GRAPH ATTENTIVE CONVOLUTIONAL NETWORKS DALAM DETEKSI TRANSAKSI ILEGAL PADA JARINGAN *BITCOIN* UNTUK KEBUTUHAN *DIGITAL FORENSIK*

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Magister Komputer (M.Kom)

LUSINDAH ERDIAN NOVA

14240010

Program Studi Informatika (S2)

Fakultas Teknologi Informasi

Universitas Nusa Mandiri

Jakarta

2026

SURAT PERNYATAAN ORISINALITAS

DAN BEBAS PLAGIARISME

Yang bertanda tangan di bawah ini :

Nama : Lusindah Erdian Nova
NIM : 14240010
Program Studi : Informatika
Fakultas : Teknologi Informasi
Jenjang : Strata Dua (S2)
Peminatan : Artificial Intelligence & *Blockchain*

Dengan ini menyatakan bahwa tesis yang telah saya buat dengan judul: "***Graph Attentive Convolutional Networks* Dalam Deteksi Transaksi Ilegal Pada Jaringan *Bitcoin* Untuk Kebutuhan *Digital forensik***" adalah hasil karya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar, serta belum pernah diterbitkan atau dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tesis yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Program Studi Informatika (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri dicabut/dibatalkan.

Jakarta, 14 Agustus 2026

Yang menyatakan,



Lusindah Erdian Nova

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya:

Nama	: Lusindah Erdian Nova
NIM	: 14240010
Program Studi	: Informatika
Fakultas	: Teknologi Informasi
Jenjang	: Strata Dua (S2)
Peminatan	: Artificial Intelligence & *Blockchain*
Jenis Karya	: Tesis

Demi pengembangan ilmu pengetahuan, dengan ini menyetujui untuk memberikan izin kepada pihak Program Studi Informatika (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri, **Hak Bebas Royalti Non-Eksklusif** ***(Non-exclusive Royalti-Free Right)*** atas karya ilmiah saya yang berjudul : **"*GRAPH ATTENTIVE CONVOLUTIONAL NETWORKS* DALAM DETEKSI TRANSAKSI ILEGAL PADA JARINGAN *BITCOIN* UNTUK KEBUTUHAN *DIGITAL FORENSIK*"** beserta perangkat yang diperlukan (apabila ada).

Dengan **Hak Bebas Royalti Non-Eksklusif** ini, pihak Universitas Nusa Mandiri berhak menyimpan, mengalih-media atau bentuk-kan, mengelolanya dalam pangkalan data *(database)*, mendistribusikannya dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Nusa Mandiri, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 14 Agustus 2026
Yang menyatakan,



Lusindah Erdian Nova

HALAMAN PERSETUJUAN DAN PENGESAHAN TESIS

Tesis ini diajukan oleh:

Nama	:	Lusindah Erdian Nova
NIM	:	14240010
Program Studi	:	Informatika
Jenjang	:	Strata Dua (S2)
Konsentrasi	:	*Artificial Intelligence & Blockchain*
Judul Tesis	:	Graph Attentive Convolutional Networks Dalam Deteksi Transaksi Ilegal Pada Jaringan Bitcoin Untuk Kebutuhan Digital Forensik

Telah dipertahankan pada periode 2026-1 dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Komputer (M.Kom) pada Program Studi Informatika (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri.

Jakarta, 14 Agustus 2026

PEMBIMBING TESIS

Pembimbing I	:	Dr. Hilman Ferdinandus Pardede, S.T., M.Eng.	Ditandatangani Digital: Hilman Ferdinandus Pardede 2026-09-17 07:32:46 .....

D E W A N P E N G U J I

Penguji I	:	Dr. Eng. Muhammad Haris, S.Kom., M.Eng.	Ditandatangani Digital: Muhammad Haris 2026-09-17 17:04:39 .....
Penguji II	:	Dr. Agus Subekti, M.T.	Ditandatangani Digital: Agus Subekti 2026-08-14 13:45:22 .....
Penguji III / Pembimbing I	:	Dr. Hilman Ferdinandus Pardede, S.T., M.Eng.	Ditandatangani Digital: Hilman Ferdinandus Pardede 2026-08-14 13:11:32 .....

UNIVERSITAS NUSA MANDIRI

	LEMBAR BIMBINGAN TESIS
	UNIVERSITAS NUSA MANDIRI

NIM : 14240010

Nama Lengkap : Lusindah Erdian Nova

Dosen Pembimbing : Prof. Dr. Eng. Hilman F. Pardede, S.T, M. Eng

Judul Tesis : Graph Attentive Convolutional Networks Dalam Deteksi Transaksi Ilegal Pada Jaringan Bitcoin Untuk Kebutuhan Digital Forensik.

No	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1.	1 April 2026	Perbaikan Proposal dan Pembahasan Dataset	
2.	14 April 2026	Seminar Proposal	
3.	21 April 2026	Revisi judul dan kerangka teori	
4.	7 Mei 2026	Saran Metodologi dan Implementasi	
5.	5 Juni 2026	Implementasi Menggunakan SVM Dan GCN	
6.	9 Juli 2026	Bab 4 hasil implementasi	
7.	20 Juli 2026	Revisi laporan hasil eksperiment	
8.	30 Juli 2026	Revisi Bab 1-5 keseluruhan	

Catatan untuk dosen pembimbing

Bimbingan Tesis

* Dimulai pada tanggal: 1 April 2026
* Diakhiri pada tanggal: 30 Juli 2026
* Jumlah pertemuan bimbingan : 8 (delapan) kali bimbingan

Disetujui oleh,

Dosen Pembimbing

Ditandatangani Digital:
Hilman Ferdinandus Pardede
2026-07-27 11:48:32

(Prof. Dr. Eng. Hilman F. Pardede, S.T, M. Eng)

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan YME, atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penelitian ini dengan baik. Penelitian yang berjudul "***Graph Attentive Convolutional Networks*** **Dalam Deteksi Transaksi Ilegal Pada Jaringan *Bitcoin* Untuk Kebutuhan *Digital forensik*** " ini disusun sebagai salah satu syarat dalam menempuh pendidikan pada Program Studi Informatika (S2), Fakultas Teknologi Informasi, Universitas Nusa Mandiri.

Penelitian ini bertujuan untuk mengembangkan model *Machine learning* dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin* dengan memanfaatkan pendekatan berbasis *graph*. Melalui penelitian ini, penulis berharap dapat memberikan kontribusi terhadap pengembangan sistem transaksi pada jaringan *Blockchain Bitcoin* yang direpresentasikan dalam bentuk *graph*.

Penulis menyadari bahwa proses penyusunan penelitian ini tidak terlepas dari bantuan, dukungan, bimbingan, serta motivasi dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor Universitas Nusa Mandiri.
2. Wakil Rektor I Bidang Akademik Universitas Nusa Mandiri.
3. Dekan Fakultas Teknologi Informasi Universitas Nusa Mandiri.
4. Ketua Program Studi Informatika (S2) Universitas Nusa Mandiri.
5. Bapak Prof. Dr. Eng. Hilman F. Pardede, S.T, M. Eng. selaku dosen pembimbing yang telah menyediakan waktu, pikiran dan tenaga dalam membimbing penulis dalam menyelesaikan tesis ini.
6. Orang tua tercinta yang telah memberikan dukungan material dan moral kepada penulis.
7. Seluruh Dosen Program Studi Informatika (S2) Fakultas Teknologi Informasi Universitas Nusa Mandiri yang telah memberikan pelajaran yang berarti bagi penulis selama menempuh studi.
8. Seluruh staf di lingkungan Universitas Nusa Mandiri yang telah melayani penulis dengan baik selama kuliah.

9. Rekan-rekan mahasiswa yang telah memberikan bantuan dan dukungan hingga terselesaikannya tesis ini.

Serta semua pihak yang terlalu banyak untuk penulis sebutkan satu persatu sehingga terwujudnya penulisan laporan tesis ini. Penulis menyadari bahwa penulisan laporan tesis ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan karya ilmiah di masa yang akan datang.

Akhir kata semoga laporan tesis ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 14 Agustus 2026

Penulis



Lusindah Erdian Nova

UNIVERSITAS
NUSA MANDIRI

ABSTRAK

Nama : Lusindah Erdian Nova

NIM : 14240010

Program Studi : Informatika

Fakultas : Teknologi Informasi

Jenjang : Strata Dua (S2)

Peminatan : Artificial Intelligence & *Blockchain*

Jenis Karya : *Graph Attentive Convolutional Networks* Dalam Deteksi Transaksi Ilegal Pada Jaringan *Bitcoin* Untuk Kebutuhan *Digital forensik*

Perkembangan teknologi Blockchain telah meningkatkan penggunaan mata uang kripto, khususnya Bitcoin, dalam berbagai aktivitas transaksi digital. Di balik berbagai manfaat yang ditawarkan, sifat pseudonim pada jaringan Bitcoin juga membuka peluang terjadinya aktivitas ilegal, seperti pencucian uang, penipuan, dan kejahatan siber lainnya. Kondisi tersebut menjadi tantangan dalam proses digital forensik karena pola transaksi ilegal umumnya tidak dapat dikenali hanya dari karakteristik setiap transaksi, tetapi juga dipengaruhi oleh hubungan antartransaksi di dalam jaringan Blockchain.

