Engine)**

Strategi ini dirancang untuk **mengoptimalkan kuota gratis** Google Gemini API (30 RPM / 200 RPD) dengan teknik Batching.

### **7.6.1. Mekanisme Pemrosesan (Batch Processing)**

Sistem **TIDAK** mengirim request ke AI per-opini (1:1), melainkan menggunakan **Aggregator**:

1. **Tampung:** Opini yang masuk ditampung di tabel sementara (staging)
2. **Batching:** Queue Worker mengambil tumpukan data (misal: 20-50 opini sekaligus)
3. **Single Prompt:** Mengirim 1 Prompt Raksasa ke Gemini: *"Analisis 50 opini JSON berikut..."*
4. **Single Response:** Menerima 1 JSON Raksasa dan memecahnya kembali ke database  
     
    **Efisiensi:** 50 Opini = 1 Request AI

### **7.6.2. Skenario Beban (Worst Case Handling)**

**Low Load:**

* Throttling halus pada UI ("Sedang berpikir...")
* **Over Load (Limit Habis) - Circuit Breaker aktif:**
  + **Free User:** AI dimatikan sementara, tampilkan visual data mentah saja
  + **Pro User:** Masuk Delay Queue, hasil dikirim via notifikasi/email saat traffic reda

### **7.6.3. Batasan Limitasi Bisnis (AI Credit)**

| **Fitur** | **Free Plan** | **Pro Plan** |
| --- | --- | --- |
| **Limit Request** | Maks 3 Batch/Hari | 50 Batch/Hari |
| **Kecepatan** | Batch Lambat (Tunggu 20 item)/5 menit | Batch Cepat (Tiap 5 item/Instan) |
| **Report Composer** | 1 Laporan/Hari | Unlimited |
| **Co-Moderator** | Regex Filter (Tanpa AI) | AI Aktif (Strict Mode) |

## **7.7. Estimasi Biaya AI (Referensi)**

Untuk menjawab pertanyaan bisnis terkait Google Gemini API.

### **7.7.1. Free Tier (Saat ini digunakan)**

* **Cost:** $0
* **Limit:** 30 Request/Menit (RPM), 200 Request/Hari (RPD)
* **Penggunaan Data:** Google berhak menggunakan data input untuk melatih model (Privacy concern untuk Enterprise)

### **7.7.2. Pay-as-you-go (Untuk User Pro Plan) / $5**

* **Model:** Gemini 2.0 flash-lite (Cukup untuk analisis teks/opini)
* **Harga Input:** $0.075 per 1 Juta Token (~700.000 kata)
* **Harga Output:** $0.30 per 1 Juta Token

**Estimasi Kasar:**

* Jika 1 User Pro memproses 1.000 opini (masing-masing 50 kata) = ~50.000 kata
* Biaya API: Kurang dari **$0.01 (Rp 150 perak)** per analisis massal tersebut

**Kesimpulan:** Margin keuntungan Pro Plan sangat besar jika kita menggunakan model Flash dengan strategi Batching.

## **7.8. Roadmap Development (Fase 1 - MVP)**

**Minggu 1:**

* Setup Laravel, DB Schema (PostgreSQL), Auth (Google OAuth)
* Template Gallery UI

**Minggu 2:**

* Development CRUD 11 Subbab Fitur (Template Gallery)
* Alur Link Publik (Auto-Close)

**Minggu 3:**

* Integrasi AI (Aggregator Engine/Queue)
* Visualisasi Chart.js

**Minggu 4:**

* Insight Studio (PDF Generator)
* Scheduler (Auto-Purge)
* Testing Beban

# 8. DOKUMEN SPESIFIKASI TEKNIS: DATA & AI (FINAL MERGE)

**Fokus:** Logika Insight Studio, Struktur Database, & AI Prompt

## 8.1. LOGIKA INSIGHT STUDIO (FITUR TAMBAHAN)

### 8.1.1. Subbab: Quick Feedback (Like/Dislike)

* **Input User:**
  + Data biner (Boolean: true untuk Like, false untuk Dislike) atau Enum (thumbs\_up, thumbs\_down)
* **Proses Backend:**
  + Tidak butuh AI. Cukup SQL Aggregation sederhana:

COUNT(value) WHERE type='thumbs\_up'

* **Output Visual (Scorecard):**
  + Backend mengirim JSON:  
    {"likes": 150, "dislikes": 20, "score": 88}
  + Frontend (Blade/JS) me-render komponen Scorecard:
    - Ikon Jempol Hijau Besar (88%)
    - Bar warna (Hijau vs Merah)
    - Angka total partisipan

### 8.1.2. Subbab: Open Forum (Leaderboard + WordCloud)

Fitur ini memproses data transkrip chat.

* **Visual 1: Leaderboard (Keaktifan User)**
  + **Proses:** SQL Query pada tabel forum\_messages
  + **Logic:**SELECT sender\_name, COUNT(\*) as total FROM forum\_messages GROUP BY sender\_name ORDER BY total DESC LIMIT 10
  + **Output:** Tabel peringkat "Top 10 User Paling Aktif".
* **Visual 2: Word Cloud (Topik Hangat)**
  + **Proses:** Menggunakan AI Batching
  + **Logic:** Mengambil 100 pesan terakhir → Kirim ke Gemini → Minta ekstraksi keyword
  + **Output:** Chart.js me-render Word Cloud berdasarkan frekuensi keyword yang dikembalikan AI

## 8.2. SKEMA DATABASE (ERD) & STRUKTUR JSON

Versi ini menggunakan konsep "Folder" dan "Activity" serta mencakup strategi Indexing untuk Auto-Purge.

