UJIAN TENGAH SEMESTER

MATA KULIAH : PENAMBANGAN DATA

KODE KULIAH : IF25-32025

NAMA ANGGOTA KELOMPOK

REYNALDI CRISTIAN SIMAMORA 122140116

AKHWAN ADIB AL-HAKIM 122140149

---

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi yang pesat pada era digital saat ini telah menyebabkan meningkatnya volume, kecepatan, dan keragaman data yang dihasilkan oleh berbagai sektor kehidupan. Data kini menjadi salah satu aset terpenting bagi organisasi dan perusahaan, karena di dalamnya terkandung informasi yang dapat digunakan untuk mendukung proses pengambilan keputusan. Namun, data yang melimpah tidak akan memiliki makna apabila tidak diolah dan dianalisis secara tepat. Oleh karena itu, muncul kebutuhan akan suatu metode yang mampu mengekstraksi pengetahuan dari data dalam jumlah besar secara efisien dan efektif. Proses inilah yang dikenal dengan istilah **data mining**.

Secara umum, **data mining** adalah proses menemukan pola, hubungan, atau informasi tersembunyi dari kumpulan data besar dengan menggunakan teknik statistik, kecerdasan buatan, dan pembelajaran mesin (*machine learning*). Data mining merupakan tahap inti dari proses yang lebih luas, yaitu **Knowledge Discovery in Database (KDD)**. KDD terdiri dari beberapa tahapan penting, yaitu: (1) *data selection* atau pemilihan data yang relevan, (2) *data preprocessing* atau pembersihan data dari kesalahan dan inkonsistensi, (3) *data transformation* untuk mengubah data menjadi format yang sesuai, (4) *data mining* sebagai tahap penerapan algoritma untuk menemukan pola, serta (5) *interpretation/evaluation* yaitu tahap interpretasi hasil agar menjadi pengetahuan yang bermakna. Dengan demikian, data mining tidak hanya berfokus pada pengolahan data, tetapi juga mencakup keseluruhan proses konversi data mentah menjadi pengetahuan yang bernilai.

Secara teoritis, data mining mengintegrasikan berbagai disiplin ilmu seperti *machine learning*, *pattern recognition*, statistik, dan sistem basis data. Beberapa teknik utama dalam data mining meliputi **klasifikasi (classification)** yang bertujuan mengelompokkan data ke dalam kategori tertentu, **klastering (clustering)** yang mengelompokkan data berdasarkan kemiripan tanpa label, **asosiasi (association rule mining)** yang menemukan hubungan antar-item, serta **prediksi (prediction)** yang memperkirakan nilai atau kejadian masa depan berdasarkan data historis. Selain itu, terdapat pula teknik **anomali detection (outlier detection)** yang berfungsi untuk menemukan data yang tidak sesuai dengan pola umum dalam dataset.

Dalam konteks data mining, **deteksi anomali (anomaly detection)** memiliki peranan penting karena mampu mengidentifikasi data atau kejadian yang menyimpang secara signifikan dari perilaku normal. Data anomali sering kali menunjukkan adanya potensi masalah, kesalahan pencatatan, penyimpangan proses, atau bahkan indikasi kecurangan (fraud). Secara umum, deteksi anomali dapat didefinisikan sebagai proses mengenali observasi atau pola data yang tidak sesuai dengan ekspektasi umum dari

suatu sistem. Penerapan deteksi anomali sangat luas, di antaranya dalam bidang keuangan untuk mendeteksi transaksi mencurigakan, dalam bidang kesehatan untuk menemukan tanda-tanda penyakit langka, dalam keamanan jaringan untuk mendeteksi serangan siber, serta dalam bidang penjualan atau persediaan untuk menemukan pola stok yang tidak wajar.

