BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian mengenai *fraud detection* telah berkembang pesat dengan memanfaatkan berbagai pendekatan *machine learning*, mulai dari metode klasifikasi tradisional hingga model berbasis *ensemble* dan *deep learning*. Dalam beberapa studi, algoritma klasifikasi seperti *Decision Tree*, *Logistic Regression*, dan *Random Forest* banyak digunakan karena kemampuannya dalam menangani data numerik maupun kategorikal serta kemudahan dalam interpretasi hasil model [37], [38], [39]. Namun, model-model tersebut memiliki keterbatasan dalam menangkap hubungan non-linear yang kompleks pada data transaksi berskala besar.

Keterbatasan tersebut mendorong penggunaan metode berbasis *ensemble*, salah satunya adalah Gradient Boosted Tree (GBT) yang terbukti mampu meningkatkan performa prediksi melalui proses *boosting* secara iteratif [40]. Beberapa penelitian menunjukkan bahwa GBT memiliki kemampuan yang lebih baik dalam menangani data dengan pola kompleks dan distribusi tidak seimbang dibandingkan model berbasis pohon tunggal [41][42].

Seiring meningkatnya kompleksitas model, kebutuhan akan interpretabilitas juga menjadi perhatian penting dalam penelitian *fraud detection*. Oleh karena itu, pendekatan Explainable Artificial Intelligence (XAI) mulai banyak digunakan untuk menjelaskan hasil prediksi model. Salah satu metode yang umum digunakan adalah SHAP (SHapley Additive exPlanations) yang mampu memberikan interpretasi terhadap kontribusi masing-masing fitur dalam proses pengambilan keputusan model [13]. Beberapa penelitian menunjukkan bahwa integrasi XAI dapat meningkatkan transparansi model serta membantu analis dalam memahami faktor penyebab terjadinya *fraud*.

Selain pendekatan berbasis tabular, penelitian terbaru juga mulai mengadopsi pendekatan berbasis graf untuk menangkap hubungan antar entitas dalam transaksi

keuangan. *Graph Neural Network* (GNN) digunakan untuk merepresentasikan relasi antara entitas seperti pengguna, *merchant*, dan transaksi dalam bentuk struktur *graph*, sehingga memungkinkan model untuk mengidentifikasi pola *fraud* yang bersifat relasional [43][44]. Pendekatan ini terbukti efektif dalam mendeteksi pola kecurangan yang tidak dapat ditangkap oleh model konvensional yang hanya berbasis fitur tabular.

Di sisi lain, perkembangan sistem terdistribusi juga mendorong munculnya pendekatan *Federated Learning* (FL) yang memungkinkan model dilatih pada beberapa sumber data secara terpisah tanpa perlu menggabungkan data secara langsung [13]. Pendekatan ini menjadi relevan dalam konteks *fraud detection* karena mempertimbangkan aspek privasi dan keamanan data [21]. Beberapa penelitian menunjukkan bahwa FL dapat menghasilkan performa yang kompetitif dibandingkan pendekatan terpusat, meskipun masih menghadapi tantangan dalam hal sinkronisasi model dan distribusi data [45].

Meskipun berbagai pendekatan tersebut telah menunjukkan potensi dalam meningkatkan performa *fraud detection*, penelitian yang mengintegrasikan beberapa pendekatan sekaligus dalam satu kerangka kerja berbasis Big Data masih terbatas. Selain itu, tidak semua penelitian menghubungkan proses pengembangan model dengan implementasi sistem secara *end-to-end*, termasuk tahap *deployment* dan konsistensi *pipeline* antara pelatihan dan inferensi.

Berdasarkan hal tersebut, penelitian ini berfokus pada pengembangan dan evaluasi model *fraud detection* berbasis Gradient Boosted Tree dengan pendekatan *Explainable AI* menggunakan SHAP, serta dibandingkan dengan pendekatan alternatif seperti *Graph Neural Network* dan *Federated Learning*. Selain itu, penelitian ini juga menekankan integrasi *pipeline* dari tahap *preprocessing* hingga *deployment* berbasis Streamlit, sehingga memberikan kontribusi tidak hanya pada aspek performa model, tetapi juga pada aspek implementasi sistem yang lebih praktis dan terintegrasi.

Tabel 2.1 Penelitian Terdahulu terkait Machine Learning terhadap Fraud Detection

No.	Judul Penelitian	Nama Jurnal, Penulis & Tahun	Metode Penelitian	Temuan Utama	Relevansi terhadap Proyek
1.	Deteksi Fraud Kartu Kredit dengan Logistic Regression, Random Forest, dan Gradient *Boosting*	Jurnal Informatika Terpadu, Herlambang Awan Irawan (2025)	*Logistic Regression, Random Forest, Gradient Boosting Classifier*, integrasi SMOTE dan PCA	*Random Forest* dan *Gradient Boosting* memberikan performa terbaik dengan skor ROC-AUC > 0,999. Fitur V14 ditemukan sebagai variabel paling berpengaruh.	Penggunaan metode ensemble learning dan teknik penanganan ketidakseimbangan data (SMOTE) [6].
2.	*Multimodal detection framework for financial fraud integrating LLMs and interpretable machine learning*	*Journal of Data and Information Science*, Nie, dkk. (2025)	Penggabungan data finansial, tata kelola, dan tekstual (laporan tahunan) menggunakan LLM (*Doubao*), algoritma GBDT, dan analisis SHAP	Fusi multimodal meningkatkan performa sebesar 7,4%. LLM efektif dalam menyarikan laporan panjang menjadi fitur semantik yang prediktif.	Integrasi data tidak terstruktur (teks) dan penggunaan AI yang dapat diinterpretasikan (*SHAP*) untuk transparansi [46].
3.	*Leveraging Blockchain and Federated Reinforcement Learning for Enhanced*	JITSI: Jurnal Ilmiah Teknologi Sistem Informasi,	*Federated Reinforcement Learning* (FRL) dikombinasikan dengan teknologi	Meningkatkan keamanan data melalui desentralisasi (data tetap di sisi klien) dan	Fokus pada privasi data pengguna dan keamanan dalam lingkungan transaksi finansial

	Fraud Detection in Financial Transactions	Tanweer Alam (2025)	*Blockchain* dan *Smart Contracts.*	memastikan integritas data melalui buku besar *blockchain* yang tidak dapat diubah.	yang terdistribusi [36].
4.	*Deep reinforcement learning-based heterogeneous Graph Neural Networks for multi-task financial fraud detection*	*Journal of Combinatorial Mathematics and Combinatorial Computing*, Xiaohan Du (2025)	*Heterogeneous Graph Neural Networks* (H-GNN), *Deep Reinforcement Learning*, dan *Variational Auto-Encoder* (VAE).	GVAE lebih stabil dan unggul dalam identifikasi transaksi fraud dibandingkan auto-encoder tradisional (AUC 0,925).	Efektif dalam menangkap hubungan kompleks antar entitas dan memproses data non-homogen dalam skenario fraud [35].
5.	*FinGuard-GNN: Dynamic Graph Neural Network Framework for Financial Fraud Detection*	*Frontiers in Business, Economics and Management*, Ruijie Huang (2025)	*Dynamic Graph Neural Network* dengan pengkodean ATP & SEW serta mekanisme *Cascaded Risk Diffusion* (CRD).	Berhasil menangani atribut *non-additive* dan memodelkan propagasi risiko hierarkis dalam jaringan transaksi (AUC 98,1%).	Solusi untuk keterbatasan algoritme grafik tradisional dalam menangkap pola pengaruh antar tipe *node* yang berbeda [34].
6.	*Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for*	*Journal of Risk and Financial Management*, Aljunaid, S. K., Almheiri, S. J., Dawood, H., & Khan, M. A. (2025)	Model Explainable *Federated Learning* (XFL) yang mengintegrasikan *Federated Learning* (FL)	Model mencapai akurasi 99,95% dan *miss rate* hanya 0,05%. Algoritma *Gradient Boosting*	Menyediakan kerangka kerja deteksi *fraud* perbankan yang menyeimbangkan antara akurasi tinggi, privasi data nasabah, dan