### 8.2.1. Grup Pengguna (Auth & Plan)

* **Table: users**
  + id (PK, BigInt)
  + display\_name (String) - Display Name
  + email (String, Unique)
  + password (String)
  + google\_id (String, Nullable)
  + plan\_type (Enum: 'free', 'pro')
  + subscription\_ends\_at (Timestamp, Nullable)
  + created\_at, updated\_at

### 8.2.2. Grup Koleksi Data (Folder & Subbab)

* **Table: folders** (Wadah Utama / BAB)
  + id (PK, BigInt), user\_id (FK → users.id)
  + module (Enum: 'voting', 'opinion', 'forum') - Penanda BAB
  + name (String) - Nama Folder (misal: "Evaluasi Q3")
  + created\_at - Acuan utama Auto-Purge Folder Kosong
* **Table: activities** (Item Fitur / SUBBAB)
  + id (PK, BigInt), folder\_id (FK → folders.id)
  + type (Enum: 'single\_choice', 'rating', 'quick\_feedback', 'complaint', 'town\_hall', etc)
  + title (String), slug (String, Unique, Indexed) - Link Publik
  + status (Enum: 'active', 'closed'), closed\_at (Timestamp) - Acuan Auto-Close
  + settings (JSONB) - Menyimpan konfigurasi fleksibel
* **Contoh JSON Settings (Voting):**{ "options": ["Opsi A", "Opsi B"], "manual\_close": false, "allow\_anonymous": true }
* **Contoh JSON Settings (Opini/Komplain):**{ "char\_limit": 500, "regex\_filter": "^[a-zA-Z0-9 ]+$", "complaint\_categories": ["Pelayanan", "Fasilitas"] }

### 8.2.3. Grup Data Mentah (Transactional)

* **Table: responses** (Hasil Input User)
  + id (PK, BigInt), activity\_id (FK → activities.id)
  + value\_data (JSONB) - Menyimpan jawaban user
    - Isi JSON (Rating): {"rating": 5}
    - Isi JSON (Komplain): {"category": "Fasilitas", "text": "AC Panas"}
    - Isi JSON (Quick Feedback): {"feedback": "thumbs\_up"}
  + is\_processed\_by\_ai (Boolean, Default: False)
  + created\_at (Indexed) - Acuan Auto-Purge Data (7 Hari)
* **Table: forum\_messages** (Chat Transcript)
  + id (PK, BigInt), activity\_id (FK → activities.id)
  + sender\_name (String), content (Text)
  + is\_pinned (Boolean, Default: False), created\_at (Indexed)

### 8.2.4. Grup Analisis AI & Laporan (Insight Engine)

* **Table: ai\_aggregates** (Cache Hasil AI)
  + id (PK, BigInt), activity\_id (FK → activities.id)
  + batch\_date (Date)
  + summary\_data (JSONB) - Menyimpan hasil olahan AI (Sentiment count, Keywords list, Summary text) agar siap visualisasi
* **Table: report\_snapshots** (Cache Laporan PDF)
  + id (PK, UUID), user\_id (FK)
  + file\_path (String)
  + created\_at, expires\_at (Timestamp) - Acuan Cron Job penghapusan file temp (24 jam)

## 8.3. AI PROMPT LIBRARY (JSON FORMAT)

Diperbarui dengan instruksi Deteksi Bahasa dan Output JSON Murni.

### 8.3.1. System Instruction Global

Ini adalah instruksi "kunci" yang dikirim di setiap request untuk memastikan format JSON dan bahasa yang tepat.

{

"role": "system",

"content": "Anda adalah mesin analisis data JSON-to-JSON untuk votim. TUGAS: 1) Terima input array data. 2) DETEKSI BAHASA dominan (Indo/Inggris). 3) Output HANYA JSON valid. JANGAN gunakan markdown. JANGAN sertakan data pribadi."

}

### 8.3.2. Prompt: Batch Analysis (Opini & Komplain)

Untuk Subbab: Open Opinion, Complaint System.

{

"task": "analyze\_opinion\_batch",

"parameters": {

"temperature": 0.2,

"max\_output\_tokens": 2000

},

"input\_context": "List opini/komplain: BATCH\_JSON\_ARRAY",

"instruction": "Analisis setiap item. 1) Tentukan sentimen. 2) Ekstrak 1 keyword utama (benda/topik). 3) Buat ringkasan global dari seluruh batch.",

"expected\_output\_schema": {

"items": [

{

"id": "Int",

"sentiment": "String (positif|negatif|netral)",

"keyword": "String"

}

],

"global\_summary": "String (Ringkasan tren masalah)",

"dominant\_sentiment": "String"

}

}

### 8.3.3. Prompt: Report Composer (Insight Studio)

Untuk menggabungkan beberapa Folder/Subbab menjadi satu narasi.

{

"task": "generate\_executive\_report",

"parameters": {

"temperature": 0.7,

"max\_output\_tokens": 1000

},

"input\_context": "Statistik gabungan: AGGREGATED\_DATA\_JSON",

"instruction": "Bertindak sebagai Analis Data. Tulis 'Executive Summary' yang menghubungkan titik-titik korelasi antar data yang diberikan. Berikan rekomendasi aksi.",

"expected\_output\_schema": {

"report\_title": "String",

"executive\_summary": "String (Paragraf narasi)",

"key\_takeaway": "String (1 kalimat rekomendasi)"

}

}

### 8.3.4. Prompt: AI Co-Moderator (Forum - Pro Only)

Untuk moderasi real-time.

{

"task": "moderate\_chat\_message",

"parameters": {

"temperature": 0.0,

"max\_output\_tokens": 100

},

"input\_text": "MESSAGE\_CONTENT",

"instruction": "Analisis pesan ini. Apakah mengandung Hate Speech, SARA, atau Ancaman Fisik? Jawab JSON.",

"expected\_output\_schema": {

"is\_flagged": "Boolean",

"category": "String (hate\_speech|safe|spam)",

"confidence\_score": "Float (0.0 - 1.0)"

}

}

### 8.3.5. Prompt: Open Forum Topic Extraction (Baru)

Untuk menghasilkan Word Cloud dari riwayat chat forum.

{

"task": "extract\_forum\_topics",

"parameters": {

"temperature": 0.3,

"max\_output\_tokens": 500

},

"input\_context": "Transkrip chat: CHAT\_BATCH\_TEXT",

"instruction": "Abaikan kata sapaan/basa-basi. Ekstrak 10 topik atau kata benda yang paling sering dibahas secara substansial.",

"expected\_output\_schema": {

"topics": [

{

"text": "String",

"frequency\_score": "Int (1-100)"

}

]

}

}

# 9. FORMAT HTTP JSON & INTERAKSI API

Ini adalah jembatan komunikasi antara: Frontend (Blade/Livewire) dengan Backend (Laravel Controller), Backend dengan AI Service (Gemini), dan Backend dengan Frontend (Visualisasi Chart.js). Tujuannya adalah mendefinisikan standar komunikasi yang konsisten, aman, dan bebas *error* (*robust*).