Penelitian ini bertujuan untuk mengembangkan model deteksi transaksi ilegal pada jaringan Bitcoin menggunakan Graph Attention Network (GAT), yaitu model Graph Neural Network yang menerapkan mekanisme attention. Mekanisme attention pada GAT memungkinkan model memberikan bobot yang berbeda terhadap setiap transaksi tetangga berdasarkan tingkat relevansinya. Model GAT kemudian dikombinasikan dengan Class Weight untuk mengatasi ketidakseimbangan data antara transaksi Licit dan Illicit. Dataset yang digunakan adalah Elliptic Bitcoin Dataset yang terdiri atas 203.769 transaksi, dengan 42.019 transaksi Licit, 4.545 transaksi Illicit, dan 157.205 transaksi Unknown, serta 234.355 hubungan antartransaksi (edge). Transaksi Unknown tidak digunakan sebagai target dalam pelatihan dan evaluasi model karena tidak memiliki ground truth, tetapi tetap dianalisis pada tahap implementasi untuk mengidentifikasi transaksi yang memiliki pola mendekati kelas Illicit.

Tahapan penelitian meliputi eksplorasi dataset, data preprocessing, pembentukan graph, pelatihan model, evaluasi performa, serta analisis transaksi Unknown. Model yang digunakan terdiri atas Support Vector Machine (SVM), Graph Convolutional Network (GCN), GCN dengan Class Weight, GAT, dan GAT dengan Class Weight. Evaluasi dilakukan menggunakan metrik Accuracy, Precision, Recall, F1-score, Confusion Matrix, dan Receiver Operating Characteristic–Area Under Curve (ROC-AUC).

Hasil penelitian menunjukkan bahwa GAT dengan Class Weight memperoleh Recall sebesar 90,43%, ROC-AUC sebesar 0,9809, False Negative sebanyak 87 transaksi, dan True Positive sebanyak 822 transaksi. Sementara itu, SVM

memperoleh Accuracy tertinggi sebesar 97,36%, tetapi memiliki kemampuan yang lebih rendah dalam mendeteksi transaksi ilegal dibandingkan model berbasis graph. Implementasi model GAT + Class Weight terhadap 157.205 transaksi Unknown menghasilkan 119.100 transaksi (75,76%) yang diprediksi Licit dan 38.105 transaksi (24,24%) yang diprediksi Illicit. Analisis struktur graph juga menemukan 4.027 transaksi Unknown yang memiliki hubungan langsung dengan transaksi Illicit, dengan 1.942 transaksi Unknown diprediksi sebagai Illicit sekaligus memiliki hubungan langsung dengan transaksi Illicit. Kelompok transaksi tersebut dapat digunakan sebagai kandidat prioritas untuk pemeriksaan lebih lanjut, namun tidak dapat langsung dinyatakan sebagai transaksi ilegal karena tidak memiliki label aktual.

Berdasarkan hasil penelitian, pendekatan berbasis Graph Neural Network, khususnya GAT yang dikombinasikan dengan Class Weight, dapat digunakan untuk mendukung deteksi transaksi ilegal pada jaringan Bitcoin. Selain mampu mendeteksi transaksi berlabel Illicit, model juga dapat digunakan sebagai alat bantu untuk menyaring dan memprioritaskan transaksi Unknown yang memiliki indikasi mencurigakan berdasarkan hasil prediksi model dan hubungan antartransaksi. Hasil tersebut dapat mendukung proses digital forensik, khususnya dalam membantu investigator menentukan transaksi yang memerlukan pemeriksaan lebih lanjut.

Kata kunci: Bitcoin, Blockchain, digital forensik, Graph Attention Network, Graph Convolutional Network, Graph Neural Network, deteksi transaksi ilegal, Elliptic Bitcoin Dataset.

UNIVERSITAS
NUSA MANDIRI

ABSTRACT

Name	: Lusindah Erdian Nova
NIM	: 14240010
Study of Program	: Informatika
Fakulty	: Teknologi Informasi
Levels	: Strata Dua (S2)
Concentration	: Artificial Intelligence & *Blockchain*
Title	:*Graph Attentive Convolutional Networks* Dalam Deteksi Transaksi Ilegal Pada Jaringan *Bitcoin* Untuk Kebutuhan *Digital forensik*

The rapid growth of Blockchain technology has significantly increased the adoption of cryptocurrencies, particularly Bitcoin, in digital financial transactions. Despite its advantages, the pseudonymous nature of Bitcoin also creates opportunities for various illicit activities, such as money laundering, fraud, and other financial crimes. This condition presents a challenge for digital forensik investigations because illicit transaction patterns may not be identified solely from the characteristics of individual transactions, but may also be reflected in the relationships among transactions within the Blockchain network.

This study aims to develop a model for detecting illicit transactions on the Bitcoin network using a Graph Attention Network (GAT), a Graph Neural Network model that applies an attention mechanism. The attention mechanism enables GAT to assign different weights to neighboring transactions according to their relevance. GAT is then combined with Class Weight to address the class imbalance between Licit and Illicit transactions. The Elliptic Bitcoin Dataset used in this study consists of 203,769 transactions, including 42,019 Licit transactions, 4,545 Illicit transactions, and 157,205 Unknown transactions, connected by 234,355 edges. Unknown transactions are not used as target labels during model training and evaluation because they do not have ground-truth labels. However, they are further analyzed during the implementation stage to identify transactions whose patterns are more similar to the Illicit class.

The research stages include dataset exploration, data preprocessing, graph construction, model training, performance evaluation, and analysis of Unknown transactions. The models evaluated in this study include Support Vector Machine (SVM), Graph Convolutional Network (GCN), GCN with Class Weight, GAT, and GAT with Class Weight. Model performance is evaluated using Accuracy, Precision, Recall, F1-score, Confusion Matrix, and Receiver Operating Characteristic–Area Under the Curve (ROC-AUC).

The results show that GAT with Class Weight achieved a Recall of 90.43%, an ROC-AUC of 0.9809, 87 False Negative transactions, and 822 True Positive transactions. Meanwhile, SVM achieved the highest Accuracy of 97.36%, but showed a lower ability to detect Illicit transactions compared with the graph-based models. The implementation of GAT with Class Weight on 157,205 Unknown transactions resulted in 119,100 transactions (75.76%) being predicted as Licit and 38,105 transactions (24.24%) being predicted as Illicit. Graph structure analysis further identified 4,027 Unknown transactions that were directly connected to Illicit transactions. Among them, 1,942 Unknown transactions were both predicted as Illicit and directly connected to Illicit transactions.

Based on these results, Graph Neural Network approaches, particularly GAT combined with Class Weight, can support the detection and analysis of illicit Bitcoin transactions. In addition to detecting labeled Illicit transactions, the model can be used as a decision-support tool to screen and prioritize Unknown transactions that show suspicious indications based on model predictions and graph relationships. The 1,942 transactions identified through both indicators may therefore be prioritized for further digital forensik examination. However, these transactions cannot be considered confirmed illicit transactions because their actual ground-truth labels are unavailable.

Keywords: *Bitcoin, Blockchain, digital forensics, Graph Attention Network, Graph Convolutional Network, Graph Neural Network, illicit transaction detection, Elliptic Bitcoin Dataset.*

UNIVERSITAS
NUSA MANDIRI

DAFTAR ISI

Halaman

HALAMAN JUDUL ii
SURAT PERNYATAAN ORISINALITAS DAN BEBAS PLAGIARISME ... iii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS iv
PERSETUJUAN DAN PENGESAHAN TESIS iii
LEMBAR BIMBINGAN TESIS vi
KATA PENGANTAR vii
ABSTRAK ix
***ABSTRACT* ..... xi**
DAFTAR ISI xxiii
DAFTAR TABEL xv
DAFTAR GAMBAR xv
DAFTAR LAMPIRAN xvxviii
BAB I PENDAHULUAN 19
1.1. LATAR BELAKANG 19
1.2. IDENTIFIKASI MASALAH 22
1.3. TUJUAN PENELITIAN 223
1.4. RUANG LINGKUP PENELITIAN 24
A. Ruang Lingkup ***24***
B. Batasan Penelitian ***25***
1.5. SISTEMATIKA PENULISAN 26
BAB II TINJAUAN PUSTAKA 28
2.1. TINJAUAN TEORI 28
2.1.1. Bitcoin ***28***
2.1.2 *BLOCKCHAIN* 30
2.1.3 TRANSAKSI ILEGAL PADA JARINGAN *BLOCKCHAIN* 30
2.1.4 DIGITAL FORENSIK 31
2.1.5 *MACHINE LEARNING* UNTUK DETEKSI TRANSAKSI ILEGAL 33
2.1.6 SUPPORT VECTOR MACHINE (SVM) 33
2.1.7 *GRAPH NEURAL NETWORK* (GNN) 34
2.1.8 *GRAPH CONVOLUTIONAL NETWORK* (GCN) 35
2.1.9 *GRAPH* ATTENTION NETWORK (GAT) 37
2.1.10 PRAPROSES DAN NORMALISASI DATA (*FEATURE SCALING*) 39
2.1.11 CLASS WEIGHT (PENANGANAN KETIDAKSEIMBANGAN KELAS) 40