	Financial Fraud Detection		untuk privasi data terdistribusi dengan *Explainable AI* (XAI) menggunakan teknik SHAP dan LIME.	*Machine* (GBM) terbukti paling unggul dibandingkan SVM dan *Logistic Regression*. Penggunaan XAI memberikan transparansi yang memungkinkan analis memahami alasan di balik prediksi *fraud*.	akuntabilitas keputusan model melalui interpretasi AI [13].
7.	*Developing advanced data science and AI models to mitigate and prevent financial fraud in real-time systems*	*World Journal of Advanced Engineering Technology and Sciences*, Fatunmbi (2024)	Tinjauan sistematis metode ML, *Deep learning* (CNN, RNN, LSTM), GAN untuk data sintetis, dan XAI (SHAP, LIME).	AI meningkatkan akurasi deteksi dan mengurangi *false positives* dibandingkan sistem berbasis aturan (*rule-based*) tradisional.	Memberikan peta jalan strategis untuk mitigasi fraud *real-time* dan konvergensi AI dengan keamanan siber [47].
8.	*Credit Card Fraud Detection Through Explainable Artificial Intelligence*	*Indatu Journal of Management and Accounting,* Muksalmina, dkk. (2025)	Perbandingan *Logistic Regression, Naive Bayes, Decision Tree, Random Forest*,	*Random Forest* memiliki performa tertinggi (99,52%); SHAP membantu	Pentingnya interpretabilitas model (XAI) untuk akuntabilitas manajerial dan kepatuhan

	for Managerial Oversight		dan aplikasi SHAP.	manajer memahami alasan di balik prediksi fraud.	terhadap regulasi finansial [33].
9.	*Online Payment Fraud Detection*	IJRASET, Tulesh Soni (2025)	ML tersupervisi (*Random Forest, Gradient Boosting*), SMOTE, dan deteksi *real-time* menggunakan Apache Kafka.	Model *ensemble* memberikan keseimbangan terbaik antara akurasi dan efisiensi untuk sistem yang sensitif terhadap waktu.	Fokus pada penguatan keamanan ekosistem pembayaran digital melalui analisis pola perilaku dan anomali [48].
10.	*Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models*	*Computational Economics*, Masad A. Alrasheedi (2025)	Perbandingan luas: SVM, *Decision Tree, Random Forest, XGBoost, AdaBoost, MLP, ANN*, dan PCA.	RF, XGBoost, dan MLP mencapai akurasi 0,99 pada data seimbang; RF tetap efektif pada data tidak seimbang (akurasi 0,97).	Menyediakan analisis komparatif algoritme lintas berbagai skenario dataset (Eropa dan Amerika Serikat) [32].

Berbagai pendekatan *machine learning* dan *deep learning* telah dilakukan seperti pada tabel 2.1 dan menunjukkan performa yang baik dalam deteksi *fraud* transaksi keuangan digital seperti Random Forest, Gradient *Boosting*, XGBoost, hingga *Graph Neural Network* (GNN). Beberapa penelitian juga memanfaatkan *framework* Big Data seperti PySpark dan Azure Databricks untuk mendukung pemrosesan data berskala besar. Selain itu, penggunaan Explainable Artificial Intelligence (XAI), khususnya SHAP semakin banyak diterapkan untuk meningkatkan interpretabilitas

model dan membantu analisis pola *fraud*. Penelitian lain turut mengeksplorasi pendekatan berbasis *graph*, *Federated Learning*, dan blockchain untuk mendukung analisis relasional, keamanan, serta privasi data pada sistem keuangan digital.

Meskipun demikian, penelitian sebelumnya masih memiliki beberapa keterbatasan seperti potensi *overfitting*, kurangnya keseimbangan antara performa klasifikasi dan interpretabilitas, tingginya kompleksitas komputasi pada pendekatan berbasis *graph*, serta tantangan sinkronisasi pada sistem terdistribusi. Selain itu, beberapa model masih mengalami kesulitan dalam mendeteksi kelas *fraud* minoritas dan belum sepenuhnya merepresentasikan kondisi transaksi nyata.

Berdasarkan hal tersebut, penelitian ini mengembangkan sistem deteksi *fraud* menggunakan Gradient Boosted Tree (GBT) yang didukung metode SHAP untuk meningkatkan interpretabilitas model. Penelitian ini juga mengeksplorasi pendekatan *Graph Neural Network* (GNN) dan *Federated Learning* (FL) dalam analisis transaksi keuangan digital berskala besar dengan dukungan *framework* PySpark sebagai alat pendukung untuk pemrosesan data secara *scalable*. Pendekatan yang diusulkan diharapkan mampu memberikan keseimbangan antara performa, interpretabilitas, skalabilitas, dan kesiapan implementasi sistem deteksi *fraud*.

2.2 Teori yang berkaitan

2.2.1 *Fraud Detection* dan *Machine Learning*

Fraud detection merupakan suatu proses identifikasi aktivitas transaksi yang bersifat mencurigakan atau tidak sah dengan tujuan untuk meminimalkan potensi kerugian finansial [49]. Dalam sistem keuangan digital modern, volume transaksi yang sangat besar serta kecepatan pemrosesan yang tinggi menyebabkan pendekatan manual menjadi tidak efektif. Maka, diperlukan pendekatan berbasis komputasi yang mampu mendeteksi pola secara otomatis dan akurat. *Machine learning* dalam *fraud detection* umumnya digunakan dalam kerangka *supervised learning*, yaitu model dilatih menggunakan data historis yang telah memiliki label kelas seperti transaksi *fraud* dan *non-fraud*, sehingga model dapat mempelajari

karakteristik dari masing-masing kelas untuk melakukan prediksi terhadap transaksi baru [31], [49]. Keunggulan pendekatan *supervised learning* pada *fraud detection* adalah kemampuannya dalam menemukan pola dan hubungan kompleks pada data transaksi yang sulit ditentukan menggunakan pendekatan berbasis *threshold* manual [31], [32].

Namun, salah satu tantangan utama dalam pengembangan model *fraud detection* adalah permasalahan ketidakseimbangan data (*class imbalance*), yaitu kondisi ketika jumlah transaksi normal jauh lebih besar dibandingkan jumlah transaksi fraud [4], [50]. Ketidakseimbangan tersebut dapat menyebabkan model *machine learning* lebih cenderung mengenali kelas mayoritas sehingga kemampuan model dalam mendeteksi transaksi *fraud* sebagai kelas minoritas menjadi berkurang [4], [7]. Akibat kondisi tersebut, penggunaan akurasi sebagai satu-satunya metrik evaluasi dapat memberikan hasil yang kurang representatif, sehingga penelitian *fraud detection* umumnya menggunakan metrik tambahan seperti *precision*, *recall*, F1-score, dan AUC untuk mengevaluasi performa model secara lebih menyeluruh [39], [42], [51].

Selain kemampuan prediksi, penelitian *fraud detection* modern juga memberikan perhatian terhadap aspek interpretabilitas model, karena hasil prediksi dari sistem berbasis *machine learning* perlu dapat dijelaskan agar mendukung proses pengambilan keputusan [13], [20]. Dalam industri finansial, model dengan tingkat akurasi tinggi tetapi tidak memiliki kemampuan interpretasi dapat menjadi kendala dalam penerapan karena pengguna bisnis membutuhkan pemahaman mengenai faktor atau karakteristik transaksi yang menyebabkan suatu transaksi dikategorikan sebagai *fraud* [15], [33].

Secara umum, pendekatan *fraud detection* dapat diklasifikasikan menjadi:

1. **Rule-Based System**

   Menggunakan aturan tetap, contohnya dilakukan *threshold* jumlah transaksi.
2. **Machine Learning-Based System**

Menggunakan algoritma *machine learning* untuk mempelajari pola historis dan memprediksi kemungkinan *fraud*.

Permasalahan utama dalam *fraud detection* terletak pada karakteristik data yang cenderung tidak seimbang (*imbalanced*), di mana jumlah transaksi *non-fraud* jauh lebih besar dibandingkan transaksi *fraud* [50]. Kondisi ini menyebabkan model klasifikasi sering kali *bias* terhadap kelas mayoritas [50]. Selain itu, pola *fraud* yang terus berkembang dapat berubah seiring waktu menuntut sistem deteksi untuk memiliki kemampuan adaptasi dan skalabilitas yang baik [52].