## 9.1. Standar Respons JSON (Backend ke Frontend)

Setiap *request* AJAX atau *Livewire dispatch* yang membutuhkan data (bukan HTML) wajib mengikuti format ini.

### 9.1.1. Format Sukses (Standard Success Response)

Digunakan untuk mengirim data visualisasi ke Chart.js atau konfirmasi aksi.

{

"status": "success",

"code": 200,

"data": {

// Payload spesifik fitur ada di sini

"chart\_data": {

"..."

},

"summary": "..."

},

"meta": {

"timestamp": "2023-10-27T10:00:00Z",

"version": "1.0"

}

}

### 9.1.2. Format Error (Standard Error Response)

Digunakan untuk menangani validasi gagal, limit habis, atau *error* sistem.

{

"status": "error",

"code": 422, // atau 400, 403, 500

"message": "Validasi Gagal", // Pesan untuk UI (Toaster/Alert)

"errors": {

"email": [

"Email wajib diisi"

], // Detail field error

"captcha": [

"Verifikasi gagal"

]

},

"error\_code": "VALIDATION\_ERROR" // Kode internal untuk debugging

}

## 9.2. Detail Payload JSON per Fitur (Frontend <-> Backend)

### 9.2.1. Payload Visualisasi: Voting & Polling

Untuk Subbab: Single/Multiple Choice, Rating, Quick Feedback.

*Request (Frontend meminta data):* GET /api/v1/insight/chart/{activity\_id}

*Response (Backend mengirim ke Chart.js):* Backend harus mengubah data mentah DB menjadi format yang dimengerti Chart.js.

{

"status": "success",

"data": {

"type": "bar", // Jenis chart (bar/pie/doughnut)

"labels": [

"Opsi A",

"Opsi B",

"Opsi C"

],

"datasets": [

{

"label": "Jumlah Suara",

"data": [

15,

25,

5

],

"backgroundColor": [

"#36A2EB",

"#FF6384",

"#FFCE56"

] // Warna tema glassmorphism

}

],

// Data tambahan untuk Gauge Meter (Rating Scale)

"meta\_score": {

"average": 4.2,

"max": 5,

"total\_votes": 45

}

}

}

### 9.2.2. Payload Visualisasi: Word Cloud (Opini)

Untuk Subbab: Open Opinion, Open Forum.

*Request:* GET /api/v1/insight/wordcloud/{activity\_id}

*Response:* Mengirim *array* *keyword* yang sudah dihitung frekuensinya (hasil agregasi AI).

{

"status": "success",

"data": {

"chart\_type": "wordCloud",

"words": [

{

"text": "Pelayanan",

"weight": 50

},

{

"text": "Lambat",

"weight": 30

},

{

"text": "Ramah",

"weight": 20

}

],

// Ringkasan AI (Teks Narasi)

"ai\_summary": "Pelanggan umumnya mengapresiasi keramahan staf, namun mengeluhkan kecepatan pelayanan yang lambat saat jam sibuk."

}

}

### 9.2.3. Payload Visualisasi: Complaint System (Treemap)

Untuk Subbab: Complaint System.

*Response:* Struktur hierarki untuk Treemap.

{

"status": "success",

"data": {

"chart\_type": "treemap",

"tree\_data": [

{

"category": "Fasilitas",

"value": 40, // Jumlah komplain

"children": [

// (Opsional) Detail sub-masalah dari AI

{

"name": "AC Panas",

"value": 25

},

{

"name": "Kursi Rusak",

"value": 15

}

]

},

{

"category": "Pelayanan",

"value": 20,

"children": [

"..."

]

}

]

}

}

### 9.2.4. Payload Transaksi: Submit Response (Input User)

Digunakan saat *user* mengisi *form* publik.

*Request:* POST /api/v1/submit/{slug}

{

"fingerprint": "hash\_browser\_uniq\_id", // Untuk rate limiting ringan

"recaptcha\_token": "token\_dari\_google", // Wajib

"responses": {

// Struktur dinamis tergantung tipe subbab

"rating": 5,

"text": "Sangat puas dengan pelayanan.",

"selected\_options": [

"Nasi Goreng",

"Es Teh"

]

}

}

## 9.3. Format Interaksi Backend <-> AI (Internal)

Format ini digunakan oleh *Queue Worker* saat berkomunikasi dengan Gemini API. Ini tidak terekspos ke publik.

### 9.3.1. Request ke Gemini (Batch Processing)

Dikirim sebagai satu *string prompt* raksasa.

{

"contents": [

{

"parts": [

{

"text": "SYSTEM: [System Instruction 8.3.1]\nTASK: [Prompt 8.3.2]\nDATA: [{\"id\":1, \"text\":\"AC panas\"}, {\"id\":2, \"text\":\"Pelayan ramah\"}]"

}

]

}

],

"generationConfig": {

"response\_mime\_type": "application/json" // Memaksa output JSON

}

}

### 9.3.2. Response dari Gemini (Raw -> Parsed)

Gemini akan membalas dengan *JSON string*. *Backend* harus mem-*parse* ini.

*Raw Response (String):*

"```json

{

"items": [

{

"id": 1,

"sentiment": "negatif",

"keyword": "AC"

},

{

"id": 2,

"sentiment": "positif",

"keyword": "Pelayanan"

}

],

"global\_summary": "Masalah utama fasilitas, namun pelayanan diapresiasi."