2.1.12 Optimisasi Model (Adam Optimizer dan *Cross-entropy loss*) 42
2.1.13 Evaluasi Model Klasifikasi 443
2.1.14 t-SNE (Visualisasi *Embedding*) 49
2.1.15 Sintesis Kerangka Teori 50
2.2. Tinjauan Penelitian Sebelumnya 51
2.3. Kerangka Konsep 61
BAB III METODOLOGI PENELITIAN 63
3.1 Pendekatan Penelitian 683
3.2 Kerangka Penelitian 65
3.3 Sumber dan Pengumpulan Data 71
3.4 Data *Preprocessing* 883
3.5 Implementasi Model *Graph* Attention Network (GAT) 90
3.6 Model Klasifikasi 95
3.7 Pelatihan dan Pengujian Model 96
3.8 Metrik Evaluasi 99
3.9 Visualisasi Representasi Model menggunakan t-SNE 102
3.10 Analisis Risiko *Neighbor* (*Neighbor* Risk *Analysis*) 102
3.11 Kerangka Investigasi *Digital forensik* 1090
BAB IV HASIL DAN PEMBAHASAN 106
4.1 Gambaran Umum Implementasi Penelitian 106
4.2 Analisis Hasil Deteksi Transaksi *Fraud* 131
4.3 Analisis Struktur *Graph* 138
4.4 Pemetaan Hasil Penelitian ke dalam Kerangka Investigasi 144
BAB V PENUTUP 146
5.1 Kesimpulan 146
5.1 saran 148
Daftar Referensi 150
Lampiran 1. Daftar Riwayat Hidup Peneliti 156
Lampiran 2. Hasil Cek Plagiarisme 157
Lampiran 3. Daftar Publikasi Yang Telah Dihasilkan 158
Lampiran 4. Code Hasil Penelitian 159

DAFTAR TABEL

Halaman

Tabel 2.1 Penelitian Terdahulu……56
Tabel 3.1 Jadwal Penelitian……..……70
Tabel 3.2 Spesifikasi Perangkat yang Digunakan ……..……..……73
Tabel 3.3 Perangkat Lunak dan Library .....……..……..……73
Tabel 3.4 Konfigurasi Hyperparameter Kelima Model.....……..…..……74
Tabel 3.5 Ringkasan Tahapan Penelitian.....……..…..……76
Tabel 3.6 Karakteristik Elliptic Bitcoin Dataset.....……78
Tabel 3.7 File Elliptic Bitcoin Dataset.....……79
Tabel 3.8 Distribusi Label Elliptic Bitcoin Dataset.....…80
Tabel 3.9 Ringkasan Hasil Eksplorasi Dataset.....…82
Tabel 3.10 Hasil Penggabungan Dataset.....…84
Tabel 3.11 Hasil Pemeriksaan Missing Value.....…86
Tabel 3.12 Hasil Transformasi Label / Label Encoding.....…87
Tabel 3.13 Pembagian Data Training dan Testing.....…90
Tabel 3.14 Hasil Perhitungan Class Weight pada Data Training…….……..98
Tabel 4.1 Ringkasan Hasil Visualisasi Struktur Graph.....108
Tabel 4.2 Perbandingan Hasil Confusion Matrix Seluruh Model.....121
Tabel 4.3 Hasil Evaluasi Model.....124
Tabel 4.4 Hasil Evaluasi Menggunakan macro dan micro avarage.....127
Tabel 4.5 Perkembangan Nilai Training Loss per Epoch..... 129
Tabel 4.6 Unknown hubungan tertinggi..... 132
Tabel 4.7 Ringkasan Hasil Deteksi Fraud..... 134
Tabel 4.8 Hasil Prediksi GAT + Class Weight Unknown..... 135
Tabel 4.9 Ringkasan Analisis Transaksi Unknown..... 136
Tabel 4.10 Transaksi Unknown dengan Probabilitas Illicit..... 137

DAFTAR GAMBAR

Halaman

Gambar 2.1 Model Input-Output UTXO pada Transaksi *Bitcoin*…….29

Gambar 2.2 Proses Digital forensik pada Jaringan Blockchain Forensics)..……32

Gambar 2.3 Arsitektur dan Alur Propagasi GCN ……37

Gambar 2.4 Mekanisme Attention dan Multi-Head Attention pada GAT…….38

Gambar 2.5 Confusion Matrix ……49

Gambar 3.1 Kerangka Penelitian Deteksi Transaksi Ilegal pada Jaringan Bitcoin Menggunakan Graph Attention Network untuk Kebutuhan Digital forensik ……..65

Gambar 3.2 Alur Proposed Method Deteksi Transaksi Ilegal pada Jaringan Bitcoin Menggunakan Graph Attention Network (GAT) dengan Class Weight untuk Mendukung Digital forensik…… 68

Gambar 3.3 Distribusi Label Transaksi Elliptic Bitcoin Dataset …..……..81

Gambar 3.4 Proses Penggabungan Dataset (Merge Dataset)…… …….83

Gambar 3.5 Hasil Penggabungan Dataset ……84

Gambar 3.6 Hasil Pemeriksaan Missing value pada Dataset …….85

Gambar 3.7 Hasil Feature Scaling Menggunakan StandardScaler ……..….87

Gambar 3.8 Struktur Graph Hasil Data Preparation ……..……..….88

Gambar 3.9 Hasil Pembagian Data Training dan Testing……..….89

Gambar 3.10 Arsitektur Model Graph Attention Network (GAT)……..…..….91

Gambar 3.11 Kerangka Investigasi Digital forensik ……..……104

Gambar 3.12 Alur Graph Attention Network (GAT) untuk Deteksi Transaksi Ilegal pada Jaringan Bitcoin ……105

Gambar 4.1 Visualisasi Struktur Graph pada Elliptic Bitcoin Dataset ……107

Gambar 4.2 Heatmap Korelasi Fitur ……..….109

Gambar 4.3 Transaction Distribution by Time step ……...….111

Gambar 4.4 Distribusi Label Unknown, Licit, dan Illicit ……..……...112

Gambar 4.5 Fraud Transaction by Time step ……..……...….113

Gambar 4.6 Confusion Matrix SVM …….……..……...….114

Gambar 4.7 Confusion Matrix GCN dan GCN + Class Weight ……..…...…116

Gambar 4.8 Confusion Matrix GAT dan GAT + Class Weight...……..…...…118

Gambar 4.9 Perbandingan Kurva ROC……...……..…...…129

Gambar 4.10 Distribusi Tingkat Risiko Neighbor …….……..131

Gambar 4.11 Analisis Pengaruh Hidden dimension GCN dan GAT …….…..138

Gambar 4.12 Analisis Pengaruh Hidden dimension GCN vs GAT…….….....139

Gambar 4.13 Perbandingan Visualisasi Embedding Transaksi Menggunakan GCN dan GCN + Class Weight…….142

Gambar 4.14 Perbandingan Visualisasi Embedding Transaksi Menggunakan GAT dan GAT + Class Weight……143

UNIVERSITAS
NUSA MANDIRI

DAFTAR LAMPIRAN

Halaman

LAMPIRAN 1. DAFTAR RIWAYAT HIDUP PENELITI ……156
LAMPIRAN 2. HASIL CEK PLAGIARISME 157
LAMPIRAN 3. DAFTAR PUBLIKASI YANG TELAH DIHASILKAN ……158
LAMPIRAN 4. CODE HASIL PENELITIAN 159

UNIVERSITAS
NUSA MANDIRI

BAB I
PENDAHULUAN

1.1. Latar Belakang

Sejak diluncurkan pada tahun 2009, jaringan *Bitcoin* merekam seluruh transaksi yang pernah terjadi di dalamnya, dan catatan tersebut bisa dilihat oleh siapa saja secara terbuka [1]. Hal inilah yang menjadi pembeda utama *Bitcoin* dibanding sistem perbankan pada umumnya. Sayangnya, keterbukaan data transaksi ini tidak diiringi dengan keterbukaan identitas pemiliknya. Setiap alamat *Bitcoin* hanya terlihat sebagai kumpulan karakter acak, tanpa nama maupun identitas resmi yang menyertainya. Perpaduan antara data yang terbuka namun identitas yang tersembunyi inilah yang akhirnya membuat *Bitcoin* cukup rentan dimanfaatkan untuk kejahatan finansial.

