

# Progress 1

## Penjelasan Arsitektur Sistem

Arsitektur sistem ini terdiri dari empat komponen utama: **Public Access**, **Local Services**, **Data Layer**, dan **External API**. Keempatnya saling terhubung untuk mendukung proses *retrieval*, *workflow automation*, pemanggilan API, serta pemrosesan dokumen.

### 1. Public Access (Public Exposure)

Bagian ini menggunakan **Ngrok** sebagai perantara agar layanan lokal dapat diakses dari internet. Ngrok menerima permintaan HTTP dari browser dan meneruskannya ke server lokal. Dengan demikian, pengguna bisa mengakses sistem tanpa perlu melakukan konfigurasi jaringan yang kompleks.

### 2. Local Services (Local Environment/Docker)

Pada lingkungan lokal, terdapat dua layanan inti:

- **n8n**: Berfungsi sebagai *workflow engine* untuk menangani otomatisasi, webhook, dan alur kerja pemrosesan data.
- **Node.js API Service**: Menyediakan berbagai fungsi seperti query Vector Database, membaca file lokal, melakukan *preprocessing*, dan memanggil API eksternal.

Keduanya biasanya berjalan melalui Docker untuk memudahkan pengelolaan, isolasi, dan deployment.

### 3. Data Layer

Lapisan data terdiri dari dua sumber utama:

- **Vector Database**: Menyimpan embedding dokumen untuk keperluan pencarian semantik, retrieval, dan RAG (Retrieval-Augmented Generation).
- **Local Files (PDF, Docs, dan lainnya)**: Menjadi sumber data mentah yang akan di-*ingest* dan diproses sebelum masuk ke VectorDB.

Node.js mengambil data dari kedua sumber ini sesuai kebutuhan proses.

### 4. External API (External Services)

Sistem juga menggunakan dua layanan eksternal:

- **OpenAI API:** Digunakan untuk membuat jawaban, merangkum konten, atau memproses hasil query VectorDB.
- **GitHub Repository:** Menyediakan tempat penyimpanan kode melalui mekanisme sinkronisasi versi. Baik Node.js maupun n8n dapat menarik pembaruan dari repo GitHub.

## Alur Koneksi dan Proses Kerja

1. **Browser** mengirim permintaan HTTP ke **Ngrok**.
2. **Ngrok** meneruskan permintaan tersebut ke **n8n** yang berjalan di **localhost:5678**.
3. **n8n** mengaktifkan workflow yang kemudian berkomunikasi dengan **Node.js**.
4. **Node.js** melakukan beberapa fungsi:
  - Query ke **VectorDB**
  - Mengambil isi dokumen dari **Local Files**
  - Melakukan pemanggilan ke **OpenAI API**
  - Menyusun kembali hasil dan mengirimnya ke pengguna melalui browser
5. Selain itu, n8n juga dapat melakukan proses *ingest* dan *preprocessing* dokumen baru ke dalam **Vector Database**.
6. **GitHub** menyediakan kontrol versi sehingga pembaruan kode secara otomatis dapat diterapkan ke Node.js dan n8n.

# Progress 2



# Laporan Pembuatan Telegram Bot dengan OpenAI & n8n (Self-Hosted)

## 1 Judul

Integrasi Bot Telegram dengan OpenAI melalui Automasi Workflow n8n (Self-Hosted)

---

## 2 Tujuan

Dokumentasi ini bertujuan untuk menjelaskan langkah-langkah teknis dalam pembuatan chatbot Telegram yang mampu menjawab pesan otomatis menggunakan model AI (OpenAI), yang di-deploy melalui platform automasi **n8n** secara self-hosted.

