# Dokumen Spesifikasi Teknis

## Platform Analisis Sentimen & Isu Publik Berbasis AI

Versi: 1.0
Tanggal: 30 Mei 2025
Penulis: Parametrix Indonesia (gunakan untuk copyright pada footer)

## 1. Pendahuluan

Dokumen ini merinci aspek teknis dari Platform Analisis Sentimen & Isu Publik Berbasis AI, termasuk arsitektur, teknologi yang direkomendasikan, komponen utama, dan pertimbangan keamanan.

## 2. Arsitektur Sistem

Sistem ini akan mengadopsi arsitektur berbasis *Microservices* atau *Event-Driven Architecture* untuk menangani volume data yang besar dan pemrosesan *real-time*.

- **Arsitektur Umum:**
  - **Data Ingestion Layer:** Mengumpulkan data dari berbagai sumber.
  - **Data Processing Layer:** Membersihkan, menormalisasi, dan menganalisis data (NLP, AI).
  - **Data Storage Layer:** Menyimpan data mentah dan hasil analisis.
  - **API Layer:** Menyediakan akses ke data analisis untuk *frontend*.
  - **Frontend Layer:** Dasbor interaktif untuk visualisasi dan konfigurasi.
  - **Notification Layer:** Mengirimkan *alert* dan laporan.

## 3. Teknologi yang Direkomendasikan

- **Frontend (Web):**
  - **Framework:** Next.Js.
  - **Styling:** Tailwind CSS.
  - **Data Visualization:** D3.js, Chart.js, atau Recharts untuk grafik interaktif.
- **Backend & Data Processing:**
  - **Bahasa Pemrograman:** Scrapy + BeautifulShop + Python (ekosistem kaya untuk AI/ML).
  - **Framework:** FastAPI (untuk API), Apache Spark/PySpark (untuk pemrosesan data besar).
  - **NLP/AI Libraries:** spaCy, Hugging Face Transformers (untuk model bahasa Indonesia), Scikit-learn (untuk klasifikasi).
  - **Message Queue:** Apache Kafka
- **Database:**
  - **NoSQL Database:** MongoDB (untuk data mentah yang fleksibel dan hasil analisis yang tidak terstruktur).
  - **Time-Series Database (Opsional):** InfluxDB atau TimescaleDB (untuk menyimpan data tren sentimen dari waktu ke waktu).
- **Cloud Platform:**
  - AWS, Google Cloud Platform (GCP), Azure, atau yang lain.
    - **Layanan yang Digunakan:**
      - **Compute:** EC2 (AWS), Compute Engine (GCP), Virtual Machines (Azure) untuk *backend* dan *processing*.
      - **Managed Kubernetes:** EKS (AWS), GKE (GCP), AKS (Azure) untuk *container orchestration* (sangat direkomendasikan untuk skalabilitas).
      - **Data Storage:** S3 (AWS), Cloud Storage (GCP), Azure Blob Storage (untuk data mentah dan *backup*).
      - **Database Services:** DocumentDB (AWS), Firestore (GCP), Cosmos DB

(Azure) untuk MongoDB.
        - **Message Queue Services:** Amazon Kinesis/SQS (AWS), Google Cloud Pub/Sub (GCP), Azure Service Bus/Event Hubs.
        - **AI/ML Services (Opsional, untuk *managed* NLP):** AWS Comprehend, Google Cloud Natural Language API, Azure Cognitive Services.
- **Version Control:** Git (GitHub/GitLab/Bitbucket).
- **CI/CD:** Jenkins, GitLab CI/CD, GitHub Actions.
- **Monitoring & Logging:** Prometheus & Grafana, ELK Stack, CloudWatch (AWS), Cloud Monitoring (GCP).

## 4. Komponen Utama Sistem

### 4.1. Modul Data Ingestion

- **Web Crawlers/Scrapers:** Dikembangkan menggunakan Python (misal: Scrapy, BeautifulSoup) untuk situs berita dan forum.
- **API Integrators:** Menggunakan API resmi dari platform media sosial (misal: Twitter/X API v2, Facebook Graph API) untuk mengumpulkan data yang diizinkan.
- **Keyword Management Service:** Mengelola *keyword* yang akan dipantau.