Skala transaksi kripto yang berkaitan dengan aktivitas ilegal terus menjadi perhatian serius dalam beberapa tahun terakhir, dengan nilai yang mencapai puluhan miliar dolar AS secara global [2]. Di sisi lain,[3], mencatat bahwa dari 46.564 transaksi Bitcoin pada dataset Elliptic yang telah memiliki label, terdapat 4.545 transaksi Illicit atau sekitar 9,76%, sedangkan sisanya merupakan transaksi Licit. Proporsi kelas Illicit yang jauh lebih kecil dibandingkan kelas Licit menunjukkan adanya ketidakseimbangan kelas yang menjadi tantangan dalam proses klasifikasi. Angkanya memang terlihat kecil, tetapi nominal uang yang terlibat tetap besar, dan setiap transaksi ilegal yang lolos dari deteksi pada dasarnya berarti ada satu kejahatan yang tidak pernah terbongkar.

Persoalan ini menjadi semakin rumit ketika dibawa ke ranah *digital forensik*. Studi [3] memperlihatkan bahwa dalam kasus penipuan di jaringan *Bitcoin*, sebuah *node* yang terlibat aktivitas ilegal sering kali tidak terlihat mencurigakan jika diamati sendirian. Pola kejanggalannya baru terlihat setelah *node* tersebut ditelusuri bersama-sama dengan *node* lain yang terhubung dengannya. Artinya, seorang investigator tidak bisa hanya berhenti pada satu transaksi yang dianggap mencurigakan, melainkan perlu menelusuri lebih jauh ke mana dana tersebut

mengalir, alamat-alamat apa saja yang ikut terlibat, dan apakah pola pergerakan dananya mengarah pada indikasi pencucian uang. Pekerjaan seperti ini akan sulit dilakukan kalau hanya mengandalkan pemeriksaan transaksi satu per satu, sebab informasi yang paling penting justru muncul dari pola hubungan antar transaksi, bukan dari satu transaksi yang berdiri sendiri.

Selama ini, deteksi transaksi ilegal pada *Blockchain* umumnya masih mengandalkan metode klasifikasi konvensional seperti *Support Vector Machine (SVM)*. Metode ini memang sudah cukup teruji untuk menangani data berdimensi tinggi dan banyak dipakai dalam berbagai penelitian klasifikasi [4], sehingga penelitian ini pun turut menjadikannya sebagai *baseline*. Hanya saja, kelemahan SVM terletak pada cara kerjanya yang menilai setiap transaksi hanya berdasarkan fiturnya sendiri, tanpa mempertimbangkan dengan siapa transaksi tersebut saling terhubung. Padahal, pada dasarnya transaksi *Bitcoin* membentuk struktur *graph*, di mana setiap transaksi berperan sebagai *node* dan aliran dana antartransaksi menjadi *edge* yang menghubungkan satu sama lain.

Penggunaan pendekatan berbasis graph dalam penelitian ini dipilih karena transaksi pada jaringan Bitcoin pada dasarnya saling terhubung satu sama lain. Dalam bentuk graph, setiap transaksi dapat digambarkan sebagai node, sedangkan hubungan antartransaksi digambarkan sebagai edge. Dengan bentuk data seperti ini, informasi yang diperoleh tidak hanya berasal dari karakteristik masing-masing transaksi, tetapi juga dari hubungan transaksi tersebut dengan transaksi lainnya.

Hal ini penting dalam mendeteksi transaksi ilegal karena suatu transaksi belum tentu terlihat mencurigakan jika hanya dilihat dari fiturnya saja. Pola transaksi ilegal dapat terlihat ketika hubungan transaksi tersebut dengan transaksi lain ikut dianalisis. Metode klasifikasi konvensional seperti SVM lebih berfokus pada fitur setiap transaksi secara individual dan tidak secara langsung memanfaatkan hubungan antartransaksi. Sementara itu, Graph Neural Network dapat mempelajari fitur transaksi sekaligus informasi dari transaksi yang saling terhubung. Oleh karena itu, pendekatan berbasis graph dipilih karena lebih sesuai

dengan karakteristik data transaksi Bitcoin dan tujuan penelitian dalam mendeteksi pola transaksi ilegal.

Untuk mengatasi keterbatasan tersebut, dikembangkanlah *Graph Convolutional Network (GCN)*. Metode yang dikenal sebagai GCN [5], ini memungkinkan sebuah model mempelajari representasi suatu *node* dengan turut memperhitungkan *node-node* tetangga di sekitarnya. Dalam konteks jaringan Bitcoin pada Elliptic Bitcoin Dataset, GCN memungkinkan model mempelajari suatu transaksi dengan mempertimbangkan informasi dari transaksi-transaksi lain yang terhubung dengannya. Dengan demikian, model tidak hanya menggunakan karakteristik suatu transaksi secara individual, tetapi juga memanfaatkan pola hubungan antarnode dalam graph.

Pendekatan berbasis *graph* untuk kasus semacam ini juga berkembang cukup pesat dalam beberapa tahun terakhir [6], menerapkan *Graph Attention Network* untuk mendeteksi pencucian uang pada data transaksi finansial, sedangkan [7] mengembangkan model CT-GCN+ untuk mengenali *node phishing* pada *graph* transaksi *cryptocurrency*. [6] kemudian memperluas pendekatan tersebut dengan menambahkan komponen *recurrent* agar mampu menangkap pola temporal pada transaksi *Bitcoin*. Sementara itu, studi terbaru [8] melaporkan bahwa model GCN mampu mencapai akurasi hingga 98,5 % dalam mengklasifikasikan transaksi ilegal pada dataset Elliptic, jauh mengungguli metode klasifikasi konvensional. Berbagai temuan inilah yang memperkuat alasan penelitian ini memilih pendekatan berbasis *graph*, bukan sekadar metode klasifikasi konvensional seperti SVM, sebagai fokus utamanya.

Penelitian ini menggunakan dataset Elliptic karena datanya sudah tersedia dalam bentuk *graph*, lengkap dengan fitur transaksi, daftar keterhubungan antar transaksi, serta label legal maupun illegal [8], Ketiga komponen tersebut cukup jarang ditemukan sekaligus pada dataset publik lain, sehingga dataset ini dinilai menjadi pilihan yang tepat untuk membangun sekaligus menguji model deteksi berbasis *graph*.

Berdasarkan uraian tersebut, penelitian ini mengembangkan model *Graph Attention Network (GAT)* untuk mendeteksi transaksi ilegal pada jaringan Bitcoin. Berbagai penelitian sebelumnya menunjukkan bahwa pendekatan berbasis *Graph Convolutional Network (GCN)* mampu memanfaatkan struktur graph untuk mengklasifikasikan transaksi ilegal secara efektif. Namun, GCN masih memiliki keterbatasan dalam membedakan tingkat kepentingan setiap transaksi tetangga karena proses agregasi informasi dilakukan secara relatif seragam. Oleh karena itu, penelitian ini menerapkan mekanisme attention pada GAT agar model dapat memberikan bobot yang berbeda terhadap setiap transaksi tetangga berdasarkan tingkat relevansinya. Selain itu, mekanisme attention tersebut dikombinasikan dengan *Class Weight* untuk mengatasi ketidakseimbangan kelas pada dataset, sehingga model diharapkan lebih sensitif dalam mendeteksi transaksi ilegal yang jumlahnya jauh lebih sedikit dibandingkan transaksi legal. Sebagai pembanding, model SVM dan GCN tetap digunakan untuk mengevaluasi peningkatan kinerja yang dihasilkan oleh pendekatan yang diusulkan[9]

1.2. Identifikasi Masalah

Berdasarkan latar belakang yang sudah dipaparkan, permasalahan dalam penelitian ini dapat diidentifikasi sebagai berikut.

1. Transaksi ilegal pada jaringan *Bitcoin*, seperti *fraud* dan pencucian uang, masih sulit diidentifikasi dengan akurasi yang tinggi karena pola transaksinya tidak terlihat secara langsung dan hanya bisa dikenali lewat hubungan antar transaksi, sehingga dibutuhkan metode deteksi dengan kinerja yang lebih baik untuk mendukung kebutuhan investigasi *digital forensik*.
2. Data pada jaringan *Blockchain*, seperti pada dataset Elliptic, memiliki karakteristik yang kompleks, mulai dari volume data yang besar hingga ketidakseimbangan jumlah antara transaksi legal dan ilegal (*imbalance*), sehingga menyulitkan proses deteksi transaksi ilegal secara akurat.
3. Pola transaksi ilegal sering kali tersembunyi di dalam hubungan antar transaksi, sehingga sulit terdeteksi apabila hanya mengandalkan pendekatan berbasis fitur.