}

```"

*Validasi Backend (Anti-Error):*

* **Strip Markdown:** Hapus karakter ```json dan di awal/akhir \*string\*.
* **JSON Decode:** Coba json\_decode()
* **Fallback:** Jika *decode* gagal (format rusak), tandai *job* sebagai *failed* dan jadwalkan *retry* (maks 3x).
* **Sanitasi:** Pastikan ID yang dikembalikan AI cocok dengan ID yang dikirim (mencegah halusinasi data).

## 9.4. Implementasi Error Handling & Edge Cases

Untuk memastikan sistem "tanpa celah *error*", kita harus menangani skenario berikut di level JSON.

### 9.4.1. Skenario: Data Kosong (Belum ada responden)

Jangan biarkan Chart.js *error* (*blank screen*). Kirim JSON "Empty State".

{

"status": "success",

"data": {

"is\_empty": true,

"message": "Belum ada data masuk. Bagikan link Anda untuk mulai mengumpulkan data.",

"chart\_data": [] // Array kosong

}

}

*Frontend:* Tampilkan ilustrasi "No Data" yang ramah, bukan *chart* kosong yang jelek.

### 9.4.2. Skenario: AI Belum Selesai (Processing)

Jika Admin membuka *dashboard* tapi *Queue* AI masih bekerja (status is\_processed = false).

{

"status": "success",

"data": {

"is\_processing": true,

"progress": 45, // Persentase (Opsional)

"message": "AI sedang menganalisis 50 opini baru...",

// Kirim data parsial yang sudah ada (jika ada) agar tidak blank total

"partial\_data": {

"..."

}

}

}

*Frontend:* Tampilkan *Loader/Spinner* di bagian *widget* AI (Word Cloud/Summary), tapi tetap tampilkan data Voting (Bar Chart) yang sudah *ready*.

### 9.4.3. Skenario: Circuit Breaker (API Limit Habis)

Jika *Backend* mendeteksi Error 429 dari Google.

{

"status": "partial\_success", // Kode khusus

"data": {

"ai\_active": false,

"message": "Analisis AI sedang sibuk. Menampilkan data visual mentah saja.",

// Kirim Word Cloud sederhana berbasis hitungan kata manual (non-AI) sebagai fallback

"fallback\_wordcloud": [

"..."

]

}

}

**Kesimpulan**

Implementasi Format JSON di atas dirancang untuk:

* **Konsistensi:** *Frontend* selalu tahu struktur data, *meta*, dan *status*.
* **Ketahanan (*Robustness*):** Menangani *loading state, empty state*, dan *error state* dengan elegan.
* **Efisiensi:** Mengirim data yang sudah "matang" (siap dimakan Chart.js) sehingga *Frontend* tidak perlu melakukan kalkulasi berat.

# 10. MANAJEMEN RISIKO (ERROR HANDLING)

## 10.1. SKENARIO ERROR & CIRCUIT BREAKER (INSIGHT STUDIO)

Kita tidak boleh membiarkan *dashboard* menjadi putih (blank) saat AI gagal. Kita harus punya **"Graceful Degradation"** (Penurunan Kualitas yang Elegan).

### 10.1.1. Logika Circuit Breaker (Backend)

*Mekanisme pemutus arus jika API Google macet agar server kita tidak hang.*

* **Ambang Batas (Threshold):** Jika terjadi **5x Error (500/429)** dari Gemini dalam **1 menit**.
* **Aksi (Open Circuit):** Stop memanggil API Gemini selama **5 menit**.
* **Fallback Strategy:** Beralih ke mode "Manual Calculation".

### 10.1.2. Skenario Kegagalan & Pesan Error (User Friendly)

Berikut adalah matriks pesan yang akan muncul di *frontend* (Toast Notification atau Card State):

| **Skenario Error** | **Penyebab Teknis** | **Pesan untuk User (Bahasa Manusia)** | **Fallback Action (Apa yang ditampilkan?)** |
| --- | --- | --- | --- |
| **Limit Habis** | Error 429 (Too Many Requests) dari Google | *"Layanan AI sedang sibuk. Menampilkan data visual dasar."* | Tampilkan Grafik Batang & Word Cloud manual (hitung frekuensi kata via SQL biasa). Sembunyikan "AI Summary". |
| **AI Halusinasi** | Output JSON rusak/tidak valid | *"Gagal merangkum opini. Silakan coba sesaat lagi."* | Tampilkan tombol "Coba Lagi" (Manual Retry). Data visual tetap muncul. |
| **Data Kosong** | User belum mengisi form | *"Belum ada data masuk. Bagikan link Anda untuk memulai!"* | Tampilkan ilustrasi "Empty State" + Tombol "Copy Link". |
| **Konten Toksik** | AI mendeteksi input admin yang melanggar policy | *"Permintaan tidak dapat diproses karena terdeteksi konten sensitif."* | Blokir request tersebut. |
| **Network Error** | Koneksi server ke Google putus | *"Koneksi terganggu. Data tersimpan aman, coba refresh."* | Tampilkan data terakhir dari *Cache Snapshot*. |

### 10.1.3. JSON Error Response (Standar Frontend)

Jika Insight Studio gagal memuat, Backend mengirim JSON ini: (Teks JSON Asli tidak disertakan, hanya judul sub-bagian)

# 

# 11. SPESIFIKASI DESAIN UI/UX: votim (SPATIAL SAAS)

**Konsep Visual:** Deep Glassmorphism + Lightweight 3D Elements. **Mood:** Profesional, Futuristik, "Living Interface" (Tidak Statis).

## 1. DESIGN SYSTEM (PONDASI VISUAL)

Sebelum masuk ke layar, kita tentukan aturan main visualnya agar konsisten.

### 1.1. Background & Atmosfer (The 3D Element)

* **Base:** #0B0F19 (Deep Space Blue - Hampir Hitam).
* **Ambient 3D:** Jangan gunakan gambar diam. Gunakan *CSS animated gradient blobs* atau *Spline 3D Objects* yang sangat sederhana (bola kaca buram, kubus tumpul) yang melayang/berputar sangat lambat di pojok layar.
  + *Tujuan:* Memberikan kesan "hidup" tanpa membebani browser.
* **Lighting:** Sumber cahaya imajiner dari pojok kiri atas, memberikan highlight putih tipis pada ujung-ujung kartu kaca.

