



BeanGuard.

Automated Inventory Reconciliation & Anomaly Detection for Kopikita Roastery

Presented by binbin nusnus



Tasya



Akbar



Melvyn

Tantangan Pengelolaan Inventaris UMKM F&B

Masalah utama bukan kurangnya data, tetapi data operasional yang belum saling terhubung.

BeanGuard.

Big Picture UMKM di Indonesia

UMKM F&B di Indonesia terus **bertumbuh**, namun **digitalisasi** operasional **masih belum terintegrasi**. Data dari jurnal Scripta Economica pada tahun 2025 menunjukkan:

64,2jt UMKM ada di Indonesia

Namun sebanyak 68% **UMKM** memilih untuk melakukan pencatatan manual dan belum menggunakan software ERP dikarenakan berbagai macam faktor hambatan

Biaya implementasi cenderung tinggi

Sistem manual dianggap lebih "mudah & murah"

Merasa belum memerlukan sistem yang lebih kompleks

Study Case: Kopikita Roastery

Salah satu UMKM yang belum menerapkan penggunaan ERP adalah **kedai Kopikita Roastery** yang sehari-harinya hanya menggunakan spreadsheet biasa. Namun saat ini sedang mengalami berbagai kendala pendataan.



Masing-masing dengan satuan ukur yang berbeda sehingga **sulit dicocokkan** secara manual, **langsung** dan **belum punya** sistem **integrasi data** yang **jelas**. Akibatnya, pemilik kedai gagal mendeteksi penyusutan barang dan mengalami kerugian.

Sudarynianto Nurathaya Sutarto, Natasha Anjanette Pranata, & Sri Rahayuningsih. (2025). Perubahan Praktik Manajemen Persediaan pada UMKM melalui Transformasi Digital di Era E-Commerce. *Journal of Economics, Management, and Accounting*, 1(2), 170–178. <https://doi.org/10.65310/JKG02470>

Kopikita memiliki 5 dataset utama, tetapi belum memiliki model data relasional yang menghubungkan transaksi, resep, stok, dan master inventory.

Overview Sumber Data

Dataset	Fungsi
sales_history.csv	Riwayat transaksi penjualan dari POS
warehouse_stock.json	Stok fisik harian di gudang
master_inventory.csv	Data pembelian dan bahan baku
recipe_BOM.json	Komposisi bahan baku setiap menu
employee.json	Data referensi karyawan

Kondisi Awal Data

- Berasal dari 5 dataset terpisah tanpa model relasional terpadu.
- Memiliki struktur dan format yang berbeda-beda.
- Belum dapat digunakan untuk rekonsiliasi inventaris secara langsung.

Masalah dan Dampaknya pada Proses Bisnis

1

Data Quality Issues

Dirty data, null values, typo, dan perubahan struktur data. Kualitas data yang buruk menurunkan akurasi laporan dan meningkatkan risiko pengambilan keputusan yang salah.

2

Unit & Recipe Conversion

Penjualan produk jadi harus dikonversi menjadi konsumsi bahan baku melalui Bill of Materials (BOM) sebelum dapat dibandingkan dengan stok. Perbedaan satuan ukur menghambat rekonsiliasi inventaris dan menyulitkan pelacakan konsumsi bahan baku secara akurat.

3

Inventory Reconciliation

Perbedaan satuan dan sumber data membuat POS, gudang, dan pembelian tidak dapat dicocokkan secara langsung. Pemilik usaha tidak memiliki visibilitas stok yang akurat dan real-time untuk mendukung pengambilan keputusan.