Dalam penelitian ini, *machine learning* menjadi solusi yang relevan karena mampu mempelajari pola historis transaksi dan mengidentifikasi anomali berdasarkan karakteristik data [53]. Pendekatan *machine learning* dalam *fraud detection* umumnya dibagi menjadi *supervised learning* dan *unsupervised learning* [54]. *Supervised learning* digunakan ketika data memiliki label, sehingga model dapat dilatih untuk membedakan antara transaksi *fraud* dan *non-fraud* [54]. Sementara itu, *unsupervised learning* digunakan untuk menemukan pola tersembunyi atau segmentasi perilaku transaksi tanpa memerlukan label sebelumnya [54].

2.2.2 *Gradient Boosted Tree* (GBT)

Gradient Boosted Tree merupakan algoritma *ensemble* yang membangun model secara iteratif dengan menggabungkan beberapa *weak learners* untuk menghasilkan model yang lebih kuat [55]. GBT bekerja dengan meminimalkan fungsi *loss* melalui pendekatan *gradient boosting*. Algoritma ini memiliki keunggulan khusus dalam menangani *evolving data streams*, di mana model mampu beradaptasi secara efektif terhadap perubahan pola data yang terjadi secara berkelanjutan melalui proses pembaruan bobot pada setiap iterasi pelatihan [40]. Selain itu, GBT terbukti memberikan performa yang sangat tangguh pada dataset dengan distribusi tidak seimbang (*imbalanced data*), sehingga sangat relevan untuk menangkap pola kecurangan yang langka pada kasus deteksi *fraud* [42].

Dalam penelitian *fraud detection*, GBT banyak digunakan karena mampu menangkap hubungan non-linear antar fitur serta menghasilkan performa

klasifikasi yang baik pada permasalahan dengan pola transaksi yang kompleks [6], [32]. Dibandingkan dengan model pohon keputusan tunggal, pendekatan *ensemble* seperti GBT memiliki kemampuan yang lebih baik dalam meningkatkan stabilitas model dan menangani variasi data, termasuk pada kondisi distribusi kelas yang tidak seimbang [5], [42]. Model GBT juga dapat diterapkan pada dataset dengan berbagai tipe fitur setelah melalui proses *preprocessing* yang sesuai, sehingga memungkinkan penggunaannya pada data transaksi finansial yang memiliki karakteristik fitur numerik maupun kategorikal [6], [40].

Selain itu, performa GBT sangat dipengaruhi oleh proses pengaturan parameter model seperti *learning rate*, kedalaman pohon (*tree depth*), dan jumlah *estimator*, karena parameter tersebut menentukan kompleksitas serta kemampuan generalisasi model terhadap data baru [40], [42]. Kemampuan tersebut membuat GBT menjadi salah satu algoritma yang sesuai untuk penelitian *fraud detection* karena mampu memberikan keseimbangan antara performa klasifikasi, efisiensi komputasi, dan kestabilan model ketika digunakan pada data transaksi dalam jumlah besar [6], [39].

Walaupun memiliki performa prediksi yang kuat, model berbasis *ensemble* seperti GBT memiliki keterbatasan dalam interpretasi karena struktur model yang terdiri dari banyak pohon keputusan membuat proses pengambilan keputusan menjadi lebih sulit dibandingkan model sederhana seperti *decision tree* tunggal [16], [33]. Oleh karena itu, penelitian *fraud detection* modern mulai mengintegrasikan metode *Explainable Artificial Intelligence* (XAI) untuk memberikan penjelasan terhadap hasil prediksi model *machine learning* yang kompleks [13], [20]. Salah satu pendekatan yang banyak digunakan adalah SHAP (*Shapley Additive Explanations*) yang dapat membantu mengidentifikasi kontribusi setiap fitur terhadap keputusan model sehingga hasil prediksi *fraud* lebih mudah dipahami oleh pengguna non-teknis [15], [16].

Dalam penelitian ini, GBT dipilih sebagai model utama karena memiliki kemampuan klasifikasi yang kuat sekaligus dapat dikombinasikan dengan metode

SHAP untuk mendukung analisis interpretabilitas terhadap faktor-faktor yang memengaruhi hasil prediksi *fraud* [6], [15].

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x)$$

Gambar 2.1 Representasi Model GBT

Berikut keterangan variabel pada rumus tersebut:

1. $\boldsymbol{F(x)}$ **:** Model akhir atau hasil prediksi final yang dihasilkan dari penggabungan seluruh model yang telah dibangun.
2. $\boldsymbol{h_m}$ **:** Model pohon keputusan individu (sering disebut *weak learner*) yang dibangun pada setiap tahapan atau iterasi ke-*m*.
3. $\boldsymbol{\gamma_m}$ **:** Bobot tertentu yang diberikan kepada setiap model pohon h_m dalam proses penjumlahan untuk membentuk model akhir.

Pada setiap iterasi, model baru dibangun untuk meminimalkan *residual* dari model sebelumnya:

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]$$

Gambar 2.2 Rumus Perhitungan Residual

Berikut keterangan variabel pada rumus tersebut:

1. $\boldsymbol{r_i^{(m)}}$**:** Pseudo-residual atau nilai target untuk melatih pohon ke-*m* pada sampel data ke-*i*.
2. $\boldsymbol{L}$**:** Fungsi kerugian (*loss function*), contohnya *Mean Squared Error* untuk regresi atau *Log-Loss* untuk klasifikasi.
3. $\boldsymbol{y_i}$**:** Nilai aktual dari data ke-*i*.

Gradient Boosted Tree dipilih karena memiliki kemampuan yang baik dalam memodelkan hubungan non-linear pada data tabular dan sering digunakan pada permasalahan klasifikasi yang melibatkan pola kompleks [40], [42]. Dibandingkan dengan algoritma lain yang juga populer pada *fraud detection* seperti *Random Forest* dan *XGBoost*, GBT dipertimbangkan karena pendekatan *boosting* mampu

menggabungkan beberapa model pohon keputusan untuk meningkatkan kemampuan prediksi pada data kompleks [5], [6], [56]. Selain itu, pada penelitian ini GBT dipandang lebih sesuai untuk menjadi *baseline* model karena fokus utama penelitian adalah menguji efektivitas *feature engineering* dan skenario keterbatasan informasi, bukan hanya melakukan optimalisasi model yang kompleks [49].

Pemilihan *Gradient Boosted Tree* juga mempertimbangkan kemampuannya dalam menangani permasalahan klasifikasi berbasis data kompleks serta kemampuannya dalam membangun model prediktif melalui pendekatan *ensemble* berbasis pohon keputusan [40], [42]. Penggunaan kerangka komputasi terdistribusi seperti PySpark dapat mendukung proses pengolahan data berskala besar melalui kemampuan pemrosesan paralel dan integrasi *pipeline machine learning* [10]. Hal ini menjadikan proses pemodelan lebih efisien dalam menangani data berskala besar, sekaligus mendukung alur eksperimen yang memerlukan pengolahan fitur, pelatihan model, dan evaluasi secara sistematis pada lingkungan *big data* [10], [49]. Dengan demikian, pemilihan GBT didasarkan pada keseimbangan antara performa model, kemampuan menangkap pola data, dan kesesuaian dengan tujuan penelitian [40], [57].

