BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Teori

Bagian ini membahas landasan teori yang menjadi dasar penelitian, mulai dari konsep dasar *Bitcoin* dan *Blockchain*, karakteristik transaksi ilegal pada jaringan *Blockchain*, konsep *digital forensik*, hingga metode *Machine learning* dan *Graph Neural Network* yang dipakai untuk mendeteksi transaksi ilegal tersebut, yaitu *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan *Graph Attention Network* (GAT). Selain itu, bagian ini juga membahas tahapan-tahapan pendukung dalam membangun model, mulai dari praproses dan normalisasi data, teknik *Class Weight* untuk menangani ketidakseimbangan kelas, proses optimisasi model saat pelatihan, metrik evaluasi model klasifikasi, hingga teknik visualisasi t-SNE yang dipakai untuk mengamati hasil representasi model.

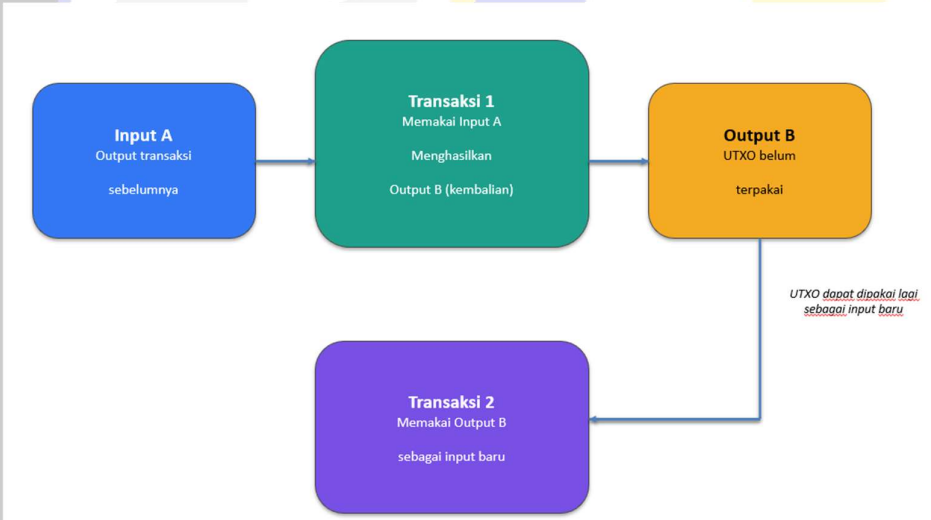
2.1.1. *Bitcoin*

Bitcoin adalah mata uang digital pertama yang diperkenalkan oleh Satoshi Nakamoto pada tahun 2008 melalui whitepaper berjudul "*Bitcoin*: *A Peer-to-Peer Electronic Cash System*". Berbeda dengan mata uang pada umumnya, *Bitcoin* tidak diterbitkan maupun dikendalikan oleh otoritas pusat seperti bank sentral atau pemerintah, melainkan dijalankan oleh jaringan komputer yang tersebar di seluruh dunia secara peer-to-peer, sehingga transaksi bisa berlangsung tanpa perlu pihak ketiga sebagai perantara [10]. Setiap transaksi *Bitcoin* dicatat secara terbuka pada buku besar bersama (l*edge*r)[11], sehingga siapa pun bisa memverifikasi keabsahannya. Karakteristik keterbukaan ini terus dikonfirmasi oleh studi-studi terbaru yang merekonstruksi seluruh riwayat jaringan transaksi *Bitcoin*[1]

Secara teknis, transaksi *Bitcoin* bekerja memakai model input dan output yang dikenal sebagai *Unspent Transaction Output (UTXO)*. Setiap transaksi mengambil satu atau lebih output dari transaksi sebelumnya sebagai input, lalu menghasilkan output baru yang bisa dipakai pada transaksi berikutnya [10]. Pola

input-output inilah yang membentuk rantai keterkaitan antar transaksi, yang apabila digambarkan dalam bentuk graf akan memperlihatkan bagaimana nilai (*value)* berpindah dari satu titik ke titik lain di dalam jaringan. Studi terbaru yang merekonstruksi seluruh jaringan transaksi *Bitcoin* sejak tahun 2009 menemukan bahwa jaringan ini memiliki karakteristik graf berskala bebas (*scale-free*) [1] dengan pola distribusi kekayaan yang tidak merata dan cenderung semakin timpang seiring waktu, sebuah temuan yang semakin memperkuat relevansi pendekatan berbasis graf dalam menganalisis perilaku transaksi *Bitcoin* [15].

Karakteristik seperti jumlah input dan output, nilai transaksi, waktu transaksi (*time step*), serta derajat keterhubungan suatu transaksi dengan transaksi lain merupakan atribut yang lazim diekstraksi sebagai fitur dalam penelitian deteksi anomali pada *Bitcoin*[3]. Fitur-fitur inilah yang menjadi dasar representasi data pada dataset Elliptic yang dipakai dalam penelitian ini beserta versi perluasannya, Elliptic++, di mana setiap transaksi direpresentasikan melalui kombinasi fitur lokal (*local features*) dan fitur agregat dari transaksi tetangga (*aggregate features*)[3].



Sumber: Diolah oleh penulis berdasarkan Song, Wei, Qu, dan Wang (2024).

Gambar 2.1 Model Input-Output UTXO pada Transaksi *Bitcoin*

2.1.2 *Blockchain*

Blockchain adalah teknologi buku besar terdistribusi *(distributed ledger)* yang mencatat data transaksi ke dalam blok-blok yang saling terhubung secara kriptografis. Setiap blok baru memuat nilai hash dari blok sebelumnya, sehingga membentuk rantai yang bersifat tahan-ubah *(immutable)*, begitu suatu blok sudah dikonfirmasi oleh jaringan, mengubah isi blok tersebut akan mengubah seluruh nilai hash pada blok-blok setelahnya, sehingga upaya manipulasi data akan mudah terdeteksi. Sifat desentralisasi ini menghilangkan kebutuhan akan otoritas pusat, sebab validitas transaksi diverifikasi secara kolektif oleh partisipan jaringan lewat mekanisme consensus. [10] menjelaskan bahwa arsitektur *Blockchain* umumnya terdiri atas tiga varian, yaitu *Blockchain* publik yang terbuka bagi siapa saja, *Blockchain* privat yang aksesnya dikendalikan oleh satu organisasi, dan *Blockchain* konsorsium yang dikelola bersama oleh beberapa entitas[3]. Ketiganya sama-sama mengandalkan algoritma konsensus, seperti *Proof-of-Work* maupun *Proof-of-Stake*, untuk menjaga keandalan pencatatan transaksi tanpa otoritas tunggal.

Meskipun sifat transparan dan tahan-ubah ini memberi jaminan keamanan yang kuat terhadap manipulasi data, *Blockchain* publik seperti *Bitcoin* juga bersifat pseudonim, artinya identitas pemilik alamat tidak langsung terhubung dengan identitas dunia nyata. Perpaduan antara keterbukaan data transaksi dan anonimitas identitas pengguna inilah yang pada akhirnya membuka celah bagi penyalahgunaan jaringan *Blockchain* untuk aktivitas ilegal, sehingga dibutuhkan mekanisme pemantauan tambahan di luar jaminan kriptografis semata.

2.1.3 Transaksi Ilegal pada Jaringan *Blockchain*

Transaksi ilegal merujuk pada aktivitas transaksi yang berkaitan dengan pelanggaran hukum, seperti pencucian uang (*money laundering*), penipuan (*fraud*), pemerasan lewat *ransomware*, skema Ponzi, hingga pembayaran pada pasar gelap *(darknet market).* Dataset Elliptic yang menjadi acuan penelitian ini, beserta versi perluasannya Elliptic++, mengelompokkan entitas seperti penambang *(miners),* penyedia dompet digital *(wallet providers),* dan *exchange* sebagai kategori sah

(licit), sementara entitas seperti *ransomware*, skema *Ponzi*, dan *malware* dikelompokkan sebagai kategori ilegal (illicit) [3].

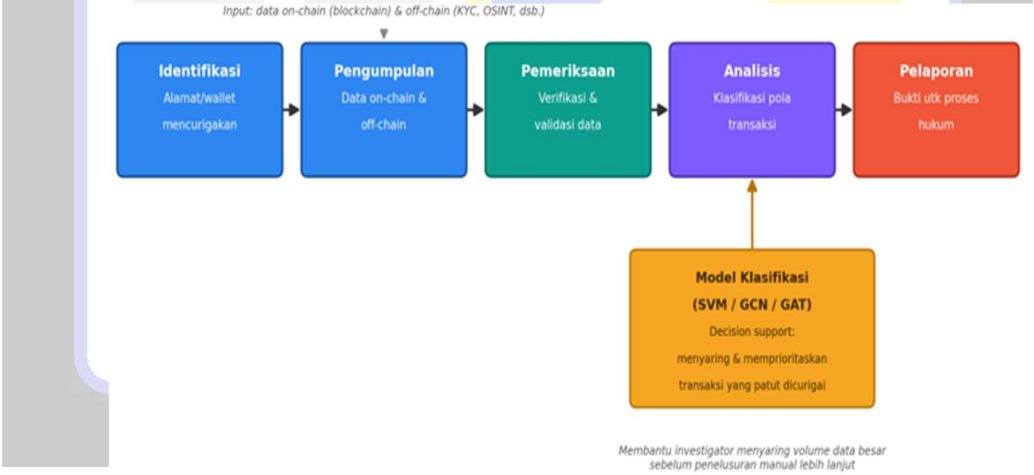
Mendeteksi transaksi ilegal pada *Blockchain* menghadapi tantangan tersendiri dibandingkan sistem keuangan konvensional. Pertama, identitas pengguna bersifat pseudonim sehingga tidak bisa langsung diasosiasikan dengan pelaku sebenarnya. Kedua, pola transaksi ilegal jarang terlihat mencolok pada satu transaksi tunggal, melainkan baru tampak jelas ketika dianalisis dalam konteks hubungan antar banyak transaksi, misalnya lewat teknik layering di mana dana ilegal dipecah dan dialirkan melalui banyak alamat perantara untuk mengaburkan asal-usulnya dalam tinjauan tentang deteksi anomali pada graf menekankan bahwa perilaku menyimpang, semacam ini sering kali hanya bisa dikenali lewat pola relasional antar entitas yang saling terhubung, bukan atribut individual semata[12]. Sejalan dengan hal tersebut menunjukkan bahwa pendekatan *Machine learning* terhadap pola transaksi *Bitcoin* mampu mengklasifikasikan pembayaran *ransomware* yang sulit dikenali hanya dari atribut transaksi secara terpisah[13].

