## **Data Governance Plan**

### **1. Objective**

To ensure the accuracy, consistency, and reliability of trade and P2P transaction data across multiple data sources, enabling trusted insights into user behaviors (e.g., conversion from P2P to trade). This plan establishes clear ownership, quality control, lineage tracking, and access management across the data pipeline — from raw to analytics layers.

### **2. Governance Scope**

Covers all processes and datasets in the pipeline:

* **Raw Layer**: Direct ingestion from multiple CSVs / source schemas.
* **Staging Layer**: Standardization, deduplication, and timezone conversion (UTC → UTC+7 Jakarta).
* **Data Warehouse Layer**: Integration of trade, user, and P2P tables for analysis and dashboards.
* **Analytics Layer**: Business-ready datasets for metrics like active traders, trade volume, and conversion rates.

### **3. Roles & Responsibilities**

| **Role** | **Responsibility** |
| --- | --- |
| **Data Owner** | Accountable for source data integrity and business rules (e.g., Operations or Product team). |
| **Data Steward** | Ensures data quality checks, manages dictionary & metadata, oversees anomaly resolution. |
| **Data Engineer** | Builds ETL pipelines, maintains schema consistency, and enforces transformations. |
| **Data Analyst** | Consumes curated data for insight generation, validates business metrics, and reports anomalies. |

### **4. Data Quality Framework**

| **Quality Dimension** | **Validation Example** | **Action** |
| --- | --- | --- |
| **Accuracy** | Validate that trade\_value > 0 and IDs exist in the user table. | Reject invalid records. |
| **Completeness** | Check for missing receiver\_id, sender\_id, or token\_id. | Fill or flag as NULL. |
| **Uniqueness** | Detects and removes duplicate trades (e.g., trade\_id or transfer\_id). | Deduplication logic in staging. |
| **Timeliness** | Apply UTC → UTC+7 timezone conversion. | Consistent timestamps for dashboard use. |
| **Consistency** | Align schema formats (string, numeric) across source schemas. | Standardize datatypes in staging. |

### **5. Metadata & Data Lineage**

Each dataset must include metadata fields like:

* source\_system, extraction\_time, last\_updated\_by, record\_count.

**Data Lineage Example (simplified):**

Source: p2p\_transfers.csv, trades.csv

↓

RAW Schema: raw\_p2p\_transfers, raw\_trades

↓

STAGING Schema: stg\_p2p\_transfers, stg\_trades

(deduplication, timezone conversion, data type standardization)

↓

DATA WAREHOUSE Schema: datawarehouse\_layer

(joins trades + p2p to derive conversion and engagement KPIs)

↓

DASHBOARD Layer: user\_behavior\_dashboard

(conversion rate, avg trade value, P2P frequency)

### **6. Scalability & Maintenance**

To ensure scalability:

* **Modular schema design**: Separate schema for raw, staging, and warehouse layers.
* **Incremental load pipelines**: Only process delta changes to reduce computation cost.
* **Partitioning by date/user\_id** for large tables (e.g., trades).
* **Reusable data models**: Design fact tables (fact\_trades, fact\_p2p) and shared dimensions (dim\_user, dim\_token).
* **Automated validation**: Daily data quality jobs to alert anomalies.

### **7. Access & Compliance**

* Use **role-based access control (RBAC)**:  
  + Analysts → dw schema (read-only)
  + Engineers → all schemas
  + Business users → dashboard only
* Ensure **PII masking** for sensitive fields (user\_id, email).
* Enable **audit logging** of queries and data changes.