

Herta's Vibe Checker

A Business Capability Model Document

Rhio Bimo | 13523123

THE INGENIOUS SOCIETY | Blue Planet, Planet Herta

DAFTAR ISI

I.	MODEL BISNIS	2
II.	LAYANAN UTAMA	3
III.	STARTEGI BISNIS	4
IV.	BUSINESS CAPABILITY MODEL	6
V.	PERSYARATAN FUNGSIONAL	10
VI.	USE CASE	12
VII.	SEQUENCE DIAGARAM	14
VIII.	ENTITY RELATIONSHIP DIAGRAM	18
IX.	APPLICATION ARCHITECTURE DIAGRAM	19

I. Model Bisnis

Herta's Vibe Checker adalah layanan berbasis teknologi Natural Language Processing (NLP) yang bertujuan untuk menganalisis “vibe” atau emosi dari teks pada platform komunitas online (Twitter, YouTube, Twitch, etc.). Layanan ini membantu streamer, moderator komunitas, maupun pengguna media sosial untuk mengecek sentimen dalam percakapan atau komentar secara otomatis. Nilai utama yang ditawarkan adalah kemampuan untuk mendeteksi secara cepat apakah sebuah chat atau komentar bernada positif, netral, atau negatif (bahkan toxic). Hal ini memungkinkan pelaku konten digital (misalnya streamer Twitch/ YouTube, admin komunitas Instagram/Twitter, dll.) mengambil tindakan proaktif seperti memoderasi komentar, memahami reaksi audiens, atau meningkatkan keterlibatan komunitas.

Dari sisi model bisnis, Herta's Vibe Checker dapat disediakan sebagai API berlangganan (SaaS) maupun sebagai fitur terintegrasi dalam platform komunitas. Target pelanggan utamanya adalah pengembang aplikasi (misalnya pembuat bot atau moderator tools), platform streaming/chat, serta perusahaan yang ingin memantau sentimen brand di media sosial. Monetisasi layanan bisa melalui skema berlangganan bulanan untuk akses API (dengan tier berdasarkan jumlah analisis dan tingkat scan vibe per bulan), ataupun integrasi freemium di mana pengguna gratis mendapat jatah analisis terbatas dan pengguna berbayar mendapat kuota lebih besar serta fitur lanjutan (seperti laporan analitik mendalam).

II. Layanan Utama

Berikut adalah layanan utama dari Madam Herta's Vibe Checker:

- Analisis Vibe Teks Tunggal:

Memeriksa sentimen atau “vibe” dari satu potongan teks (misalnya satu komentar). Layanan ini memberikan hasil berupa label sentimen (positif/negatif/ netral) beserta skor kepercayaannya. Fitur ini cocok untuk memeriksa tweet individu, komentar tunggal, atau pesan privat.

- Analisis Vibe Teks Banyak (Batch):

Memeriksa vibe dari sekumpulan teks sekaligus, misalnya beberapa baris chat Twitch atau kumpulan komentar YouTube. Layanan ini memproses setiap teks dalam batch dan mengembalikan daftar hasil analisis. Fitur ini bermanfaat untuk moderator yang ingin menilai atmosfer keseluruhan dalam satu sesi chat atau kolom komentar secara cepat.

- Riwayat dan Laporan:

Penyimpanan hasil analisis vibe sebelumnya, sehingga pengguna (misalnya pengembang bot) dapat mengambil riwayat vibe dari teks-teks yang telah dianalisis. Dengan riwayat ini, bisa disusun laporan seperti tren sentimen seiring waktu atau statistik berapa banyak komentar positif vs negatif yang terdeteksi.

- Integrasi Multi-Platform:

Layanan disiapkan untuk mudah diintegrasikan dengan berbagai platform populer. Contohnya, integrasi dengan API Twitter untuk menganalisis vibe tweets trending, atau integrasi dengan Twitch chat melalui bot. Dengan demikian, Herta's Vibe Checker dapat menjadi komponen pendukung di ekosistem platform lain (seperti plugin chatbot atau moderation tool).

- Layanan Kesehatan Sistem:

Termasuk endpoint health-check untuk memantau status sistem (apakah layanan berjalan), serta endpoint statistik layanan untuk melihat metrik kinerja (misal total request terproses, uptime, dsb). Layanan pendukung ini memastikan baik pengguna maupun tim internal dapat memonitor kesehatan dan performa sistem dengan mudah.

III. Startegi Bisnis

Strategi bisnis Herta's Vibe Checker berfokus pada penyediaan teknologi AI yang mudah diadopsi oleh komunitas online. Let me cook on this:

- Fokus Komunitas & Kemitraan:

Menggarap komunitas streamer dan media sosial dengan menawarkan solusi moderasi dan insight. Misalnya, bekerja sama dengan streamer Twitch populer atau komunitas gaming untuk menguji coba layanan sebagai bagian dari moderasi chat mereka. Dengan membangun studi kasus keberhasilan di komunitas tersebut, layanan ini mendapat eksposur dan kepercayaan lebih luas.

- API-First Approach:

Menyediakan layanan dalam bentuk API yang aman dan terdokumentasi baik, sehingga para developer dan platform bisa dengan cepat mengintegrasikannya. Dokumentasi lengkap (mirip standar bank BCA API) dan sandbox disiapkan agar pengembang dapat mencoba layanan dengan mudah. Pendekatan API-first ini membuka peluang adopsi luas tanpa perlu mengembangkan UI sendiri yang kompleks.

