

## 1. Problem understanding & Context definition.

### 1.1 Business Context

Perusahaan memiliki proses internal dalam penggantian biaya atau *reimbursement* yang memungkinkan karyawan mengajukan klaim atas pengeluaran terkait aktivitas bisnis, seperti penggunaan kartu kredit perusahaan, perjalanan dinas, konsumsi, dan kebutuhan operasional yang lainnya. Proses ini umumnya melibatkan pengajuan bukti transaksi, seperti struk atau nota pembelian atau dokumen perjalanan, kemudian diverifikasi oleh tim keuangan atau divisi terkait.

Seiring meningkatnya volume transaksi dan kompleksitas operasional, proses *reimbursement* menjadi rentan terhadap berbagai bentuk penyalahgunaan dan manipulasi. Contoh permasalahan yang sering terjadi antara lain penggunaan kartu kredit perusahaan untuk kepentingan pribadi, manipulasi pengajuan klaim, pemalsuan perjalanan dinas, serta praktik pengajuan biaya berulang. Permasalahan yang akan menimbulkan masalah finansial, tapi juga meningkatkan beban audit internal, serta menurunkan efektivitas operasional internal perusahaan.

### 1.2 Problem Statement

Permasalahan utama yang dihadapi perusahaan bukan se-simple adanya tindakan *fraud*, melainkan juga keterbatasan perusahaan dalam mendeteksi dan mencegah aktivitas *reimbursement* yang bersifat manipulatif atau *fraud* secara akurat, konsisten, dan dalam skala yang masif.

Pendekatan yang saat ini umum masih menggunakan pada aturan yang standar, seperti batas nominal pengeluaran tertentu, daftar *merchant* tertentu, atau pemeriksaan manual oleh tim terkait. Pendekatan ini memiliki beberapa kelemahan utama, antara lain sulit beradaptasi jika ada pola *fraud* yang baru, memerlukan upaya manual yang signifikan (*resources heavy*), serta kecenderungan bersifat reaktif. Akibatnya, banyak pola kecurangan yang bersifat halus dan bertahap yang tidak mudah terdeteksi.

### 1.3 Objectives

Dari POV bisnis dan rekayasa sistem, solusi yang diharapkan memiliki beberapa tujuan utama sebagai berikut:

#### A. Deteksi Dini:

Mengidentifikasi klaim *reimbursement* yang berpotensi *fraud* sebelum proses persetujuan atau pembayaran dilakukan.

B. Prioritasi Resiko:

Memberikan penilaian resiko terhadap klaim maupun pengguna, sehingga tim terkait dapat memfokuskan proses *review* pada kasus-kasus dengan resiko lebih tinggi.

C. Skalabilitas dan Konsistensi:

Meningkatkan kemampuan deteksi tanpa menambah beban kerja manual secara proporsional seiring bertambahnya volume transaksi.

D. Auditabilitas:

Menyediakan alasan atau indikator yang dapat dijelaskan atas setiap klaim yang ditandai, guna mendukung kebutuhan audit dan kepatuhan internal.

## 2. AI/ML Approach, Kebutuhan Data, dan Infrastruktur.

### 2.1 Strategies.

Permasalahan *fraud* dalam proses *reimbursement* memiliki karakteristik yang cukup kompleks. Tidak semua kasus *fraud* memiliki *pattern* yang jelas. Oleh karena itu, pendekatan yang mengandalkan satu metode AI/ML memiliki resiko tidak cukup *robust* dalam menghadapi variasi dan perkembangan pola *fraud*.

*Approach* yang diusulkan adalah arsitektur multi metode (*hybrid*) yang mengombinasikan:

1. *Supervised Learning* untuk pola fraud yang sudah diketahui
2. *Unsupervised* atau *Semi-supervised Learning* untuk mendeteksi anomali dan pola baru.
3. Analisis perilaku (*behavioral profiling*) user/pengguna.

Dengan pendekatan multi layer tersebut, memungkinkan sistem untuk menangani kasus *fraud* yang sudah eksplisit atau perilaku kecurangan yang bersifat halus dan bertahap.

### 2.2 Dekomposisi masalah menjadi task AI/ML.

Berdasarkan strategi pengambilan keputusan atas metode yang digunakan pada 2.1, permasalahan dapat dipecah dalam beberapa tugas spesifik AI/ML, yaitu:

#### 2.2.1 *Claim Risk Classification* (Klasifikasi Resiko Klaim)

Setiap klaim *reimbursement* dinilai berdasarkan probabilitas atau tingkat resiko fraud.

- *Methods: Supervised Classification*
- *Output:* Skor resiko atau kelas (Low, Medium, High)
- Manfaat: Mendukung keputusan approve, reject, atau manual review.

Pendekatan dengan model ini diharapkan efektif untuk mendeteksi pola *fraud* yang telah terjadi sebelumnya dan terdokumentasi.

#### 2.2.2 Deteksi Anomali dalam Transaksi (*anomaly detection*)

Digunakan untuk mendeteksi klaim atau transaksi yang menyimpang dari pola normal, meskipun belum pernah diklasifikasikan sebagai *fraud*.