---

## 3 Alat & Teknologi yang Digunakan

Komponen	Keterangan
Telegram Bot	Media komunikasi bot dengan user
BotFather	Untuk membuat token API Bot Telegram
n8n Self-Hosted (localhost)	Platform automasi untuk membuat workflow
Node AI Agent – OpenAI Chat Model	Model AI sebagai mesin respon
Node Send Message (Telegram)	Untuk mengirim balasan otomatis dari AI ke user

---

## 4 Diagram Alur Sistem

User → Telegram → Telegram Trigger Node → AI Agent (OpenAI) → Send Telegram Message → User

---

## 5 Langkah-Langkah Pembuatan

### ✳ A. Membuat Bot Telegram

1. Buka Telegram dan chat @BotFather
2. Ketik /newbot
3. Masukkan nama bot dan username bot
4. Salin **Bot Token**, contoh:

1234567890:ABCdefGHIjkLmnoPQrsTUvwxyZ

---

### ✳ B. Setup Workflow di n8n

#### 1 Node – Telegram Trigger

- Fungsi: Menangkap pesan masuk dari user Telegram
- Konfigurasi:
  - Updates: message

**Output yang diterima:**

```
{  
  "message": {  
    "text": "bismillah"  
  }  
}
```

---

#### 2 Node – AI Agent (OpenAI Model)

Pilih parameter:

Source of Prompt (User Message): Define Below

-

Isi Prompt Expression:

```
{{ $json.message.text }}
```

- 
- Model: OpenAI Chat Model (misal GPT-4o atau gpt-3.5-turbo)

**Contoh prompt lengkap (opsional):**

User said: "{{ \$json.message.text }}"

Reply in a friendly and helpful tone.



Catatan:  
Jika memilih "Connected Chat Trigger Node", AI Agent akan error "No prompt specified" karena Telegram Trigger tidak mengirim field `chatInput`.

---

### 3 Node – Send a text message (Telegram)

Field "Chat ID" → pilih expression:

```
{{ $json.message.chat.id }}
```

- 

Field "Text" → isi:

```
{{ $json.output }}
```

- 

(output AI muncul di field `output`, lihat tab Output AI Agent setelah test)

---

## 6 Pengujian Bot

1. Klik **Execute Workflow** di n8n
2. Buka Telegram dan kirim pesan ke bot, contoh:

```
/start  
halo
```

hai  
bismillah

3. Bot membalas otomatis berdasarkan AI:

**Contoh respon:**

Hello! How can I assist you today?  
This message was sent automatically with n8n

---

## 7 Hasil / Output

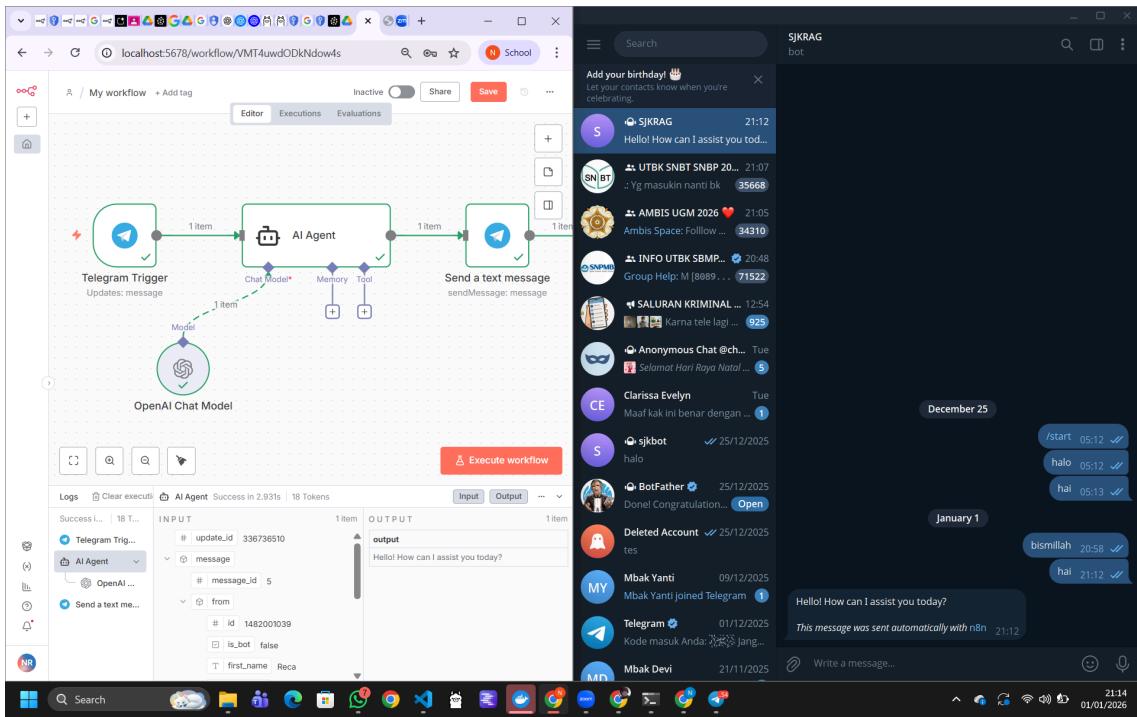
Bot berhasil:

- ✓ menerima pesan dari Telegram
  - ✓ mengirimkan pesan ke AI Agent
  - ✓ menampilkan respon AI ke Telegram secara otomatis
  - ✓ berjalan stabil via n8n self-hosted
- 

## 8 Kesimpulan

Melalui workflow sederhana n8n → OpenAI → Telegram, dapat dibuat sistem chatbot cerdas yang mampu membalas pesan pengguna secara real-time tanpa perlu menulis kode backend.

Metode ini cocok untuk pembuatan layanan customer service otomatis, bot edukasi, reminder, dan sistem Q&A ringan.



# Progress 3

# LAPORAN IMPLEMENTASI WEBAPP CHATBOT – INTEGRASI N8N & OPENAI

## 1. Judul Proyek

Implementasi Sistem Chatbot Web Menggunakan n8n Automation Tool dan OpenAI Model

## 2. Latar Belakang

Artificial Intelligence (AI) saat ini banyak diimplementasikan dalam berbagai bidang, salah satunya chatbot untuk meningkatkan efektivitas layanan informasi. Untuk mempermudah proses pembangunan sistem otomatisasi, digunakan platform *n8n* sebagai workflow automation engine yang terintegrasi dengan model AI dari OpenAI. Melalui integrasi ini, user dapat mengirim pesan melalui web, dan sistem akan merespons secara otomatis menggunakan AI.

## 3. Tujuan

Tujuan dari proyek ini adalah:

1. Membangun web aplikasi sederhana untuk mengirimkan pesan ke AI.
2. Membuat workflow otomatis di n8n yang menerima input melalui Webhook.
3. Menghubungkan webhook ke OpenAI Message Model untuk menghasilkan respons AI.
4. Menampilkan hasil jawaban AI kembali ke website secara real-time.

## 4. Alat & Teknologi

Alat dan teknologi yang digunakan dalam implementasi ini yaitu:

Teknologi / Tools	Fungsi
-------------------	--------

n8n Automation Tool	Sistem backend & workflow otomatis
Webhook Node	Menerima request POST dari web
OpenAI – Message Model Node	Menghasilkan jawaban AI
Respond to Webhook Node	Mengirimkan output kembali ke web
HTML + JavaScript	Tampilan frontend untuk pengguna
Browser (localhost)	Media uji coba aplikasi web

## 5. Arsitektur Sistem

Alur komunikasi data sistem adalah sebagai berikut:

```
User → Website → POST Request → n8n Webhook
      → OpenAI Message Model → Generate Response
      → Respond to Webhook → JSON → Ditampilkan ke UI Web
```

## 6. Langkah Implementasi

### 6.1 Pembuatan Workflow pada n8n

Workflow terdiri dari 3 node utama:

1. **Webhook (POST)** → digunakan untuk menerima pesan yang dikirim dari web UI.
2. **Message Model (OpenAI)** → node ini mengirimkan data yang diterima webhook ke model AI dan menghasilkan respons berdasarkan input.
3. **Respond to Webhook** → mengembalikan output AI kembali ke website.