1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan model Machine Learning yang dapat mendeteksi transaksi ilegal pada jaringan Bitcoin menggunakan pendekatan berbasis graph, khususnya *Graph Attention Network (GAT)*. Dalam penelitian ini dilakukan pengujian terhadap lima konfigurasi model, yaitu Support *Vector Machine (SVM), Graph Convolutional Network (GCN), GCN* dengan *Class Weight, Graph Attention Network (GAT),* dan GAT dengan *Class Weight*. GAT dengan *Class Weight* digunakan sebagai model utama yang diusulkan untuk meningkatkan kemampuan deteksi terhadap kelas Illicit pada kondisi data yang tidak seimbang.

Secara lebih rinci, tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis karakteristik transaksi pada jaringan *Bitcoin* untuk memahami perbedaan antara transaksi normal dan transaksi ilegal.

2. Mengembangkan model deteksi transaksi ilegal menggunakan metode *Graph Attention Network (GAT)* dan GAT dengan penambahan *Class Weight* (GAT + *Class Weight*) yang mampu memanfaatkan hubungan antartransaksi dalam bentuk *graph* sekaligus mengatasi ketidakseimbangan data antara transaksi normal dan ilegal.

3. Mengolah dan mempersiapkan data dari dataset Elliptic, termasuk menangani ketidakseimbangan data dan memastikan kualitas data sebelum digunakan dalam proses pelatihan model.

4. Mengevaluasi kinerja model dalam mendeteksi transaksi ilegal menggunakan Accuracy, Precision, Recall, F1-score, Confusion Matrix, dan ROC-AUC, dengan Recall sebagai salah satu metrik utama karena berkaitan langsung dengan jumlah False Negative.

5. Membandingkan performa model GAT dengan beberapa metode lain, yaitu *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight (GCN + Class Weight)*, untuk melihat sejauh mana peningkatan kinerja yang dihasilkan oleh model yang diusulkan.

6. Mengkaji hasil deteksi model GAT dalam konteks *digital forensik*, khususnya sebagai alat bantu dalam proses identifikasi dan analisis transaksi ilegal pada jaringan *Blockchain*.

1.4. Ruang Lingkup Penelitian

Penelitian ini berfokus pada pengembangan model deteksi transaksi ilegal menggunakan *Graph Attention Network (GAT)* sebagai model utama, yang dibandingkan dengan *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight (GCN + Class Weight)* sebagai *baseline*, dengan ruang lingkup dan batasan sebagai berikut.

A. Ruang Lingkup

Penelitian ini berfokus pada pengembangan model *Machine learning* untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin* dengan memanfaatkan pendekatan berbasis *graph*. Ruang lingkup penelitian ini mencakup hal-hal berikut.

1. Objek penelitian berupa transaksi pada jaringan *Blockchain Bitcoin* yang direpresentasikan dalam bentuk *graph*, dengan setiap transaksi sebagai *node* dan hubungan antartransaksi sebagai *edge*.

2. Dataset yang digunakan adalah dataset Elliptic, yang mencakup fitur transaksi, hubungan antartransaksi (*edgelist*), serta label transaksi (legal dan ilegal).

3. Fitur yang digunakan berasal dari karakteristik transaksi, seperti informasi statistik dan pola transaksi yang tersedia pada dataset.

4. Model yang digunakan terdiri atas SVM sebagai baseline berbasis fitur, GCN dan GCN dengan *Class Weight* sebagai model graph pembanding, serta GAT dan GAT dengan *Class Weight* sebagai model utama yang dievaluasi.

5. Proses penelitian meliputi tahap praproses data, pembentukan *graph*, pelatihan model, serta evaluasi performa model.

6. Evaluasi model dilakukan menggunakan metrik klasifikasi seperti akurasi, *precision*, *recall*, dan *F1-score*.

UNIVERSITAS NUSA MANDIRI

B. Batasan Penelitian

Agar penelitian lebih terarah dan tidak melebar ke mana-mana, ditetapkan batasan-batasan sebagai berikut.

1. Penelitian ini hanya berfokus pada pengembangan dan evaluasi model deteksi transaksi ilegal, serta tidak mencakup pembangunan sistem deteksi real-time atau implementasi aplikasi.

2. Dataset yang digunakan terbatas pada dataset Elliptic, sehingga hasil penelitian tidak dapat digeneralisasi secara langsung ke seluruh jenis *Blockchain*.

3. Data yang digunakan bersifat statis (offline dataset) dan tidak mencerminkan kondisi transaksi secara real-time.

4. Klasifikasi transaksi dibatasi pada kategori label yang tersedia dalam Elliptic Bitcoin Dataset, yaitu Licit dan Illicit. Penelitian ini tidak melakukan klasifikasi terpisah berdasarkan jenis tindak kejahatan pada masing-masing transaksi Illicit.

5. Model yang digunakan difokuskan pada *Graph Attention Network (GAT)* sebagai metode utama, tanpa membahas secara mendalam metode *graph* lainnya di luar GAT dan metode pembanding yang telah ditentukan.

6. Metode *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight (GCN + Class Weight)* hanya digunakan sebagai pembanding (*baseline*), bukan sebagai fokus utama penelitian.

7. Penanganan ketidakseimbangan data (*imbalance*d data) dilakukan sesuai kebutuhan eksperimen dan tidak menjadi fokus utama dalam pengembangan metode.

8. Penelitian ini tidak membahas aspek implementasi forensik secara operasional, melainkan lebih menitikberatkan pada analisis hasil model dalam mendukung proses identifikasi transaksi ilegal.

1.5. Sistematika Penulisan

Laporan penelitian ini disusun secara sistematis ke dalam beberapa bab agar pembahasannya lebih terarah dan mudah dipahami. Sistematika penulisannya adalah sebagai berikut:

BAB I	PENDAHULUAN	Bab ini memuat latar belakang masalah yang menjelaskan pentingnya deteksi transaksi ilegal pada jaringan *Bitcoin* dalam konteks *digital forensik*. Selain itu, bab ini juga berisi identifikasi masalah, tujuan penelitian, ruang lingkup dan batasan penelitian, serta sistematika penulisan sebagai gambaran umum penelitian.
BAB II	TINJAUAN PUSTAKA	Bab ini membahas teori-teori yang mendasari penelitian, mulai dari konsep *Blockchain*, transaksi *Bitcoin*, transaksi ilegal, *digital forensik*, hingga metode *Machine learning* yang digunakan, khususnya *Graph Attention Network (GAT)* dan *Support Vector Machine (SVM)*. Bab ini juga menyajikan penelitian-penelitian terdahulu yang relevan sebagai dasar pengembangan penelitian.
BAB III	METODOLOGI PENELITIAN	Bab ini menjelaskan tahapan penelitian yang dilakukan, mulai dari pengumpulan data, praproses data, pembentukan *graph* dari dataset Elliptic, penjelasan teori mengenai mekanisme attention, hingga proses pengembangan model menggunakan *Graph Attention Network* (GAT). Bab ini juga menjelaskan penggunaan SVM, GCN, dan GCN + *Class Weight* sebagai *baseline* serta metode evaluasi yang dipakai untuk mengukur kinerja model.
BAB IV	HASIL DAN PEMBAHASAN	Bab ini menyajikan hasil proses pelatihan dan pengujian model dalam mendeteksi transaksi ilegal.

		Hasil tersebut kemudian dianalisis dan dibandingkan antara model GAT, GAT + *Class Weight*, SVM, GCN, dan GCN + *Class Weight* untuk melihat perbedaan performanya. Bab ini juga membahas hasil penelitian dalam konteks *digital forensik* untuk memahami pola transaksi ilegal yang terdeteksi.
BAB V	KESIMPULAN DAN SARAN	Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan, serta saran untuk pengembangan penelitian selanjutnya, khususnya dalam pengembangan model deteksi transaksi ilegal berbasis *graph* pada jaringan *Blockchain*.

UNIVERSITAS
NUSA MANDIRI

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.

