



# Jalur Obat

## AI-Based Counterfeit Medicine Distribution Detection System

---

### 1. Project Overview

**Jalur Obat** is an AI-based application designed to monitor and analyze medicine distribution journeys in order to detect counterfeit risks.

The system combines:

- A **web application**
- A **Node.js backend**
- A **PostgreSQL database**
- A **Python-based AI service**

The application records medicine journey data, evaluates risk using AI scoring, and classifies the overall safety status of distributed medicines.

---

### 2. System Requirements

Before running the project, ensure the following tools are installed:

- **Visual Studio Code**
- **Git**

- **Node.js (npm included)**
  - **Python (with venv support)**
  - **PostgreSQL**
  - **pgAdmin 4**
- 

## 3. Project Setup

### 3.1 Clone the Repository

1. Open a terminal.
2. Navigate to the directory where you want to store the project:

```
cd Documents
```

3. Clone the repository:

```
git clone https://github.com/DlegendZ/Jalur-Obat.git
```

4. Enter the project directory:

```
cd jalur-obat
```

---

### 3.2 Install Node.js Dependencies

1. Open a terminal inside the project directory.
2. Install required dependencies:

```
npm install express pg dotenv cors
```

3. Confirm that the `node_modules` folder is created.
- 

### 3.3 Backend Python Environment (AI Service)

1. Navigate to the backend folder:

```
cd backend
```

2. Create a virtual environment:

```
python -m venv venv
```

3. Activate the virtual environment:

```
venv\Scripts\activate
```

4. Install required Python packages:

```
pip install -r requirement.txt
```

---

## 4. Database Configuration

### 4.1 Create the Database

1. Open **pgAdmin 4**.
2. Create a new database (example name: `medicine_journey` ).

---

### 4.2 Create Tables

Execute the following **DDL script** inside the newly created database:

```
CREATE TABLE medicine_info (  
    report_id SERIAL PRIMARY KEY,  
    serial_number TEXT NOT NULL,  
    medicine_name TEXT NOT NULL,  
    current_location TEXT NOT NULL,  
    quantity INTEGER NOT NULL,  
    additional TEXT,  
    temperature NUMERIC(5,2) NOT NULL,  
    humidity NUMERIC(5,2) NOT NULL,  
    ai_score_fake_result NUMERIC(5,2),  
    ai_journey_score TEXT NOT NULL CHECK (  
        ai_journey_score IN ('Safe', 'Need Attention', 'Full Attention', 'Faked', 'U  
nknown')  
    ),  
    overall_status TEXT NOT NULL CHECK (
```

```
        overall_status IN ('Safe', 'Need Attention', 'Bad')
    ),
    expedition_type TEXT NOT NULL CHECK (
        expedition_type IN ('Land', 'Air', 'Sea')
    ),
    created_at TIMESTAMP DEFAULT NOW(),
    officer_id TEXT NOT NULL,
    journey_status TEXT NOT NULL CHECK (
        journey_status IN ('start', 'update', 'end')
    )
);
```

Grant privileges (replace `YOUR_USERNAME` with your PostgreSQL username):

```
GRANT ALL PRIVILEGES ON TABLE medicine_info TO YOUR_USERNAME;
GRANT USAGE, SELECT ON SEQUENCE medicine_info_report_id_seq TO YOUR_USERNAME;
```

## 4.3 Environment Variables

Update the `.env` file with your PostgreSQL credentials:

```
DB_USER=your_pgadmin_username
DB_PASSWORD=your_pgadmin_password
DB_NAME=medicine_journey
DB_HOST=localhost
DB_PORT=5432
```

## 5. Running the System

### 5.1 Start the Web Application

1. Open a terminal in the directory containing `package.json`.
2. Run:

```
npm run dev
```

3. Open the application in a browser:

```
http://localhost:3000
```

---

## 5.2 Start the AI Service

1. Open **Visual Studio Code**.
2. Open the `jalur-obat` project.
3. Open the terminal and navigate to the backend folder:

```
cd backend
```

4. Activate the virtual environment:

```
venv\Scripts\activate
```

5. Start the AI server:

```
uvicorn app.main:app --reload
```

---

## 5.3 Start the Database Server

1. Navigate to the `app` folder (contains `server.js`).
2. Open a system terminal (CMD).
3. Run:

```
node server.js
```

4. The backend is now connected to the database.

---

## 6. System Workflow Summary

- **Frontend** collects medicine journey data.
- **Node.js backend** stores and retrieves data from PostgreSQL.
- **AI service** evaluates risk and generates AI scores.

- **Database** stores all journey records and evaluation results.
- 

## 7. Conclusion

Jalur Obat provides an integrated solution for tracking medicine distribution and detecting counterfeit risks through structured data collection and AI-based evaluation.

Proper setup of dependencies, database configuration, and service execution is required to run the system successfully.