Pada node **Respond to Webhook**, struktur output JSON harus disesuaikan agar frontend dapat membaca respons. Format yang digunakan adalah:

```
{
  "answer": "{$json.message.content}"
}
```

Field "`answer`" ini digunakan agar JavaScript di web dapat menampilkan hasil tanpa error `Undefined`.

## 6.2 Web Frontend

Website menggunakan HTML + JavaScript sederhana, di mana user memasukkan teks dan mengirimkannya ke webhook. JavaScript menerima respons JSON dan menampilkan nilai "`answer`" pada halaman.

## 7. Pengujian

Pengujian dilakukan dengan menekan tombol **Execute Workflow** di n8n dan mengirimkan input dari form web. Hasil pengujian menunjukkan:

Parameter	Nilai
Status Workflow	Success
Waktu Eksekusi	± 6.8 detik
Output AI	"Hello! How can I assist you today?"

Setelah penyesuaian JSON di node Respond to Webhook, teks berhasil ditampilkan dengan benar pada web tanpa error `undefined`.

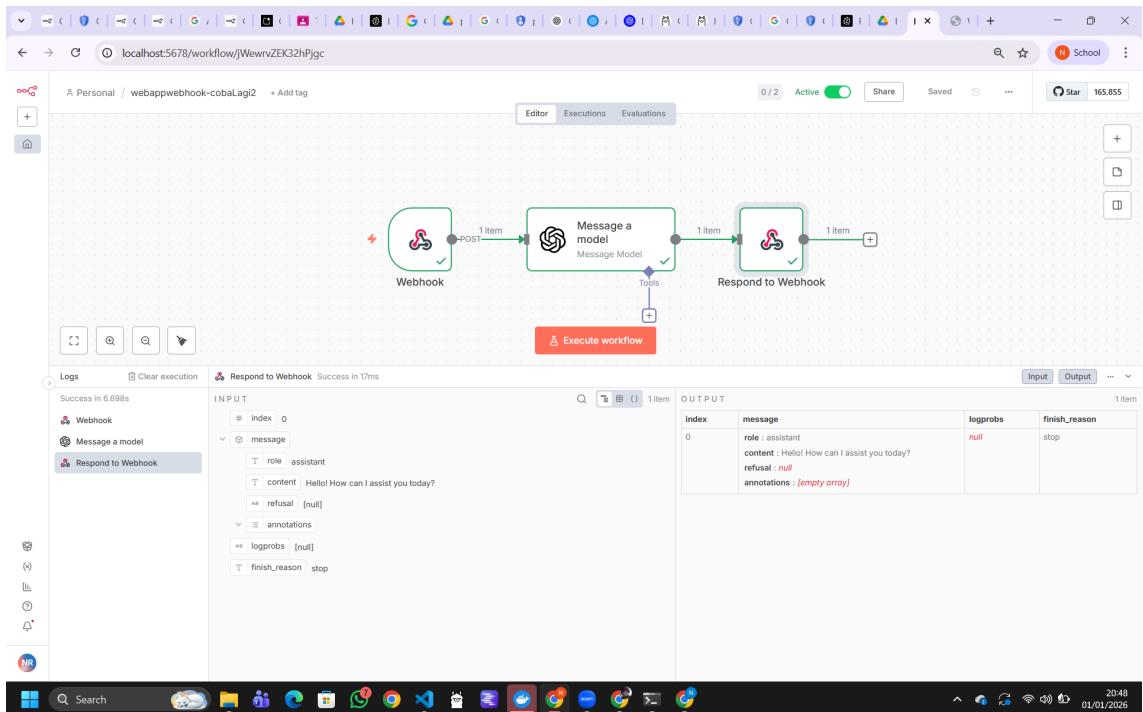
## 8. Hasil & Analisis

Berdasarkan implementasi dan pengujian:

- Integrasi antara web frontend, webhook, dan OpenAI berhasil dilakukan.
- Workflow dapat berjalan otomatis setiap kali request web dikirim.
- Respons AI berhasil diterima dan ditampilkan kembali pada browser.