- *Methods: Unsupervised / Semi-supervised learning.*
- *Output:* Skor anomali
- Manfaat: Menangkap pola *fraud* baru atau manipulasi yang belum dikenali.

*Approach* ini menjadi penting karena kecurangan bersifat adaptif terhadap aturan dan sistem yang ada.

#### 2.2.3 *User Behavioral Analysis*.

Tidak hanya melihat klaim secara individu, juga menganalisis pola perilaku pengguna dari waktu ke waktu.

- *Methods: Time-series analysis / behavioral modeling*
- *Output:* Skor risiko pengguna
- Manfaat: Mendeteksi penyalahgunaan yang bersifat bertahap atau sistematis

Contohnya adalah peningkatan frekuensi klaim tertentu atau pola perjalanan dinas yang tidak wajar.

### 2.3 *Proposed AI/ML Methods*

Dengan mempertimbangkan kebutuhan akurasi, interpretabilitas, dan efisiensi, metode berikut diusulkan sebagai baseline:

#### 2.3.1 *Supervised Learning*(Wang et al., 2025)

- *Logistic Regression* (baseline yang *interpretable*)
- *Gradient Boosting* (*XGBoost/LightGBM*)

- *Random Forest*

Alasan pemilihan:

- Bekerja baik pada data tabular
- Mendukung feature importance
- Mudah diintegrasikan ke sistem produksi

#### 2.3.2 *Unsupervised / Semi-Supervised Learning (Fariha et al., 2025; Kennedy et al., 2025)*

- *Isolation Forest*
- *One-Class SVM*
- *Autoencoder* (jika data besar dan kompleks)

Alasan pemilihan:

- Tidak bergantung pada label fraud lengkap
- Efektif untuk deteksi outlier

#### 2.3.3 Analisis Teks dan Dokumen

Jika tersedia data struk dalam bentuk teks atau gambar:

- OCR untuk ekstraksi teks struk
- NLP sederhana untuk mendeteksi pola manipulasi teks

### 2.4 *Dataset yang dibutuhkan*

#### 2.4.1 Transaksi Reimbursement

- ID klaim
- Tanggal transaksi dan pengajuan
- Jumlah klaim
- Jenis pengeluaran
- Merchant/vendor
- Metode pembayaran

#### 2.4.2 Data Pengguna

- ID Karyawan

- Departemen/Role
- Riwayat klaim
- Historis pengeluaran

#### 2.4.3 Data Label

- Status klaim (example: approve, rejected)
- Hasil historis audit (example: fraud/tidak fraud)
- Alasan penolakan/fraud

#### 2.4.4 Data Pendukung

- Kebijakan reimbursement
- Jadwal Perjalanan/Timeline pengeluaran
- Lokasi transaksi

### 2.5 *Feature Engineering*

Case fitur-fitur yang relevan:

1. Deviasi nominal klaim dari rata rata historis pengguna
2. Frekuensi klaim per periode
3. Pola waktu pengajuan (Jam, hari, bulan)
4. Konsistensi merchant/vendor dan kategori biaya
5. Rasio klaim yang direview atau ditolak sebelumnya

### 2.6 *Infrastruktur Minimum yang Diperlukan*

#### 2.6.1 Penyimpanan Data

- Relational Database (PostgreSQL/MySQL)
- Data warehouse sederhana (applicable jika volume data besar) seperti Redshift, atau Google BigQuery.

#### 2.6.2 Training Model Environment

- Server CPU/GPU (GPU akan bisa diutilize ketika hingga mengimplementasikan OCR/NLP/LLM)
- Python Framework (Tensorflow, Scikit-learn, Pandas, etc)

Sidenote: Dikarenakan model yang digunakan hybrid antara *Supervised* dan *Semi/Unsupervised Learning* direkomendasikan untuk menggunakan conda environment ketika training dan model development. Agar setiap task memiliki environment yang berbeda/separated jadi lebih mudah untuk di maintain dan dalam pengelolaan python framework dan library, terutama jika hingga implementasi OCR atau NLP. Conda Env akan lebih robust dan practical ketimbang venv.

#### 2.6.3 Inference dan Integrasi

- Batch inference harian atau *near real-time*
- REST API untuk scoring klaim
- Integrasi dengan sistem reimbursement yang sudah ada.

#### 2.6.4 Monitoring Dasar

- Logging input dan output model
- Tracking distribusi skor resiko
- Feedback dari hasil review manual

### 3. Arsitektur End-to-End dan MLOps Lifecycle.

#### 3.1 Prinsip design arsitektur

Ketika merancang arsitektur, beberapa prinsip perlu ditekankan,

1. *Human -in-the-loop* : Sistem AI berfungsi sebagai pendukung keputusan, bukan pengambil keputusan utama dan final
2. Modular dan bertahap: Setiap komponen dapat dikembangkan, di uji, dan di improve secara independen.
3. *Explainability* dan *Audability*: Setiap skor resiko atau flag harus ditelusuri kembali ke daya dan fitur yang mendasari keputusan tersebut.
4. Ground-up Requirement: Sistem dan development dapat dimulai dengan batch processing kemudian di tingkatkan ke near real-time jika dibutuhkan.