2.2.3 *Explainable Artificial Intelligence* (SHAP)

SHAP *(Shapley Additive Explanations*) merupakan metode interpretabilitas yang digunakan untuk menjelaskan kontribusi setiap fitur terhadap prediksi model [13]. Metode ini bekerja berdasarkan konsep *Shapley value* dari teori permainan yang secara matematis menghitung kontribusi marginal setiap fitur saat bergabung ke dalam berbagai kombinasi subset fitur lainnya guna mencapai hasil akhir keputusan model [16]. Dalam penerapannya, SHAP tidak menghasilkan "skor interpretabilitas" yang bersifat subyektif, melainkan memberikan nilai penjelasan kontribusi (*contribution explanation values*) yang menunjukkan besaran pengaruh masing-masing atribut dalam keputusan klasifikasi [15], [16]. Pendekatan tersebut memungkinkan SHAP memberikan interpretasi secara global, yaitu memahami pola pengaruh fitur pada keseluruhan model, maupun secara lokal, yaitu

menjelaskan alasan di balik satu prediksi tertentu [15], [33]. Nilai SHAP untuk suatu fitur dapat dirumuskan sebagai:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

Gambar 2.3 Rumus Nilai Shapley

Berikut keterangan variabel pada rumus tersebut:

1. $\boldsymbol{\phi i}$ **:** Nilai Shapley untuk variabel/pemain ke-*i* (kontribusi atau nilai akhir yang menjadi haknya).
2. $\boldsymbol{N}$ **:** Himpunan seluruh variabel atau pemain yang ada dalam sistem (jumlah total = |N|).
3. $\boldsymbol{i}$ **:** Variabel spesifik yang sedang dihitung nilai kontribusinya.
4. $\boldsymbol{S}$ **:** Himpunan bagian (sub-koalisi) dari pemain yang bergabung *sebelum* pemain i (jumlah anggota = |S|).
5. $\boldsymbol{|N|!}$ **:** Seluruh pemain di luar pemain i.
6. $\boldsymbol{f(S)}$ **:** Nilai atau hasil dari kerja sama subset S (sebelum variabel i ditambahkan).
7. $\boldsymbol{f(S \cup \{i\})}$ **:** Nilai atau hasil kerja sama setelah variabel i ditambahkan ke dalam subset S.
8. $\boldsymbol{[f(S \cup \{i\}) - f(S)]}$ **:** Bagian ini disebut sebagai kontribusi marginal (tambahan nilai yang diberikan secara langsung oleh variabel i saat bergabung ke dalam kelompok S)

Dengan pendekatan ini, SHAP mampu memberikan interpretasi yang konsisten dan transparan terhadap keputusan model, sehingga memudahkan dalam memahami faktor-faktor yang mempengaruhi terjadinya *fraud* [15]. Selain itu, metode ini dapat diintegrasikan ke dalam arsitektur pembelajaran terdistribusi untuk menciptakan sistem Explainable *Federated Learning* (XFL) yang menjamin akuntabilitas pada operasional perbankan [15]. Penggunaan nilai Shapley juga terbukti efektif sebagai strategi pemilihan fitur (*feature selection*) guna mengoptimalkan performa model deteksi melalui identifikasi atribut yang paling berpengaruh [16].

Keunggulan SHAP adalah kemampuannya dalam menyediakan interpretasi yang dapat diterapkan pada berbagai model *machine learning*, sehingga metode ini banyak digunakan untuk meningkatkan transparansi pada model prediktif yang memiliki tingkat kompleksitas tinggi [13], [20]. Pada model berbasis pohon keputusan seperti *Gradient Boosted Trees* (GBT), SHAP dapat digunakan melalui pendekatan Tree SHAP yang memanfaatkan struktur pohon sehingga proses perhitungan kontribusi fitur dapat dilakukan secara lebih efisien dibandingkan pendekatan interpretasi umum [15], [16]. Dalam penelitian *fraud detection*, penggunaan SHAP berperan sebagai penghubung antara performa prediksi model GBT dengan kebutuhan interpretasi, sehingga hasil klasifikasi *fraud* dapat dijelaskan secara lebih mudah kepada pengguna bisnis maupun pihak terkait [13], [15], [33].

2.2.4 *Graph Neural Network* (GNN)

Graph Neural Network merupakan metode *deep learning* yang digunakan untuk memodelkan data dalam bentuk graf, di mana data direpresentasikan sebagai *node* dan *edge* [17]. Berbeda dengan model *machine learning* berbasis data tabular yang memproses data berdasarkan representasi fitur individual, *Graph Neural Network* (GNN) dirancang untuk memanfaatkan struktur graf yang terdiri dari *node* dan *edge* sehingga mampu mempelajari hubungan antar entitas dalam suatu data [17], [18]. Dalam konteks *fraud detection*, struktur graf dapat merepresentasikan hubungan seperti keterkaitan antar akun, transaksi, perangkat, maupun entitas lainnya sehingga pola hubungan yang kompleks dapat dianalisis oleh model [18], [19]. Mekanisme utama pada GNN adalah *message passing*, yaitu proses ketika setiap *node* mengumpulkan dan menggabungkan informasi dari node tetangganya untuk menghasilkan representasi *node* baru yang lebih informatif [17]. Melalui mekanisme tersebut, GNN mampu menangkap pola struktural dan hubungan antar entitas yang sulit diperoleh menggunakan model *machine learning* berbasis fitur tabular konvensional [17], [19]. Proses pembaruan representasi *node* pada GNN dapat dirumuskan sebagai:

$$h_v^{(k+1)} = \sigma\left(\sum_{u \in N(v)} W^{(k)} h_u^{(k)}\right)$$

Gambar 2.4 Rumus *Node* GNN

Berikut keterangan variabel pada rumus tersebut:

1. $\boldsymbol{h_v^k}$ **:** Representasi *node* (vektor fitur) pada lapisan (layer) ke-*k*.
2. $\boldsymbol{N(v)}$**:** Himpunan tetangga (*neighbors*) dari *node* tersebut. Variabel ini menunjukkan dari mana saja informasi dikumpulkan atau diagregasi dalam struktur graf.
3. $\boldsymbol{W^k}$ **:** Parameter model atau bobot yang dipelajari oleh sistem pada lapisan ke-*k* untuk mengolah informasi fitur.
4. $\boldsymbol{\sigma}$ **:** Fungsi aktivasi seperti ReLU atau Sigmoid yang memberikan sifat non-linear pada pemrosesan data.

Pendekatan ini memungkinkan model untuk memahami pola hubungan yang kompleks yang sering kali tidak dapat ditangkap oleh model berbasis fitur tabular [18]. Melalui penggabungan informasi dari *node* tetangga, GNN mampu menangkap pola relasional yang kompleks dalam data transaksi untuk meningkatkan akurasi deteksi anomali [18]. Meskipun memiliki kemampuan dalam memodelkan hubungan kompleks, GNN memiliki beberapa tantangan seperti kebutuhan konstruksi graf yang tepat, peningkatan kompleksitas komputasi, serta kesulitan interpretasi akibat struktur model yang lebih kompleks dibandingkan model *machine learning* tradisional [17], [43]. Oleh karena itu, penerapan GNN dalam *fraud detection* sering digunakan sebagai pendekatan alternatif atau komplementer terhadap model berbasis tabular, terutama ketika data memiliki hubungan relasional yang kuat antar entitas [18], [19], [44]. Dalam penelitian *fraud detection*, penggunaan GNN dapat membantu mengevaluasi apakah informasi hubungan antar entitas dalam bentuk graf mampu memberikan peningkatan performa dibandingkan pendekatan berbasis fitur tabular seperti model *ensemble tree-based* [18], [29].

2.2.5 *Federated Learning* (FL)

Federated Learning merupakan pendekatan pembelajaran terdistribusi yang memungkinkan model dilatih pada beberapa klien tanpa memindahkan data ke pusat [20]. Setiap klien melatih model secara lokal, kemudian parameter model dikirim ke server untuk dilakukan agregasi. Keunggulan utama *Federated Learning* (FL) adalah kemampuannya dalam melakukan pelatihan model secara kolaboratif tanpa harus memindahkan data mentah dari masing-masing sumber ke lokasi terpusat, sehingga dapat membantu menjaga privasi data pengguna [20], [21]. Kemampuan tersebut menjadi penting dalam *fraud detection* karena data transaksi finansial umumnya memiliki karakteristik sensitif sehingga proses pengelolaan dan pertukaran data perlu mempertimbangkan aspek keamanan serta privasi [13], [20].