## 9. Kesimpulan

Sistem chatbot berbasis web dengan integrasi n8n dan OpenAI berhasil dibuat dan berfungsi dengan baik. Sistem dapat menerima input dari user, memproses melalui automation engine, dan menghasilkan jawaban AI secara otomatis.



# Progress 4

# LAPORAN DOKUMENTASI SISTEM

## Integrasi Google Drive – n8n – Ollama – Pinecone Vector Store

---

### 1. Pendahuluan

Dokumen ini merupakan laporan dokumentasi sistem yang menjelaskan alur kerja (workflow) otomatis berbasis **n8n** untuk melakukan pemrosesan dokumen PDF dari **Google Drive**, melakukan **pemecahan teks (text splitting)**, menghasilkan **embedding menggunakan Ollama**, dan menyimpan hasil embedding ke dalam **Pinecone Vector Store**. Sistem ini dirancang untuk mendukung implementasi *Retrieval Augmented Generation (RAG)* pada aplikasi tanya-jawab berbasis AI.

---

### 2. Tujuan Sistem

Tujuan utama dari sistem ini adalah:

1. Mengotomatisasi pengambilan file PDF dari Google Drive.
  2. Mengekstraksi dan memecah isi dokumen menjadi potongan teks yang terstruktur.
  3. Menghasilkan embedding vektor menggunakan model embedding dari Ollama.
  4. Menyimpan embedding ke Pinecone sebagai basis data vektor.
  5. Menyediakan fondasi untuk pencarian semantik dan sistem tanya-jawab berbasis AI.
- 

### 3. Arsitektur Sistem

Sistem terdiri dari beberapa komponen utama berikut:

1. **Google Drive** – Sumber dokumen PDF.
2. **n8n** – Workflow automation sebagai orkestrator.
3. **Text Splitter** – Memecah dokumen menjadi chunk teks.
4. **Ollama Embeddings** – Mengubah teks menjadi vektor embedding.
5. **Pinecone Vector Store** – Menyimpan embedding untuk pencarian semantik.

---

## 4. Alur Kerja (Workflow)

### 4.1 Google Drive Trigger

Node ini berfungsi sebagai pemicu awal workflow.

- Trigger aktif ketika ada file baru yang diunggah ke Google Drive.
- File yang diproses adalah file dengan format PDF.

### 4.2 Loop Over Items

Node ini digunakan untuk:

- Melakukan iterasi apabila terdapat lebih dari satu file.
- Memastikan setiap file diproses satu per satu.

### 4.3 Download File

Node ini berfungsi untuk:

- Mengunduh file PDF dari Google Drive.
- Menyediakan file sebagai input ke tahap pemrosesan berikutnya.

### 4.4 Default Data Loader

Node ini bertugas untuk:

- Mengekstraksi teks dari file PDF.
- Menghasilkan metadata seperti judul dokumen, jumlah halaman, dan konten teks.

### 4.5 Recursive Character Text Splitter

Node ini digunakan untuk:

- Memecah teks panjang menjadi potongan (chunk) lebih kecil.
- Menjaga konteks teks agar tetap relevan.
- Memudahkan proses embedding dan penyimpanan vektor.

### 4.6 Embeddings Ollama

Node ini berfungsi untuk:

- Mengonversi setiap potongan teks menjadi embedding vektor.

- Menggunakan model embedding lokal dari Ollama.