2.1.4 *Digital forensik*

Digital forensik adalah proses identifikasi, pengumpulan, pemeriksaan, analisis, dan pelaporan bukti digital untuk mendukung proses investigasi maupun penegakan hukum. Salah satu kerangka kerja forensik digital yang umum diadopsi adalah *Guide to Integrating Forensic Techniques into Incident Response* yang diterbitkan oleh *National Institute of Standards and Technology (NIST)*, yang membagi proses investigasi forensik digital ke dalam empat tahap utama, yaitu *collection* (pengumpulan bukti digital), *examination* (pemeriksaan teknis bukti), *analysis* (analisis untuk menarik kesimpulan), dan *reporting* (penyajian hasil analisis). Kerangka ini menjadi acuan dalam memetakan tahapan teknis penelitian ke dalam alur investigasi *forensik digital*, sebagaimana dijelaskan lebih lanjut pada subbab 3.11. Dalam konteks *Blockchain*, cabang ini juga dikenal sebagai *financial forensics* atau *Blockchain forensics*[3], yaitu penerapan teknik forensik digital khusus untuk menelusuri aliran dana, mengelompokkan alamat-alamat yang diduga terkait, dan mengidentifikasi pola transaksi mencurigakan pada jaringan mata uang

kripto. Dalam tinjauan literatur sistematis mengenai Blockchain forensics, menjelaskan bahwa sifat Blockchain yang terdesentralisasi dan immutable menghadirkan tantangan tersendiri bagi investigator digital forensik. Di satu sisi, sifat immutable membantu menjaga integritas catatan transaksi tetapi di sisi lain, sifat pseudonim pada alamat Blockchain menyebabkan proses atribusi terhadap identitas dunia nyata menjadi lebih kompleks. Oleh karena itu, tinjauan tersebut menekankan pentingnya penggabungan teknik forensik tradisional dengan pendekatan kecerdasan buatan untuk mempercepat proses analisis pola transaksi berskala besar yang mustahil dilakukan secara manual.[2], [14]

Model klasifikasi transaksi ilegal yang dibangun dalam penelitian ini, baik berbasis SVM, GCN, maupun GAT, pada dasarnya berfungsi sebagai alat bantu *(decision support)* dalam proses *digital forensik*, yakni membantu *investigator* menyaring dan memprioritaskan transaksi yang patut dicurigai dari volume data yang sangat besar, sebelum dilakukan penelusuran dan pembuktian lebih lanjut secara manual oleh pihak berwenang.



Sumber: diolah oleh penulis, disintesis dari Regina O., Ramli K., Amarullah A.H. (2025) dan Atlam H.F., Ekuri N., Azad M.A., Lallie H.S. (2024).

Gambar 2.2 Proses *Digital forensik* pada Jaringan *Blockchain* (*Blockchain Forensics*)

2.1.5 *Machine learning* untuk Deteksi Transaksi Ilegal

Pendekatan *supervised learning* dipilih karena dataset Elliptic telah menyediakan label transaksi berupa licit dan illicit. Berbagai penelitian menunjukkan bahwa metode *supervised learning* mampu mengklasifikasikan transaksi ilegal pada jaringan Bitcoin dengan memanfaatkan data berlabel sebagai proses pembelajaran. Meskipun demikian, metode *machine learning konvensional* yang hanya mengandalkan fitur setiap transaksi masih memiliki keterbatasan dalam menangkap hubungan antartransaksi serta informasi kontekstual yang terdapat pada struktur graph blockchain. Oleh karena itu, perkembangan *Graph Neural Network (GNN)* dimanfaatkan untuk mengintegrasikan informasi fitur dan hubungan antarnode sehingga mampu menghasilkan representasi data yang lebih informatif dan meningkatkan performa klasifikasi transaksi ilegal.[15]

Berbeda dengan metode klasifikasi konvensional yang hanya memanfaatkan atribut setiap transaksi secara individual, pendekatan berbasis *Graph Neural Network* memanfaatkan struktur hubungan antartransaksi yang terbentuk pada jaringan Bitcoin. Dalam penelitian ini dipilih *Graph Attention Network (GAT)* karena mekanisme *attention* memungkinkan model memberikan bobot yang berbeda terhadap setiap transaksi tetangga sesuai tingkat pengaruhnya dalam proses klasifikasi. Dengan demikian, informasi yang paling relevan pada lingkungan graph dapat dimanfaatkan secara lebih efektif untuk meningkatkan kemampuan deteksi transaksi ilegal. Sebagai pembanding, model SVM dan GCN digunakan untuk mengevaluasi peningkatan kinerja yang diperoleh dari penerapan mekanisme *attention*.[16]

2.1.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah algoritma klasifikasi yang bekerja dengan mencari *hyperplane* pemisah optimal antarkelas data pada ruang fitur berdimensi tinggi. Prinsip utama SVM adalah memaksimalkan margin, yaitu jarak antara *hyperplane* pemisah dengan titik data terdekat dari masing-masing kelas (*support vectors*), sehingga model memiliki kemampuan generalisasi yang baik terhadap data baru. Hingga saat ini, SVM masih lazim dipakai sebagai metode

pembanding pada berbagai penelitian klasifikasi transaksi ilegal berbasis dataset Elliptic [8].

Untuk data yang tidak dapat dipisahkan secara linear, *Support Vector Machine (SVM)* memanfaatkan fungsi *kernel*, seperti *Radial Basis Function (RBF)*, untuk memetakan data ke ruang berdimensi lebih tinggi sehingga lebih mudah dipisahkan. Kinerja model SVM dipengaruhi oleh pemilihan kernel dan pengaturan *hyperparameter* yang sesuai. Dalam penelitian mengenai klasifikasi transaksi Bitcoin, SVM masih banyak digunakan sebagai model pembanding *(baseline)* karena mampu menangani data berdimensi tinggi dan memberikan performa yang stabil pada proses klasifikasi berbasis fitur. Namun, SVM memperlakukan setiap transaksi sebagai data yang berdiri sendiri sehingga belum mampu memanfaatkan informasi hubungan antartransaksi yang tersimpan pada *struktur graph blockchain*. Keterbatasan tersebut mendorong pengembangan pendekatan berbasis *Graph Neural Network* yang mampu memanfaatkan informasi relasional untuk meningkatkan akurasi deteksi transaksi ilegal.[17]

2.1.7 *Graph Neural Network (GNN)*

Graph Neural Network (GNN) merupakan kelompok metode deep learning yang dirancang untuk memproses data yang direpresentasikan dalam bentuk graph, yaitu kumpulan *node* (simpul) dan *edge* (sisi) yang menggambarkan hubungan antarentitas. Berbeda dengan metode pembelajaran konvensional yang memperlakukan setiap data sebagai sampel yang berdiri sendiri, GNN memanfaatkan informasi struktural pada *graph* untuk mempelajari representasi setiap node berdasarkan karakteristiknya serta hubungan dengan node lain di sekitarnya. Pendekatan ini memungkinkan model menangkap informasi relasional yang tidak dapat dimanfaatkan oleh metode *machine learning konvensional*.[18]

Prinsip dasar yang mendasari sebagian besar arsitektur GNN adalah *message passing* atau agregasi tetangga (*neighborhood aggregation*): setiap *node* memperbarui representasinya dengan mengumpulkan dan menggabungkan informasi dari *node-node* tetangganya secara berulang pada tiap lapisan *(layer).* Dengan mekanisme ini, representasi akhir suatu *node* tidak hanya mencerminkan

atribut *node* itu sendiri, tetapi juga konteks lingkungan di sekitarnya. Prinsip pembelajaran representasi semacam ini turut relevan dengan pendekatan dengan mekanisme message passing, representasi akhir suatu node tidak hanya mencerminkan atribut node itu sendiri, tetapi juga informasi dari node-node yang terhubung dengannya, mekanisme ini relevan karena transaksi saling berhubungan melalui aliran transaksi sehingga pola aktivitas ilegal dapat muncul dari hubungan antarnode, bukan hanya dari atribut transaksi secara individual.yang dikembangkan pada domain lain, yang menggaris bawahi pentingnya pembelajaran fitur tanpa bergantung sepenuhnya pada label eksplisit sebelum diteruskan ke tahap klasifikasi [19]. Dalam konteks jaringan *Bitcoin*, di mana transaksi saling terhubung melalui aliran dana input-output, pendekatan ini sangat relevan karena mampu menangkap pola interaksi antartransaksi yang tidak bisa ditangkap oleh metode berbasis fitur semata seperti SVM. Dua varian GNN yang dipakai dan dibandingkan dalam penelitian ini adalah *Graph Convolutional Network (GCN)* dan *Graph Attention Network (GAT)*, yang masing-masing dijelaskan pada subbab berikut.[5]

Penggunaan Graph Neural Network pada penelitian ini didasarkan pada karakteristik dataset Elliptic yang memiliki informasi fitur transaksi dan hubungan antartransaksi. Setiap transaksi direpresentasikan sebagai node, sedangkan hubungan antartransaksi direpresentasikan sebagai edge. Dengan struktur tersebut, model tidak hanya mempelajari karakteristik dari satu transaksi, tetapi juga dapat memanfaatkan informasi dari transaksi lain yang terhubung dengannya. Kemampuan ini penting karena pola transaksi ilegal dapat terbentuk dari hubungan antartransaksi dan tidak selalu terlihat apabila transaksi dianalisis secara terpisah. Oleh karena itu, Graph Neural Network dinilai sesuai untuk digunakan dalam penelitian ini.