Selain menjaga privasi, FL memungkinkan beberapa entitas atau sumber data yang berbeda untuk melakukan pembaruan model secara terdistribusi melalui pertukaran parameter model tanpa berbagi dataset asli [20], [36]. Karakteristik tersebut membuat FL sesuai diterapkan pada lingkungan finansial yang memiliki banyak sumber data tersebar seperti institusi atau sistem yang perlu melakukan kolaborasi analisis *fraud* tanpa menggabungkan seluruh data transaksi dalam satu penyimpanan terpusat [13], [21]. Meskipun memiliki keunggulan dalam aspek privasi, penerapan FL masih menghadapi beberapa tantangan seperti perbedaan distribusi data antar klien, kebutuhan sinkronisasi pembaruan model, serta kompleksitas koordinasi antar sumber data yang berpartisipasi dalam proses pelatihan [22], [36]. Selain itu, variasi karakteristik data dan keterbatasan fitur yang tersedia pada masing-masing klien dapat memengaruhi kualitas pembelajaran model serta menyebabkan perbedaan performa antar lingkungan penerapan [21], [22]. Proses agregasi model dalam FL biasanya menggunakan metode *Federated Averaging* (FedAvg):

$$w = \sum_{k=1}^{K} \frac{n_k}{n} w_k$$

Gambar 2.5 Rumus Federated Averaging (FedAvg)

Berikut keterangan variabel pada rumus tersebut:
w: Parameter model global, yaitu hasil akhir dari proses agregasi seluruh parameter yang dikirimkan oleh klien ke server pusat.

$\boldsymbol{w_k}$**:** Parameter model global yaitu hasil akhir dari proses agregasi seluruh parameter yang dikirimkan oleh klien ke server pusat.

$\boldsymbol{n_k}$**:** Parameter dari klien ke-*k* yang dihasilkan setelah setiap klien melakukan pelatihan model secara lokal menggunakan data mereka sendiri.

$\boldsymbol{n}$**:** Total data dari seluruh klien yang berpartisipasi dalam proses pembelajaran terdistribusi tersebut.

Pendekatan ini memungkinkan pelatihan model dilakukan secara efisien dengan tetap mempertimbangkan aspek privasi dan distribusi data [58]. Implementasi FL sangat krusial dalam menjamin aspek privasi dan keamanan dalam transaksi finansial yang bersifat sensitif, sehingga memungkinkan kolaborasi antar institusi tanpa risiko kebocoran informasi [36]. Dengan pendekatan ini, pelatihan model tetap dapat dilakukan tanpa menggabungkan data secara langsung, sehingga jauh lebih aman dari sisi privasi bagi pengguna sistem finansial [20].

2.2.6 *User Acceptance Testing* (UAT)

User Acceptance Testing (UAT) adalah tahap akhir dalam siklus pengujian perangkat lunak yang melibatkan pengguna akhir untuk memastikan bahwa produk atau sistem yang dikembangkan dapat memenuhi kriteria bisnis dan berfungsi sebagaimana mestinya di lingkungan nyata [59]. *Software testing* merupakan tahap krusial dalam pengembangan sistem informasi untuk memastikan aplikasi dapat berfungsi sesuai kebutuhan pengguna [60]. UAT sering dilakukan setelah sistem melewati tahapan pengujian teknis seperti *Functional Testing*, *Integration Testing*, dan *System Testing* [61]. Tujuan utama dari UAT adalah untuk mengonfirmasi bahwa sistem atau aplikasi yang dikembangkan dapat digunakan dengan mudah oleh pengguna dan dapat mendukung tujuan bisnis yang sudah ditentukan [61]. UAT juga memastikan bahwa sistem memenuhi kebutuhan fungsional yang

ditentukan pengguna, meminimalkan risiko kegagalan produk dan meningkatkan kualitas perangkat lunak [61].

Pada umumnya, UAT dilakukan melalui pemberian skenario pengujian atau instrumen evaluasi kepada pengguna untuk menilai aspek seperti kemudahan penggunaan, kesesuaian fungsi sistem, serta pengalaman pengguna selama berinteraksi dengan aplikasi [60], [62]. Penilaian dalam UAT sering menggunakan kuesioner dengan skala pengukuran tertentu seperti skala Likert sehingga persepsi dan tingkat penerimaan pengguna terhadap sistem dapat dianalisis secara kuantitatif [60]. Hasil pengujian UAT dapat digunakan sebagai indikator bahwa sistem telah sesuai dengan kebutuhan pengguna serta mampu digunakan secara efektif dalam lingkungan penerapan yang sebenarnya [60], [61], [62]. Secara umum, nilai hasil UAT dapat dihitung dengan rumus persentase berikut:

$$\text{Persentase UAT (\%)} = \left(\frac{\text{Total Skor yang Diperoleh}}{\text{Skor Maksimum Ideal}} \right) \times 100\%$$

Gambar 2.6 Persentase Skala Likert

Berikut keterangan variabel pada rumus tersebut:

1. Skor yang diperoleh adalah jumlah seluruh nilai jawaban responden dari kuesioner UAT.

2. Skor maksimal adalah skor tertinggi yang mungkin diperoleh jika semua responden memilih nilai tertinggi pada seluruh item pernyataan.

3. Hasil persentase kemudian diinterpretasikan untuk melihat tingkat penerimaan pengguna terhadap sistem.

Metodologi pengujian UAT meliputi beberapa tahap yaitu perencanaan UAT, menyiapkan pertanyaan pengujian, dan analisis hasil pengujian [62]. UAT *Planning* adalah tahap perencanaan UAT dengan menentukan tujuan, ruang lingkup, jadwal, sumber daya, dan kriteria keberhasilan. Tahap *Preparing Test Questions* adalah menyiapkan pertanyaan pengujian dan membuat skenario uji melalui Google Forms *questionnaire* atau dokumen pengujian. Tahap *Conducting Tests* adalah

pelaksanaan pengujian dimana *end-user* menjalankan skenario uji yang sudah direncanakan di lokasi pengguna akhir [62].

2.2.7 Fraud Transaction Characteristics

Fraud transaction characteristics merupakan karakteristik atau pola transaksi yang menunjukkan adanya perilaku tidak normal, penyimpangan, atau indikasi aktivitas penipuan dalam suatu sistem keuangan [1], [31]. Dalam lingkungan finansial digital, *fraud* dapat muncul dalam berbagai bentuk seperti penyalahgunaan identitas, transaksi tidak sah, *account takeover*, serta bentuk manipulasi transaksi lainnya yang memanfaatkan kelemahan pada proses pembayaran maupun autentikasi pengguna [1], [3].

Karakteristik transaksi *fraud* umumnya memiliki perbedaan dibandingkan transaksi normal karena menunjukkan pola yang menyimpang seperti perubahan perilaku pengguna, nilai transaksi yang tidak sesuai kebiasaan, frekuensi transaksi yang tidak wajar, maupun aktivitas yang berbeda dari histori transaksi sebelumnya [3], [32]. Karakteristik transaksi *fraud* menjadi aspek penting dalam pembangunan model deteksi *fraud* berbasis *machine learning* karena model membutuhkan representasi fitur yang mampu menggambarkan pola perilaku transaksi secara informatif [6], [31]. Fitur yang digunakan dalam *fraud detection* dapat mencakup informasi seperti jumlah transaksi, waktu transaksi, jenis transaksi, lokasi, histori aktivitas pengguna, serta karakteristik lain yang dapat membantu model membedakan transaksi normal dan transaksi mencurigakan [6], [32].

Pada penelitian *fraud detection* terkini, karakteristik transaksi tersebut dianalisis oleh model *machine learning* untuk mengidentifikasi hubungan dan pola kompleks antara transaksi normal dan transaksi *fraud* secara otomatis [6], [31]. Secara konseptual, transaksi *fraud* sering dipandang sebagai bentuk penyimpangan atau anomali terhadap pola transaksi mayoritas, sehingga pendeteksian *fraud* membutuhkan kemampuan model dalam mengenali perbedaan karakteristik antara kelas normal dan kelas *fraud* [31], [51].

Karakteristik *fraud* tidak selalu sama pada setiap dataset karena dipengaruhi oleh jenis transaksi, lingkungan bisnis, perilaku pengguna, serta pola serangan yang terjadi, tetapi umumnya memiliki ciri berupa ketidakwajaran nilai transaksi, perubahan perilaku, dan pola aktivitas yang sulit dikenali secara manual [1], [3]. Oleh karena itu, pemahaman terhadap karakteristik transaksi *fraud* menjadi faktor penting dalam pengembangan sistem deteksi *fraud* agar model dapat menangkap sinyal anomali secara lebih akurat dan menghasilkan prediksi yang lebih sesuai dengan kondisi transaksi sebenarnya [31], [32]. Dalam penelitian yang menggunakan fitur berbasis perilaku transaksi, proses deteksi *fraud* dilakukan dengan memanfaatkan penyimpangan karakteristik transaksi terhadap pola historis normal sebagai dasar bagi model *machine learning* dalam melakukan klasifikasi. [6], [32].