### 4.2. Modul Data Processing (AI/NLP Pipeline)

- **Pre-processing:** Tokenisasi, *stop word removal*, *stemming/lemmatization* (menggunakan NLTK/spaCy).
- **Sentiment Analysis Model:**
    - Model klasifikasi teks berbasis *Machine Learning* (misal: SVM, Naive Bayes) atau *Deep Learning* (misal: BERT, RoBERTa) yang telah dilatih pada dataset bahasa Indonesia.
    - Dapat menggunakan *pre-trained models* dari Hugging Face dan *fine-tuning* dengan data spesifik politik Indonesia.
- **Topic Modeling:** Menggunakan Latent Dirichlet Allocation (LDA) atau Non-negative Matrix Factorization (NMF) untuk mengidentifikasi topik.
- **Named Entity Recognition (NER):** Mengidentifikasi entitas seperti nama orang, organisasi, lokasi.
- **Influencer Identification:** Algoritma untuk menghitung metrik pengaruh (misal: jumlah *follower*, *engagement*, *retweet*).

### 4.3. Modul Data Storage

- **Raw Data Storage:** Menyimpan data mentah yang dikumpulkan (misal: JSON dari API, HTML dari *scraper*) di *object storage* (S3/Cloud Storage).
- **Processed Data Storage:** Menyimpan hasil analisis sentimen, topik, entitas, dan metrik di MongoDB.

### 4.4. Modul API (Backend)

- **RESTful API:** Menyediakan *endpoint* untuk *frontend* mengambil data analisis.
- **Authentication & Authorization:** JWT dan RBAC.

### 4.5. Modul Dasbor & Visualisasi

- **Frontend Application:** Mengonsumsi data dari API *backend*.
- **Charting Libraries:** Menggunakan D3.js atau Recharts untuk visualisasi tren, distribusi, *word cloud*, dll.
- **User Interface:** Desain responsif untuk tampilan di desktop dan *mobile*.

### 4.6. Modul Notifikasi & Laporan

- **Alerting Service:** Memantau metrik sentimen dan volume, memicu *alert* jika ambang batas terlampaui.
- **Reporting Service:** Menghasilkan laporan PDF/Excel secara terjadwal.
- **Integrasi:** Email Service API (SendGrid/Mailgun), WhatsApp Gateway API.

## 5. Persyaratan Keamanan

- **Data Encryption:** Enkripsi data saat istirahat (at rest) menggunakan *disk encryption* dan dalam transit (in transit) menggunakan SSL/TLS.
- **API Key Management:** Pengelolaan API key yang aman untuk akses ke platform media sosial.
- **Access Control:** RBAC yang ketat untuk akses ke dasbor dan konfigurasi.
- **Input Validation:** Validasi dan *sanitization* semua input pengguna untuk mencegah serangan.
- **Rate Limiting:** Menerapkan *rate limiting* pada API untuk mencegah penyalahgunaan.
- **Audit Logging:** Mencatat semua aktivitas pengguna dan sistem.
- **Compliance:** Memastikan kepatuhan terhadap Terms of Service dari platform sumber data (terutama media sosial).

## 6. Skalabilitas & Performa

- **Distributed Processing:** Menggunakan *message queue* dan *worker processes* (misal: Celery dengan Redis/RabbitMQ) untuk pemrosesan data secara paralel.
- **Auto-scaling:** Mengonfigurasi *auto-scaling* untuk *compute resources* berdasarkan beban kerja.
- **Database Sharding/Replication:** Untuk MongoDB, pertimbangkan *sharding* dan replikasi untuk skalabilitas dan ketersediaan tinggi.
- **Caching:** Menggunakan Redis atau Memcached untuk *caching* hasil analisis yang sering diakses.
- **CDN:** Untuk mendistribusikan aset *frontend* dan mempercepat *load time*.

## 7. Lingkungan Pengembangan & Deployment

- **Development Environment:** Menggunakan Docker untuk konsistensi lingkungan.
- **Staging Environment:** Lingkungan terpisah untuk pengujian integrasi dan performa.
- **Production Environment:** Lingkungan *live* di *cloud platform* dengan konfigurasi *high-availability*.
- **CI/CD Pipeline:** Otomatisasi *build*, *test*, dan *deployment* menggunakan GitLab CI/CD atau GitHub Actions.