#### 3.2 End-to-end Architecture

##### 3.2.1 Data Source Layer

berisi seluruh sumber data yang relevan:

- Sistem reimbursement internal

- Sistem kartu kredit perusahaan
- Data perjalanan dinas
- Sistem HR (informasi karyawan)

Data berasal dari transaksi historis dan transaksi yang sedang berjalan.

### 3.2.2 Data Ingestion dan Storage Layer

Fungsi utama lapisan ini adalah mengumpulkan dan menyimpan data secara terstruktur.

- Ingestion:
  - Batch ingestion harian atau periodik
  - API atau ETL sederhana
- Storage:
  - Relational database untuk data operasional
  - Data warehouse untuk data historis dan analitik

Layer yang menjadi fundamental untuk training dan inference model.

### 3.2.3 Feature Engineering Layer

Layer ini untuk:

- Data Cleaning
- Menggabungkan data dari berbagai sumber
- Menghasilkan fitur numerik dan kategorikal

Contoh:

- Agregasi klaim per pengguna per periode
- Statistik historis (mean, median, deviasi)
- Fitur berbasis waktu

Untuk tahap awal, *feature engineering* dapat diimplementasikan sebagai batch jobs terjadwal.

### 3.2.4 Model training & eval layer

mencakup proses:

- Training model *supervised* dan *unsupervised*
- Validasi dan evaluasi performa model

- Seleksi model terbaik

Training dilakukan secara offline dengan data historis, dan hasilnya disimpan sebagai model versi tertentu.

### 3.2.5 Model Registry dan Versioning

Model yang telah dilatih dan divalidasi disimpan dalam *registry* untuk:

- *Versioning*
- *Rollback*
- *Reproducibility*

Setiap model terhubung dengan:

- Dataset versi tertentu
- *Feature schema*
- Metriks evaluasi

### 3.2.6 Inference dan Decision Support Layer

Model digunakan untuk melakukan scoring pada klaim baru:

- Batch Inference  
Scoring dilakukan secara periodik (misalnya setiap jam atau harian)
- Near Real-Time Inference (Opsional)  
Klaim diskor saat diajukan melalui API

Output:

- Skor risiko klaim
- Skor risiko pengguna
- Indikator anomali

Hasil ini diteruskan ke sistem *reimbursement* untuk mendukung keputusan.

### 3.2.7 Review dan Feedback Layer

Krusial untuk MLOps practices:

- Tim keuangan melakukan review klaim yang ditandai
- Keputusan akhir dicatat (valid / fraud / pelanggaran)
- Feedback ini dikumpulkan sebagai data label baru

Layer ini juga untuk menutup loop pembelajaran model.

Dalam pengembangan model, termasuk pendekatan arsitektur dan layer yang digunakan merupakan kombinasi dari *decision-centric AI Development*, *layered* atau *modular System Architecture*, *Human-in-the-loop ML System design*, *MLOps Lifecycle Development* dan *Agile Development for ML*. Dimana model AI/ML berfungsi sebagai pendukung pengambilan keputusan bisnis, bukan sebagai pengambil keputusan akhir. Dan penggunaan layer dalam development juga bertujuan agar mudah dikembangkan dan di maintain serta setiap layer dapat diubah tanpa merusak layer/sistem yang lain. Kemudian dengan *human-in-the-loop*, dipastikan bahwa keputusan bersifat akuntabel, dapat diaudit, dan adaptif terhadap perubahan pola *fraud*, serta pengembangan model ML/AI yang mengikuti model lifecycle, bukan one-time training, sehingga menjadi model AI yang *production level* bukan hanya eksperimen dalam notebook.

Dan *Agile for ML* digunakan untuk pengembangan secara iteratif dan inkremental untuk meminimalisasi resiko teknis serta memastikan nilai bisnis dapat divalidasi sejak tahap awal.

Reference:

- Fariha, N., Khan, M. N. M., Hossain, M. I., Reza, S. A., Bortty, J. C., Sultana, K. S., Jawad, M. S. I., Safat, S., Ahad, M. A., & Begum, M. (2025). Advanced fraud detection using machine learning models: enhancing financial transaction security. *International Journal of Accounting and Economics Studies*, 12(2), 85–104. <https://doi.org/10.14419/c73kcb17>
- Kennedy, R. K. L., Villanustre, F., & Khoshgoftaar, T. M. (2025). Unsupervised feature selection and class labeling for credit card fraud. *Journal of Big Data* 2025 12:1, 12(1), 111-. <https://doi.org/10.1186/S40537-025-01154-1>
- Wang, C., Nie, C., & Liu, Y. (2025). *Evaluating Supervised Learning Models for Fraud Detection: A Comparative Study of Classical and Deep Architectures on Imbalanced Transaction Data*. <https://arxiv.org/pdf/2505.22521>