2.2.8 Class Imbalance

Class imbalance merupakan kondisi ketika distribusi kelas dalam suatu dataset tidak seimbang, yaitu ketika jumlah sampel pada salah satu kelas jauh lebih besar dibandingkan kelas lainnya [4], [51]. Dalam *fraud detection*, permasalahan *class imbalance* sering terjadi karena transaksi normal biasanya memiliki jumlah yang jauh lebih besar dibandingkan transaksi *fraud* yang jumlahnya relatif sedikit [4], [7]. Ketidakseimbangan tersebut dapat menyebabkan model *machine learning* lebih cenderung mempelajari pola dari kelas mayoritas sehingga kemampuan dalam mengenali kelas minoritas, yaitu transaksi *fraud* menjadi kurang optimal [4], [5]. Kondisi ini menjadi permasalahan penting karena kelas minoritas yang memiliki jumlah data lebih sedikit merupakan kelas yang menjadi fokus utama dalam sistem deteksi *fraud* [5], [51].

Penggunaan akurasi sebagai satu-satunya metrik evaluasi pada dataset yang tidak seimbang dapat menghasilkan interpretasi yang kurang tepat karena model dapat memperoleh nilai akurasi tinggi meskipun gagal dalam mendeteksi sebagian besar transaksi *fraud* [39], [57]. Oleh karena itu, evaluasi model *fraud detection* pada *data imbalanced* lebih tepat menggunakan metrik seperti *precision, recall*, F1-score, dan

AUC untuk memberikan gambaran performa yang lebih representatif terhadap kemampuan model dalam mendeteksi kelas *fraud* [39], [42], [57].

Dalam penelitian *fraud detection*, permasalahan *class imbalance* perlu ditangani secara khusus agar model tidak hanya menghasilkan prediksi yang baik terhadap transaksi normal, tetapi juga mampu mengenali transaksi *fraud* secara efektif [5], [7]. Salah satu pendekatan yang dapat digunakan untuk menangani *class imbalance* adalah *class weighting*, yaitu memberikan bobot yang berbeda pada setiap kelas sehingga kesalahan pada kelas minoritas memiliki pengaruh yang lebih besar selama proses pelatihan model [5], [7]. Melalui pendekatan tersebut, model dapat lebih memperhatikan pola transaksi *fraud* tanpa harus mengubah distribusi data asli secara langsung, sehingga membantu mengurangi kecenderungan model terhadap kelas mayoritas [7].

Pada pendekatan *weighted loss*, salah satu formulasi yang dapat digunakan adalah *weighted binary cross-entropy*, yaitu fungsi *loss* yang memberikan bobot lebih besar terhadap kesalahan klasifikasi pada kelas tertentu sesuai kebutuhan model [5]. Rumus *weighted binary cross-entropy* untuk keseluruhan dataset rata-rata dapat dituliskan sebagai berikut:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[w_1 \cdot y_i \log(\hat{y}_i) + w_0 \cdot (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Gambar 2.7 Rumus *Weighted Binary Cross-Entropy* (WBCE)

Berikut keterangan variabel pada rumus tersebut:

1. ***N:*** Jumlah total sampel data.
2. ***y*:** Label sebenarnya (*Ground Truth*), bernilai 1 (Kelas Positif) atau 0 (Kelas Negatif).
3. **$\hat{y}$:** Probabilitas prediksi model (bernilai antara () dan 1).
4. **log:** Logaritma natural (ln).
5. **w_1:** Bobot yang diberikan untuk kelas positif.
6. **w_0:** Bobot yang diberikan untuk kelas negatif.

Pada beberapa penelitian, strategi *class weighting* dilakukan dengan memberikan bobot kelas yang berbanding terbalik terhadap jumlah sampel pada masing-masing kelas, sehingga kelas dengan jumlah data lebih sedikit memperoleh bobot yang lebih besar selama proses pelatihan model [5], [7]. Pemberian bobot tersebut bertujuan untuk mengurangi dominasi kelas mayoritas dan meningkatkan perhatian model terhadap pola pada kelas minoritas yang memiliki jumlah data lebih terbatas [5]. Dengan pendekatan ini, model dapat lebih sensitif dalam mengenali karakteristik transaksi *fraud* yang jumlahnya relatif kecil dibandingkan transaksi normal [4], [7].

Dalam penelitian *fraud detection*, penanganan *class imbalance* menjadi aspek penting karena distribusi transaksi *fraud* dan *non-fraud* yang tidak seimbang dapat memengaruhi proses pembelajaran model serta menyebabkan penurunan kemampuan deteksi terhadap kelas *fraud* [4], [51]. Oleh karena itu, penerapan metode penanganan *imbalance* seperti *class weighting* diperlukan agar model mampu menghasilkan performa klasifikasi yang lebih seimbang antara transaksi normal dan transaksi *fraud* [5], [7].

2.2.9 Evaluasi *Fraud Detection*

Evaluasi *fraud detection* merupakan proses untuk mengukur kemampuan model dalam membedakan transaksi *fraud* dan *non-fraud* berdasarkan data pengujian yang telah disiapkan [39], [42]. Dalam penelitian deteksi *fraud*, evaluasi tidak hanya berfokus pada jumlah prediksi yang benar, tetapi juga perlu mempertimbangkan kemampuan model dalam mengenali transaksi *fraud* yang biasanya memiliki jumlah jauh lebih sedikit dibandingkan transaksi normal [4], [51]. Hal tersebut menjadi penting karena kesalahan dalam mengklasifikasikan transaksi *fraud* sebagai transaksi normal (*false negative*) dapat menyebabkan transaksi mencurigakan tidak terdeteksi, sedangkan kesalahan dalam mengklasifikasikan transaksi normal sebagai *fraud* (*false positive*) dapat meningkatkan jumlah alarm palsu [39], [57]. Oleh karena itu, evaluasi model *fraud detection* umumnya menggunakan kombinasi beberapa metrik agar kemampuan model dalam mendeteksi kedua kelas dapat dinilai secara lebih menyeluruh [39], [42].

Pada penelitian terkini, model *fraud detection* sering dievaluasi menggunakan metrik seperti *accuracy, precision, recall*, F1-score, dan AUC-ROC untuk memperoleh gambaran performa klasifikasi yang lebih lengkap [39], [57]. *Accuracy* menunjukkan proporsi keseluruhan prediksi yang berhasil diklasifikasikan dengan benar, tetapi metrik ini dapat memberikan gambaran yang kurang tepat ketika diterapkan pada dataset dengan distribusi kelas yang tidak seimbang [4], [57]. *Precision* menunjukkan tingkat ketepatan prediksi positif model, yaitu proporsi transaksi yang diprediksi sebagai *fraud* dan benar-benar merupakan fraud. [64]. *Recall* menunjukkan kemampuan model dalam menemukan seluruh transaksi *fraud* yang sebenarnya terdapat pada dataset, sehingga metrik ini menjadi penting ketika tujuan utama adalah meminimalkan transaksi *fraud* yang terlewatkan [39], [57]. F1-score digunakan untuk menggabungkan *precision* dan *recall* dalam satu nilai evaluasi sehingga dapat memberikan keseimbangan antara kemampuan mendeteksi *fraud* dan menghindari kesalahan prediksi positif [57]. Sementara itu, AUC-ROC digunakan untuk mengukur kemampuan model dalam membedakan kelas *fraud* dan *non-fraud* pada berbagai nilai *threshold* klasifikasi [39], [42].