### 1.2. Material Kaca (Glass Cards)

* **Surface:** background: rgba(255, 255, 255, 0.03); + backdrop-filter: blur(20px);.
* **Border:** 1px solid rgba(255, 255, 255, 0.08); (Sangat tipis).
* **Shadow:** box-shadow: 0 8px 32px 0 rgba(0, 0, 0, 0.37); (Bayangan lembut berwarna).
* **Active State:** Saat diklik/hover, border menyala menjadi warna aksen (#8B5CF6).

### 1.3. Typography & Iconography

* **Font:** **Plus Jakarta Sans** (Geometris, modern, cocok untuk angka).
* **Icons:** **Phosphor Icons** (Style: Duotone). Memberikan kedalaman lebih daripada ikon solid biasa.

## 2. STRUKTUR NAVIGASI (GLOBAL SHELL)

Layout aplikasi ini menggunakan **L-Shape Layout** (Sidebar Kiri + Top Bar + Konten Utama).

### 2.1. Sidebar Dinamis (Kiri)

* **Lebar:** 260px (Fixed).
* **Material:** Kaca buram penuh setinggi layar.
* **Menu Items (5 Fitur Utama):**
  + Setiap item memiliki ikon di kiri + teks.
  + **State Active:** Background pill transparan dengan *glow* ungu neon di kiri.
  + **Urutan:**
    1. **Insight Studio** (Ikon: ChartPolaris - Bersinar).
    2. **Voting / Polling** (Ikon: Gavel).
    3. **Public Opinion** (Ikon: ChatTeardropText).
    4. **Room Forum** (Ikon: UsersThree).
    5. **Settings** (Ikon: Gear - di bagian bawah).
* **Micro-Interaction:** Saat hover, ikon sedikit membesar (scale 1.1).

## 3. DETAIL LAYAR PER FITUR (THE FLOW)

### LAYAR 1: INSIGHT STUDIO (The Command Center)

*Ini adalah halaman default. Harus terasa seperti kokpit pesawat.*

**A. Header Area (Top Bar)**

* **Kiri:** Judul "Insight Studio".
* **Kanan (AI Quota):** Sebuah *capsule widget* kecil.
  + Visual: Barometer lingkaran kecil + Teks "AI Credits: 850/1000".
  + Warna: Hijau (Aman) -> Kuning (Menipis) -> Merah (Habis).
  + *Tidak mengganggu, tapi terlihat.*

**B. Main Stage: "The Report Composer" (Area Utama)**

* **Visual:** Sebuah area *drop-zone* besar di tengah dengan garis putus-putus menyala (*dashed glow border*).
* **Action:** Tombol besar di tengah "Compose New Report".
* **Interaksi (Saat tombol diklik):**
  + Muncul **Panel "Source Selector"** (Melayang di kanan).
  + **UI Breadcrumbs Checkbox:** Tampilan pohon (Tree View) folder & subbab.
    - [ ] Voting (Bab)
      * [v] Folder: Evaluasi Q3 (User mencentang ini)
        + [v] Rating Kantin (Subbab)
        + [v] Voting Ketua (Subbab)
  + User bisa memilih folder utuh atau memecah isinya.

**C. Widget Area (Bawah Composer)**

* **Latest Activity:** List horizontal (*carousel*) kartu kecil berisi koleksi data terakhir yang aktif.
* **Quick Stats:** 3 Kartu kecil berisi total responden hari ini.

### LAYAR 2: FITUR MODUL (Voting / Opinion / Forum)

*Contoh kasus: User mengklik menu "Voting / Polling".*

**A. Header & Limit Indicator**

* Judul: "Voting & Polling".
* **Free Plan Limit Bar:** Di sebelah judul, ada *progress bar* tipis.
  + Teks: "Active Collections: 2/3 (Free Plan)".
  + Warna: Jika 3/3, bar berubah merah dan muncul tombol kecil "Upgrade".

**B. Folder Management (Bagian Atas)**

* Tampilan Grid ikon Folder (Seperti MacOS Finder).
* Folder memiliki label nama (misal: "Event Desember").
* **Tombol "Create Folder":** Ikon Folder dengan tanda Plus.

**C. Activity List (Bagian Bawah - Di dalam Folder)**

* Jika user mengklik Folder, tampilan masuk ke dalam (Breadcrumb: Voting > Event Desember).
* **Tombol Utama (CTA):** **"New Activity"** (Besar, Gradient Purple).
* **List Data:** Tabel daftar voting yang sudah dibuat (Judul | Tipe | Status | Responden | Tanggal).

### LAYAR 3: TEMPLATE GALLERY (The Creation Modal)

*Muncul saat user klik "New Activity". Ini harus visual, jangan cuma teks.*

**A. Modal Window**

* Overlay gelap (Blur background).
* Jendela kaca besar di tengah.

**B. Content: The Gallery**

* Judul: "Pilih Tipe Voting".
* **Grid Card (Pilihan Subbab):**
  + Setiap kartu mewakili 1 Subbab (Single Choice, Rating, Ranking, Quick Feedback).
  + **Isi Kartu:**
    - **Ikon 3D Besar/Ilustrasi:** Misal untuk "Rating", gambar 3 bintang melayang 3D.
    - **Judul:** "Rating Scale".
    - **Deskripsi:** "Cocok untuk survei kepuasan."
* **Hover Effect:** Kartu terangkat (levitate) dan border bersinar saat mouse lewat.

### LAYAR 4: CONFIGURATION & PREVIEW (Split Screen)

*User sudah memilih template (misal: Rating Scale). Sekarang mengeditnya.*

**Layout: Terbagi Dua (Kiri: Editor, Kanan: Live Preview)**

**A. Panel Kiri: Konfigurasi (Formulir)**

* Input Judul Pertanyaan.
* **Tab Settings (Accordion):**
  + **General:** Ikon Bintang/Hati, Skala 1-5 atau 1-10.
  + **Restrictions:** Toggle "Allow Anonymous", "Limit Character" (untuk opini).
  + **Advanced (Regex):** Input field dengan ikon tanda tanya (?) untuk tooltip bantuan.
  + **Lifecycle:** Toggle "Manual Close" & "Auto-Close Timer".

**B. Panel Kanan: Live Preview (Tampilan HP)**

* Mockup bingkai HP (Phone Frame).
* Di dalamnya menampilkan bagaimana form akan terlihat oleh audiens secara *real-time*.
* Saat admin mengetik judul di kiri, teks di mockup kanan berubah.

### LAYAR 5: PUBLIC VIEW (Tampilan Audiens)

*Tampilan di HP responden.*

**A. Clean Interface**

* Background: Polos/Gradient sangat halus (agar fokus).
* **Card Tengah:**
  + Logo votim kecil di atas (atau Logo Pro).
  + Pertanyaan Besar.