4

Anomaly Detection

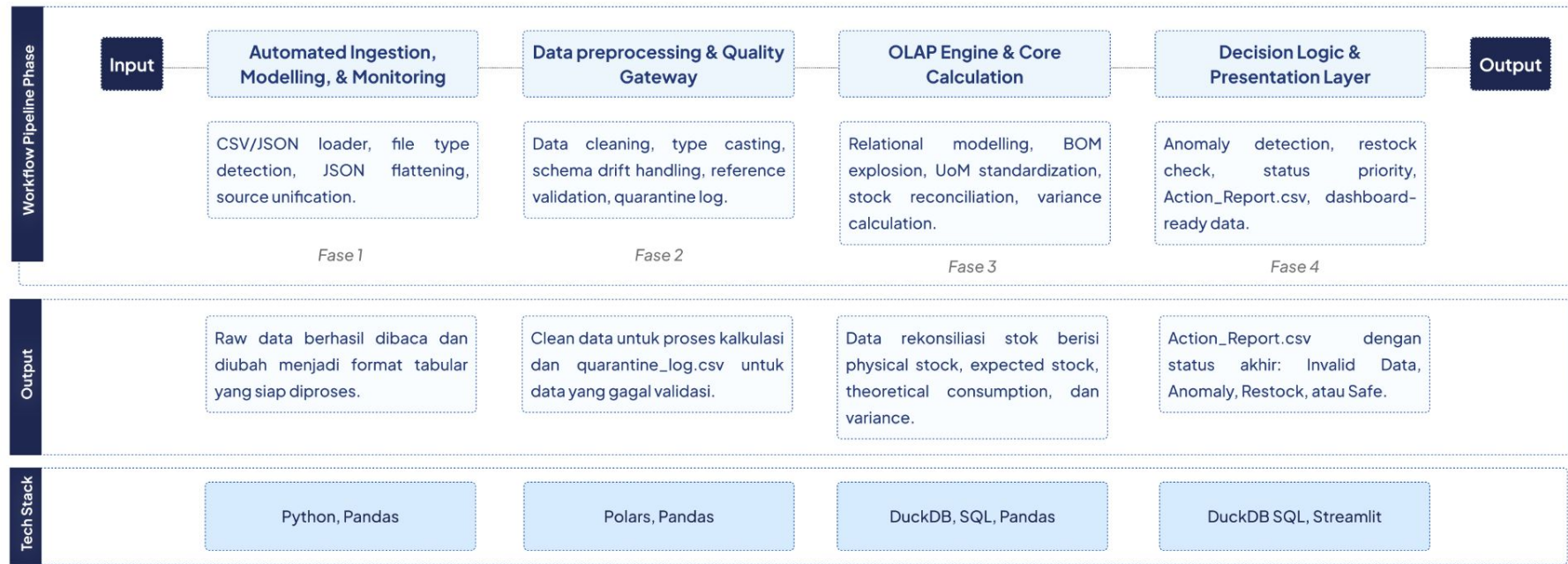
Sistem belum mampu membedakan selisih operasional yang wajar dengan indikasi anomali atau shrinkage sehingga tidak mampu membedakan penyusutan normal dengan anomali inventaris dan berpotensi menimbulkan kerugian finansial.

From Fragmented Data to Actionable Insights

BeanGuard.

BeanGuard mengubah data operasional yang terpisah dan tidak rapi menjadi laporan aksi yang dapat langsung digunakan untuk pengambilan keputusan.

Objective: Membangun lightweight automated inventory pipeline yang mengubah data POS, gudang, dan inventory yang terfragmentasi menjadi keputusan stok yang akurat melalui UoM normalization, BOM mapping, dan anomaly detection.



Introduction

Problem Identification

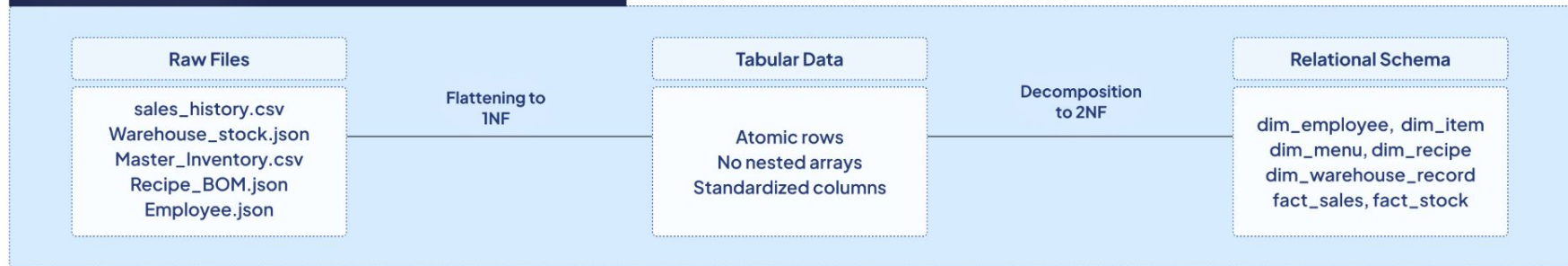
Solution Approach

Result

Data Modelling: Membangun Data Relasional

Data mentah dinormalisasi menjadi struktur relasional agar transaksi, resep, item, employee, dan stok dapat saling terhubung.