## 4.7 Pinecone Vector Store

Node ini merupakan tahap akhir workflow:

- Menyimpan embedding vektor ke dalam Pinecone.
  - Menyertakan metadata dokumen untuk keperluan pencarian.
  - Menghasilkan beberapa item embedding sesuai jumlah chunk teks.
- 

## 5. Contoh Output Sistem

Output dari sistem berupa:

- Data embedding dalam bentuk vektor numerik.
- Metadata dokumen (nama file, tipe file, jumlah halaman).
- Konten teks per potongan (pageContent).

Data ini tersimpan di Pinecone dan siap digunakan untuk pencarian semantik atau sistem RAG.

---

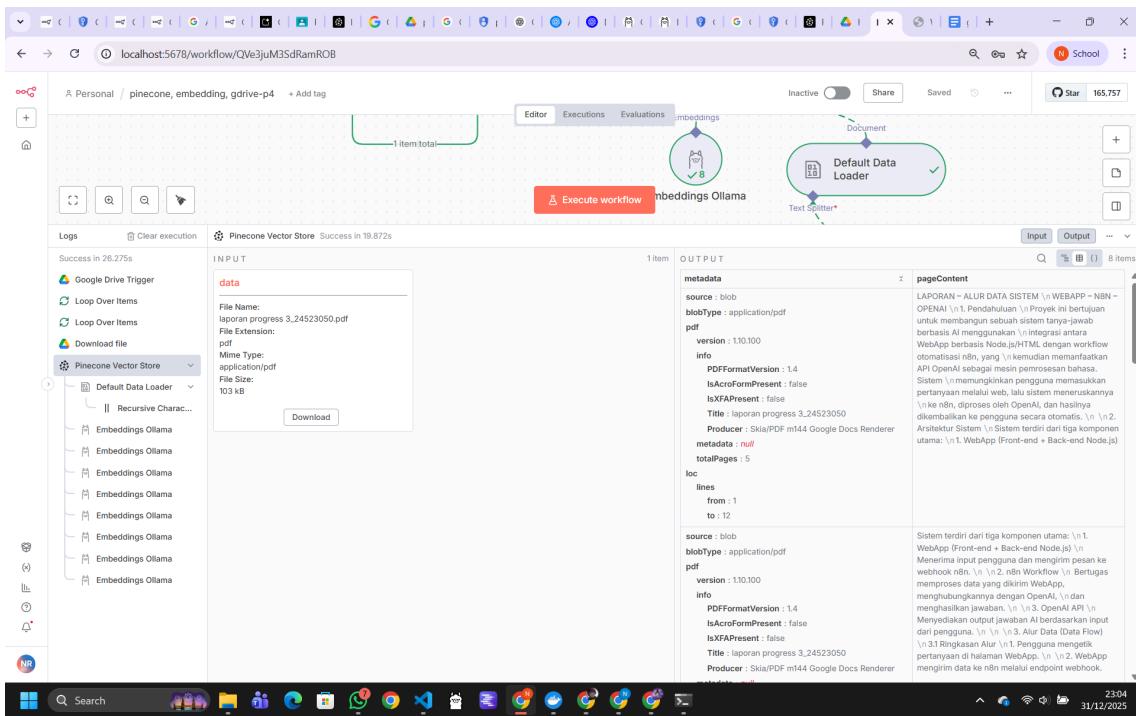
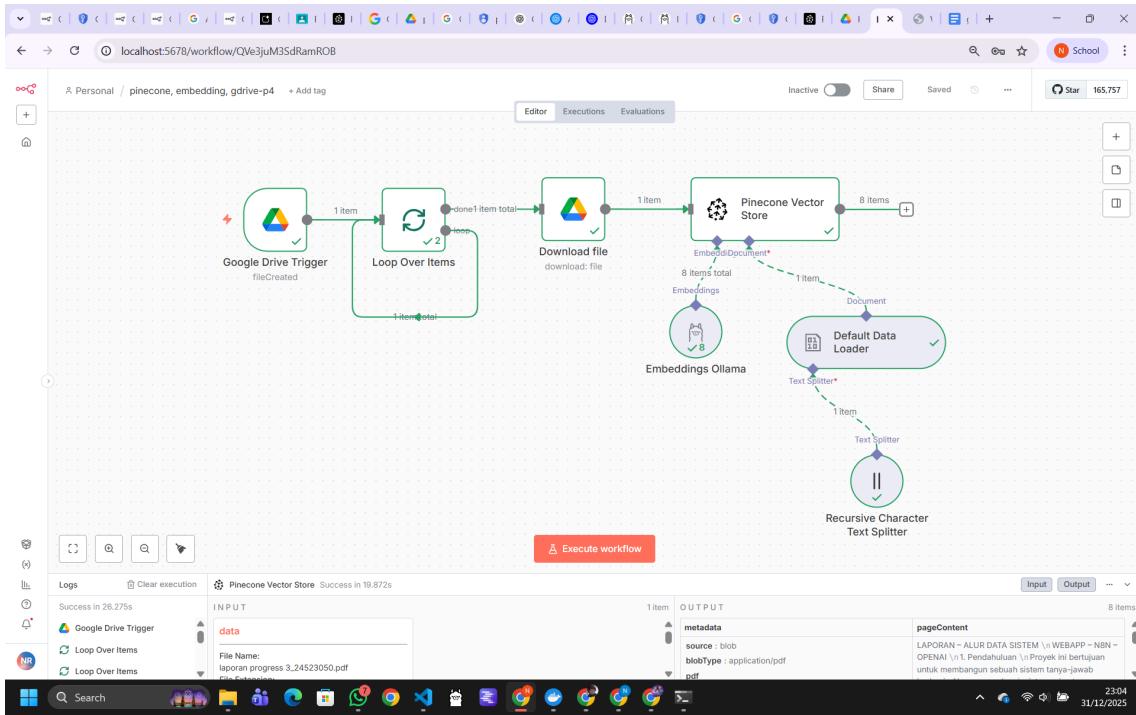
## 6. Keunggulan Sistem