2.1.8 *Graph Convolutional Network* (GCN)

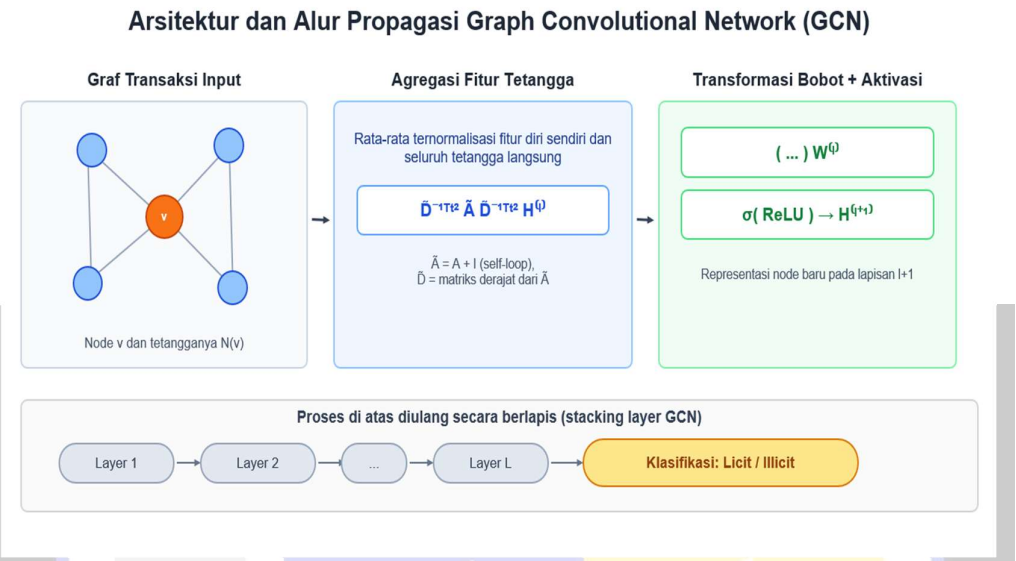
Graph Convolutional Network (GCN) merupakan salah satu arsitektur *Graph Neural Network* yang paling banyak dipakai sebagai pendekatan efisien untuk pembelajaran semi-supervised pada data berstruktur graf [5][8]. GCN bekerja dengan menggeneralisasi operasi konvolusi, yang lazim dipakai pada data grid

seperti citra, ke dalam domain graf melalui pendekatan *aproksimasi konvolusi spektral* orde pertama. Secara matematis, aturan propagasi pada satu lapisan GCN dirumuskan sebagai berikut.

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}) \quad \text{2.1}$$

dengan $\tilde{A} = A + I$ merupakan matriks ketetanggaan *(adjacency matrix)* yang sudah ditambahkan *self-loop*, $\tilde{D}$ adalah matriks derajat dari $\tilde{A}$, $H^{(l)}$ adalah matriks representasi *node* pada lapisan ke-l, $W^{(l)}$ adalah matriks bobot yang dipelajari, dan σ adalah fungsi aktivasi non-linear seperti ReLU. Secara intuitif, setiap *node* memperbarui representasinya dengan merata-ratakan (secara ternormalisasi) *fitur* dirinya sendiri beserta *fitur* seluruh *node* tetangga langsungnya, kemudian mentransformasikannya melalui bobot yang dipelajari.

Dalam konteks deteksi transaksi ilegal pada jaringan *Bitcoin, Graph Convolutional Network (GCN)* memungkinkan model mempelajari karakteristik setiap transaksi sekaligus hubungan antartransaksi yang direpresentasikan dalam bentuk *graph*. Dengan memanfaatkan informasi struktural tersebut, GCN mampu menangkap menangkap pola interaksi antarnode yang berkaitan dengan aktivitas ilegal, termasuk pola transaksi yang dapat menjadi perhatian dalam proses *anti-money laundering (AML)*. yang sulit dikenali apabila setiap transaksi dianalisis secara terpisah. Pada dataset Elliptic, representasi *graph* memberikan informasi mengenai keterkaitan antartransaksi sehingga dapat dimanfaatkan untuk meningkatkan kemampuan model dalam mengklasifikasikan transaksi legal dan ilegal. Selain itu, pendekatan berbasis GCN menunjukkan kinerja yang lebih baik dibandingkan beberapa metode *machine learning konvensional*, seperti *Logistic Regression (LR)*, *Random Forest (RF)*, dan *Multilayer Perceptron (MLP)*, karena mampu memanfaatkan informasi hubungan antartransaksi dalam jaringan Bitcoin.[20]



Sumber: disintesis dari Asiri A., Somasundaram K. (2025), "Graph convolution network for fraud detection in Bitcoin transactions", Scientific Reports, 15(1), 11076, doi: 10.1038/s41598-025-95672-w

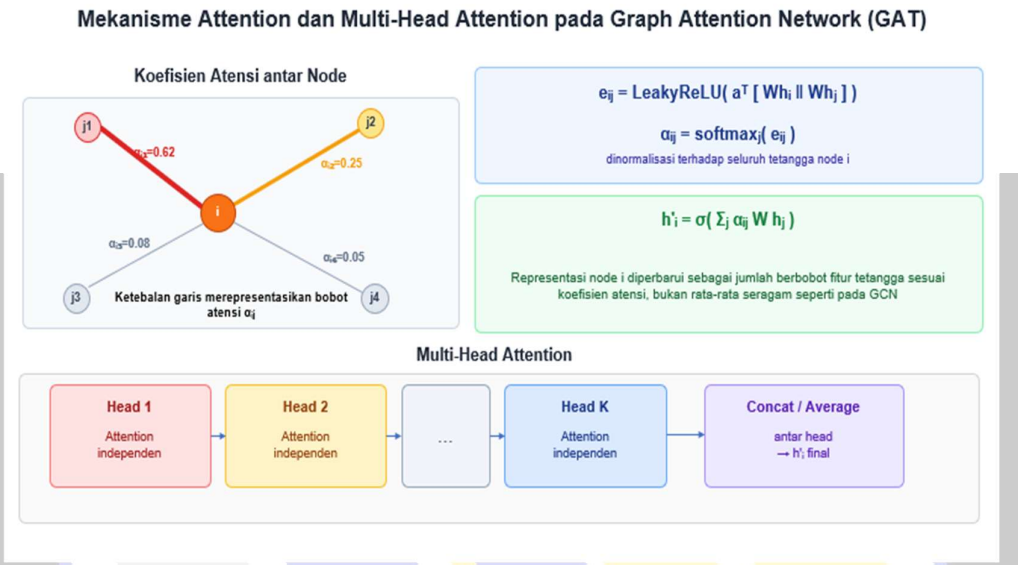
Gambar 2.3 Arsitektur dan Alur Propagasi *Graph Convolutional Network* (GCN)

2.1.9 *Graph* Attention Network (GAT)

Graph Attention Network (GAT) merupakan varian GNN yang memperkenalkan mekanisme *self-attention* ke dalam proses agregasi fitur antar*node*. Berbeda dengan GCN yang memberi bobot tetangga berdasarkan normalisasi derajat *node* (setiap tetangga diperlakukan relatif setara), GAT mempelajari bobot kepentingan (*attention coefficient*) yang berbeda untuk setiap tetangga secara adaptif selama proses pelatihan. Pendekatan ini pernah diterapkan pada konteks deteksi pencucian uang pada data transaksi finansial [16], yang menunjukkan bahwa mekanisme atensi mampu menonjolkan hubungan transaksi yang paling mencurigakan dibandingkan dengan memperlakukan seluruh tetangga secara seragam.

Koefisien atensi antar*node* i dan *node* tetangganya j umumnya dihitung melalui persamaan $e_{ij} = \text{LeakyReLU}(a^T [Wh_i \| Wh_j])$, yang kemudian dinormalisasi memakai fungsi *Softmax* terhadap seluruh tetangga *node* i, sehingga total bobot atensi berjumlah satu. Untuk menstabilkan proses pembelajaran, GAT umumnya

menerapkan *multi-head attention*, yaitu menghitung beberapa perangkat koefisien atensi secara paralel lalu menggabungkan hasilnya.[21]



Sumber: disintesis dari Sheu G.-Y., Li C.-Y. (2022), "On the Potential of a *Graph* Attention Network in *Money laundering* Detection", Journal of *Money laundering* Control, 25(3), 594-608, doi: 10.1108/JMLC-07-2021-0076

Gambar 2.4 Mekanisme *Attention* dan *Multi-Head Attention* pada *Graph Attention Network (GAT)*

Kelebihan utama *Graph Attention Network (GAT) dibandingkan Graph Convolutional Network (GCN)* adalah kemampuannya memberikan bobot perhatian *(attention weight)* yang berbeda pada setiap node tetangga berdasarkan tingkat kepentingannya. Dengan mekanisme tersebut, informasi dari node yang lebih relevan dapat memberikan kontribusi yang lebih besar terhadap proses pembelajaran dibandingkan node yang kurang relevan. Dalam deteksi transaksi ilegal pada jaringan Bitcoin, mekanisme *attention* memungkinkan model lebih efektif menangkap hubungan transaksi yang mencerminkan aktivitas ilegal sehingga meningkatkan kemampuan klasifikasi dibandingkan pendekatan yang menggunakan *agregasi* seragam [22]

2.1.10 Praproses dan Normalisasi Data (*Feature Scaling*)

Sebelum data transaksi dimasukkan ke dalam model, tahap praproses merupakan langkah yang tidak dapat diabaikan, terutama pada dataset seperti Elliptic yang memiliki banyak fitur numerik dengan rentang nilai yang berbeda-beda. Perbedaan skala antarfitur dapat menyebabkan fitur dengan nilai yang lebih besar mendominasi proses pembelajaran sehingga kontribusi fitur lain menjadi berkurang. Oleh karena itu, proses *feature scaling* dilakukan untuk menyamakan rentang nilai setiap fitur agar proses pembelajaran model menjadi lebih optimal[23]