Dengan demikian, evaluasi *fraud detection* perlu mempertimbangkan kondisi ketidakseimbangan data agar hasil pengukuran performa model tidak hanya mencerminkan dominasi kelas mayoritas [4], [51]. Dikarenakan *fraud detection* merupakan permasalahan klasifikasi dengan konsekuensi kesalahan yang berbeda antara *false positive* dan *false negative*, model yang baik tidak hanya ditentukan berdasarkan nilai *accuracy* yang tinggi, tetapi juga kemampuan mempertahankan performa deteksi *fraud* secara efektif [39], [57]. Dalam praktiknya, pemilihan metrik evaluasi perlu disesuaikan dengan kebutuhan sistem karena setiap metrik memberikan informasi yang berbeda mengenai karakteristik performa model [42], [57]. Oleh karena itu, penelitian *fraud detection* umumnya menerapkan pendekatan evaluasi *multi-metric* untuk memperoleh gambaran yang lebih komprehensif mengenai kemampuan model dalam melakukan klasifikasi transaksi *fraud* [39], [42].

2.2.10 Evaluasi Metrik

Evaluasi metrik merupakan proses pengukuran performa model klasifikasi menggunakan indikator kuantitatif untuk menggambarkan kemampuan model dalam menghasilkan prediksi terhadap data yang diuji [39], [42]. Dalam penelitian *fraud detection*, beberapa metrik yang umum digunakan adalah *accuracy, precision, recall*, F1-score, dan AUC-ROC karena masing-masing metrik memberikan perspektif yang berbeda terhadap kemampuan model dalam melakukan klasifikasi [39], [57].

Pemahaman terhadap berbagai metrik evaluasi diperlukan agar hasil pengujian model dapat diinterpretasikan secara tepat dan tidak hanya bergantung pada satu indikator performa, terutama pada kondisi dataset yang tidak seimbang [4], [42]. *Accuracy* merupakan metrik yang mengukur proporsi jumlah prediksi yang benar terhadap keseluruhan jumlah data pengujian. [64]. Berikut rumus *accuracy*:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Gambar 2.8 Rumus *Accuracy*

Berikut keterangan variabel pada rumus tersebut:

1. ***TP*** = *True Positive*, jumlah transaksi *fraud* yang diprediksi *fraud*.
2. ***TN***= *True Negative*, jumlah transaksi *non-fraud* yang diprediksi *non-fraud*.
3. ***FP***= *False Positive*, jumlah transaksi *non-fraud* yang diprediksi *fraud*.
4. ***FN***= *False Negative*, jumlah transaksi *fraud* yang diprediksi *non-fraud*.

Accuracy mudah dipahami dan sering digunakan sebagai evaluasi awal performa model, tetapi metrik ini dapat memberikan hasil yang kurang representatif pada dataset dengan distribusi kelas yang tidak seimbang karena model dapat memperoleh nilai tinggi dengan hanya memprediksi kelas mayoritas [4], [57]. *Precision* merupakan metrik yang menunjukkan proporsi prediksi positif yang

benar dibandingkan dengan seluruh prediksi positif yang dihasilkan oleh model. [64]. Berikut rumus *precision*:

$$Precision = \frac{TP}{TP + FP}$$

Gambar 2.9 Rumus *Precision*

Berikut keterangan variabel pada rumus tersebut:

1. ***TP*** = *True Positive*, jumlah transaksi *fraud* yang diprediksi *fraud*.
2. ***FP***= *False Positive*, jumlah transaksi *non-fraud* yang diprediksi *fraud*.

Dalam *fraud detection*, *precision* penting karena menunjukkan tingkat ketepatan model ketika memberikan prediksi *fraud*, sehingga *precision* yang tinggi menunjukkan jumlah kesalahan berupa *false alarm* dapat dikurangi [39], [57]. *Recall* merupakan metrik yang mengukur kemampuan model dalam mendeteksi transaksi *fraud* yang sebenarnya terdapat dalam dataset [57]. Berikut rumus *recall*:

$$Recall = \frac{TP}{TP + FN}$$

Gambar 2.10 Rumus *Recall*

Berikut keterangan variabel pada rumus tersebut:

1. ***TP*** = *True Positive*, jumlah transaksi *fraud* yang diprediksi *fraud*.
2. ***FN***= *False Negative*, jumlah transaksi *fraud* yang diprediksi *non-fraud*.

Recall menjadi salah satu metrik penting dalam *fraud detection* karena menunjukkan kemampuan model dalam menangkap transaksi *fraud* dan mengurangi jumlah kasus *fraud* yang tidak terdeteksi (*false negative*) [39], [57]. Pada beberapa kasus *fraud detection*, *recall* dapat menjadi perhatian utama karena kegagalan mendeteksi transaksi *fraud* dapat memiliki konsekuensi yang lebih besar dibandingkan menghasilkan beberapa prediksi *fraud* yang salah [39]. F1-score merupakan metrik yang menggabungkan *precision* dan *recall* menggunakan rata-rata sehingga dapat memberikan keseimbangan antara kemampuan model dalam mendeteksi *fraud* dan mengurangi *false positive* [57]. Berikut rumus *F1-score*:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Gambar 2.11 Rumus *F1-Score*

Berikut keterangan variabel pada rumus tersebut:

1. ***Precision*** = Nilai *Precision.*
2. ***Recall*** = Nilai *Recall.*

F1-score sering digunakan pada dataset yang mengalami *class imbalance* karena memberikan evaluasi yang lebih seimbang dibandingkan hanya menggunakan *accuracy* [4], [57]. AUC-ROC merupakan metrik yang digunakan untuk mengukur kemampuan model dalam membedakan kelas positif dan negatif pada berbagai nilai *threshold* klasifikasi [39], [42]. Kurva ROC (*Receiver Operating Characteristic*) menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai *threshold* prediksi. [41]. Berikut rumusnya secara umum:

$$\text{TPR} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}$$

Gambar 2.12 Rumus TPR (*True Positive Rate*)

$$\text{FPR} = \frac{\text{False Positive (FP)}}{\text{False Positive (FP)} + \text{True Negative (TN)}}$$

Gambar 2.13 Rumus FPR (*False Positive Rate*)

Berikut keterangan variabel pada rumus tersebut:

1. ***TPR***= *True Positive Rate* atau *recall*.
2. ***FPR***= *False Positive Rate*.
3. ***TP,FN,FP,TN*** = elemen *Confusion Matrix*.

AUC (*Area Under the Curve*) merupakan luas area di bawah kurva ROC dengan nilai antara 0 hingga 1, di mana nilai yang mendekati 1 menunjukkan kemampuan model yang lebih baik dalam membedakan kelas *fraud* dan *non-fraud* [39], [42]. Dalam *fraud detection*, AUC-ROC sering digunakan sebagai salah satu metrik evaluasi karena mampu menggambarkan kemampuan diskriminasi model pada

berbagai *threshold*, meskipun interpretasinya tetap perlu mempertimbangkan kondisi ketidakseimbangan kelas pada dataset [4], [39].

2.3 Framework/Algoritma yang digunakan

2.3.1 *Cross-Industry Standard Process for Data mining* (CRISP-DM)

Metodologi yang digunakan dalam penelitian *data mining* memiliki peran penting untuk memastikan bahwa proses pengolahan data, pembangunan model, dan evaluasi hasil dilakukan secara sistematis serta terstruktur. Penerapan metodologi yang jelas dapat membantu penelitian berbasis *machine learning* dalam mengelola tahapan analisis data hingga menghasilkan model prediksi yang dapat dievaluasi secara objektif [56], [63]. Salah satu metodologi yang banyak digunakan dalam penelitian *data mining* dan *machine learning* adalah CRISP-DM (*Cross Industry Standard Process for Data Mining*), yaitu pendekatan yang menyediakan kerangka kerja terstruktur dalam proses pengembangan model berbasis data [56], [63]. CRISP-DM merupakan kerangka kerja standar yang dikembangkan untuk membantu peneliti dan praktisi dalam mengelola proses analisis data secara sistematis mulai dari pemahaman masalah hingga implementasi model [64].

Metodologi CRISP-DM terdiri dari enam tahapan utama yang saling berhubungan, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Enam tahap ini tidak bersifat linier mutlak, melainkan iteratif artinya peneliti dapat kembali ke tahap sebelumnya apabila ditemukan hasil yang belum sesuai dengan kebutuhan [56]. Struktur ini membuat CRISP-DM cocok digunakan dalam penelitian *machine learning* karena mendukung proses pengembangan model yang lebih fleksibel namun tetap sistematis [56].