  + **Input Element:** (Misal: 5 Bintang yang bisa diklik).
  + Tombol "Submit" besar.
* **Feedback:** Setelah submit, animasi "Confetti" halus + Pesan "Terima kasih".

### LAYAR 6: REPORT RESULT (Insight Studio Output)

*Setelah Admin klik "Generate" di Insight Studio.*

**A. Layout Dokumen (PDF Preview)**

* Tampilan seperti kertas A4 digital di tengah layar.
* **Header:** Judul Laporan & Tanggal.
* **AI Summary Box:** Area dengan background berbeda (misal: ungu sangat muda transparan) berisi teks narasi. Ada ikon "Sparkles" menandakan ini AI.
* **Charts:**
  + **Chart Utama:** Grafik Batang/Pie Chart (menggunakan library Chart.js tapi di-styling CSS agar terlihat modern).
  + **Word Cloud:** Awan kata dengan warna-warni sesuai tema.
* **Action Bar (Floating di bawah):** Tombol "Download PDF", "Share Link".

### 4. ALUR INTERAKSI (MICRO-INTERACTIONS)

Agar tidak membosankan (static), tambahkan ini:

1. **Loading State (AI Processing):**
   * Jangan gunakan *spinner* putar biasa.
   * Gunakan animasi **"Breathing Gradient"** pada kartu Insight Studio saat AI sedang berpikir. Tulisan: *"AI is connecting the dots..."*
2. **Folder Opening:**
   * Saat folder diklik, ia tidak sekadar pindah halaman. Folder tersebut "zoom in" (membesar) dan memudar, lalu isinya muncul (*fade in*). Transisi mulus (0.3s).
3. **Hover Cards:**
   * Semua elemen yang bisa diklik memiliki efek *glass reflection* (seperti cahaya lewat di atas kaca) saat di-hover.

**Rangkuman untuk Desainer/Developer:** Dokumen ini menggabungkan struktur data "Folder" yang telah kita sepakati dengan estetika "Spatial Glass". Tampilannya bersih (SaaS standar) namun memiliki *delight* visual melalui elemen 3D minimalis dan efek kaca, sesuai dengan permintaan Anda untuk menghindari kesan membosankan. Batasan Free Plan ditempatkan strategis di titik pembuatan (Bab Fitur) untuk mendorong upgrade, sementara Insight Studio dibiarkan bersih sebagai pusat nilai (*value center*).

## 12. LEGALITAS & PRIVASI (COMPLIANCE)

Ini adalah tameng hukum Anda. Karena kita pakai AI Gratis (Google menggunakan data untuk training), kita harus transparan.

### 12.1. Terms of Service (ToS) - Poin Kunci

Jangan *copy-paste* buta, tapi pastikan ada poin ini:

1. **"As-Is" Service:** Layanan disediakan "sebagaimana adanya". Anda tidak bertanggung jawab jika AI salah menganalisis sentimen yang menyebabkan keputusan bisnis yang buruk.
2. **AI Usage Disclaimer:** "Fitur analisis menggunakan teknologi AI eksperimental. Hasil mungkin tidak 100% akurat dan dapat mengandung bias."
3. **Data Processing:** Pengguna menyetujui bahwa data teks (opini) akan diproses oleh pihak ketiga (Google Gemini API) untuk tujuan analisis.

### 12.2. Privacy Policy - Poin Kunci (Zero Capital Strategy)

1. **Data Retention:** "Kami HANYA menyimpan data mentah Anda selama **7 hari**. Setelah itu, data akan dihapus permanen dari server kami demi keamanan dan efisiensi." (Ini melindungi kita dari tuntutan jika user minta data lama).
2. **Third-Party Sharing:** "Data anonim diproses melalui API Google Gemini sesuai dengan kebijakan privasi Google."
3. **User Content:** "Anda dilarang mengunggah data pribadi sensitif (NIK, No. Rekening) ke dalam form."

### 12.3. Cara Testing User (UAT - User Acceptance Testing)

Sebelum rilis publik, lakukan ini:

1. **Alpha Test (Internal):** Anda sendiri berpura-pura jadi 5 user berbeda. Coba "hancurkan" sistem (input karakter aneh, input 1000 kata, *spam* tombol).
2. **Beta Test (Friends):** Ajak 5 teman.
   * Minta 1 orang jadi Admin (Buat Voting).
   * Minta 4 orang jadi Audiens (Isi Voting).
   * Tanya Admin: "Apakah grafiknya dimengerti?", "Apakah PDF-nya berguna?".
3. **Stress Test (AI):** Coba input 50 opini sekaligus, lihat apakah *Queue* berjalan mulus atau macet.

**Penutup: Kesiapan Proyek**

Dengan dokumen Error Handling dan Resep Desain ini, Anda sudah memiliki paket lengkap:

1. **Otak:** Logika Bisnis & AI Prompt.
2. **Tulang:** Skema Database & JSON.
3. **Wajah:** Design System Glassmorphism.
4. **Tameng:** Legalitas & Error Handling.

# ADDENDUM: OPERATIONAL & SUPPORT REQUIREMENTS

Status: Final Checklist (Non-Core Features). Fokus: Stabilitas, Edukasi, Keamanan Bisnis, dan UX Mobile.

## 1. Sistem Notifikasi & Alert (Async Transactional)

***Tujuan:*** *Komunikasi sistem ke user tanpa memblokir proses utama.*

* **Use Case:** Notifikasi "Laporan Report Composer Siap" (karena proses AI lama), verifikasi email, dan *alert* link kedaluwarsa.
* **Implementasi:**
  + Driver: SMTP Standar (Gunakan Free Tier Brevo atau Mailtrap untuk MVP).
  + Queue: Pengiriman email wajib via Queue emails agar user tidak menunggu loading saat klik tombol.
  + In-App Notification: Menggunakan tabel notifications bawaan Laravel untuk lonceng notifikasi di dashboard.

## 2. Super Admin Dashboard (God Mode)

***Tujuan:*** *Monitoring dan kontrol penuh pemilik SaaS.*

* **Tech Stack:** FilamentPHP (Pilihan terbaik. Cepat, gratis, terintegrasi Laravel).
* **Fitur Wajib:**
  + User Management: Ban user, lihat status langganan.
  + Quota Monitoring: Widget sederhana menampilkan total request AI hari ini vs Limit Google.