Salah satu teknik *feature scaling* yang paling umum digunakan adalah standardisasi (*StandardScaler*), yaitu metode yang mengubah setiap fitur agar memiliki nilai rata-rata (*mean*) sebesar 0 dan simpangan baku (*standard deviation*) sebesar 1[23]. Proses ini bertujuan agar seluruh fitur berada pada skala yang sebanding sehingga algoritma pembelajaran tidak memberikan bobot yang berlebihan pada fitur yang memiliki rentang nilai lebih besar

Berbagai penelitian menunjukkan bahwa pengaruh *feature scaling* bergantung pada algoritma yang digunakan. Model berbasis *ensemble* seperti *Random Forest*, *XGBoost*, *CatBoost*, dan *LightGBM* cenderung tidak terlalu dipengaruhi oleh proses *scaling*. Sebaliknya, algoritma seperti *Logistic Regression, Support Vector Machine (SVM),* dan *Multi-Layer Perceptron (MLP)* menunjukkan perubahan kinerja yang signifikan bergantung pada metode *feature scaling* yang diterapkan. Temuan tersebut menunjukkan bahwa standardisasi merupakan tahap praproses yang penting, khususnya pada model yang memanfaatkan perhitungan jarak maupun optimasi berbasis gradien[22]**.** Pada penelitian ini digunakan metode *StandardScaler* untuk melakukan standardisasi terhadap seluruh fitur numerik pada dataset Elliptic sebelum proses pelatihan model SVM maupun GAT. Nilai rata-rata dan simpangan baku dihitung menggunakan data latih, kemudian *parameter* tersebut diterapkan secara konsisten pada data latih dan data uji untuk menghindari terjadinya *data leakage*. Dengan demikian, seluruh fitur memiliki skala yang seragam sehingga proses pelatihan model menjadi lebih stabil dan konvergensi dapat berlangsung lebih cepat.

2.1.11 *Class Weight* (Penanganan Ketidakseimbangan Kelas)

Salah satu tantangan utama dalam deteksi transaksi ilegal adalah ketidakseimbangan kelas (*class imbalance*), yaitu kondisi ketika jumlah data pada satu kelas jauh lebih sedikit dibandingkan kelas lainnya. Pada dataset Elliptic misalnya, transaksi yang berlabel illicit hanya berkisar sekitar 9-10% dari keseluruhan transaksi berlabel, sementara sisanya tergolong licit. [24] dalam tinjauan terbaru mengenai pembelajaran pada data tidak seimbang menjelaskan bahwa algoritma klasifikasi standar, yang umumnya dilatih untuk meminimalkan *error rate* secara keseluruhan, cenderung bias terhadap kelas mayoritas pada kondisi seperti ini, karena model bisa mencapai akurasi tinggi hanya dengan cenderung memprediksi kelas mayoritas, sementara kemampuannya mengenali kelas minoritas (illicit), yang justru paling penting dalam konteks anti-pencucian uang, menjadi rendah.

Dalam tinjauan yang sama, [24] mengelompokkan solusi yang umum dipakai untuk menangani data tidak seimbang ke dalam dua kategori utama, yaitu pendekatan berbasis data seperti *oversampling* dan *undersampling*, serta pendekatan berbasis algoritma seperti *cost-sensitive learning* atau *Class Weight*ing.

Class Weight merupakan salah satu pendekatan yang digunakan untuk menangani ketidakseimbangan kelas dengan memberikan bobot yang berbeda pada setiap kelas selama proses pelatihan model. Kelas yang memiliki jumlah data lebih sedikit diberikan bobot yang lebih besar, sedangkan kelas dengan jumlah data lebih banyak diberikan bobot yang lebih kecil. Tujuannya agar kesalahan prediksi pada kelas minoritas memberikan penalti yang lebih besar pada fungsi *loss*, sehingga model tidak terlalu dominan mempelajari kelas mayoritas. Pendekatan pembobotan ini dapat digunakan tanpa menambah atau mengurangi jumlah data seperti pada metode *oversampling* maupun *undersampling*. pada data tidak seimbang, metode klasifikasi cenderung bias terhadap kelas mayoritas, sehingga pemberian bobot pada kelas minoritas dapat digunakan untuk meningkatkan perhatian model terhadap kelas yang jumlah datanya lebih sedikit.

$$w_c = \frac{N}{K \times N_c} \quad \text{.....2.2}$$

wc adalah bobot untuk kelas ke-c, N adalah jumlah seluruh data training, K adalah jumlah kelas, dan Nc adalah jumlah data training pada kelas ke-c. Berdasarkan persamaan tersebut, kelas yang memiliki jumlah data lebih sedikit akan memperoleh bobot yang lebih besar. Sebaliknya, kelas yang memiliki jumlah data lebih banyak akan memperoleh bobot yang lebih kecil. Bobot tersebut kemudian digunakan pada fungsi *loss* selama proses pelatihan model.

Class Weight bekerja dengan memberikan bobot penalti yang lebih besar terhadap kesalahan klasifikasi pada kelas minoritas di dalam fungsi *loss*, sehingga model terdorong untuk memberikan perhatian yang lebih besar terhadap kelas tersebut selama proses pelatihan tanpa perlu melakukan penambahan maupun pengurangan data melalui teknik *resampling*. Pendekatan ini banyak digunakan untuk mengatasi permasalahan *class imbalance* karena mampu mengurangi kecenderungan model memprediksi kelas mayoritas secara berlebihan [25].

Skema pembobotan yang umum digunakan adalah *balanced class weight*, yaitu pemberian bobot yang berbanding terbalik dengan jumlah sampel pada setiap kelas. Dengan demikian, semakin sedikit jumlah data pada suatu kelas, semakin besar bobot yang diberikan pada kelas tersebut selama proses optimasi. Penerapan *Class Weight* bertujuan meningkatkan kemampuan model dalam mengenali kelas minoritas, yang umumnya ditunjukkan melalui peningkatan nilai *recall*, meskipun pada beberapa kasus dapat disertai *trade-off* terhadap nilai *precision* sehingga diperlukan keseimbangan dalam proses pelatihan[26]

Class imbalance merupakan kondisi ketika jumlah sampel pada suatu kelas jauh lebih banyak dibandingkan kelas lainnya. Pada kondisi tersebut, proses pelatihan model dapat lebih banyak dipengaruhi oleh kelas mayoritas sehingga kemampuan model dalam mengenali kelas minoritas menjadi lebih rendah. Permasalahan ini menjadi penting dalam deteksi fraud karena transaksi ilegal pada umumnya memiliki jumlah yang lebih sedikit dibandingkan transaksi normal.

Salah satu pendekatan yang dapat digunakan untuk menangani kondisi tersebut adalah *cost-sensitive learning* melalui pemberian *Class Weight*. Pendekatan ini memberikan bobot yang berbeda pada setiap kelas berdasarkan distribusi data. Kelas dengan jumlah sampel yang lebih sedikit diberikan bobot yang lebih besar sehingga kesalahan prediksi terhadap kelas minoritas menghasilkan penalti *loss* yang lebih besar. Dengan demikian, model didorong untuk tidak hanya mengoptimalkan prediksi terhadap kelas mayoritas, tetapi juga meningkatkan sensitivitas terhadap kelas minoritas.

Dalam penelitian berbasis graph, pendekatan *cost-sensitive learning* telah digunakan untuk menangani ketidakseimbangan kelas pada kasus deteksi fraud. GAT-COBO dengan menggabungkan *Graph Attention Network* dan *cost-sensitive learning* untuk meningkatkan perhatian model terhadap kelas minoritas pada graph yang tidak seimbang. Pendekatan tersebut menunjukkan bahwa pemberian biaya yang lebih besar terhadap kesalahan klasifikasi kelas minoritas dapat membantu model menangani permasalahan *graph imbalance*.[27]

2.1.12 Optimisasi Model (*Adam Optimizer* dan *Cross-entropy loss*)

Setelah arsitektur model ditentukan, proses pelatihan (*training*) membutuhkan sebuah algoritma optimisasi yang bertugas memperbarui bobot model secara bertahap agar nilai fungsi loss semakin kecil. Salah satu algoritma optimisasi yang paling banyak dipakai dalam pelatihan *deep learning*, termasuk *Graph Neural Network*, adalah Adam (*Adaptive Moment Estimation*). Berbeda dengan *Stochastic Gradient Descent* (SGD) konvensional yang memakai satu nilai learning rate tetap untuk seluruh *parameter*, Adam menghitung *learning rate adaptif* untuk setiap *parameter* secara individual berdasarkan estimasi momen pertama (rata-rata *gradien*) dan momen kedua (rata-rata kuadrat *gradien*) dari *gradien* sebelumnya [28]. dalam tinjauan terhadap berbagai algoritma optimisasi pada jaringan saraf modern menjelaskan bahwa keunggulan utama Adam terletak pada kemampuannya menyesuaikan laju pembelajaran secara otomatis untuk setiap parameter, sehingga proses konvergensi cenderung lebih cepat dan lebih stabil dibandingkan algoritma *gradient descent konvensional*, terutama pada model

dengan jumlah parameter yang besar maupun data yang jarang (*sparse*). Karakteristik inilah yang membuat Adam menjadi pilihan *default* pada banyak arsitektur *Graph Neural Network*, termasuk GAT yang dipakai pada penelitian ini.