Metode deteksi anomali dapat diklasifikasikan menjadi beberapa pendekatan, antara lain pendekatan **statistik**, **berbasis jarak (distance-based)**, **berbasis densitas (density-based)** seperti *Local Outlier Factor (LOF)*, serta pendekatan **berbasis pembelajaran mesin (machine learning)** seperti *Isolation Forest* dan *Autoencoder*. Pemilihan metode yang tepat sangat bergantung pada karakteristik data dan tujuan analisis. Dengan menerapkan metode deteksi anomali yang sesuai, organisasi dapat mengidentifikasi penyimpangan lebih awal, meningkatkan keandalan sistem, serta mengoptimalkan pengambilan keputusan berbasis data.

Secara keseluruhan, data mining, khususnya dalam konteks deteksi anomali, memiliki peranan strategis dalam pengolahan data modern. Melalui analisis mendalam terhadap data transaksi atau operasional, organisasi dapat menemukan pola tersembunyi yang selama ini tidak terlihat, sekaligus mendeteksi ketidakwajaran yang dapat berdampak pada kinerja maupun keamanan sistem. Oleh karena itu, penerapan data mining tidak hanya menjadi alat bantu analisis, tetapi juga bagian integral dari strategi transformasi digital dan manajemen pengetahuan di berbagai bidang.

---

2. Percobaan Analisa Dataset

Dalam percobaan analisis dataset transaksi penjualan, pembelian, dan stok obat, pemilihan metode analisis yang tepat sangat penting untuk memperoleh informasi yang akurat dan relevan. Salah satu metode yang digunakan dalam penelitian ini adalah **Anomaly Detection** atau **Deteksi Anomali**. Metode ini dipilih karena memiliki kemampuan untuk mengidentifikasi pola-pola data yang tidak wajar atau menyimpang dari perilaku normal sistem.

Pada konteks **data transaksi obat**, setiap entri data umumnya merepresentasikan kegiatan operasional seperti pembelian dari pemasok, penjualan kepada konsumen, serta perubahan stok yang terjadi di gudang atau apotek. Secara ideal, setiap proses tersebut mengikuti pola tertentu, misalnya jumlah pembelian yang sesuai dengan kebutuhan stok, harga jual yang stabil, dan pergerakan barang yang konsisten dengan frekuensi transaksi. Namun, dalam praktiknya, data transaksi sering kali mengandung berbagai ketidakwajaran, baik yang disebabkan oleh **kesalahan pencatatan (human error)**, **ketidaksesuaian sistem**, maupun **indikasi kecurangan (fraudulent behavior)**. Contoh bentuk anomali tersebut antara lain:

- Nilai stok yang negatif atau jauh berbeda dari rata-rata normal.
- Transaksi penjualan dengan harga atau jumlah yang tidak sesuai dengan pola historis.
- Data pembelian dengan kuantitas sangat besar atau sangat kecil dibandingkan periode sebelumnya.
- Pencatatan transaksi ganda atau tanggal yang tidak konsisten dengan urutan logis operasional.

Dengan menerapkan metode **deteksi anomali**, pola-pola data yang menyimpang tersebut dapat diidentifikasi secara sistematis. Pendekatan ini tidak hanya membantu dalam menemukan kesalahan teknis pada sistem pencatatan data, tetapi juga berguna untuk **mendeteksi potensi kebocoran stok, kesalahan distribusi, atau praktik manipulatif** yang mungkin terjadi dalam proses bisnis apotek maupun distributor obat.