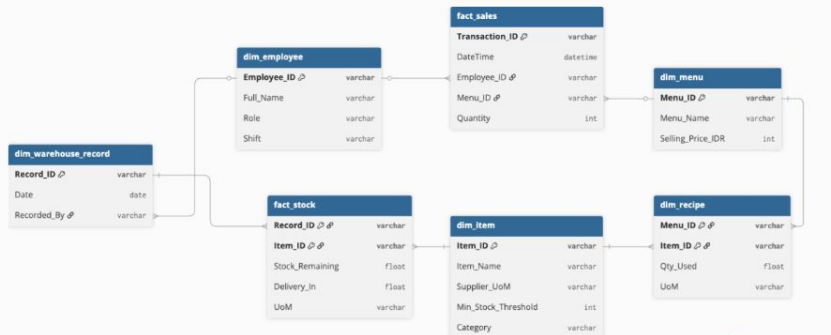
Proses Normalisasi



Hasil Akhir Tabel

Kami berhenti di 2NF karena kebutuhan utama untuk analytical processing bukan membangun transactional database yang terlalu kompleks. Pada tahap 2NF, data sudah cukup rapi untuk menghilangkan partial dependency, memisahkan master data dari data transaksi, dan mendukung proses utama seperti BOM calculation, stock reconciliation, serta anomaly detection.

Selain itu karena sistem ini harus ringan, mudah dijalankan, dan efisien untuk analisis harian, 2NF menjadi titik optimal antara data integrity dan processing efficiency.

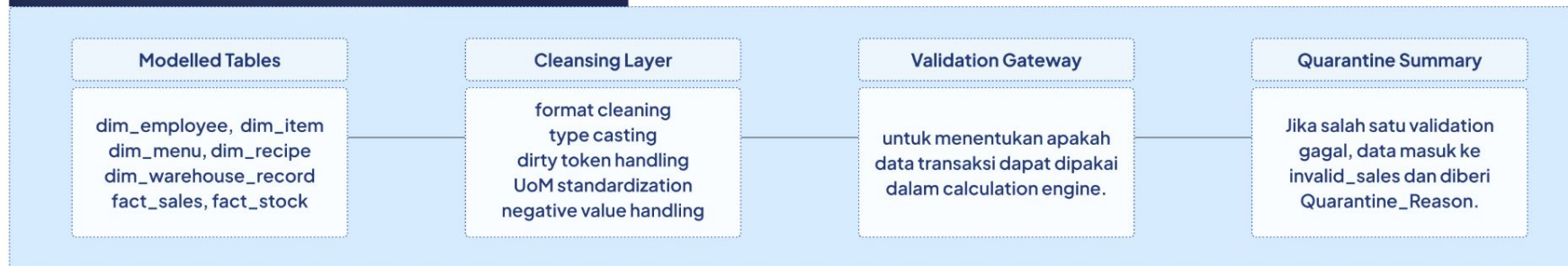


Data Preprocessing & Quality Gateway

BeanGuard.

BeanGuard menstandarisasi format dan memisahkan data valid dan invalid agar pipeline tetap berjalan tanpa intervensi manual.

Proses Preprocessing & Validation



Final Output



valid_sales → Calculation Engine

valid_sales berisi transaksi POS yang sudah lolos seluruh aturan validasi. Data ini memiliki `Transaction_ID`, `DateTime`, `Quantity`, `Employee_ID`, dan `Menu_ID` yang valid, sehingga aman digunakan untuk lanjut ke proses perhitungan berikutnya.



invalid_sales → Quarantine Summary

invalid_sales berisi transaksi yang gagal melewati validation gateway. Setiap baris invalid diberi `Quarantine_Reason` agar penyebab error dapat diketahui. Data ini diringkas menjadi quarantine summary untuk menunjukkan jumlah dan persentase data invalid berdasarkan penyebabnya.

UoM Standardization: Converting Units into Comparable Base Units

standarkan dan mengonversi satuan ukur dari berbagai sumber data agar stok gudang, resep, dan master inventory dapat dibandingkan dalam basis perhitungan yang sama.

String Normalization

Untuk menyamakan variasi penulisan satuan menjadi format standar. Contoh:

- Gram, GRAM, gr → g
- Kilogram, KG, kilo → kg

Dimension Classification

Setiap satuan diklasifikasikan berdasarkan dimensinya. Contoh:

- ml, l → volume
- pcs, karton → count

Base Unit Conversion

Setiap dimensi dikonversi ke base unit agar dapat dibandingkan. Contoh:

- mass → g
 - volume → ml
- Conversion factor:
- kg × 1000 → g
 - l × 1000 → ml

Base Unit Conversion

Jika item dicatat dalam karton, sistem mengestimasi jumlah pcs per karton berdasarkan kategori.