Fungsi *loss* merupakan komponen penting dalam proses pelatihan model *deep learning* karena digunakan untuk mengukur tingkat kesalahan prediksi terhadap label sebenarnya. Pada setiap iterasi pelatihan, nilai *loss* dihitung berdasarkan selisih antara hasil prediksi model dan label target, kemudian digunakan sebagai dasar untuk memperbarui *parameter* model melalui proses optimisasi. Semakin kecil nilai *loss* yang diperoleh, semakin dekat hasil prediksi model dengan label sebenarnya sehingga menunjukkan bahwa proses pembelajaran berlangsung dengan baik [29] Pada tugas klasifikasi, salah satu fungsi *loss* yang paling banyak digunakan adalah *Cross-Entropy Loss*. Fungsi ini mengukur perbedaan antara distribusi probabilitas hasil prediksi model dengan distribusi label sebenarnya. Apabila probabilitas yang diberikan model terhadap kelas yang benar semakin tinggi, maka nilai *Cross-Entropy Loss* akan semakin kecil. Sebaliknya, apabila prediksi model jauh dari label sebenarnya, nilai *loss* akan meningkat sehingga model memperoleh penalti yang lebih besar selama proses pelatihan

Pada penelitian ini, *Cross-Entropy Loss* digunakan sebagai fungsi *objektif* untuk melatih model *Graph Attention Network (GAT)*. Mengingat dataset Elliptic memiliki distribusi kelas yang tidak seimbang, perhitungan *loss* dikombinasikan dengan *Class Weight* sehingga kesalahan klasifikasi pada kelas minoritas diberikan bobot yang lebih besar dibandingkan kelas mayoritas. Pendekatan ini membantu model memberikan perhatian yang lebih besar terhadap transaksi ilegal (*illicit transaction*) tanpa perlu melakukan proses *resampling* pada data latih [30]

2.1.13 Evaluasi Model Klasifikasi

Untuk mengetahui seberapa baik model dalam mengenali transaksi ilegal pada jaringan *Bitcoin*, diperlukan metrik evaluasi yang mampu menggambarkan performa model secara menyeluruh. Pada dataset yang memiliki distribusi kelas tidak seimbang, penggunaan satu metrik saja, seperti *Accuracy*, tidak cukup untuk merepresentasikan kemampuan model dalam mengenali kelas minoritas. Oleh

karena itu, evaluasi model umumnya dilakukan menggunakan beberapa metrik yang saling melengkapi, seperti *Confusion Matrix*, *Accuracy*, *Precision*, *Recall*, *F1-score*, dan *ROC-AUC*, sehingga memberikan gambaran yang lebih komprehensif terhadap performa klasifikasi [31]

Confusion Matrix merupakan dasar dalam evaluasi model klasifikasi karena menyajikan jumlah prediksi yang benar maupun salah melalui empat komponen utama, yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. Keempat komponen tersebut menjadi dasar dalam perhitungan berbagai metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, *F1-score*, Sementara itu, ROC-AUC digunakan untuk mengukur kemampuan model membedakan kelas positif dan negatif pada berbagai nilai *threshold* berdasarkan skor atau probabilitas prediksi, sehingga memungkinkan analisis performa model secara lebih rinci [32]**.** Berdasarkan nilai pada *Confusion Matrix*, beberapa metrik evaluasi dapat dihitung. *Accuracy* menunjukkan proporsi prediksi yang benar terhadap seluruh data, namun pada dataset yang tidak seimbang metrik ini dapat memberikan gambaran yang kurang representatif. *Precision* mengukur proporsi prediksi positif yang benar, sedangkan *Recall* menunjukkan kemampuan model dalam menemukan seluruh data positif yang sebenarnya. *F1-score* merupakan rata-rata harmonik antara *Precision* dan *Recall*, sedangkan *ROC-AUC* mengukur kemampuan model dalam membedakan kelas positif dan negatif pada berbagai nilai *threshold*. Oleh karena itu, kombinasi beberapa metrik lebih disarankan dibandingkan hanya menggunakan *Accuracy* [33]

Pada penelitian ini, *Confusion Matrix*, *Accuracy*, *Precision*, *Recall*, *F1-score*, dan *ROC-AUC* digunakan secara bersamaan untuk mengevaluasi dan membandingkan performa model SVM, GCN, GCN dengan *Class Weight*, GAT, serta GAT dengan *Class Weight*. Mengingat tujuan utama penelitian adalah meningkatkan kemampuan deteksi transaksi ilegal sebagai kelas minoritas, maka metrik *Recall* dan *F1-score* menjadi indikator utama dalam mengevaluasi keberhasilan model [34]. Selain metrik evaluasi tersebut, pendekatan *Macro Average* dan *Micro Average* juga digunakan untuk memberikan gambaran performa model secara lebih menyeluruh. *Macro Average* menghitung nilai *Precision*, *Recall*,

dan *F1-score* pada setiap kelas secara terpisah, kemudian merata-ratakannya sehingga setiap kelas memiliki kontribusi yang sama tanpa dipengaruhi jumlah sampelnya. Sebaliknya, *Micro Average* menghitung seluruh *True Positive*, *False Positive*, dan *False Negative* dari semua kelas secara keseluruhan sebelum menghitung nilai evaluasi, sehingga hasilnya lebih dipengaruhi oleh kelas dengan jumlah data yang lebih besar. Kedua pendekatan tersebut banyak digunakan untuk mengevaluasi model pada dataset dengan distribusi kelas yang tidak seimbang [35]

a) *Classification Report*

Klasifikasi dalam pembelajaran mesin merupakan alat evaluasi yang penting untuk menilai kinerja model dalam memprediksi kelas dari objek baru berdasarkan data yang ada. Proses klasifikasi biasanya melibatkan dua tahap utama: pembelajaran untuk membuat model dan pengujian untuk mengevaluasi model tersebut. Kinerja model diukur dengan menggunakan berbagai metrik, seperti *Accuracy, presisi, recall,* dan *F1-score*, yang semuanya diperoleh dari tabel *Confusion Matrix*. Tabel ini membantu dalam mengetahui jumlah prediksi yang tepat dan salah, sehingga mempermudah dalam mengevaluasi efektivitas model klasifikasi [36] [37].

Accuracy adalah metrik fundamental dalam evaluasi model pembelajaran mesin, yang diukur sebagai proporsi prediksi yang benar dibandingkan dengan total prediksi yang dilakukan. Secara matematis, akurasi dihitung dengan rumus:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad \text{.....2.3}$$

Accuracy mengukur seberapa banyak prediksi model yang benar dibandingkan dengan total semua prediksi. Formula ini menjumlahkan jumlah *True Positives (TP)* dan *True Negatives (TN)*, lalu membaginya dengan total semua prediksi *(TP + TN + FP + FN)*, kemudian dikalikan 100 agar hasilnya dalam *persentase*. Semakin tinggi nilai *Accuracy*, semakin baik performa model dalam mengklasifikasikan data dengan benar.

Akurasi tinggi menunjukkan bahwa model memiliki kinerja yang baik dalam mengklasifikasikan data. Namun, akurasi bisa menyesatkan dalam konteks *dataset* yang tidak seimbang. Dalam kondisi tersebut, metrik lain seperti *presisi* dan *recall* menjadi lebih relevan untuk memberikan wawasan yang lebih mendalam mengenai kemampuan model dalam mendeteksi kelas positif. Beberapa studi menunjukkan bahwa dengan menggunakan model yang berbeda, mereka dapat mencapai akurasi yang substansial. Misalnya, sebuah penelitian menggunakan algoritma untuk mengidentifikasi berbagai patologi menunjukkan akurasi dari 90% hingga 96% [38]Fokus pada pengukuran akurasi dan penggunaan metrik ini telah diperluas untuk mencakup analisis mendalam dari *dataset* dan model yang berbeda.[32]

Presisi (precision) adalah *metrik* yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam pembelajaran mesin. *Metrik* ini dihitung dengan membandingkan jumlah prediksi positif yang benar (*True Positives*) dengan jumlah total prediksi positif yang dihasilkan oleh model, termasuk yang salah (*False Positives*). Secara matematis, presisi dapat dinyatakan dengan rumus:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positives\ (TP)}}{\boldsymbol{True\ Positives\ (TP)+False\ Positives\ (FP)}} \quad \text{.....2.4}$$

Precision merupakan metrik evaluasi yang digunakan untuk mengukur ketepatan model dalam menghasilkan prediksi positif. Metrik ini menunjukkan proporsi data yang diprediksi sebagai kelas positif dan benar-benar termasuk ke dalam kelas positif. Nilai precision yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan False Positive yang rendah, sehingga prediksi positif yang dihasilkan lebih dapat dipercaya

Precision sangat bermanfaat dalam situasi di mana kesalahan dalam klasifikasi positif memiliki dampak besar, misalnya dalam konteks diagnosis medis atau deteksi penipuan, di mana hasil positif palsu dapat menyebabkan biaya yang signifikan atau kesalahpahaman. Melalui analisis metrik ini, para peneliti dapat menentukan efektivitas model dalam memastikan bahwa sebagian besar dari prediksi positif yang dilakukan adalah benar, yang meningkatkan kepercayaan

pengguna terhadap sistem. Selain itu, presisi sering digunakan bersamaan dengan metrik lain seperti *recall* untuk mendapatkan evaluasi kinerja yang lebih menyeluruh dari model.

Recall merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam menemukan seluruh data yang sebenarnya termasuk ke dalam kelas positif. Nilai Recall menunjukkan seberapa besar proporsi data positif yang berhasil diklasifikasikan dengan benar oleh model dibandingkan dengan seluruh data yang sebenarnya positif. Dalam perhitungannya, Recall menggunakan nilai *True Positive* (TP) dan *False Negative* (FN) yang diperoleh dari *confusion matrix*. *True Positive* menunjukkan jumlah data positif yang berhasil diprediksi dengan benar sebagai positif, sedangkan *False Negative* menunjukkan jumlah data positif yang salah diprediksi sebagai negatif. Recall dapat dihitung menggunakan persamaan berikut [39]

$$\boldsymbol{Recall} = \frac{TP}{TP+FN} \quad \text{.....2.5}$$

TP = *True Positive*, yaitu jumlah data positif yang berhasil diprediksi dengan benar sebagai positif.
FN = *False Negative*, yaitu jumlah data positif yang salah diprediksi sebagai negatif.