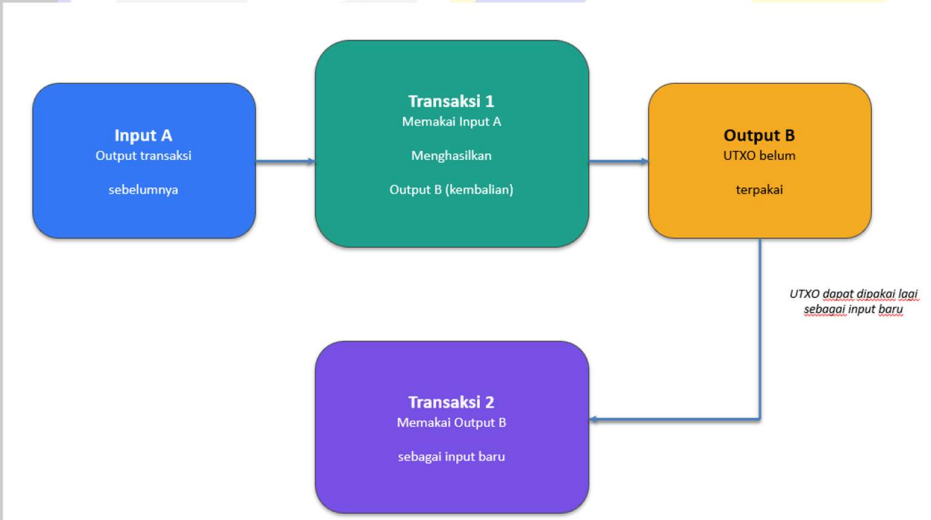
2.1.1. *Bitcoin*

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].

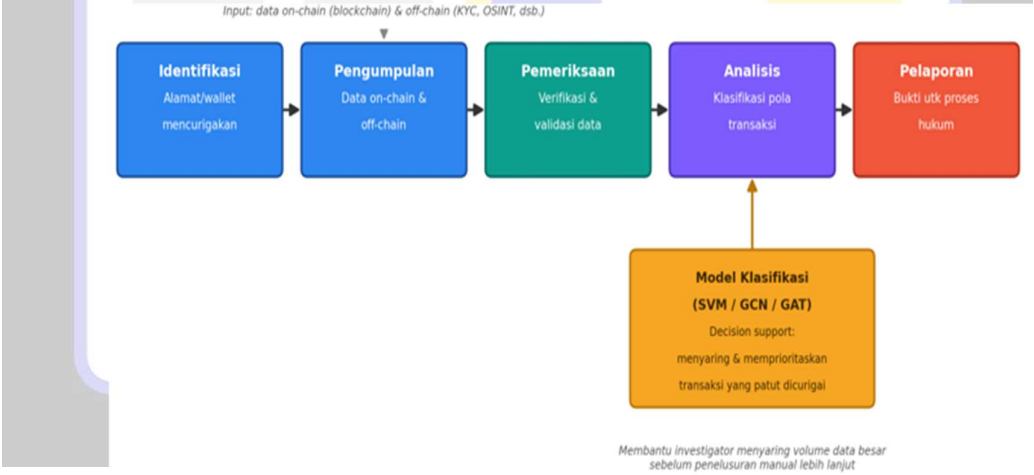
Mendeteksi transaksi ilegal pada *Blockchain* menghadapi tantangan tersendiri dibandingkan sistem keuangan konvensional. Pertama, identitas pengguna bersifat pseudonim sehingga tidak bisa langsung diasosiasikan dengan pelaku sebenarnya. Kedua, pola transaksi ilegal jarang terlihat mencolok pada satu 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)

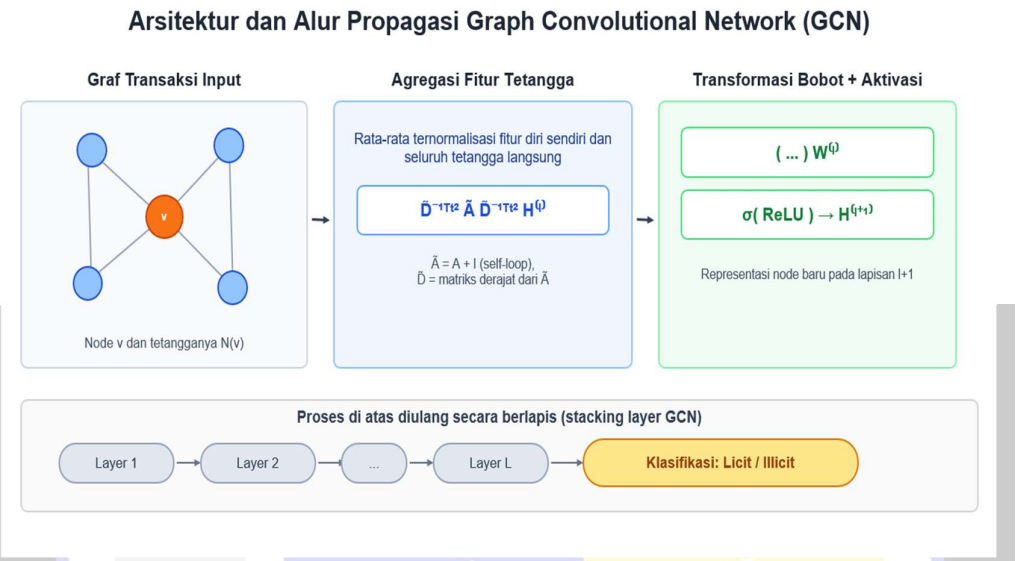
Graph Convolutional Network (GCN) merupakan salah satu arsitektur *Graph Neural Network* yang paling banyak dipakai sebagai pendekatan efisien untuk pembelajaran semi-supervised pada data berstruktur graf [5][8]. GCN bekerja dengan menggeneralisasi operasi konvolusi, yang lazim dipakai pada data grid

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

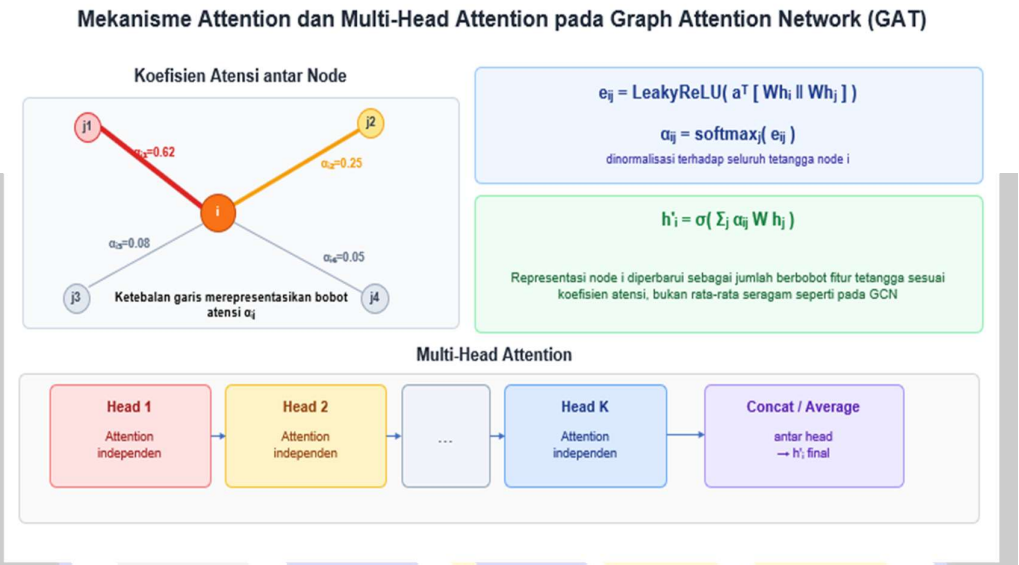
Gambar 2.3 Arsitektur dan Alur Propagasi *Graph Convolutional Network* (GCN)

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].

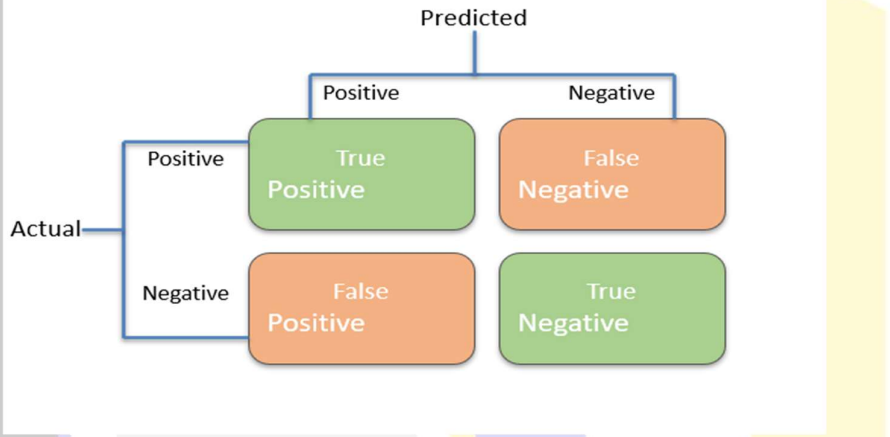
Secara matematis, dihitung dengan rumus berikut:

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

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

BAB III
METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini termasuk penelitian eksperimental kuantitatif dengan pendekatan komparatif, di mana beberapa model *Machine learning* dibangun dan diuji dalam kondisi yang sama untuk dibandingkan performanya. Model utama yang dikembangkan adalah *Graph Attention Network* (GAT) dengan *Class Weight*, yang kemudian dibandingkan dengan model lain, yaitu *Support Vector Machine (SVM)*, GAT, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight*, menggunakan Elliptic *Bitcoin* Dataset sebagai data eksperimen. Penelitian ini berfokus pada permasalahan deteksi transaksi ilegal pada jaringan *Blockchain*, khususnya pada *Bitcoin*. Meskipun *Blockchain* dikenal aman dan transparan, dalam praktiknya teknologi ini tetap dimanfaatkan untuk berbagai aktivitas ilegal seperti penipuan (*fraud*) dan pencucian uang (*money laundering*). Aktivitas tersebut sulit dideteksi karena pola transaksi yang kompleks serta sifat pseudonim dari pengguna *Bitcoin*.