  + Global Purge Button: Tombol darurat untuk membersihkan file temp/data lama jika storage server penuh mendadak.

## 3. SEO & Social Sharing (Dynamic Open Graph)

***Tujuan:*** *Meningkatkan Click-Through-Rate (CTR) saat link dibagikan di WA/Twitter.*

* **Implementasi:** Komponen Blade Layout dinamis.
  + og:title: "Ikuti Voting: [Judul Koleksi]"
  + og:description: "Berikan suaramu sekarang di votim..."
  + og:image: Generate gambar sederhana on-the-fly atau gunakan logo votim default.

## 4. Onboarding & Edukasi User

***Tujuan:*** *Mengurangi kebingungan user awam terhadap fitur teknis.*

* **Micro-Interaction:** Tooltip (?) pada kolom Regex dan Kategori Komplain (Hover state).
* **Product Tour:** Menggunakan library Driver.js (Ringan, Vanilla JS).
  + *Skenario:* Saat pertama kali masuk Insight Studio, sorot tombol "Generate Report" dan beri teks "Klik di sini untuk membuat laporan gabungan".

## 5. Mobile Strategy (Insight Studio)

***Tujuan:*** *Mengatasi keterbatasan layar HP untuk visualisasi data kompleks.*

* **Portrait Mode (Default HP):** Tampilkan Simplified View.
  + Jangan render Chart kompleks. Ganti dengan List/Card View (Angka ringkasan).
  + Alasan: Chart.js di lebar 360px seringkali tidak terbaca.
* **Landscape Hint:** Jika user ingin melihat Chart penuh, tampilkan overlay: *"Putar HP Anda ke Landscape untuk visualisasi detail."*

## 6. Maintenance & Downtime Strategy

***Tujuan:*** *Komunikasi profesional saat terjadi gangguan.*

* **Custom 503 Page:** Edit file view error Laravel (errors/503.blade.php).
* **Desain:** Glassmorphism dengan pesan ramah: *"Sistem sedang istirahat sebentar untuk upgrade. Kembali dalam 15 menit."*
* **Command:** Gunakan php artisan down --secret="kode-rahasia-admin" agar Anda tetap bisa akses saat user lain melihat maintenance.

## 7. Backup & Disaster Recovery

***Tujuan:*** *Keamanan data untuk User Pro (Selling Point).*

* **Library:** Spatie Laravel Backup.
* **Strategi:**
  + Backup Database + File JSON penting.
  + Frekuensi: Mingguan (Otomatis via Scheduler).
  + Destinasi: Google Drive (via API Gratis) atau S3 Compatible Storage (Cloudflare R2 - Free Tier besar).

## 8. Fair Usage Policy (FUP) & Quota Logic

***Tujuan:*** *Mencegah Pro User melakukan spamming yang mematikan server.*

* **Aturan Bisnis:** "Unlimited" tidak pernah benar-benar tak terbatas.
* **Limitasi:**
  + Hard Limit: Misal 5.000 Request AI / Hari.
  + Action: Jika lewat, kirim email peringatan dan *throttle* kecepatan (perlambat queue), jangan langsung ban.
  + Legalitas: Cantumkan klausul FUP ini di *Terms of Service*.

### KESIMPULAN PROYEK & CHECKPOINT TERAKHIR

* Feature Completeness: 100% (Core + Advance + Support).
* Technical Clarity: 100% (Stack, ERD, JSON, AI Prompt).
* Business Logic: 100% (Freemium, FUP, Lifecycle Data).
* Risk Management: 100% (Error Handling, Backup, Compliance).

## 9. STRATEGI MITIGASI "SURVIVAL MODE" (AI & DATABASE)

Status: Wajib Implementasi untuk Skalabilitas Zero Capital.

Fokus: Menangani beban tinggi (100+ Admin) dan limitasi free tier resource.

### 9.1. Database Survival Strategy

*Tujuan: Mencegah database penuh atau crash karena koneksi berlebih.*

1. **Ephemeral Storage (Anti-BLOB):**
   * **Aturan:** DILARANG KERAS menyimpan file biner (PDF/Gambar) di database.
   * **Mekanisme:** File laporan disimpan di /tmp server (RAM/Disk sementara).
   * **Lifecycle:** File dihapus otomatis oleh Cron Job setiap 1 jam setelah diunduh.
2. **Aggressive Job Pruning:**
   * **Masalah:** Tabel jobs dan failed\_jobs membengkak cepat.
   * **Solusi:** Set failed\_jobs agar otomatis terhapus setelah **24 jam**. Gunakan DB::transaction untuk menghapus *row* job sukses secara instan.
3. **Connection Pooling:**
   * **Implementasi:** Menggunakan *PgBouncer* (bawaan Neon/Supabase) untuk membagi koneksi database yang terbatas kepada ratusan *request* konkuren dalam milidetik.

### 9.2. Aggregator Engine V2 & Fault Tolerance

*Tujuan: Menghemat token AI dan mencegah kegagalan massal.*

1. **Validasi Pra-Batch (Gatekeeper):**
   * Validasi ketat di Controller sebelum data masuk Queue. Tolak input >500 karakter atau JSON cacat di awal (400 Bad Request).
2. **Token Limiter (Output Control):**
   * Set parameter max\_output\_tokens pada API Request untuk membatasi jawaban Gemini (misal: maks 20 kata per item). Mencegah AI "mengarang bebas" yang boros biaya.
3. **Handling "Batch Poisoning" (Bisect Retry):**
   * **Skenario:** 1 data rusak menyebabkan 1 batch (isi 50) gagal.
   * **Solusi:** Strategi Belah Dua (*Bisect*). Jika batch gagal, pecah jadi 2 sub-batch (25+25) dan coba lagi secara rekursif hingga data "beracun" terisolasi dan dibuang.

### 9.3. Arsitektur Asynchronous Worker

*Tujuan: Memisahkan jalur prioritas agar Free User tidak memacetkan Pro User.*

1. **Worker 1 (VIP - Pro Plan):** --queue=high --timeout=60. Jalur eksklusif, selalu kosong dan cepat.
2. **Worker 2 (Ekonomi - Free Plan):** --queue=low --timeout=120. Memproses data secara *Batch* besar saat server longgar.
3. **Worker 3 (Maintenance):** --queue=default. Khusus untuk tugas bersih-bersih (hapus file temp, email) agar tidak antri di belakang user.