Nilai Recall berada pada rentang 0 sampai 1 atau dapat dinyatakan dalam bentuk persentase. Semakin tinggi nilai Recall, semakin besar kemampuan model dalam menemukan data yang benar-benar termasuk kelas positif. Sebaliknya, nilai Recall yang rendah menunjukkan bahwa masih terdapat banyak data positif yang tidak berhasil dikenali oleh model atau menghasilkan *False Negative*. Dalam penelitian ini, kelas positif merupakan transaksi Illicit**,** sedangkan kelas negatif merupakan transaksi Licit**.** Oleh karena itu, Recall menunjukkan kemampuan model dalam mendeteksi transaksi yang sebenarnya Illicit. Recall menjadi salah satu metrik yang diprioritaskan karena penelitian berfokus pada deteksi transaksi ilegal untuk mendukung kebutuhan digital forensik. Nilai Recall yang tinggi menunjukkan semakin sedikit transaksi Illicit yang tidak berhasil terdeteksi atau menjadi *False*

Negative. Meskipun demikian, Recall tetap dianalisis bersama dengan Precision, F1-score, Accuracy, dan ROC-AUC agar performa model dapat dievaluasi secara lebih menyeluruh, khususnya karena dataset yang digunakan memiliki distribusi kelas yang tidak seimbang.

F1-score merupakan metrik evaluasi yang banyak digunakan untuk mengukur kinerja model klasifikasi, terutama ketika menghadapi dataset dengan distribusi kelas yang tidak seimbang. Nilai *F1-score* diperoleh dari rata-rata harmonik (*harmonic mean*) antara *precision* dan *recall*, sehingga mampu memberikan ukuran yang seimbang terhadap kemampuan model dalam menghasilkan prediksi positif yang benar sekaligus mengenali seluruh data positif yang sebenarnya. Oleh karena itu, *F1-score* sering digunakan sebagai metrik utama ketika precision dan recall memiliki tingkat kepentingan yang sama dalam proses evaluasi model [40].

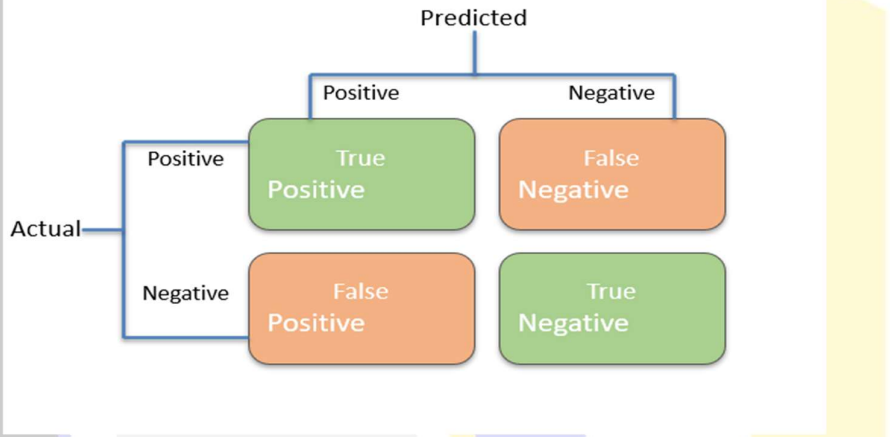
Secara matematis, dihitung dengan rumus berikut:

$$\boldsymbol{F1-score} = 2 \times \frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Precision + Recall}} \quad \text{.....2.6}$$

Persamaan tersebut menunjukkan bahwa nilai F1-score akan tinggi apabila nilai precision dan recall sama-sama tinggi. Sebaliknya, apabila salah satu metrik memiliki nilai rendah, maka nilai F1-score juga akan menurun. Karakteristik ini menjadikan F1-score lebih representatif dibandingkan hanya menggunakan accuracy, khususnya pada permasalahan klasifikasi dengan distribusi kelas yang tidak seimbang.[31] Pada penelitian ini, F1-score digunakan sebagai salah satu indikator utama untuk mengevaluasi performa model SVM, GCN, GCN dengan Class Weight, GAT, dan GAT dengan Class Weight. Pemilihan metrik ini didasarkan pada karakteristik dataset Elliptic yang memiliki ketidakseimbangan distribusi kelas, sehingga diperlukan ukuran evaluasi yang mampu memberikan gambaran seimbang terhadap kemampuan model dalam mengenali transaksi ilegal tanpa hanya bergantung pada nilai accuracy

b) *Confusion Matrix*

Confusion Matrix memainkan peran penting dalam analisis kinerja model, karena memberikan gambaran yang komprehensif tentang prediksi yang dihasilkan. Dengan analisis tersebut, para peneliti dapat menentukan kapan model bekerja baik dan kapan tidak, serta melakukan perbaikan yang diperlukan[32]. Pendekatan ini tidak hanya mendasar untuk meningkatkan akurasi, tetapi juga untuk memahami bagaimana algoritma melakukan klasifikasi dalam konteks data yang tidak seimbang dan kompleks.[41]



Sumber: Diolah oleh penulis berdasarkan Opitz (2024), *A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice*.

Gambar 2.5 *Confusion Matrix*

2.1.14 t-SNE (Visualisasi *Embedding*)

Selain dievaluasi secara kuantitatif, representasi yang dipelajari oleh model berbasis graph juga diamati secara visual untuk memahami pemisahan transaksi Licit dan Illicit dalam ruang *embedding*. Dalam penelitian ini, visualisasi dilakukan terhadap GCN, GCN + *Class Weight*, GAT, dan GAT + *Class Weight* menggunakan *t-distributed Stochastic Neighbor Embedding (t-SNE),* sebuah teknik reduksi dimensi *non-linear* yang dirancang khusus untuk keperluan visualisasi data berdimensi tinggi [42] dalam tinjauan mengenai t-SNE menjelaskan bahwa teknik

ini bekerja dengan memodelkan kemiripan antar titik data pada ruang berdimensi tinggi sebagai distribusi probabilitas, kemudian berusaha mencari representasi pada ruang berdimensi rendah (biasanya dua dimensi) yang distribusi probabilitasnya semirip mungkin dengan distribusi aslinya. Dengan cara ini, titik-titik data yang saling berdekatan pada ruang berdimensi tinggi akan cenderung digambarkan berdekatan pula pada ruang berdimensi rendah, sementara titik-titik yang jauh berbeda akan digambarkan saling berjauhan, sehingga pola pengelompokan (*clustering*) data menjadi lebih mudah diamati secara visual.

Dalam konteks penelitian ini, t-SNE dipakai untuk memvisualisasikan *embedding node* hasil pembelajaran model GCN, guna mengamati apakah transaksi berlabel illicit membentuk kelompok (*cluster*) yang cukup terpisah dari transaksi berlabel licit setelah melalui proses pelatihan. Visualisasi ini menjadi pelengkap penting bagi metrik evaluasi kuantitatif pada subbab 2.1.12, karena mampu memberi gambaran intuitif mengenai bagaimana model "memahami" perbedaan pola antar kedua kelas transaksi tersebut, sekaligus membantu mengidentifikasi apakah masih ada tumpang tindih (*overlap*) antarkelas yang berpotensi menjadi sumber kesalahan klasifikasi.

2.1.15 Sintesis Kerangka Teori

Berdasarkan tinjauan teori di atas, dapat disintesiskan bahwa transaksi pada jaringan *Bitcoin* secara alamiah membentuk struktur graf (2.1.1), yang di dalamnya bisa tersembunyi pola transaksi ilegal yang hanya bisa dikenali lewat hubungan antar transaksi, bukan atribut tunggal (2.1.3). Kebutuhan untuk menelusuri dan membuktikan pola tersebut menjadikan pendekatan ini relevan dengan kerja *digital forensik* (2.1.4). Metode *Machine learning* konvensional seperti SVM (2.1.6) terbukti efektif sebagai *baseline*, namun memiliki keterbatasan karena tidak memanfaatkan struktur relasional data. Pendekatan *Graph Neural Network*, baik lewat GCN (2.1.8) maupun GAT (2.1.9), dirancang untuk mengatasi keterbatasan tersebut dengan memanfaatkan informasi topologi graf transaksi.

Sebelum data tersebut bisa dipelajari oleh model, tahap praproses dan normalisasi data (2.1.10) diperlukan agar setiap fitur berkontribusi secara proporsional dan proses pelatihan berjalan stabil. Karena distribusi kelas pada data transaksi ilegal sangat tidak seimbang, penerapan teknik *Class Weight* (2.1.11) turut diperlukan agar model tidak bias terhadap kelas mayoritas. Proses pelatihan model itu sendiri bergantung pada algoritma optimisasi Adam beserta fungsi *cross-entropy loss* (2.1.12) untuk memperbarui bobot model secara bertahap. Setelah model selesai dilatih, performanya diukur secara kuantitatif lewat *Confusion Matrix*, *Accuracy*, *precision*, *recall*, *F1-score*, dan ROC-AUC (2.1.13), serta diamati secara visual lewat teknik t-SNE (2.1.14) untuk melihat sejauh mana kelas legal dan ilegal terpisah dalam ruang representasi model.

Keseluruhan landasan teori tersebut menjadi dasar dalam perancangan, implementasi, dan evaluasi model pada penelitian ini, sebagaimana dijabarkan lebih lanjut pada Bab III, yaitu *Bitcoin*, *Blockchain*, SVM, GCN, GAT, praproses dan normalisasi data, *Class Weight*, optimisasi model, evaluasi model klasifikasi, *digital forensik*, serta *t-SNE*, yang menjadi dasar perancangan dan evaluasi model pada penelitian ini, sebagaimana akan dijabarkan lebih lanjut pada Bab III.

2.2. Tinjauan Penelitian Sebelumnya

Penulis telah melakukan tinjauan penelitian untuk benar-benar mengonfirmasi metode klasifikasi yang digunakan dalam penelitian sebelumnya. Semua hasil tinjauan penelitian tersebut dapat dijadikan perbandingan dengan penelitian yang akan dilakukan. Hasil tinjauan ini diharapkan dapat menjadi acuan bagi setiap penulis terhadap metode klasifikasi yang telah dipelajari sebelumnya.