- Teknologi AI/NLP Nous:

Memastikan algoritma analisis vibe menggunakan model AI terkini agar hasilnya akurat dan kontekstual. Strategi ini termasuk investasi pada riset NLP (misalnya memanfaatkan model pre-trained terkini dan melatihnya pada dataset percakapan internet), sehingga Herta's Vibe Checker unggul dalam mendeteksi konteks sarkasme, ironi, atau slang yang sering ada di komunitas online.

- Skalabilitas & Keandalan:

Menerapkan infrastruktur cloud yang mampu skala otomatis untuk menangani lonjakan trafik (misalnya saat event streaming besar). Layanan juga dibangun dengan resiliensi tinggi (multi-region deployment, load balancing, dsb.) agar uptime terjaga. Strategi ini penting karena target pengguna (platform komunitas) membutuhkan layanan real-time yang handal.

- Keamanan & Privasi:

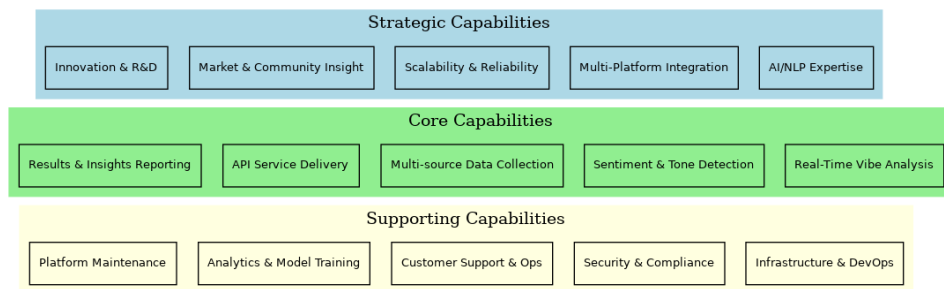
Mengadopsi standar keamanan tinggi seperti OAuth2 untuk autentikasi API serta enkripsi data. Kepercayaan komunitas digital sangat bergantung pada keamanan, apalagi layanan ini mungkin memproses data percakapan yang sensitif. Dengan memastikan hanya pengguna

terotorisasi yang dapat mengakses API dan data disimpan secara aman, Herta's Vibe Checker membangun reputasi sebagai layanan tepercaya.



IV. Business Capability Model

Business Capability Model menguraikan kapabilitas utama bisnis Herta's Vibe Chekcer dalam tiga kategori: Strategic, Core, dan Supporting Capabilities. Berikut diagram BCM yang menunjukkan kapabilitas-kapabilitas tersebut:



Gambar 1. Business Capability Model untuk Herta's Vibe Checker.

Lapisan "*Strategic*" berisi kemampuan strategis yang menjadi keunggulan kompetitif (misalnya integrasi *multi-platform*, pemanfaatan AI/NLP, dsb.). Lapisan "*Core*" mencakup kapabilitas inti yang langsung memberikan nilai ke pelanggan (misalnya analisis vibe *real-time*, penyediaan API, pengumpulan data multi-sumber). Lapisan "*Supporting*" mencakup kapabilitas penunjang internal (seperti infrastruktur TI, keamanan, dukungan pelanggan, dan *maintenance platform*).

Pada lapisan Strategic Capabilities, kapabilitas yang dimiliki meliputi:

- AI/NLP Expertise:

Kemampuan strategis perusahaan dalam membangun dan memanfaatkan teknologi AI untuk NLP. Ini memastikan layanan vibe check akurat dan adaptif terhadap bahasa gaul atau konteks percakapan. (Teknologi pendukung: model AI terkini, library NLP seperti Transformers, dll.)

- Multi-Platform Integration:

Kemampuan integrasi dengan berbagai platform eksternal (Twitter, Twitch, YouTube, dsb.). Ini memungkinkan jangkauan layanan luas dan menambah proposisi nilai melalui kemitraan. (Teknologi pendukung: API client untuk platform-platform tersebut, webhooks, SDK integrasi).

- Scalability & Reliability:

Kemampuan untuk menskalakan layanan dan menjaga keandalannya di segala kondisi. Secara strategis, ini menjaga reputasi layanan di mata pelanggan enterprise. (Teknologi:

arsitektur microservices, container orchestration/ Kubernetes, auto-scaling cloud services, monitoring tools).

- **Market & Community Insight:**

Kemampuan memahami tren pasar dan kebutuhan komunitas pengguna. Dengan insight ini, produk dapat berevolusi sesuai yang dibutuhkan streamer/ komunitas. (Pendukung: analisis data penggunaan, feedback user, riset pasar).

- **Innovation:**

Kemampuan terus berinovasi, misalnya mengembangkan fitur vibe analysis lebih mendalam (deteksi emosi spesifik) atau AI model yang lebih canggih. Strategi jangka panjang terwujud melalui R&D berkelanjutan. (Pendukung: tim R&D AI, kemitraan penelitian, dll.)

Pada lapisan Core Capabilities, kapabilitas inti operasional yang langsung menghasilkan layanan ke pengguna antara lain:

- **Real-Time Vibe Checking:**

Kapabilitas menganalisis teks secara real-time untuk menentukan tonenya. Inilah jantung layanan – engine NLP yang melakukan sentiment analysis terhadap input teks secara cepat dan akurat. (Teknologi: pipeline NLP real-time, kemungkinan menggunakan model pre-trained seperti BERT atau model khusus yang dilatih untuk percakapan.)