Item: Paper Cup
Category: packaging
Supplier_UoM: karton

Rule:
packaging → 50 pcs/karton

Result:
1 karton Paper Cup = 50 pcs

Karton-to-Pcs Mapping

Karena karton tidak selalu punya isi yang sama, sistem tidak bisa pakai 1 fixed factor untuk semua item. Jadi BeanGuard pakai rule berdasarkan kategori item.

Contoh:

- KG → kg → mass → g → valid
- sachet → UNKNOWN → need review

BOM & Stock Reconciliation: Turning Sales into Inventory Variance

OLAP untuk mengubah transaksi penjualan menjadi konsumsi bahan baku, lalu membandingkannya dengan stok fisik gudang untuk menghitung variance.

BOM Explosion

Transaksi POS hanya mencatat produk jadi yang terjual, seperti Iced Latte atau Croissant. Agar dapat dibandingkan dengan stok gudang, setiap menu harus dibongkar menjadi bahan baku menggunakan tabel resep.

Input

fact_sales_valid

Berisi transaksi POS yang sudah lolos validation gateway.

dim_recipe_clean

Berisi komposisi bahan baku per menu,

Process

$\text{Quantity Sold} \times \text{Qty Used} = \text{Calculated Usage}$

Output

calculated_usage untuk menunjukkan jumlah bahan baku yang seharusnya terpakai berdasarkan transaksi penjualan.

Stock Reconciliation

Setelah konsumsi bahan baku dihitung lanjut untuk membandingkan stok yang seharusnya tersisa dengan stok fisik yang dicatat di gudang.

Input

fact_stock_clean

Berisi stok fisik dan delivery item.

dim_warehouse_record_clean

Berisi tanggal pencatatan gudang.

calculated_usage

Berisi konsumsi bahan baku hasil BOM Explosion.

Process

get Stock_Awal using LAG() function

$\text{Expected Stock} = \text{Stock_Awal} + \text{Delivery_In} - \text{Calculated_Usage}$

$\text{Variance} = \text{Physical_Stock} - \text{Expected_Stock}$

Output

Physical_Stock, Expected_Stock, Variance, data_quality_flag

Anomaly Detection: Membedakan Selisih Wajar dan Anomali Inventaris

BeanGuard memakai threshold berlapis agar tidak semua selisih langsung dianggap anomali.

Proses Logic Anomaly Detection



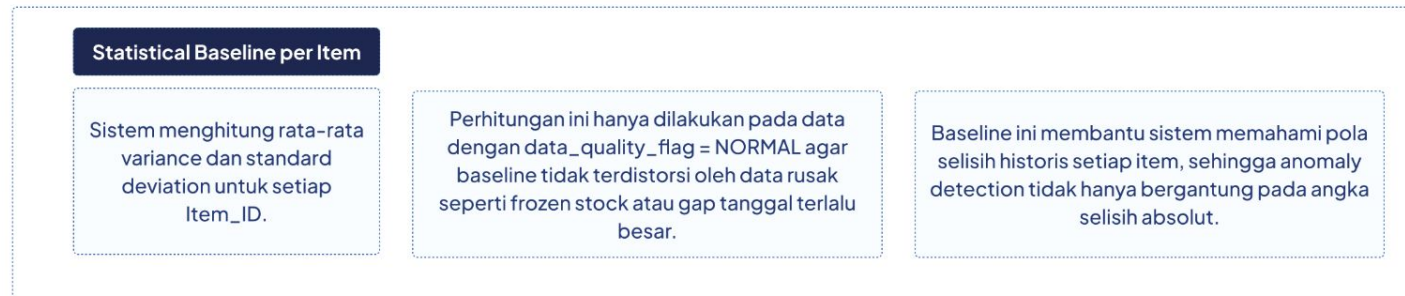
Anomaly Detection: Membedakan Selisih Wajar dan Anomali Inventaris

BeanGuard memakai threshold berlapis agar tidak semua selisih langsung dianggap anomali.