Preprocessing

Sebelum melakukan analisis dataset, perlu dilakukan pembersihan dataset. Tujuannya agar dataset dapat diformat dalam csv, dan spasi kosong yang besar pada dataset bisa ternormalisasi menjadi baik. Seluruh proses dalam kode ini bertujuan untuk mengubah data transaksi pembelian obat yang semula tidak terstruktur menjadi dataset terorganisasi yang siap dianalisis. Pada tahap awal, sistem memastikan bahwa file input tersedia dan dapat diakses, kemudian membaca data mentah menggunakan modul csv. Setiap baris dinormalisasi agar memiliki sembilan kolom tetap, mencakup informasi produk, tanggal, nomor transaksi, serta nilai dan kuantitas barang masuk maupun keluar. Setelah itu, data dikonversi kembali ke format teks menyerupai laporan pembelian asli agar sesuai dengan pola yang akan diproses menggunakan ekspresi reguler (regex). Tahap berikutnya mendefinisikan dua pola utama: satu untuk mendeteksi header produk dan satu lagi untuk mendeteksi baris transaksi individual, sementara fungsi clean_num() digunakan untuk menstandarkan format angka agar dapat dikonversi menjadi nilai numerik yang valid. Proses parsing kemudian dilakukan dengan membaca setiap baris dan mengidentifikasi apakah baris tersebut merupakan informasi produk, total kuantitas, atau transaksi individu. Data yang berhasil dikenali diklasifikasikan menjadi dua bagian utama — transaksi harian dan total kuantitas per produk. Kedua hasil tersebut disimpan ke dalam dua DataFrame, df_transaksi dan df_total, yang kemudian diekspor ke dalam file CSV terpisah. Dengan demikian, proses ini memastikan bahwa data pembelian obat yang awalnya tidak terstruktur dapat diproses secara sistematis dan siap digunakan untuk tahap analisis lanjutan seperti deteksi anomali, pemantauan stok, serta evaluasi pola pembelian dan penjualan obat secara efisien.`

In [ ]:
```python
#Parsing Dataset Pembelian → CSV Bersih

import os, re, csv, pandas as pd

file_path = "pembelian_transaksi.csv"
if not os.path.exists(file_path):
    raise FileNotFoundError(f"{file_path} not found in {os.getcwd()}")

lines, csv_rows = [], []

# Baca CSV & normalisasi kolom
with open(file_path, "r", encoding="utf-8", newline="") as _f:
    reader = csv.reader(_f)
    next(reader, None)
    for row in reader:
        row += [""] * (9 - len(row))
        kode, nama, unit, tanggal, no_trx, qty_masuk, nilai_masuk, qty_keluar, nilai_keluar
        csv_rows.append({
            "kode": kode.strip(), "nama_produk": nama.strip(), "unit": unit.strip(),
            "tanggal": tanggal.strip(), "no_transaksi": no_trx.strip(),
            "qty_masuk": qty_masuk.strip(), "nilai_masuk": nilai_masuk.strip(),
            "qty_keluar": qty_keluar.strip(), "nilai_keluar": nilai_keluar.strip()