1. **Otomatis** – Tidak memerlukan input manual setelah workflow aktif.
  2. **Skalabel** – Mampu menangani banyak dokumen.
  3. **Efisien** – Menggunakan chunking untuk mengoptimalkan embedding.
  4. **Siap RAG** – Terintegrasi langsung dengan Vector Database.
- 

## 7. Kesimpulan

Workflow n8n yang dibangun berhasil mengintegrasikan Google Drive, Ollama, dan Pinecone dalam satu alur otomatis. Sistem ini sangat cocok digunakan sebagai backend untuk aplikasi AI berbasis dokumen, seperti chatbot akademik, pencarian dokumen cerdas, dan sistem tanya-jawab berbasis konteks.

---



# Progress 5

# LAPORAN IMPLEMENTASI WORKFLOW AI AGENT TERINTEGRASI TELEGRAM

## 1. Pendahuluan

Laporan ini disusun untuk mendokumentasikan implementasi dan pengujian workflow *AI Agent* yang terintegrasi dengan Telegram menggunakan platform **n8n**. Workflow ini bertujuan untuk membangun chatbot berbasis AI yang mampu merespons pesan pengguna secara otomatis, serta mendukung integrasi *Retrieval-Augmented Generation (RAG)* melalui penyimpanan vektor menggunakan **Pinecone** dan pengelolaan dokumen dari **Google Drive**.

## 2. Tujuan Implementasi

Tujuan utama dari implementasi workflow ini adalah:

1. Mengembangkan chatbot Telegram berbasis AI yang dapat merespons pesan pengguna secara real-time.
2. Mengintegrasikan model bahasa besar (*Large Language Model*) melalui node **OpenAI Chat Model**.
3. Menyediakan kemampuan pencarian pengetahuan menggunakan **Pinecone Vector Store**.
4. Mengelola sumber data berbasis dokumen yang berasal dari **Google Drive**.
5. Memastikan workflow berjalan stabil dan berhasil dieksekusi tanpa error.