Selain itu, transaksi dalam *Blockchain* tidak berdiri sendiri, tetapi saling terhubung dan membentuk suatu struktur *graph*. Pola transaksi ilegal sering kali tidak terlihat dari satu transaksi saja, melainkan tersembunyi dalam hubungan antartransaksi tersebut [14]. Hal ini menjadi tantangan bagi metode deteksi konvensional yang hanya mengandalkan fitur dari masing-masing transaksi secara individual.

Metode *Machine learning* konvensional seperti *Support Vector Machine (SVM)* memiliki keterbatasan karena tidak mampu memanfaatkan hubungan antartransaksi dalam jaringan. Akibatnya, pola transaksi ilegal yang bersifat kompleks dan saling terhubung menjadi sulit untuk diidentifikasi secara optimal apabila hanya mengandalkan fitur numerik transaksi[18]. Istilah *Graph Attentive Convolutional Network* dalam penelitian ini mengacu pada pendekatan *graph* convolution yang menggunakan mekanisme attention, yang secara spesifik

diimplementasikan menggunakan *Graph Attention Network (GAT)* sebagai model utama.

Dalam konteks *digital forensik*, kemampuan untuk mengidentifikasi dan menganalisis transaksi ilegal menjadi sangat penting sebagai bagian dari proses investigasi aktivitas kriminal berbasis data digital. Oleh karena itu, diperlukan pendekatan yang tidak hanya mampu melakukan klasifikasi, tetapi juga memahami hubungan antartransaksi dalam jaringan sekaligus dapat diintegrasikan ke dalam alur kerja investigasi forensik digital.

Pendekatan berbasis graph dipilih karena data transaksi pada Elliptic Bitcoin Dataset memiliki hubungan antartransaksi yang dapat direpresentasikan dalam bentuk node dan edge. Representasi tersebut memungkinkan model GCN dan GAT memanfaatkan fitur transaksi sekaligus hubungan antartransaksi dalam proses klasifikasi. Hal ini menjadi dasar penggunaan Graph Neural Network dalam penelitian karena pola transaksi ilegal tidak hanya dapat dilihat dari fitur setiap transaksi, tetapi juga dari pola hubungan transaksi di dalam jaringan.

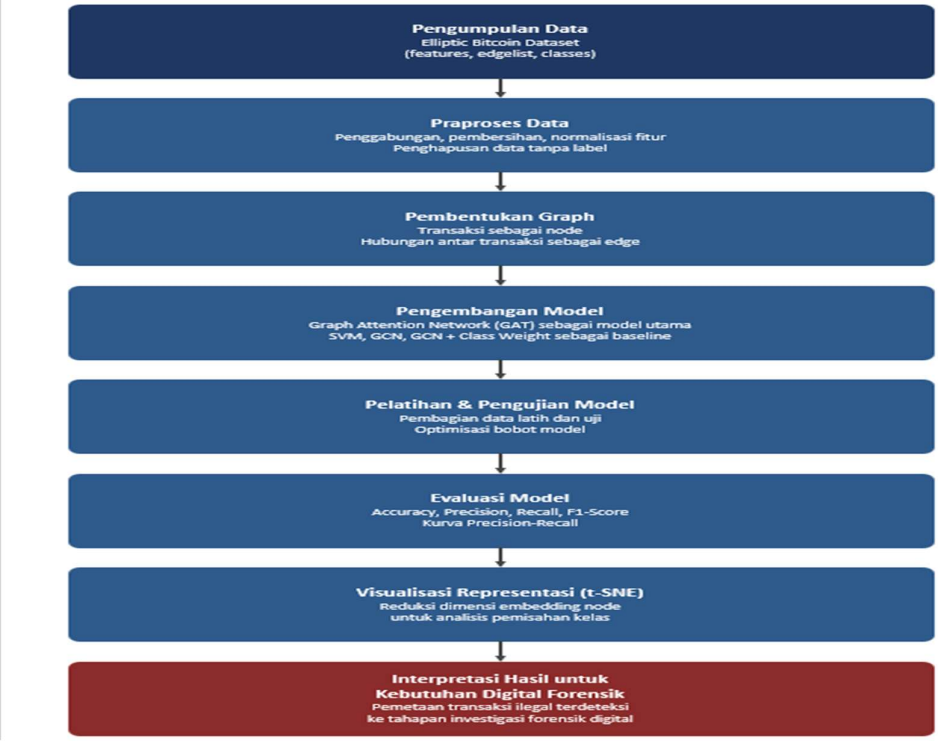
Berdasarkan pendekatan penelitian yang digunakan, penelitian ini menetapkan hipotesis sebagai dugaan awal yang akan dibuktikan melalui proses implementasi, pelatihan, dan evaluasi model klasifikasi. Hipotesis penelitian dirumuskan sebagai berikut:

1. H_0 (Hipotesis Nol)
   Penerapan *Graph Attention Network (GAT)* dengan *Class Weight* tidak memberikan peningkatan kemampuan deteksi kelas Illicit dibandingkan model pembanding, khususnya berdasarkan *Recall* dan jumlah *False Negative*.
2. H_1 (Hipotesis Alternatif)
   Penerapan *Graph Attention Network (GAT)* dengan *Class Weight* memberikan peningkatan kemampuan deteksi kelas Illicit dibandingkan model pembanding, yang ditunjukkan terutama melalui peningkatan Recall dan penurunan jumlah *False Negative*.

Melalui mekanisme attention, GAT mampu memberikan bobot yang berbeda pada setiap hubungan antartransaksi berdasarkan tingkat relevansinya. Dengan pendekatan ini, diharapkan model mampu menangkap pola transaksi ilegal secara lebih akurat dibandingkan metode konvensional. Tahapan-tahapan yang ditempuh untuk mewujudkan pendekatan ini disusun secara sistematis dalam suatu kerangka penelitian, sebagaimana disajikan pada subbab 3.2.

3.2 Kerangka Penelitian

Untuk memberikan gambaran yang sistematis mengenai tahapan-tahapan yang ditempuh dalam penelitian ini, disusun sebuah kerangka penelitian sebagaimana ditunjukkan pada Gambar 3.1.



Sumber: Diolah oleh penulis berdasarkan Pan (2024), Hoang et al. (2023), dan Marasi & Ferretti (2024).

Gambar 3.1 : Kerangka Penelitian Deteksi Transaksi Ilegal pada Jaringan *Bitcoin* Menggunakan *Graph* Attention Network untuk Kebutuhan *Digital forensik*

Gambar tersebut menunjukkan alur penelitian untuk mendeteksi transaksi *Bitcoin* ilegal menggunakan metode *Machine learning* berbasis graf.

Proses dimulai dari pengumpulan dataset Elliptic *Bitcoin*, lalu data dibersihkan, digabungkan, dan dinormalisasi. Setelah itu, transaksi dibentuk menjadi *graph*: setiap transaksi menjadi *node*, sedangkan hubungan antartransaksi menjadi *edge*.

Model utama yang digunakan adalah *Graph Attention Network (GAT)* beserta variannya GAT dengan *Class Weight*, kemudian dibandingkan dengan beberapa model *baseline*, yaitu SVM, GCN, dan GCN dengan *Class Weight*. Model dilatih dan diuji, lalu dievaluasi dengan metrik *Accuracy*, *Precision*, *recall*, *F1-score*, *Confusion Matrix*, serta ROC-AUC.

Hasilnya divisualisasikan menggunakan t-SNE untuk melihat pemisahan kelas transaksi. Tahap terakhir adalah interpretasi untuk kebutuhan forensik digital, yaitu membantu memetakan transaksi ilegal yang terdeteksi ke proses investigasi.

3.2.1 Metode yang Diusulkan (*Graph* Attention Network)

Graph Attention Network (GAT) merupakan salah satu metode pengembangan dari *Graph Neural Network (GNN)* yang menggunakan mekanisme attention untuk mempelajari hubungan antar*node* pada sebuah *graph*. Berbeda dengan *Graph Convolutional Network (GCN)* yang memperlakukan seluruh *node* tetangga dengan bobot yang sama saat proses agregasi informasi, GAT mampu memberikan bobot yang berbeda pada setiap *node* tetangga sesuai dengan tingkat kepentingannya [53]. Dengan cara ini, informasi yang dianggap lebih penting akan memberikan pengaruh yang lebih besar terhadap proses pembelajaran model.