        })

# Bentuk ulang ke format teks seperti laporan
KELUAR_SPACE_THRESHOLD = 18
current_kode = None
for r in csv_rows:
    if r["kode"] != current_kode:
        lines.append(f"{r['kode']} {r['nama_produk']} {r['unit']}")
        current_kode = r["kode"]
    if r["qty_keluar"]:
        ws, qty, nilai = " " * KELUAR_SPACE_THRESHOLD, r["qty_keluar"], r["nilai_keluar"]
    else:
        ws, qty, nilai = " ", r["qty_masuk"], r["nilai_masuk"]
    if r["tanggal"] and r["no_transaksi"]:
        lines.append(f"{r['tanggal']} {r['no_transaksi']}{ws}{qty} {nilai}".rstrip())

# Regex & pembersih angka
```

```
pattern_header_produk = re.compile(r"^([A-Z0-9]+)\s+([A-Za-z0-9\s.\-]+?)\s+([A-Z]{2,5})\s*$
pattern_transaksi = re.compile(r"^\s*(\d{2}-\d{2}-\d{2})\s+([\d.\-]+)(\s+)([\d,.]+)\s+([\d,

def clean_num(x):
    if not x: return None
    x = x.replace(".", "").replace(",", ".")
    try: return float(x)
    except: return None

# Parsing baris teks → data transaksi
transaksi_data, total_data, current_product = [], [], None

for line in lines:
    mh = pattern_header_produk.match(line)
    if mh:
        current_product = {"kode": mh[1].strip(), "nama_produk": mh[2].strip(), "unit": mh[
        continue
    if not current_product or re.match(r"^\s*-{5,}\s*$", line): continue

    if re.match(r"^\s+[\d,]+\s+[\d,]+\s*$", line):
        nums = re.findall(r"[\d,]+", line)
        if len(nums) >= 2:
            total_data.append({
                "kode": current_product["kode"], "nama_produk": current_product["nama_produ
                "unit": current_product["unit"], "total_qty_masuk": clean_num(nums[0]),
                "total_qty_keluar": clean_num(nums[1])
            })
        continue

    mt = pattern_transaksi.match(line)
    if mt:
        tanggal, no_trx, ws, qty, nilai = mt.groups()
        is_keluar = len(ws) >= KELUAR_SPACE_THRESHOLD
        qty, nilai = clean_num(qty), clean_num(nilai)
        transaksi_data.append({
            "kode": current_product["kode"], "nama_produk": current_product["nama_produk"],
            "unit": current_product["unit"], "tanggal": tanggal, "no_transaksi": no_trx,
            "qty_masuk": None if is_keluar else qty, "nilai_masuk": None if is_keluar else
            "qty_keluar": qty if is_keluar else None, "nilai_keluar": nilai if is_keluar el
        })

# Simpan hasil ke CSV
df_transaksi = pd.DataFrame(transaksi_data)
df_total = pd.DataFrame(total_data)
df_transaksi.to_csv("transaksi_clean.csv", index=False)
df_total.to_csv("transaksi_total.csv", index=False)

# 6️⃣ Ringkasan
print(f"✅ {len(df_transaksi)} baris transaksi & {len(df_total)} baris total disimpan.")
display(df_transaksi.head())
```

```
✅ 138352 baris transaksi & 0 baris total disimpan.
```