- **Tone Detection:**

Kemampuan mendeteksi kategori sentimen (positif/negatif/netral) serta nada/emosi (marah, senang, sarkastik, dsb. jika diterapkan). Ini bagian dari logika AI untuk memahami “vibe” secara lebih halus daripada sekadar polaritas sentimen. (Teknologi: algoritma klasifikasi teks multi-label, kamus kata emotif, model deep learning).

- **Multi-source Data Collection:**

Kapabilitas mengumpulkan data teks dari berbagai sumber (misal mengumpulkan chat Twitch secara live, mengambil tweet terbaru via API, dsb.). Ini penting agar layanan dapat di-feed data dengan mudah dari sumber manapun. (Teknologi: modul konektor API untuk tiap platform, stream listeners, job scheduler untuk pooling data).

- **API Service Delivery:**

Kemampuan expose hasil analisis melalui API RESTful yang aman dan cepat. Core capability ini menyangkut desain, implementasi, dan pengoperasian endpoint API agar bisa

dipanggil klien kapanpun. (Teknologi: framework web (Python Flask/FastAPI, Node Express, Go Fiber, dll.), sistem autentikasi OAuth2, gateway API.)

- **Insights Reporting:**

Kapabilitas menyajikan hasil dan insight, misalnya melalui dashboard atau laporan. Ini termasuk penyimpanan hasil analisis dan penyediaan ringkasan (berapa % positif vs negatif, dsb.) kepada pengguna. (Teknologi: database untuk menyimpan hasil, modul analitik untuk agregasi data, kemungkinan UI dashboard front-end).

Terakhir, ada lapisan Supporting Capabilities yang mencakup hal-hal pendukung internal yang memastikan core capabilities berjalan optimal:

- **Infrastructure & DevOps:**

Kemampuan menyiapkan infrastruktur TI, CI/CD, deployment container, dan pemantauan sistem. (Teknologi: layanan cloud (AWS/GCP/Azure), Docker/ Kubernetes, pipeline CI, monitoring (Prometheus/Grafana), dsb.)

- **Security & Compliance:**

Kapabilitas memastikan keamanan data dan kepatuhan (misal terhadap aturan platform atau privasi pengguna). Ini termasuk autentikasi OAuth2 yang kuat, enkripsi data, proteksi HMAC signature seperti standar BCA API, dsb. (Teknologi: OAuth2 server, modul enkripsi, audit logging, firewall).

- **Customer Support & Ops:**

Kemampuan memberikan dukungan ke pelanggan/pengguna (support center, technical account manager) dan operasi bisnis sehari-hari (billing, account management). (Teknologi: ticketing system, CRM, portal dev support).

- **Analytics & Model Training:**

Kapabilitas melakukan analytics internal dan melatih ulang model AI secara periodik. Mendukung core NLP dengan memperbaiki model dari waktu ke waktu berdasarkan data baru. (Teknologi: data pipeline, Jupyter notebooks, ML ops, GPU training instances).

- **Platform Maintenance:**

Kapabilitas melakukan maintenance platform secara berkala, termasuk upgrade sistem, perbaikan bug, scaling database, dll. (Teknologi: management tools, version control, testing frameworks untuk memastikan perubahan tidak merusak layanan).

Dengan kapabilitas-kapabilitas di atas, perusahaan dapat menjalankan proses bisnis Herta's Vibe Checker secara efektif. Tiap kapabilitas ditopang oleh teknologi spesifik. Misalnya AI/NLP memerlukan framework pembelajaran mesin canggih, integrasi multi-platform perlu modul konektor/SDK, dan keamanan mengandalkan protokol OAuth2 serta enkripsi. Kombinasi semua kapabilitas ini menjadikan bisnis mampu memberikan layanan analisis vibe yang unik sekaligus beroperasi secara andal di ekosistem digital yang luas.



V. Persyaratan Fungsional

Berdasarkan model bisnis dan layanan di atas, berikut adalah persyaratan fungsional (*Functional Requirements*) dari sistem Vibe Checker ini:

- Analisis Vibe Otomatis:

Ini muncul terus yak dari awal :v. Sistem harus dapat menerima input teks dan menghasilkan output kategori vibe (mis. “positif”, “negatif”, atau “netral”) beserta metrik kepercayaan. Analisis ini dilakukan menggunakan metode NLP secara otomatis setiap kali ada permintaan.

- Pemrosesan Batch:

Sistem harus mendukung pemrosesan beberapa teks dalam satu permintaan (batch processing). Ini berarti API dapat menerima sekumpulan teks dan mengembalikan analisis untuk masing-masing teks dalam sekali panggil, dengan performa yang tetap optimal.

- RESTful API & Response Format:

Sistem harus menyediakan antarmuka RESTful API dengan endpoint-endpoint sesuai use case (cek vibe tunggal, cek vibe batch, lihat riwayat, dll.). Setiap respons harus dalam format terstruktur (misalnya JSON) yang mencakup data yang diminta atau pesan error yang jelas jika gagal.

- Penyimpanan Hasil & Riwayat:

Sistem harus menyimpan hasil analisis yang telah dilakukan (setidaknya untuk jangka waktu tertentu atau sejumlah entri terakhir) agar dapat diambil kembali melalui endpoint “previous vibes”. Data yang disimpan minimal mencakup teks yang dianalisis, hasil vibe, skor, timestamp, dan identitas klien yang meminta (untuk isolasi data per pengguna API).