Proses Logic Anomaly Detection



Penjelasan Tahapan Proses



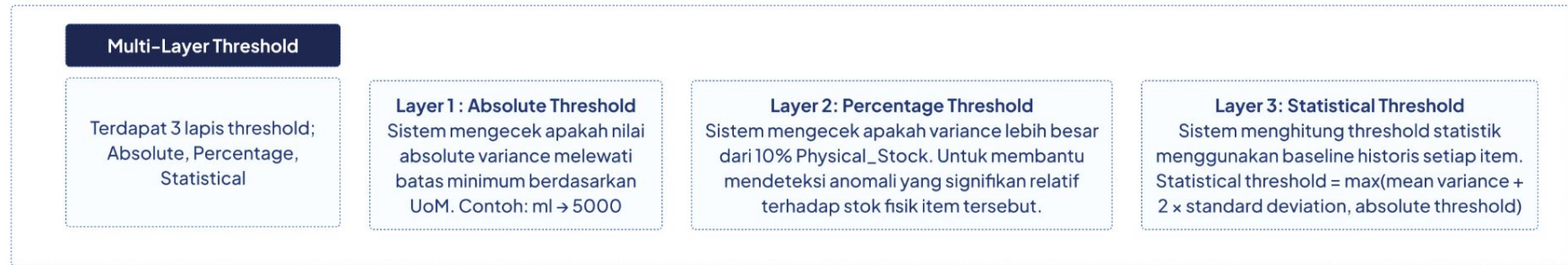
Anomaly Detection: Membedakan Selisih Wajar dan Anomali Inventaris

BeanGuard memakai threshold berlapis agar tidak semua selisih langsung dianggap anomali.

Proses Logic Anomaly Detection



Penjelasan Tahapan Proses



Label Assignment: Turning Calculation Results into Business Actions

Setiap item diklasifikasikan ke dalam satu status akhir berdasarkan prioritas aturan bisnis.

Proses Logic Anomaly Detection



Label Assignment: Turning Calculation Results into Business Actions

Setiap item diklasifikasikan ke dalam satu status akhir berdasarkan prioritas aturan bisnis.

Proses Logic Anomaly Detection



Penjelasan Tahapan Proses



Output Utama: Action_Report.csv

BeanGuard menghasilkan laporan aksi yang langsung menunjukkan status setiap item.

BeanGuard.

Contoh Hasil Output

Date	Item_ID	Action_Status
2025-01-02	INV-0001	Anomaly
2025-01-02	INV-0002	Safe
2025-01-02	INV-0007	Safe

Action_Report.csv merangkum keputusan akhir untuk setiap item inventaris.

Output ini dibuat sederhana agar pemilik bisnis dapat langsung melihat item mana yang aman, perlu restock, bermasalah secara data, atau membutuhkan investigasi anomali.

Output Detail untuk Audit dan Dashboard

Selain laporan utama, BeanGuard menyediakan detail variance dan anomaly reason agar hasil keputusan dapat ditelusuri.

Contoh Hasil Output

Date	Item_ID	Action_Status	Physical_Stock	Expected_Stock	Variance	Min_Stock_Threshold	Threshold_Converted	Supplier_UoM	Pcs_Per_Karton	is_anomaly	anomaly_reason	data_quality_flag
2025-01-02	INV-0001	Anomaly	32846.1	20223.9	12622.2	8	8000	Kilogram		TRUE	Variance > 1,000 unit	NORMAL
2025-01-02	INV-0002	Safe	24000.0	24000.0	0.0	6	6000	Kilogram		FALSE	Normal variance	NORMAL
2025-01-02	INV-0006	Safe	40000.0	40000.0	0.0	10	10000	Liter		FALSE	Normal variance	NORMAL

Tabel detail ini menjelaskan alasan di balik setiap Action_Status.

INV-0001 ditandai sebagai Anomaly karena variance mencapai 12.622,2 unit dan melewati batas anomali.

Sebaliknya, INV-0002 dan INV-0007 berstatus Safe karena stok fisik sesuai dengan expected stock serta masih berada di atas minimum threshold setelah konversi satuan.

Impact: Faster, Cleaner, and More Actionable Inventory Control

BeanGuard.

BeanGuard membantu UMKM mengurangi proses manual, mempercepat deteksi masalah stok, dan meningkatkan akurasi keputusan operasional.

Our Values

BeanGuard bukan hanya pipeline ETL, tetapi juga inventory decision engine yang ringan, yang menggabungkan normalisasi satuan ukur, pemetaan BOM, dan deteksi anomali untuk UMKM F&B.

Faster

Mengotomatisasi rekonsiliasi stok dari data POS, gudang, dan inventaris yang terfragmentasi dalam satu siklus pipeline.

Cleaner

Memvalidasi data kotor, ketidaksesuaian satuan ukur, dan referensi yang tidak valid sebelum proses perhitungan agar keputusan tidak dibuat dari data yang tidak reliabel.

Actionable

Mengubah data yang sudah dibersihkan dan direkonsiliasi menjadi tindakan inventaris yang jelas: Safe, Restock, Anomaly, dan Invalid Data.



Thank You!

Any Questions?