	kode	nama_produk	unit	tanggal	no_transaksi	qty_masuk	nilai_masuk	qty_keluar	nilai_ke
0	A000001	ANATON TAB	STRIP	06-07-21	1.13-210706.0908-003	100.0	25200.0	NaN	
1	A000001	ANATON TAB	STRIP	12-07-21	2.6-210712.1519-097	NaN	NaN	10.0	300
2	A000001	ANATON TAB	STRIP	12-07-21	2.11-210712.1633-013	NaN	NaN	10.0	300
3	A000001	ANATON TAB	STRIP	12-07-21	2.13-210712.1807-013	NaN	NaN	10.0	300
4	A000001	ANATON TAB	STRIP	12-07-21	2.11-210712.1855-018	NaN	NaN	10.0	300

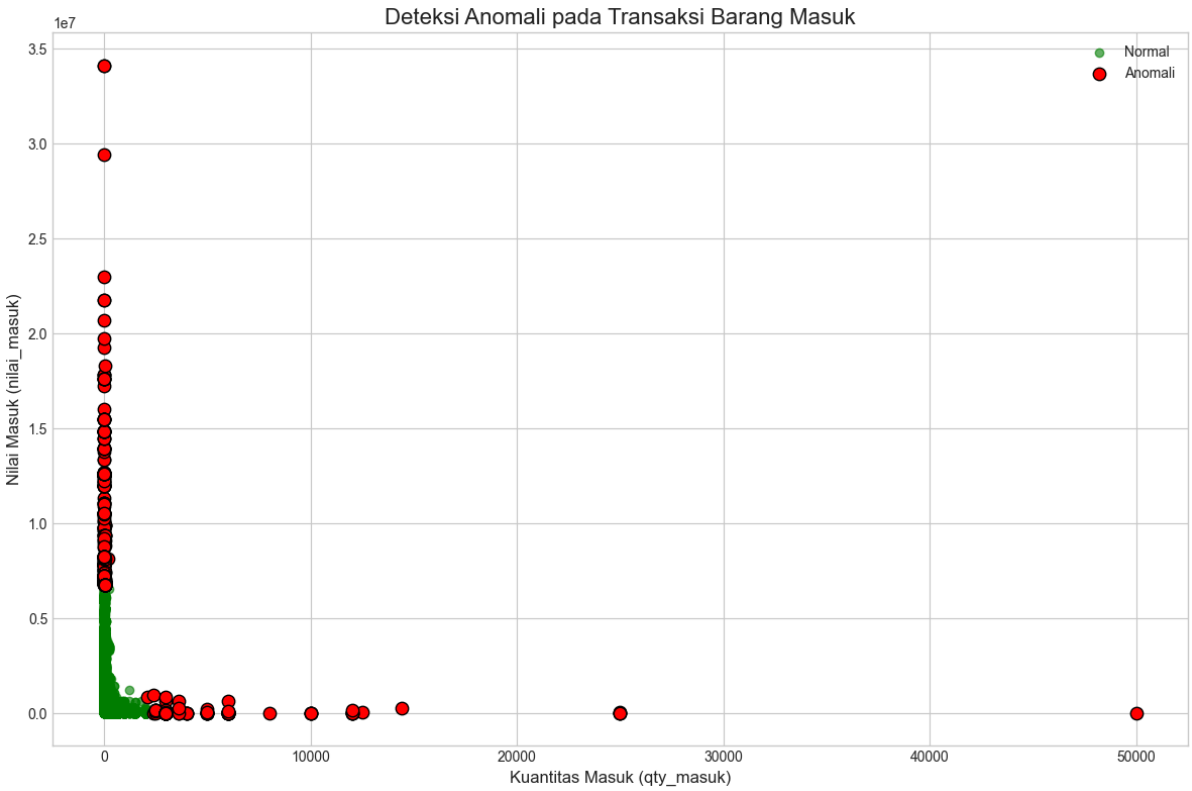
Implementasi Rule Based dan Isolation Forest

Bagian ini merupakan tahap deteksi anomali pada transaksi pembelian obat (barang masuk) menggunakan kombinasi dua pendekatan utama, yaitu machine learning dengan Isolation Forest dan metode Rule-Based Detection. Proses dimulai dengan memuat dataset hasil pre-processing, kemudian memisahkan transaksi yang memiliki nilai masuk untuk difokuskan pada analisis pembelian. Data numerik seperti kuantitas dan nilai transaksi dinormalisasi menggunakan StandardScaler agar memiliki skala yang sebanding sebelum diterapkan model Isolation Forest, yang berfungsi mendeteksi outlier berdasarkan distribusi data. Selain itu, pendekatan berbasis aturan diterapkan untuk menandai transaksi dengan nilai atau jumlah yang melebihi ambang batas tertentu (misalnya top 2% tertinggi atau di atas 10 juta). Hasil dari kedua metode tersebut kemudian dikombinasikan untuk mengidentifikasi transaksi yang benar-benar dianggap anomali. Proses ini diakhiri dengan visualisasi persebaran data dan penyimpanan hasil akhir ke file CSV agar dapat digunakan untuk analisis lebih lanjut.

In [ ]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import IsolationForest
import pandas as pd
import matplotlib.pyplot as plt

# Muat dataset hasil pre-processing
df = pd.read_csv("pembelian_transaksi.csv")
print(f"✅ Dataset awal dimuat. Jumlah baris: {len(df)}")

# Pisahkan data: fokus pada transaksi barang masuk
df_masuk = df[df['nilai_masuk'] > 0].copy()
if df_masuk.empty:
    raise ValueError("Tidak ada data transaksi masuk untuk dianalisis.")
print(f"✅ Data transaksi masuk dipisahkan. Jumlah baris: {len(df_masuk)}")