- Keamanan & OAuth2:

Semua endpoint (kecuali mungkin health-check publik) harus dilindungi oleh mekanisme autentikasi OAuth2.0. Sebelum menggunakan layanan, pengguna (client application) harus memperoleh akses token OAuth2 yang valid. Sistem harus memverifikasi token pada setiap permintaan dan menolak akses jika token invalid atau tidak disertakan. Selain itu, pertukaran data sensitif (client secret, token) harus dienkripsi (HTTPS).

- Health Monitoring:

Sistem harus menyediakan mekanisme untuk memeriksa kesehatan (health) layanan, misalnya endpoint GET /health yang mengembalikan status ("OK") dan mungkin informasi versi. Ini memudahkan integrator atau sistem monitoring otomatis untuk memastikan API up.

- Statistik Layanan:

Sistem harusnya menyediakan endpoint untuk mengambil beberapa statistik penggunaan, seperti total jumlah teks yang sudah dianalisis, distribusi sentimen yang terdeteksi, atau statistik kinerja (uptime, response time rata-rata). Ini membantu pengguna memahami performa layanan dan membantu internal tim dalam transparansi layanan.

- Waktu Respons Real-Time:

Untuk penggunaan di konteks live (misal chat streaming), sistem harus mampu memberikan respons cepat (real-time). Targetnya, analisis satu teks biasanya diselesaikan dalam waktu < 1 detik. Untuk batch, waktu respons skalabel tergantung jumlah teks, namun tetap dioptimalkan agar tidak menghambat alur aplikasi pengguna.

- Skalabilitas & RoBUSTness:

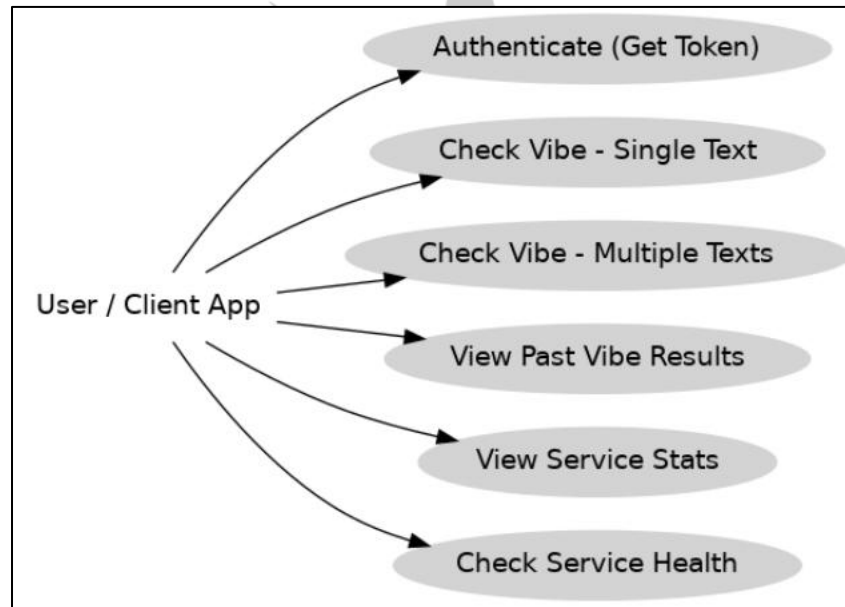
Sistem harus mampu menangani beban tinggi, seperti ratusan request per detik di saat puncak, tanpa penurunan kinerja signifikan. Persyaratan ini berarti rancangan sistem harus stateless pada layer aplikasi (untuk mudah di-scale horizontal), serta pemulihan otomatis jika terjadi kegagalan komponen (failover).

- Dukungan Multi-Bahasa (Yes, I wanna kill myself):

Sebagai pengembangan lanjutan, sistem sebaiknya dapat mendukung analisis vibe dalam berbagai bahasa (tidak hanya Bahasa Inggris). Hal ini penting bila komunitas pengguna global. Model NLP harus mampu mendeteksi bahasa atau ada parameter bahasa di API, lalu menggunakan model/strategi analisis sesuai bahasa tersebut.

VI. Use Case

Berikut adalah identifikasi aktor dan use case utama dalam sistem Herta's Vibe Checker. Aktor di sini adalah User / Client Application, yaitu aplikasi atau pengguna (misalnya developer) yang mengonsumsi API ini. Use case utamanya meliputi: Autentikasi (mendapatkan token), Cek Vibe (Teks Tunggal), Cek Vibe (Teks Jamak), Melihat Riwayat Hasil Vibe, Melihat Statistik Layanan, dan Memeriksa Kesehatan Layanan. Hubungan aktor dan use case ditunjukkan pada diagram berikut:



Gambar 2. Diagram Use Case untuk Herta's Vibe Checker.

Satu aktor eksternal ("User / Client App") melakukan berbagai use case melalui pemanggilan API: memperoleh token autentikasi, mengirim teks tunggal atau banyak untuk dianalisis vibenya, meminta data riwayat hasil analisis sebelumnya, mengambil statistik layanan, serta melakukan health-check. Semuanya difasilitasi oleh sistem Herta's Vibe Checker melalui endpoint-endpoint yang sesuai.