### 9.4. Real-time & Moderasi (Async Processing)

*Tujuan: Mencegah Error 429 (Rate Limit) dan lag pada fitur chat.*

1. **Moderasi Post-Processing (Async):**
   * Chat muncul **instan** di layar (via Pusher) untuk pengalaman mulus.
   * Job AI berjalan di *background* (detik berikutnya). Jika melanggar, pesan ditarik (*Soft Delete*) dan user diberi peringatan.
2. **Free Plan Limitation:** Fitur Moderasi AI dimatikan. Menggunakan **Regex Filter (PHP)** lokal yang instan dan gratis.

### 9.5. Pembaruan Batasan Bisnis (AI Credit Limits V2)

Berdasarkan *Worst Case Scenario* (Overload) limit perlu diperketat agar sistem tidak *down* untuk semua orang.

**Tabel Limitasi Ketat (Survival Mode):**

| **Aturan** | **Free Plan (90 User)** | **Pro Plan (10 User)** | **Alasan Teknis** |
| --- | --- | --- | --- |
| **Data Collection** | Maks 3 Folder Aktif | Unlimited | Menghemat *row* database. |
| **Visualisasi Chart** | **Cached Only** (Update tiap 1 jam) | **Real-time** (Update instan) | Mencegah Free User spamming tombol refresh yang membebani query DB. |
| **Insight AI (Batch)** | Maks **3 Batch per Hari** | **50 Batch per Hari** | 3 Batch cukup untuk 1x analisis (misal 60 opini). Mencegah spam request AI. |
| **Report Composer** | **1x per Hari** | **Unlimited** | Generasi PDF memakan CPU server tinggi. Free user tidak butuh laporan harian. |
| **Forum** | Regex Filter Chat | AI Moderated Chat | Menjaga kuota API Gemini agar tidak habis oleh chat sampah. |

# ADDENDUM II: TECHNICAL SURVIVAL ARCHITECTURE

**Status:** Wajib Implementasi. **Fokus:** Mitigasi risiko fatal akibat limitasi infrastruktur *Zero Capital* (Hosting/Database Gratisan) dan beban konkuren.

## 1. STRATEGI DATABASE & KONEKSI (ANTI-CRASH)

*Tujuan: Mencegah error Too Many Connections pada PostgreSQL Free Tier yang limitnya kecil (20-50 koneksi).*

1. **Connection Pooling (Wajib):**
   * Jangan konek langsung ke port 5432. Gunakan **PgBouncer** (biasanya port 6543 di Neon/Supabase) untuk mem multiplexing ratusan request HTTP menjadi sedikit koneksi fisik ke DB.
2. **Pemisahan Driver Cache & Session:**
   * **DILARANG** menyimpan *Session* dan *Cache* di Database.
   * Gunakan **File Driver** (local disk) atau **Cookie Session** (client-side) untuk mengurangi beban I/O database secara drastis pada setiap *refresh* halaman.

## 2. AGGREGATOR ENGINE ROBUSTNESS (ANTI-HANG)

*Tujuan: Mencegah tumpukan cron job macet dan memori server penuh.*

1. **Atomic Locks:** Tambahkan ->withoutOverlapping() pada setiap command Scheduler untuk mencegah job baru berjalan jika job lama belum selesai (mencegah tumpukan ganda).
2. **Chunking Statis:** Worker wajib menggunakan LIMIT 20 per eksekusi. Proses 20 item, lalu mati. Biarkan Scheduler menit berikutnya mengambil 20 lagi. Ini menjaga penggunaan RAM tetap rendah dan stabil.
3. **Timeout Trap:** Set timeout HTTP request ke Gemini API maksimal **30 detik**. Jika macet, langsung *Fail* agar worker tidak *hang* selamanya.

## 3. FRONTEND & MOBILE OPTIMIZATION

*Tujuan: UX lancar di perangkat low-end tanpa membebani server.*

1. **Simplified Mobile View:** Deteksi *User Agent*. Jika Mobile, **JANGAN** render Chart.js yang berat. Tampilkan **List/Card View** (Angka Ringkasan) saja.
2. **Lazy Loading:** Tombol "Lihat Grafik" manual. Jangan load JS chart berat saat halaman pertama dibuka.
3. **Fake 3D UI:** Hindari library 3D (Three.js) yang memakan RAM browser. Gunakan **Gambar PNG/WebP** statis dan efek CSS (*Shadow/Blur*) untuk meniru kedalaman 3D.

## 4. PENYIMPANAN & FILE HANDLING (ZERO STORAGE)

*Tujuan: Mengatasi disk space hosting gratis yang sangat terbatas.*

1. **Streamed PDF Response:**
   * Ubah logika export: **JANGAN PERNAH SIMPAN PDF DI DISK/DB**.
   * Generate di RAM -> Stream ke Browser User -> Hapus dari RAM.
   * Beban Storage = 0 Byte.
2. **CSS Print Fallback:**
   * Jika library PDF server-side (*dompdf*) gagal karena RAM kecil (OOM Killer), sediakan tampilan **CSS @media print**. User cukup tekan Ctrl+P -> Save as PDF di browser mereka. Ini solusi paling hemat resource.

## 5. AUTHENTICATION & SECURITY (ANTI-SPAM)

*Tujuan: Keamanan maksimal dengan biaya operasional nol.*

1. **Google OAuth Only:** Hapus fitur Register/Login via Email & Password untuk menghemat kuota SMTP (verifikasi email) dan mencegah akun bot spam.
2. **Guest Mode with Fingerprint:** Izinkan pengisian form tanpa login, tapi catat **Cookie Abadi** + Hash (IP + User Agent) untuk mencegah spamming sederhana tanpa memberatkan user.

## 6. REAL-TIME LIMITATION (WEBSOCKET)

*Tujuan: Menghemat koneksi concurrent Pusher/Reverb gratisan.*

1. **Hybrid Approach:** Gunakan Websocket **HANYA** untuk notifikasi krusial (Chat baru).
2. **Livewire Polling:** Untuk update angka statistik/chart, gunakan wire:poll.10s (Polling tiap 10 detik). Ini jauh lebih hemat koneksi daripada membuka socket terus-menerus.

## 7. ARSITEKTUR WORKER (SINGLE PROCESS)

*Tujuan: Menghindari CPU throttling pada hosting gratis (1 vCPU).*

1. **Single Process Worker:** Jalankan **HANYA 1** proses queue:work. Jangan jalankan banyak worker paralel karena akan berebut CPU dan mematikan server. Biarkan antrian memanjang (delay) untuk Free User, ini adalah *trade-off* yang wajar.