# Deteksi anomali dengan Isolation Forest
numeric_cols = ['qty_masuk', 'nilai_masuk']
X = df_masuk[numeric_cols].fillna(0)
X_scaled = StandardScaler().fit_transform(X)

iso_forest = IsolationForest(n_estimators=100, contamination=0.05, random_state=42)
iso_forest.fit(X_scaled)
df_masuk['iforest_anomaly'] = iso_forest.predict(X_scaled)
```

```python
df_masuk['iforest_anomaly'] = df_masuk['iforest_anomaly'].apply(lambda x: 1 if x == -1 else

# Deteksi berbasis aturan
def rule_based_anomaly(row):
    conditions = [
        row['nilai_masuk'] > 10_000_000,
        row['qty_masuk'] > df_masuk['qty_masuk'].quantile(0.98),
        row['nilai_masuk'] > df_masuk['nilai_masuk'].quantile(0.98)
    ]
    return 1 if any(conditions) else 0

df_masuk['rule_based_anomaly'] = df_masuk.apply(rule_based_anomaly, axis=1)

# Gabungkan kedua metode
df_masuk['final_anomaly'] = (
    (df_masuk['iforest_anomaly'] == 1) &
    (df_masuk['rule_based_anomaly'] == 1)
).astype(int)

print("\n✅ Deteksi anomali selesai.")
print(f"Isolation Forest: {df_masuk['iforest_anomaly'].sum()} anomali")
print(f"Rule Based: {df_masuk['rule_based_anomaly'].sum()} anomali")
print(f"Gabungan: {df_masuk['final_anomaly'].sum()} anomali")

# Visualisasi hasil
plt.style.use('seaborn-v0_8-whitegrid')
fig, ax = plt.subplots(figsize=(12, 8))

normal = df_masuk[df_masuk['final_anomaly'] == 0]
anomali = df_masuk[df_masuk['final_anomaly'] == 1]

ax.scatter(normal['qty_masuk'], normal['nilai_masuk'], c='green', alpha=0.6, label='Normal'
ax.scatter(anomali['qty_masuk'], anomali['nilai_masuk'], c='red', edgecolor='k', s=80, labe

ax.set_title('Deteksi Anomali pada Transaksi Barang Masuk', fontsize=16)
ax.set_xlabel('Kuantitas Masuk (qty_masuk)')
ax.set_ylabel('Nilai Masuk (nilai_masuk)')
ax.legend()
plt.tight_layout()
plt.show()

# Simpan hasil anomali
output_path = "hanya_anomali_barang_masuk.csv"
anomali.to_csv(output_path, index=False)
print(f"\n✅ Hasil deteksi disimpan ke: {output_path}")
```

```
✅ Dataset awal dimuat. Jumlah baris: 138352
✅ Data transaksi masuk dipisahkan. Jumlah baris: 15675

