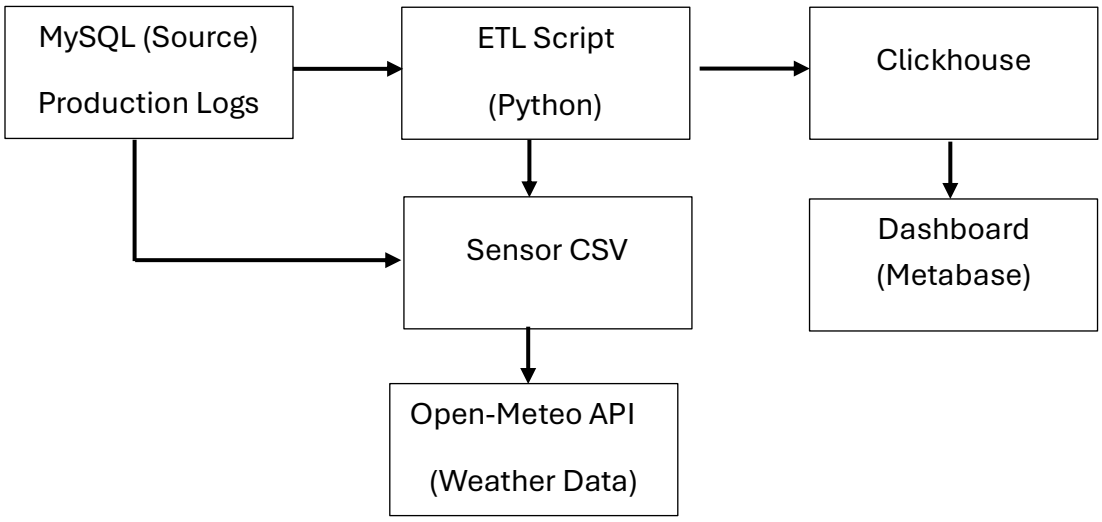
# Febriyeni Susi
# AI Engineer

**Pipeline Design**




| Component | Technology | Purpose |
|---|---|---|
| **Source Database** | MySQL | Stores raw production_logs data |
| **IoT Sensor Input** | CSV | Logs real-time equipment metrics |
| **Weather Data** | Open-Meteo API | Provides historical daily weather data for Berau, Kalimantan, Indonesia |
| **ETL** | Python | Orchestrates extraction, transformation, and loading processes |
| **Data Warehouse** | ClickHouse | Stores daily-level computed metrics for analytical |
| **Dashboard** | Metabase | Provides user-friendly visualizations and reports. |
| **Containerization** | Docker, Docker Compose | Simplifies deployment and environment consistency. |

**ETL Process**

The ETL process is fully automated and consists of several sequential stages:

1. Extraction

- Production Logs: Retrieved from a MySQL database (production_logs table) with fields date, mine_id, shift, tons_extracted, quality_grade
- Sensor Data: Loaded from a CSV file equipment_sensors.csv with fields timestamp, equipment_id, status, fuel_consumption, maintenance alert
- Weather Data: Fetched daily data from the Open-Meteo API using the date as the query parameter and coordinates of the mining site (latitude and longitude). Extracted fields are temperature_2m_mean, precipitation_sum

2. Transformation

Daily metrics are computed as follows:

| Metric | Description |
|---|---|
| total_production_daily | Total coal extracted (tons_extracted) per day |
| average_quality_grade | Average of quality_grade per day |
| equipment_utilization | Total active equipment hours / total available hours (percentage) |
| fuel_efficiency | Fuel used per ton of coal mined (total_fuel / total_production_daily) |
| temperature_mean | Mean daily temperature (°C) from Open-Meteo |
| precipitation | Daily rainfall (mm) from Open-Meteo |
| weather_impact | Correlation between precipitation and total_production_daily |

After metrics are computed:

a. Sensor data timestamps are aggregated into daily summaries.
b. Fuel consumption and operational status are joined per equipment and per date.
c. Weather data is joined on date for enrichment.

3. Loading

Final data is inserted into a ClickHouse table called daily_production_metrics. The schema:

```
CREATE TABLE daily_production_metrics (
        date Date,
        total_production_daily Float64,
        average_quality_grade Float64,
        equipment_utilization Float64,
        fuel_efficiency Float64,
        temperature_mean Float32,
        precipitation Float32
    ) ENGINE = MergeTree() ORDER BY date
```

**Data Validation**

Before loading data, strict validation checks are applied:

| Check Description | Validation Logic |
|---|---|
| Valid production values | total_production_daily >= 0 |
| Equipment utilization within bounds | 0 <= equipment_utilization <= 100 |
| Weather completeness | No missing temperature_mean or precipitation values |
| Fuel division safety | Division by zero handled in fuel_efficiency computation |
| Error logging | All anomalies are logged in etl_errors.log |

If any check fails, the ETL process is halted and the error is logged with timestamps for debugging.