Deskripsi Singkat:

- Authenticate (Get Token):

Aktor mendapatkan token akses OAuth2. Ini dilakukan dengan mengirim kredensial (client_id & secret) ke layanan autentikasi API. Sistem memverifikasi kredensial dan mengembalikan access token (Bearer token) yang akan digunakan untuk pemanggilan API selanjutnya. (Termasuk mekanisme tantangan matematika unik yang dijelaskan di bagian flow).

- Check Vibe (Single):

Aktor mengirim satu teks (misal satu komentar atau kalimat) ke API untuk dianalisis. Sistem memproses teks tersebut melalui engine NLP dan mengembalikan hasil vibe (label + skor).

- Check Vibes (Multiple):

Aktor mengirimkan sejumlah teks sekaligus (misal array of strings) untuk dianalisis dalam satu permintaan. Sistem akan memproses setiap teks (secara paralel atau berurutan secara efisien) dan mengembalikan daftar hasil vibe yang masing-masing berkorespondensi dengan teks input.

- View Traces (Past Results):

Aktor mengirimkan sejumlah teks sekaligus (misal array of strings) untuk dianalisis dalam satu permintaan. Sistem akan memproses setiap teks (secara paralel atau berurutan secara efisien) dan mengembalikan daftar hasil vibe yang masing-masing berkorespondensi dengan teks input.

- View Light Cones (Service Stats):

Aktor meminta data statistik layanan (mungkin terbatas untuk aktor jika terkait kuota penggunaan, atau statistik global non-sensitif). Sistem mengembalikan info seperti total analisis yang telah dijalankan, persentase positif/negatif terdeteksi, uptime, dsb. Ini memberikan transparansi dan informasi tambahan kepada pengguna API.

- Check Service Health:

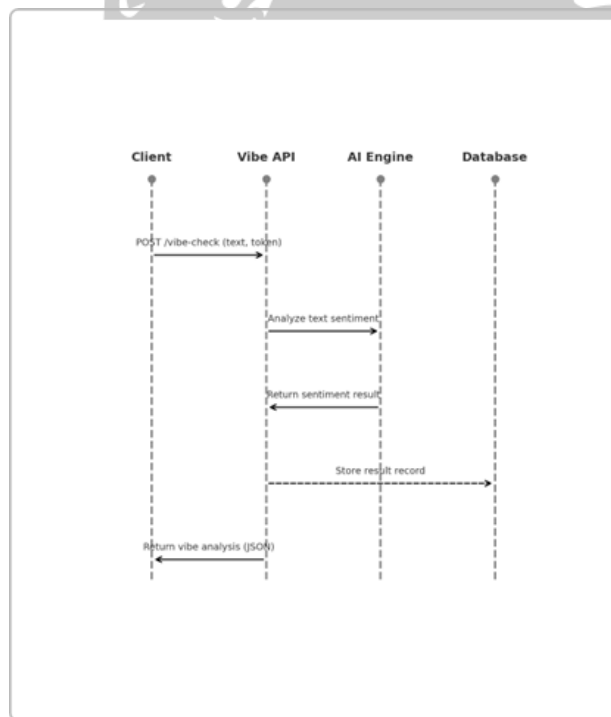
Aktor (bisa berupa sistem monitoring ataupun developer) memanggil endpoint health-check. Sistem merespons singkat (misal "OK" atau detail versi) menandakan layanan berjalan normal. Use case ini penting untuk memastikan integrasi tetap berfungsi dan mendeteksi dini jika layanan down.

VII. Sequence Diagram

Untuk menggambarkan alur interaksi antar komponen dalam sistem, berikut disajikan beberapa Sequence Diagram untuk use case utama. Sequence diagram menunjukkan bagaimana aktor (client) berinteraksi dengan komponen internal sistem (API, engine AI, database, dsb.) sekuensial dari waktu ke waktu.

7.1. Check Vibe (Single)

Use case ini dimulai ketika client (misal aplikasi pengguna) mengirim permintaan analisis vibe untuk satu teks. Terlihat bahwa client harus menyertakan token OAuth2 dalam permintaan (setelah sebelumnya melalui proses autentikasi). Berikut alur detailnya: client memanggil endpoint API /vibe-check dengan teks di body. API Server memvalidasi token lalu meneruskan teks tersebut ke modul AI Vibe Analysis Engine untuk diproses; engine menganalisis sentimen teks dan mengembalikan hasil (misalnya “positif” skor 0.82); API Server menyimpan hasil tersebut ke database (untuk keperluan historis/statistik) lalu mengirimkan respons JSON kembali ke client berisi hasil analisis.