Adapun sepuluh penelitian terdahulu yang dijadikan acuan dalam penelitian ini dilakukan oleh:

Asiri dan Somasundaram [8] mengembangkan model *Graph* Convolution Network (GCN) untuk mendeteksi transaksi ilegal (*fraud*) pada jaringan *Bitcoin* dalam konteks anti pencucian uang (anti-*money laundering*). Penelitian ini menggunakan Elliptic *Bitcoin* Dataset, yaitu sebuah dataset graf transaksi *Bitcoin* berskala besar dengan 203.769 *node* dan 234.355 *edge*, di mana setiap *node*

merepresentasikan sebuah transaksi yang dikaitkan dengan 166 fitur. Dataset memiliki 166 atribut numerik setelah txId, yang terdiri atas 1 atribut time step dan 165 fitur transaksi dan diberi label licit, illicit, atau *Unknown*. Kategori seperti miners, wallet providers, dan exchanges dikelompokkan sebagai licit, sedangkan kategori seperti *ransomware*, skema *Ponzi*, dan *malware* dikelompokkan sebagai illicit. Transaksi *Bitcoin* dikonstruksi sebagai graf berarah, di mana setiap transaksi menjadi simpul *(node)* dan aliran transaksi dari satu transaksi ke transaksi lain direpresentasikan sebagai sisi (*edge*). Model GCN kemudian digunakan untuk mempelajari struktur graf tersebut guna mengklasifikasikan setiap transaksi ke dalam kategori legal maupun ilegal.

Ouyang et al. [43] mengembangkan metode *Bitcoin Money laundering* Detection via Sub*graph* Contrastive Learning untuk mendeteksi aktivitas pencucian uang (*money laundering*) pada jaringan *Bitcoin*. Berbeda dengan penelitian sebelumnya yang umumnya hanya mengklasifikasikan transaksi secara individual (*node* classification), penelitian ini berfokus pada identifikasi kelompok transaksi (sub*graph*) yang memiliki pola perilaku serupa dalam aktivitas pencucian uang. Untuk mengatasi karakteristik data *Bitcoin* yang kompleks, heterogen, dan mengandung banyak *noise*, peneliti mengusulkan model Bit-CHetG (*Bitcoin* Contrastive Heterogeneous *Graph*) yang dibangun menggunakan heterogeneous *Graph Neural Network* dan sub*graph* contrastive learning. Model ini membentuk address–transaction heterogeneous *graph*, yaitu graf yang menghubungkan alamat dompet (wallet address) dengan transaksi *Bitcoin*, kemudian memanfaatkan metapath untuk membangun subgraf yang merepresentasikan hubungan antartransaksi dan alamat dompet. Selanjutnya, model menghasilkan representasi (*embedding*) dari subgraf transaksi dan alamat dompet menggunakan *Graph Neural Network*, kemudian menerapkan *supervised contrastive learning* untuk memperkecil pengaruh *noise* dengan mendekatkan subgraf yang berasal dari kelas yang sama dan menjauhkan subgraf dari kelas yang berbeda. Hasil pengujian pada dua dataset *Bitcoin* menunjukkan bahwa model Bit-CHetG mampu meningkatkan nilai Micro *F1-score* sedikitnya 5% dibandingkan beberapa metode sebelumnya seperti *Random Forest*, GCN, *Inspection-L*, SubGNN, HAN, MAGNN, dan TSGN,

sehingga dinilai lebih efektif dalam mendeteksi pola aktivitas pencucian uang pada jaringan *Bitcoin*.

Alarab dan Prakoonwit[6] mengembangkan model recurrent *Graph Neural Network* (RecGNN) yang mengekstrak informasi temporal sekaligus topologi graf dari data transaksi *Bitcoin* untuk melakukan klasifikasi *node* sebagai transaksi licit atau illicit. Model ini melakukan prediksi secara sekuensial dengan memanfaatkan fitur-fitur dari transaksi bertetangga yang telah diberi label terlebih dahulu (*antecedent neighbouring features*), sehingga mampu menangkap peristiwa penutupan pasar gelap (*dark market shutdown*) yang tercatat dalam dataset Elliptic namun sulit dideteksi oleh algoritma pembelajaran konvensional menggunakan fitur aslinya. Pengujian pada dataset Elliptic menunjukkan model RecGNN mencapai akurasi sebesar 98,99% dan *F1-score* sebesar 91,75%, hasil yang tercatat sebagai *state-of-the-art* dibandingkan berbagai model pembanding lainnya. Penulis juga melakukan studi kasus dengan memvisualisasikan cuplikan graf transaksi *Bitcoin* dan proses penalaran mundur (*backward reasoning*) untuk menjelaskan bagaimana suatu transaksi diklasifikasikan sebagai ilegal.

Li dan He [44] mengusulkan model pelacakan transaksi *Bitcoin* berbasis *Graph Neural Network* yang diberi nama BT2 (*Bitcoin Transaction Tracking Model*) untuk mengatasi kesulitan melacak hubungan transaksi antar pengguna akibat sifat anonimitas dan penyimpanan terdistribusi pada teknologi *Blockchain*. Model ini menerapkan metode agregasi induktif *(inductive aggregation)* untuk menghasilkan *embedding node* yang kaya informasi, termasuk bagi sejumlah kecil *node* yang belum teramati sebelumnya. Selanjutnya, vektor *embedding node* tersebut digunakan untuk menghasilkan informasi sisi (*edge*) antar*node* melalui fungsi *message-passing*, sehingga diperoleh relasi antar akun *Bitcoin*. Penelitian ini turut mengevaluasi pengaruh berbagai *parameter,* seperti kedalaman jaringan (network depth) dan jumlah iterasi, terhadap performa model. Hasil eksperimen pada dataset dunia nyata menunjukkan model BT2 mampu mencapai rata-rata AUC (area under the ROC curve) hingga 0,93 dan akurasi rata-rata sebesar 86%, yang menunjukkan performa lebih baik dibandingkan metode-metode *state-of-the-art* lainnya.

Cheng dkk.[45] mengembangkan sistem deteksi anti pencucian uang (AML) berbasis deep *graph* learning yang memperhitungkan perilaku kelompok (group-aware), dengan menggabungkan analisis jaringan (network *analysis*) dan pembelajaran terawasi (*supervised learning*) dalam satu kerangka deteksi yang dirancang untuk aplikasi dunia nyata. Pendekatan ini memodelkan tahapan proses pencucian uang, mulai dari placement (memasukkan sejumlah besar uang kotor ke dalam sistem keuangan) hingga *layering* (upaya penyamaran melalui transaksi berulang lintas akun, mata uang, atau pasar), dengan memanfaatkan representasi graf untuk mengenali pola kelompok mencurigakan yang tidak dapat ditangkap oleh model berbasis transaksi individual. Model diuji pada dataset transaksi nyata dari salah satu aliansi kartu bank terbesar di dunia. Hasil eksperimen menunjukkan bahwa metode yang diusulkan mengungguli metode-metode *state-of-the-art* lainnya, baik dalam mode operasi offline maupun online, sehingga membuktikan efektivitas pendekatan berbasis kesadaran kelompok dalam mendeteksi aktivitas pencucian uang secara lebih akurat.

Wan dan Li [46] mengusulkan model Dynamic *Graph* Convolutional *Neural Network* and *Long Short-Term Memory* (DGCNN-LSTM) untuk memprediksi aktivitas pencucian uang (*money laundering*) pada jaringan *Blockchain*. Pendekatan ini mengombinasikan *Dynamic Graph Convolutional Neural Network* (DGCNN) untuk mempelajari hubungan antartransaksi dalam struktur *graph* dengan *Long Short-Term Memory (LSTM)* untuk menangkap pola transaksi yang berubah dari waktu ke waktu. Melalui kombinasi kedua metode tersebut, model mampu memanfaatkan informasi spasial dan temporal secara bersamaan sehingga representasi transaksi menjadi lebih akurat. Hasil penelitian menunjukkan bahwa model DGCNN-LSTM memiliki kemampuan yang lebih baik dalam mengidentifikasi transaksi yang terindikasi sebagai aktivitas pencucian uang dibandingkan beberapa metode pembanding, sehingga berpotensi meningkatkan efektivitas proses deteksi pada jaringan *Blockchain*.

Lo dkk. [47] mengusulkan *Inspection-L*, sebuah *framework Graph Neural Network* yang menggabungkan metode *self-supervised Deep Graph Infomax (DGI)* dan *Graph Isomorphism Network (GIN)* dengan algoritma pembelajaran terawasi

Random Forest (RF) untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Penelitian ini merupakan penelitian pertama yang menerapkan pendekatan *self-supervised* GNN pada permasalahan anti pencucian uang (AML) di *Bitcoin*, dengan tujuan menghasilkan *embedding node* berkualitas tinggi meskipun data berlabel yang tersedia sangat terbatas, sebuah kondisi yang umum terjadi pada dataset transaksi kriptocurrency di dunia nyata. Dengan memanfaatkan representasi yang dipelajari secara *self-supervised* sebagai fitur tambahan bagi *classifier Random Forest*, pendekatan ini dirancang untuk meningkatkan kemampuan model dalam mengenali pola-pola transaksi mencurigakan dibandingkan pendekatan *supervised* murni.

Liu dkk.[48] mengusulkan metode *Graph Embedding*-Based *Money laundering Detection for Ethereum* untuk mendeteksi aktivitas pencucian uang pada jaringan Ethereum. Penelitian ini memanfaatkan teknik *graph embedding* untuk merepresentasikan hubungan antar alamat dompet dan transaksi ke dalam bentuk vektor berdimensi rendah sehingga karakteristik jaringan *Blockchain* dapat dipelajari secara lebih efektif. Representasi *graph* tersebut kemudian digunakan sebagai masukan bagi model klasifikasi untuk membedakan transaksi normal dan transaksi yang terindikasi sebagai aktivitas pencucian uang. Hasil penelitian menunjukkan bahwa pendekatan berbasis *graph embedding* mampu meningkatkan performa deteksi dibandingkan metode yang hanya memanfaatkan fitur transaksi, karena mampu menangkap hubungan struktural antar entitas dalam jaringan *Blockchain*.