Tahap pertama adalah *business understanding*, yaitu tahap untuk memahami tujuan bisnis atau tujuan penelitian serta menentukan masalah apa yang ingin diselesaikan [56], [63]. Dalam konteks *fraud detection*, tahap ini digunakan untuk mengidentifikasi kebutuhan utama, misalnya mendeteksi transaksi mencurigakan, mengurangi kerugian finansial, dan mendukung pengambilan keputusan pada institusi keuangan [31], [47]. Tahap ini penting karena arah penelitian harus selaras

dengan permasalahan nyata yang dihadapi, bukan sekadar menghasilkan model dengan performa tinggi [49].

Tahap kedua adalah *data understanding*, yaitu proses mengenali data yang akan digunakan termasuk sumber data, struktur data, kualitas data, distribusi atribut, serta kemungkinan adanya *missing value* atau ketidakseimbangan kelas [56], [63]. Pada penelitian *fraud detection*, tahap ini sangat penting karena data transaksi keuangan biasanya memiliki karakteristik khusus seperti jumlah data *fraud* yang jauh lebih sedikit dibanding data normal [4], [7]. Pemahaman terhadap karakteristik data sejak awal membantu peneliti menentukan strategi pengolahan dan pemodelan yang tepat [50].

Tahap ketiga adalah *data preparation*, yaitu tahap menyiapkan data agar siap digunakan dalam proses modeling [56], [63]. Proses ini dapat mencakup pembersihan data, transformasi data, seleksi fitur, penghapusan data yang tidak relevan, dan penanganan data tidak seimbang [16], [50]. Dalam penelitian *fraud detection*, tahap ini sering menjadi tahap yang paling penting karena kualitas data sangat menentukan hasil model [50]. Jika dataset memiliki kelas *fraud* yang sangat kecil, maka teknik seperti SMOTE, *undersampling*, atau *feature engineering* dapat digunakan untuk meningkatkan performa model [4], [21].

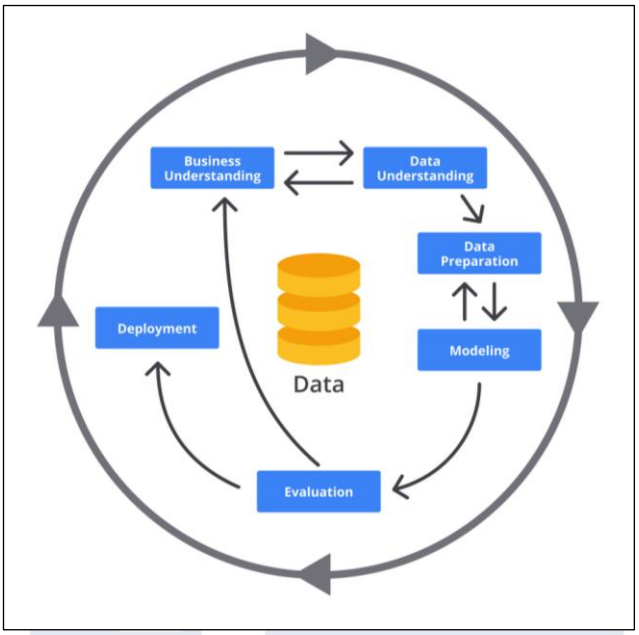
Tahap keempat adalah *modeling*, yaitu proses membangun model analitik atau *machine learning* berdasarkan data yang telah disiapkan [56], [63]. Pada tahap ini, peneliti memilih algoritma yang sesuai dengan tujuan penelitian seperti *Logistic Regression*, *Random Forest*, *Gradient Boosting*, *XGBoost*, atau model *ensemble* lainnya [5], [6], [56]. Model dapat diuji beberapa kali dengan variasi parameter untuk memperoleh hasil terbaik [32]. Dalam penelitian *fraud detection*, *modeling* bertujuan untuk mengidentifikasi pola transaksi yang normal dan mencurigakan secara otomatis [31].

Tahap kelima adalah *evaluation*, yaitu proses menilai apakah model yang dibangun sudah sesuai dengan tujuan penelitian [56], [63]. Evaluasi dilakukan menggunakan metrik yang relevan seperti *accuracy, precision, recall*, F1-score, ROC-AUC, atau

MCC [57]. Pada kasus *fraud detection*, evaluasi tidak cukup hanya menggunakan *accuracy* karena data pada umumnya *imbalanced*, sehingga metrik yang lebih sensitif terhadap kelas minoritas sangat diperlukan [57]. Evaluasi ini membantu memastikan bahwa model tidak hanya baik secara statistik, tetapi juga bermanfaat secara praktis dalam mendeteksi *fraud* [57].

Tahap terakhir adalah *deployment*, yaitu tahap penerapan model ke dalam lingkungan nyata atau sistem yang dapat digunakan dalam operasional [56], [63]. Pada konteks *fraud detection*, *deployment* dapat berupa integrasi model ke dalam sistem transaksi digital, sistem *monitoring* bank, atau *dashboard* pendukung keputusan untuk mendeteksi transaksi berisiko secara otomatis [3], [48]. Tahap ini penting karena hasil penelitian tidak berhenti pada model, tetapi dapat dimanfaatkan langsung untuk mendukung proses pengawasan dan pencegahan fraud [3].

Dengan menggunakan metodologi CRISP-DM, proses penelitian *data mining* dapat dilakukan secara lebih terstruktur dan sistematis. Metodologi ini membantu peneliti dalam mengorganisasi setiap tahapan penelitian mulai dari pemahaman masalah hingga evaluasi model. Oleh karena itu, CRISP-DM digunakan sebagai kerangka kerja dalam penelitian yang melibatkan analisis data dan pembangunan model *machine learning* termasuk dalam penelitian yang berfokus pada deteksi *fraud* pada transaksi keuangan.



Gambar 2.14 Framework CRISP-DM

Sumber: Dicoding, 2024.

2.4 Tools/software yang digunakan

2.4.1 PySpark & Jupyter Notebook

Penelitian ini menggunakan beberapa *tools* pendukung untuk menunjang proses pengolahan data, pemodelan, serta evaluasi hasil eksperimen. Bahasa pemrograman utama yang digunakan adalah Python karena fleksibilitasnya dalam mendukung berbagai *library machine learning* dan analisis data. Selain itu, digunakan PySpark sebagai *tools* pendukung untuk membantu proses pengolahan dan transformasi data, khususnya dalam menangani *dataset* berukuran besar. Penggunaan PySpark dalam penelitian ini tidak menjadi fokus utama, melainkan hanya sebagai alat bantu untuk meningkatkan efisiensi komputasi pada beberapa tahapan *preprocessing*.

Lingkungan pengembangan yang digunakan dalam penelitian ini adalah Jupyter Notebook yang dijalankan melalui Visual Studio Code. Penggunaan Jupyter Notebook memungkinkan proses eksplorasi data, pengembangan model, serta pengujian algoritma dilakukan secara interaktif. Selain itu, pendekatan berbasis

notebook memudahkan dokumentasi eksperimen, visualisasi hasil analisis, serta pengujian kode secara bertahap selama proses penelitian berlangsung.

Dataset transaksi keuangan dalam penelitian ini diolah menggunakan struktur DataFrame berbasis Python, sehingga memudahkan proses manipulasi data, transformasi fitur, serta integrasi dengan berbagai *library machine learning*. Pendekatan ini memungkinkan proses pengolahan data dilakukan secara efisien dalam lingkungan komputasi lokal.

Evaluasi model dilakukan menggunakan metrik klasifikasi yang umum digunakan pada permasalahan *fraud detection* dengan karakteristik data tidak seimbang (*imbalanced dataset*), yaitu Area Under the Curve (AUC), Precision, Recall, dan F1-Score. Penggunaan beberapa metrik evaluasi tersebut bertujuan untuk memberikan penilaian yang lebih komprehensif terhadap performa model, khususnya dalam mendeteksi transaksi *fraud* secara akurat tanpa mengabaikan keseimbangan antara kelas. Dengan kombinasi *tools* tersebut, penelitian ini berfokus pada pengembangan dan evaluasi model *machine learning*, sementara penggunaan PySpark hanya berperan sebagai pendukung dalam proses pengolahan data dan tidak menjadi bagian utama dalam analisis penelitian.