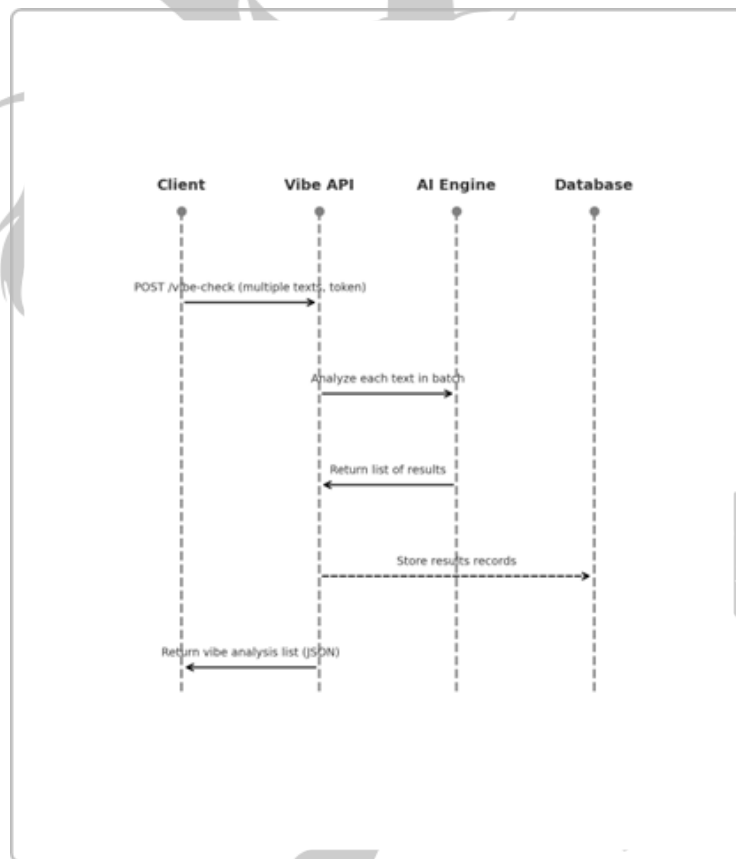


Gambar 3. Sequence diagram untuk use case “Check Vibe”

Terlihat interaksi: Client mengirim request dengan teks (dan token autentikasi) ke Vibe API, yang lalu memanggil AI Engine untuk menganalisis teks. Hasil analisis dikembalikan ke API, disimpan ke database, lalu hasil berupa vibe dikirim kembali ke Client.

7.2. Check Vibes (Multiple)

Use case ini mirip dengan single text, tetapi client mengirim beberapa teks sekaligus dalam satu permintaan (misal array JSON berisi banyak kalimat/chat). Diagram di bawah memperlihatkan: client memanggil endpoint (misal /vibe-check/batch) dengan kumpulan teks; API Server memforward data ini ke AI Engine; engine melakukan analisis sentimen pada masing-masing teks (secara batch); engine mengembalikan daftar hasil ke API; API menyimpan masing-masing hasil ke database (dapat dilakukan per item atau bulk insert); kemudian API merespons client dengan list hasil vibe (format JSON berisi array, di mana urutan sesuai urutan input).



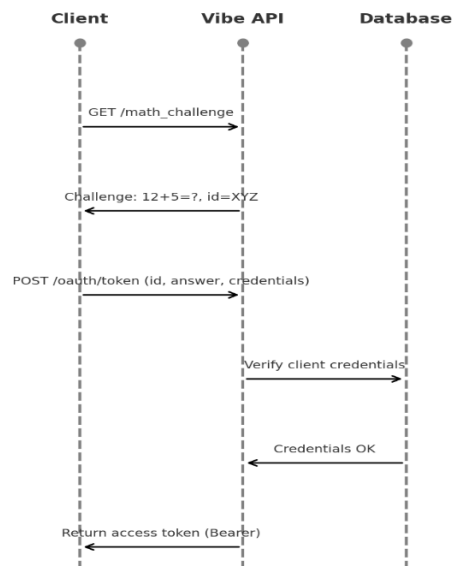
Gambar 4. Sequence diagram untuk "Check Vibes"

Alurnya serupa dengan single text, tetapi AI Engine memproses batch teks dan mengembalikan kumpulan hasil. API Server menyimpan sekumpulan hasil tersebut dan mengirim kembali ke Client daftar hasil analisis vibe.

7.3. Autentikasi (OAuth2 dengan Math Challenge {sponsored by sister})

Sebelum bisa menggunakan endpoint analisis, client perlu melakukan autentikasi untuk mendapatkan access token. Proses autentikasi di sistem ini memiliki dua langkah utama: pertama, client melakukan GET ke `/math_challenge` untuk mendapatkan soal matematika sederhana (misal $69+420=$) beserta `challenge_id` unik; kedua, client mengirim POST ke `/oauth/token` dengan menyertakan jawaban soal tersebut serta `client_id` dan `client_secret` miliknya. Tujuan dari tantangan matematika ini adalah menambah lapisan keamanan (memastikan peminta token adalah manusia atau aplikasi terotorisasi yang minimal bisa memecahkan tantangan sederhana, dan mencegah flood request token). Sequence diagram berikut menggambarkan proses ini:

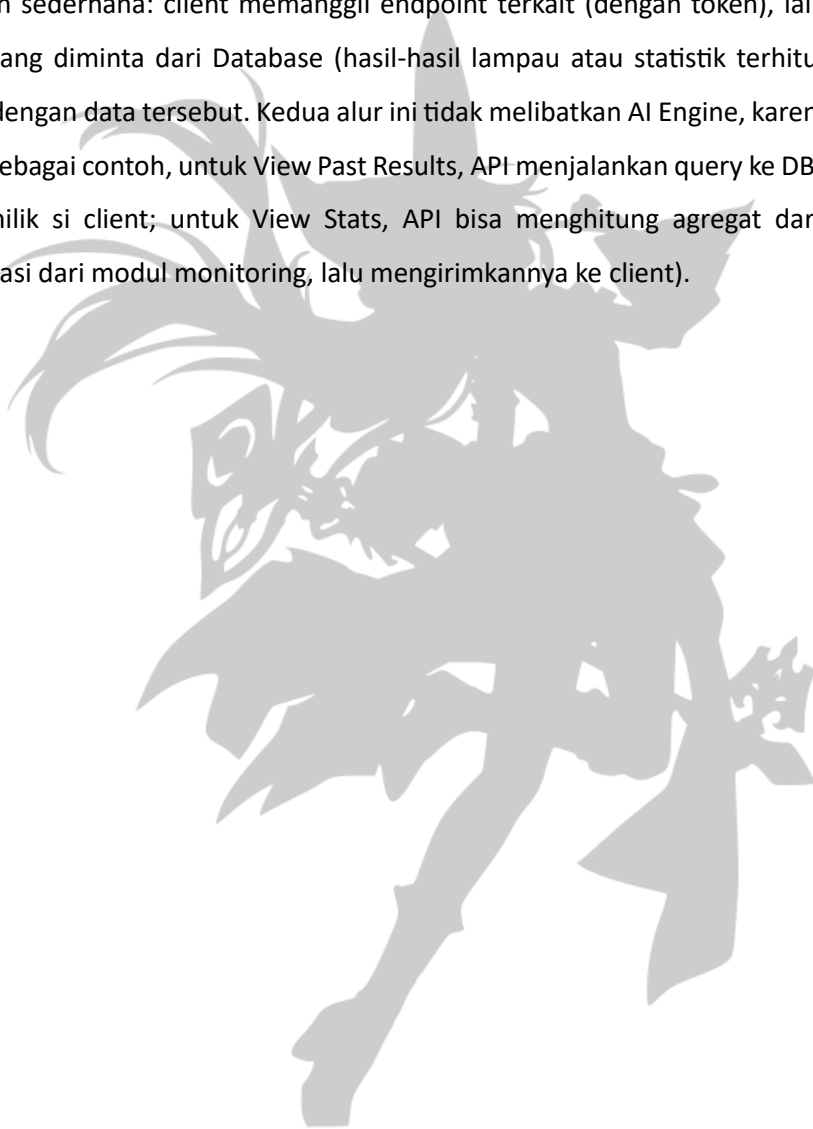
Client meminta challenge ke API -> API Server menghasilkan soal (misal " $69+80016=?$ ", `id=XYZ`) dan menyimpannya sementara (atau meng-encrypt-nya dalam token satu kali pakai) -> API mengirim challenge ke client. Lalu client mengirim request token dengan jawaban challenge dan kredensialnya -> API Server memverifikasi `client_id` dan `client_secret` ke Database (disimpan saat registrasi aplikasi) -> jika valid, API memverifikasi jawaban challenge (80085) dan id challengenya -> jika benar, API mengeluarkan access token dan mengembalikannya ke client.



Gambar 5. Sequence diagram untuk proses autentikasi OAuth2 dengan math challenge.

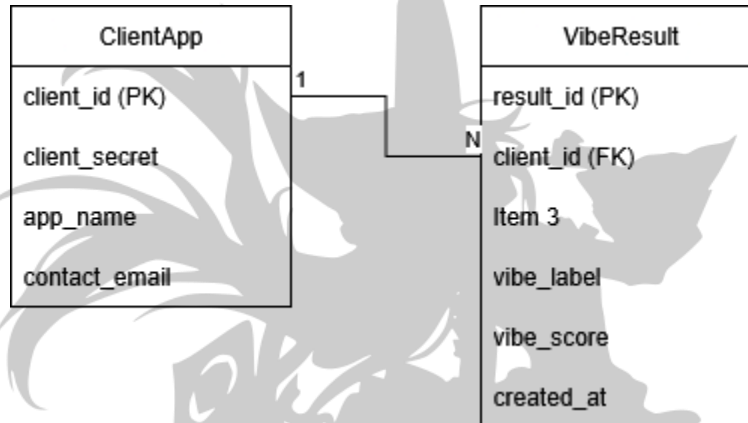
Terlihat bahwa selain verifikasi kredensial ke Database, API menambahkan langkah challenge Q&A. Client mendapatkan challenge “12+5=?” dari API, kemudian mengirim jawaban (17) bersama client_id/secret untuk mendapatkan token. API memvalidasi kredensial (dengan DB) dan jawaban challenge sebelum akhirnya merespons dengan access token (Bearer).

Catatan: Sequence diagram untuk use case “View Past Vibe Results” dan “View Service Stats” secara umum lebih sederhana: client memanggil endpoint terkait (dengan token), lalu API Server akan mengambil data yang diminta dari Database (hasil-hasil lampau atau statistik terhitung) dan langsung merespons client dengan data tersebut. Kedua alur ini tidak melibatkan AI Engine, karena hanya read data yang sudah ada. (Sebagai contoh, untuk View Past Results, API menjalankan query ke DB untuk mengambil N hasil terbaru milik si client; untuk View Stats, API bisa menghitung agregat dari table hasil atau mengambil informasi dari modul monitoring, lalu mengirimkannya ke client).



VIII. Entity Relationship Diagram

Berikut adalah diagram Entity-Relationship yang menggambarkan model data sederhana untuk sistem Herta's Vibe Checker. Terdapat dua entitas utama: ClientApp (aplikasi/pengguna yang terdaftar menggunakan API, pemegang client_id dan client_secret) dan VibeResult (hasil analisis setiap teks yang diproses). Relasi antara keduanya adalah one-to-many. Satu ClientApp dapat memiliki banyak VibeResult (well.. duh...).