✅ Deteksi anomali pada data BARANG MASUK selesai.
Jumlah anomali (Isolation Forest): 784
Jumlah anomali (Rule Based): 471
Jumlah anomali (Gabungan): 460
```



✅ Hasil deteksi (hanya anomali barang masuk) disimpan ke: hanya_anomali_barang_masuk222.csv

Analisa Anomali Detection

Dalam proses deteksi anomali pada dataset transaksi pembelian obat, penggunaan parameter seperti contamination=0.05 pada Isolation Forest dan batas ambang atas (quantile 0.98 serta nilai transaksi > Rp10.000.000) pada metode rule-based didasarkan pada pertimbangan statistik dan konteks bisnis. Parameter contamination sebesar 5% digunakan untuk mengidentifikasi sekitar 5% data yang paling menyimpang dari pola umum, karena dalam praktik transaksi farmasi, kemungkinan data ekstrem atau salah input relatif kecil namun signifikan untuk diawasi. Ambang batas quantile 0.98 digunakan untuk mendeteksi transaksi dengan nilai dan kuantitas tertinggi yang termasuk 2% teratas, sementara batas nilai Rp10 juta dipilih karena transaksi di atas angka tersebut secara empiris jauh lebih jarang dan biasanya mencerminkan pembelian besar atau potensi kesalahan input.

Berdasarkan hasil analisis, anomali yang terdeteksi dapat dikategorikan menjadi dua jenis utama. Pertama, anomali akibat kesalahan input, yang ditandai oleh ketidakkonsistenan antara jumlah (qty) dan nilai (nilai_masuk) sehingga menghasilkan harga per unit yang tidak masuk akal, bahkan di bawah satu rupiah. Misalnya, kasus Amoxicillin 500mg dengan 10.000 strip senilai Rp29.504 menghasilkan harga Rp2,95 per strip — jauh di bawah harga normal ribuan rupiah. Demikian pula, Kana White 30gr dan Masker Karet Nian menunjukkan pola serupa, mengindikasikan kemungkinan kesalahan titik desimal atau salah kolom input. Kedua, anomali transaksi sah bernilai besar, yaitu transaksi dengan nilai tinggi namun rasional secara bisnis, seperti pembelian Appeton WG Adult Coklat, Blackmores Fish Oil, dan Minyak Kutus Kutus. Meskipun total transaksinya besar (mencapai puluhan juta rupiah), harga per unit masih masuk akal sesuai dengan karakteristik produk impor premium.

Dengan demikian, penggunaan parameter tersebut memungkinkan model membedakan antara anomali murni akibat kesalahan input data dan anomali yang wajar secara bisnis. Secara konseptual, pendekatan ini tidak hanya mengandalkan perhitungan statistik semata, tetapi juga mempertimbangkan logika bisnis farmasi dan konsistensi harga per unit. Hal ini penting agar sistem deteksi anomali tidak menghasilkan false positive pada transaksi sah, namun tetap sensitif terhadap kesalahan input yang dapat mempengaruhi validitas analisis stok dan laporan keuangan.

Kesimpulan

Berdasarkan hasil penerapan metode deteksi anomali menggunakan Isolation Forest dan Rule-Based Detection, dapat disimpulkan bahwa tidak semua data yang teridentifikasi sebagai anomali merupakan kesalahan. Dalam konteks transaksi pembelian obat, anomali dapat diklasifikasikan menjadi dua kategori utama, yaitu:

Anomali Akibat Kesalahan Input Data Jenis anomali ini muncul ketika terdapat ketidakkonsistenan antara jumlah (qty) dan nilai transaksi (nilai_masuk) sehingga menghasilkan harga per unit yang tidak masuk akal. Contohnya, pada produk Amoxicillin 500mg, Kana White 30gr, dan Masker Karet Nian, harga per unit tercatat jauh di bawah nilai wajar pasar (bahkan kurang dari Rp1). Hal ini mengindikasikan adanya kesalahan teknis pada proses input data, seperti penggunaan tanda desimal yang keliru, salah kolom entri, atau format konversi numerik yang tidak tepat. Anomali semacam ini harus segera diperiksa dan dikoreksi karena berpotensi memengaruhi keakuratan laporan stok dan nilai pembelian secara keseluruhan.

Transaksi Sah Bernilai Besar Berbeda dengan kesalahan input, kategori ini mencakup transaksi dengan nilai total tinggi namun tetap logis secara bisnis, di mana harga per unit masih sesuai dengan nilai pasar produk. Contohnya terdapat pada produk Appeton WG Adult Coklat, Blackmores Fish Oil, dan Minyak Kutus Kutus, yang memang memiliki harga tinggi karena merupakan produk impor premium. Transaksi seperti ini bukanlah kesalahan, melainkan pembelian sah bernilai besar yang mencerminkan aktivitas operasional normal dengan volume atau nilai barang yang tinggi.

Lampiran

Tautan Video : Youtube

Tautan Repositori Gitub : Github