Qianyu dkk.[49] mengusulkan RMGANets, yaitu jaringan *graph*-aware berbasis atensi multi-relasi (*multi-relational attention*) yang ditingkatkan dengan *reinforcement learning* untuk mendeteksi aktivitas pencucian uang. Mengingat sifat anonim dan kompleksitas transaksi ilegal yang membuat metode *deep learning konvensional* kesulitan membangun korelasi antar alamat transaksi, model ini dibangun dengan cara membentuk sejumlah subgraf dari jaringan transaksi, kemudian mempelajari representasi dari subgraf-subgraf tersebut beserta *node-node* yang saling berhubungan (converging *node*s) untuk memperkaya informasi lokal, global, dan kontekstual. Penggunaan *node* yang berulang di antara subgraf-

subgraf tersebut turut meningkatkan interaktivitas antar subgraf, mengurangi ambiguitas intra-kelas, dan mempertegas perbedaan antar-kelas. Mekanisme *reinforcement learning* digunakan untuk mengarahkan proses pembelajaran representasi agar lebih adaptif terhadap pola-pola pencucian uang yang kompleks, sehingga meningkatkan akurasi deteksi pada jaringan transaksi yang besar dan anonim.

Lin dkk. [50] mengembangkan model *ChronoWave-GNN* melalui penelitian berjudul Detecting Illicit Transactions in *Bitcoin*: A Wavelet-Temporal *Graph* Transformer Approach for Anti-*Money laundering* untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Penelitian ini menggunakan Elliptic *Bitcoin* Dataset yang merepresentasikan hubungan antartransaksi dalam bentuk *graph*. Model yang diusulkan menggabungkan Wavelet Transform, Temporal Encoding, dan *Graph Transformer* untuk mempelajari karakteristik struktural serta pola temporal transaksi secara bersamaan. Pendekatan tersebut memungkinkan model menangkap perubahan pola transaksi ilegal yang berkembang dari waktu ke waktu sehingga menghasilkan representasi *graph* yang lebih akurat. Hasil penelitian menunjukkan bahwa model *ChronoWave-GNN* memperoleh akurasi sebesar 98,02% dan *F1-score* sebesar 97,99%, serta mampu memberikan performa yang lebih baik dibandingkan beberapa metode *Graph Neural Network* sebelumnya dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*.

Tabel 2.1 Penelitian Terdahulu

No	Pengarang (Tahun)	Judul	Hasil Penelitian
1	Asiri & Somasundara	*Graph* convolution network for *fraud* detection	Penerapan GCN pada Elliptic *Bitcoin* Dataset berhasil mengklasifikasikan transaksi ke dalam kategori legal dan ilegal dengan memanfaatkan hubungan antartransaksi dalam struktur graf, menegaskan bahwa pendekatan berbasis graf lebih mampu menangkap pola transaksi

	m (2025)	in *Bitcoin* transactions	mencurigakan dibandingkan analisis transaksi yang berdiri sendiri.
2	Ouyang, Bai, Feng, & Hu (2024)	*Bitcoin Money laundering* Detection via Sub*graph* Contrastive Learning	Model Bit-CHetG terbukti unggul dalam mendeteksi pola pencucian uang berbasis kelompok transaksi, dengan peningkatan Micro *F1-score* minimal 5% dibandingkan metode *Random Forest*, GCN, *Inspection-L*, SubGNN, HAN, MAGNN, dan TSGN pada dua dataset *Bitcoin* yang diuji.
3	Alarab & Prakoonwit (2023)	Robust recurrent *Graph Convolutional Network* approach based sequential prediction of illicit transactions in cryptocurrencies	RecGNN mencatatkan hasil *state-of-the-art* dengan akurasi 98,99% dan *F1-score* 91,75% pada dataset Elliptic, serta berhasil mengenali peristiwa penutupan pasar gelap yang luput dari deteksi algoritma konvensional.

UNIVERSITAS NUSA MANDIRI

4	Li & He (2023)	*Graph Neural Network*-Based *Bitcoin* Transaction Tracking Model (BT2)	Model BT2 mencapai rata-rata AUC hingga 0,93 dan akurasi rata-rata 86% pada data transaksi dunia nyata, mengungguli metode-metode pelacakan transaksi *Bitcoin state-of-the-art* lainnya.
5	Cheng, Ye, Xiang, Ma, Zhang, & Jiang (2023)	Anti-*Money laundering* by Group-Aware Deep *Graph* Learning	Pendekatan *deep graph learning* berbasis kesadaran kelompok terbukti mengungguli metode-metode AML *state-of-the-art*, baik pada mode operasi *offline* maupun *online*, saat diuji pada data transaksi nyata dari aliansi kartu bank besar.
6	Wan & Li (2024)	A Novel *Money laundering* Prediction Model Based on a Dynamic *Graph* Convolutional Neural Network and *Long Short-*	Kombinasi DGCNN dan LSTM terbukti lebih baik dalam mengidentifikasi transaksi pencucian uang dibandingkan metode klasifikasi konvensional, dengan memanfaatkan informasi spasial dan temporal secara bersamaan.

UNIVERSITAS NUSA MANDIRI

		Term Memory	
7	Lo, Kulatilleke, Sarhan, Layeghy, & Portmann (2023)	*Inspection-L*: *Self-supervised* GNN *node embeddings* for *money laundering* detection in *Bitcoin*	Sebagai penelitian pertama yang menerapkan *self-supervised* GNN pada AML *Bitcoin*, *Inspection-L* berhasil menghasilkan *embedding node* berkualitas tinggi meski data berlabel sangat terbatas, sehingga meningkatkan kemampuan classifier *Random Forest* mengenali pola transaksi mencurigakan.
8	Liu, Yin, Wang, Wu, Lan, Zhou, & Ge (2023)	*Graph Embedding*-Based *Money laundering* Detection for Ethereum	Pendekatan *graph embedding* terbukti meningkatkan performa deteksi pencucian uang pada jaringan *Ethereum* dibandingkan metode berbasis fitur transaksi saja, karena mampu menangkap hubungan struktural antar entitas dalam jaringan.
9	Qianyu, Wei-Tek Tsai, Bowen (2024)	RMGANets: reinforcement learning-enhanced multi-relational attention	RMGANets berhasil meningkatkan akurasi deteksi pencucian uang pada jaringan transaksi berskala besar dan anonim, dengan mengurangi ambiguitas intra-kelas serta mempertegas perbedaan antar-kelas melalui interaksi antar subgraf.

		graph-aware network for anti-*money laundering* detection	
10	Wang, Zhang, Zhao, & Liu (2023)	Detecting Illicit Transactions in *Bitcoin*: A Wavelet-Temporal *Graph* Transformer Approach for Anti-*Money laundering*	Model ChronoWave-GNN memperoleh akurasi 98,02% dan *F1-score* 97,99% pada Elliptic *Bitcoin* Dataset, melampaui performa sejumlah model *Graph Neural Network* sebelumnya dalam mendeteksi transaksi ilegal.

UNIVERSITAS NUSA MANDIRI

2.3. Kerangka Konsep

Kerangka konsep dalam penelitian ini digunakan untuk menjelaskan alur pengolahan data transaksi *Blockchain* hingga menghasilkan deteksi transaksi ilegal dalam konteks *digital forensik*.

Penelitian ini diawali dengan penggunaan data transaksi Bitcoin yang berasal dari Elliptic Bitcoin Dataset. Dataset tersebut terdiri atas fitur transaksi, hubungan antartransaksi (edgelist), dan label kelas yang merepresentasikan transaksi legal maupun ilegal. Karakteristik tersebut memungkinkan setiap transaksi direpresentasikan sebagai node, sedangkan hubungan antartransaksi direpresentasikan sebagai edge, sehingga membentuk struktur graph yang dapat dimanfaatkan dalam proses pembelajaran model[18], [20]

Sebelum proses pelatihan model dilakukan, data terlebih dahulu melalui tahap praproses yang meliputi pembersihan data, normalisasi fitur, dan pemisahan transaksi berdasarkan ketersediaan label. Transaksi berlabel Licit dan Illicit digunakan sebagai data berlabel dalam proses pelatihan dan evaluasi model, sedangkan transaksi Unknown tidak digunakan sebagai target pelatihan karena tidak memiliki ground truth. Data Unknown tetap dipertahankan untuk tahap eksplorasi dan analisis lanjutan menggunakan model yang telah dilatih [23].

Graph yang telah terbentuk kemudian digunakan sebagai masukan bagi model *Graph Attention Network (GAT)* sebagai metode utama dalam penelitian ini. GAT merupakan salah satu pengembangan *Graph Neural Network (GNN)* yang memanfaatkan mekanisme *attention* untuk memberikan bobot yang berbeda terhadap setiap *node* tetangga selama proses agregasi informasi. Dengan mekanisme tersebut, model mampu mempelajari hubungan lokal sekaligus mempertimbangkan tingkat kepentingan masing-masing *node,* sehingga representasi yang dihasilkan menjadi lebih informatif untuk tugas klasifikasi pada data berbentuk *graph*[51]

Pada penelitian ini, representasi graph dimanfaatkan untuk menangkap hubungan antartransaksi *Bitcoin* yang tidak dapat diperoleh apabila setiap transaksi dianalisis

secara terpisah. Pendekatan berbasis *Graph Neural Network* memungkinkan model memanfaatkan informasi struktural jaringan sehingga lebih efektif dalam mendeteksi transaksi ilegal dibandingkan pendekatan yang hanya menggunakan atribut transaksi. Berbagai penelitian menunjukkan bahwa model berbasis GAT dan *Graph Neural Network* mampu meningkatkan kemampuan deteksi transaksi ilegal pada jaringan Bitcoin, terutama pada data dengan distribusi kelas yang tidak seimbang [52] Performa model GAT kemudian dibandingkan dengan *Support Vector Machine (SVM)***,** *Graph Convolutional Network (GCN)*, serta variasi *GCN + Class Weight* dan *GAT + Class Weight*. Perbandingan dilakukan menggunakan metrik *Accuracy, Precision, Recall, F1-score***,** dan *ROC-AUC* untuk memperoleh gambaran yang komprehensif mengenai kemampuan masing-masing model dalam mengklasifikasikan transaksi legal dan ilegal pada dataset yang tidak seimbang

UNIVERSITAS
NUSA MANDIRI