Gambar 6. ERD sederhana Herta's Vibe Checker.

Terlihat bahwa clientApp memiliki atribut kunci primer (Primary Key, PK) client_id, berelasi 1-to-N dengan table VibeResult yang menyimpan hasil analisis. Setiap record VibeResult dikaitkan dengan client yang melakukan request tersebut.

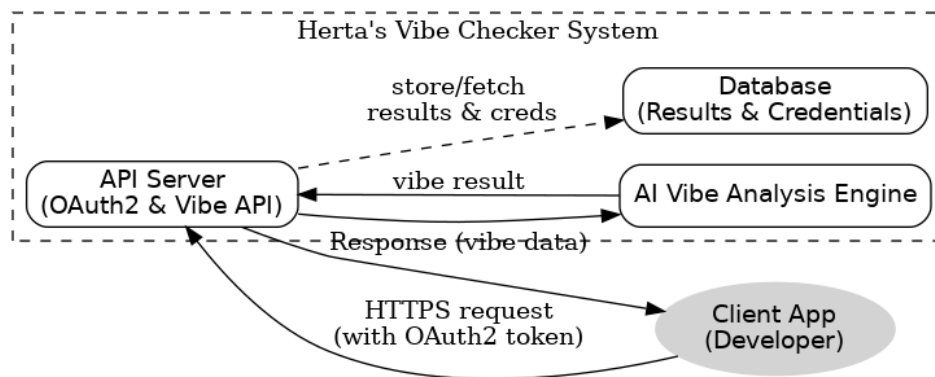
8.1. Penjelasan data:

Setiap kali ada analisis vibe baru, sebuah record di tabel VibeResult akan dibuat. Kolom client_id (FK) mengaitkan siapa pengguna/API client yang meminta analisis tersebut. Ini menjamin isolasi data sehingga satu client hanya dapat menarik riwayat miliknya sendiri. Atribut text menyimpan teks input (bisa berupa hash/dokumen referensi, dsb), vibe_label menyimpan kategori vibes (i.e. vibing, not-vibing, crashing-out), vibe score menyimpan skor confidence atau intensitas (i.e. 0.82 for absolute vibes), dan created_at mencatat waktu analisis. Pada tabel ClientApp, client_secret disimpan (terenkripsi/hashed), beserta app_name (nama aplikasi client) dan contact_email untuk admin. Tabel ini digunakan saat proses OAuth2 untuk memvalidasi kredensial client.

(Catatan: Model data di atas dapat diperluas sesuai kebutuhan. Misalnya, jika ingin mendukung multi-user atau end-user tertentu, bisa ditambah tabel User. Namun untuk lingkup dasar, dua entitas tersebut sudah mencukupi. Becoz I ain't got too much time.)

IX. Application Architecture Diagram

Di bawah ini disajikan diagram arsitektur aplikasi Herta's Vibe Checker secara keseluruhan. Komponen- komponen utama meliputi Client Application (pihak pengguna API), API Server (yang meng-handle request, termasuk komponen OAuth2 dan endpoint vibe), AI Vibe Analysis Engine (komponen atau layanan internal yang menjalankan algoritma NLP untuk analisis sentimen), serta Database (penyimpanan data hasil dan kredensial). Alur dasar: client mengirim request HTTPS dengan menyertakan token; API Server memverifikasi token (menggunakan komponen OAuth2 internal atau service terpisah), lalu memproses request (misal meneruskan teks ke AI engine, atau mengambil data dari DB); setelah itu API Server mengembalikan response ke client.



Gambar 7. Diagram arsitektur aplikasi Herta's Vibe Checker.

Di sisi kiri, Client App (Developer) mengirim request HTTPS dengan OAuth2 token ke API Server. API Server (yang mencakup modul autentikasi OAuth2 dan layanan API vibe) berada di tengah sebagai pintu masuk utama ke sistem. Untuk permintaan analisis, API Server berkomunikasi dengan AI Vibe Analysis Engine yang menjalankan pemrosesan NLP dan mengembalikan hasil vibe. API Server juga berinteraksi dengan Database (misalnya untuk menyimpan kredensial client dan hasil analisis, serta mengambil data riwayat atau stats). Response akhir berupa data vibe atau informasi lain dikirim kembali ke Client. Semua komunikasi berlangsung melalui protokol HTTPS yang aman.

Arsitektur di atas dapat di-deploy di lingkungan cloud. Setiap komponen dapat diskalakan secara terpisah: misalnya, API Server bisa terdiri dari beberapa instance belakang load balancer untuk menangani banyak request simultan. AI Engine bisa dijalankan sebagai microservice tersendiri (bahkan dimungkinkan menggunakan layanan serverless atau model inference server khusus untuk AI). Database bisa menggunakan cluster terdistribusi untuk keandalan. OAuth2 service bisa diimplementasi sebagai bagian dari API Server atau layanan terpisah (dalam diagram digambarkan terintegrasi dengan API Server).

Komunikasi antara komponen internal menggunakan protokol aman (misal gRPC internal atau REST internal) dan data sensitif (seperti `client_secret`) disimpan terenkripsi di database. Secara keseluruhan, desain ini memisahkan concerns: front-facing API handling, heavy AI processing, dan persistent storage, sehingga mempermudah maintenance dan scaling.