## 3. Arsitektur dan Alur Workflow

Workflow yang diimplementasikan terdiri dari beberapa komponen utama sebagai berikut:

### 3.1 Trigger

- **Telegram Trigger**
  - Berfungsi untuk menangkap pesan masuk dari pengguna Telegram.
  - Contoh pesan yang diterima: "halo".
- **Webhook (opsional)**
  - Digunakan untuk mendukung integrasi eksternal selain Telegram.

## 3.2 AI Agent

- Node **AI Agent** berperan sebagai pusat pemrosesan logika.
- AI Agent menerima input dari Telegram Trigger dan meneruskannya ke:
  - **OpenAI Chat Model** untuk menghasilkan respons berbasis AI.
  - **Pinecone Vector Store** untuk pengambilan konteks tambahan (RAG).

## 3.3 Model AI

- **OpenAI Chat Model**
  - Digunakan untuk memproses teks dan menghasilkan jawaban alami.
  - Terbukti mampu merespons pesan sederhana seperti sapaan pengguna.

## 3.4 Manajemen Knowledge Base

- **Google Drive Trigger**
  - Mendeteksi file baru atau perubahan file di Google Drive.
- **Download File & Loop Over Items**
  - Mengunduh file dari Google Drive.
  - Memproses file satu per satu.
- **Text Splitter & Embeddings (Ollama)**
  - Memecah dokumen menjadi potongan teks kecil.
  - Mengonversi teks menjadi vektor embedding.
- **Pinecone Vector Store**
  - Menyimpan embedding dokumen.
  - Digunakan oleh AI Agent untuk menjawab pertanyaan berbasis data.

## 3.5 Output

- **Send a Text Message (Telegram)**
  - Mengirimkan respons AI kembali ke pengguna Telegram.
  - Contoh output: "*Halo! Ada yang bisa saya bantu hari ini?*"

## 4. Hasil Pengujian Workflow

Berdasarkan hasil eksekusi workflow pada tanggal **1 Januari 2026 pukul 22:16:40**, diperoleh hasil sebagai berikut:

- Status eksekusi: **Succeeded**
- Waktu eksekusi: **±2,579 detik**
- Jumlah token yang diproses AI Agent: **69 token**
- Tidak ditemukan error pada eksekusi terakhir.

### 4.1 Pengujian Telegram Bot

- Pengguna mengirim pesan: "halo".
- Sistem merespons secara otomatis melalui Telegram dengan pesan: "*Halo! Ada yang bisa saya bantu hari ini?*"
- Pesan berhasil dikirim dan diterima dengan baik.

## 4.2 Monitoring Error

- Terdapat satu eksekusi sebelumnya yang mengalami error (durasi ±86 ms).
- Error berhasil diperbaiki dan tidak muncul kembali pada eksekusi selanjutnya.

# 5. Analisis dan Evaluasi

## 5.1 Kelebihan

- Workflow berjalan stabil dan responsif.
- Integrasi Telegram dengan AI Agent berhasil.
- Sistem siap dikembangkan menjadi chatbot berbasis knowledge base.

## 5.2 Kekurangan

- Respons masih bersifat umum (sapaan awal).
- Belum dilakukan pengujian beban (*stress testing*) untuk banyak pengguna.

# 6. Kesimpulan

Implementasi workflow AI Agent terintegrasi Telegram menggunakan n8n telah berhasil dilakukan. Sistem mampu menerima pesan dari pengguna, memprosesnya menggunakan model AI, dan mengirimkan respons kembali secara otomatis. Integrasi dengan Pinecone dan Google Drive membuka peluang pengembangan chatbot berbasis data dan dokumen di masa mendatang.

# 7. Rekomendasi Pengembangan

1. Menambahkan *prompt engineering* agar respons lebih kontekstual.
  2. Mengaktifkan penuh fitur RAG dengan dokumen Google Drive.
  3. Menambahkan logging dan monitoring lanjutan.
  4. Menguji performa untuk skala pengguna yang lebih besar.
-