Pada jaringan *Bitcoin*, setiap transaksi direpresentasikan sebagai sebuah *node*, sedangkan hubungan antartransaksi direpresentasikan sebagai *edge*. Hubungan tersebut membentuk struktur *graph* yang cukup kompleks karena satu transaksi dapat terhubung dengan banyak transaksi lainnya. Akibatnya, karakteristik suatu transaksi tidak hanya dipengaruhi oleh atribut yang dimilikinya, tetapi juga oleh transaksi-transaksi yang saling terhubung. Oleh karena itu, mekanisme attention pada GAT menjadi penting karena mampu menentukan transaksi mana yang memiliki pengaruh lebih besar terhadap proses klasifikasi.

Proses kerja GAT diawali dengan melakukan transformasi terhadap fitur setiap *node* menggunakan matriks bobot yang dipelajari selama proses pelatihan *(learnable weight matrix).* Setelah itu, model menghitung nilai *attention* untuk setiap pasangan *node* yang saling terhubung. Nilai tersebut kemudian dinormalisasi menggunakan fungsi *Softmax* sehingga diperoleh koefisien attention yang menunjukkan tingkat kepentingan masing-masing *node* tetangga[53].

Koefisien *attention* tersebut selanjutnya digunakan sebagai bobot dalam proses *agregasi informasi*. Dengan pendekatan ini, *node* yang memiliki hubungan lebih relevan akan memberikan kontribusi yang lebih besar dibandingkan *node* lain yang kurang berpengaruh. Hal ini menjadi salah satu keunggulan GAT dibandingkan GCN karena model dapat lebih selektif dalam memanfaatkan informasi dari lingkungan sekitarnya.

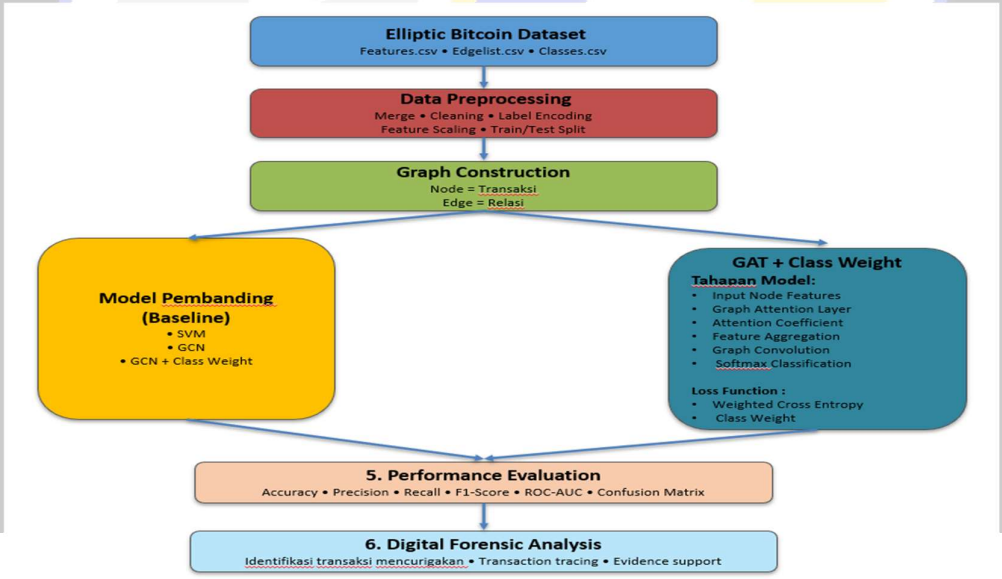
Dalam penelitian ini, *Graph Attention Network* dipilih karena dinilai lebih mampu memanfaatkan struktur *graph* yang dimiliki oleh Elliptic *Bitcoin* Dataset. Selain memanfaatkan fitur transaksi, GAT juga mempertimbangkan hubungan antartransaksi sehingga diharapkan dapat meningkatkan kemampuan model dalam membedakan transaksi legal dan ilegal. Penelitian ini juga mengombinasikan GAT dengan *Class Weight* untuk mengatasi ketidakseimbangan jumlah data pada dataset, di mana jumlah transaksi ilegal jauh lebih sedikit dibandingkan transaksi legal. Pemberian bobot kelas bertujuan agar model lebih memperhatikan kelas minoritas sehingga mampu meningkatkan nilai *Recall* dan mengurangi jumlah *False Negative*, yang merupakan aspek penting dalam proses *digital forensik*.

Implementasi model pada penelitian ini menggunakan dua lapisan *(two-layer GAT).* Lapisan pertama menerapkan mekanisme *multi-head attention* untuk menghasilkan representasi fitur yang lebih baik, sedangkan lapisan kedua digunakan untuk melakukan klasifikasi transaksi ke dalam kelas legal atau ilegal menggunakan fungsi *Softmax*. Seluruh parameter model dioptimasi menggunakan Adam Optimizer, sedangkan fungsi kerugian (*loss function*) yang digunakan adalah *Weighted Cross Entropy Loss* agar proses pembelajaran lebih seimbang terhadap kedua kelas.

Secara umum, tahapan metode *Graph Attention Network* yang digunakan dalam penelitian ini meliputi beberapa langkah sebagai berikut:

1. Membentuk struktur *graph* berdasarkan hubungan transaksi pada Elliptic *Bitcoin* Dataset.
2. Mengubah fitur setiap transaksi menjadi representasi *node*.
3. Menghitung nilai attention pada setiap hubungan antar*node*.
4. Melakukan agregasi informasi dari *node* tetangga berdasarkan bobot attention yang diperoleh.
5. Mengklasifikasikan setiap transaksi menjadi transaksi legal atau ilegal menggunakan fungsi *Softmax*.

Melalui tahapan tersebut, *Graph Attention Network* diharapkan mampu memanfaatkan informasi dari fitur transaksi maupun struktur hubungan antartransaksi secara lebih efektif. Dengan demikian, model diharapkan dapat memberikan hasil deteksi transaksi ilegal yang lebih baik dibandingkan metode klasifikasi konvensional maupun *Graph Convolutional Network (GCN)*.



Sumber: Diolah oleh penulis.

Gambar 3.2 : Alur Proposed Method Deteksi Transaksi Ilegal pada Jaringan *Bitcoin* Menggunakan *Graph Attention Network (GAT)* dengan Class Weight untuk Mendukung *Digital forensik*

Gambar 3.2 menunjukkan alur metode yang diusulkan dalam penelitian ini untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin* menggunakan *Graph Attention Network (GAT)* dengan penambahan *Class Weight*. Selain menggunakan metode yang diusulkan, penelitian ini juga melibatkan beberapa model pembanding agar dapat diketahui sejauh mana peningkatan performa yang dihasilkan oleh model GAT. Tahapan penelitian dimulai dari penggunaan Elliptic *Bitcoin* Dataset yang terdiri atas tiga file utama, yaitu features.csv, *edgelist*.csv, dan classes.csv. Dataset ini berisi informasi mengenai karakteristik transaksi, hubungan antar transaksi, serta label transaksi legal dan ilegal yang digunakan sebagai data penelitian.

Setelah dataset diperoleh, dilakukan proses data *preprocessing* yang meliputi penggabungan data (*merge*), pembersihan data (*cleaning*), perubahan label ke bentuk numerik (*label encoding*), normalisasi fitur menggunakan *Feature Scaling*, serta pembagian data menjadi data pelatihan dan data pengujian (*train/test split*). Tahapan ini dilakukan untuk memastikan data yang digunakan memiliki kualitas yang baik sehingga dapat diproses oleh model dengan lebih optimal.

Tahap berikutnya adalah *graph* construction, yaitu membentuk struktur *graph* dari dataset. Pada tahap ini setiap transaksi direpresentasikan sebagai *node*, sedangkan hubungan antartransaksi direpresentasikan sebagai *edge*. Representasi *graph* ini memungkinkan model mempelajari hubungan antartransaksi sehingga pola transaksi ilegal dapat dikenali dengan lebih baik dibandingkan hanya menggunakan informasi fitur transaksi saja. Setelah *graph* berhasil dibentuk, proses penelitian dibagi menjadi dua jalur. Jalur pertama menggunakan model pembanding (*baseline*), yaitu *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight*. Model-model tersebut digunakan sebagai acuan untuk membandingkan hasil yang diperoleh dari metode yang diusulkan.

Jalur kedua merupakan *proposed method* yang menggunakan *Graph Attention Network (GAT)* dengan *Class Weight*. Pada metode ini, data diproses melalui beberapa tahapan, dimulai dari input *node features*, kemudian setiap *node* dihitung tingkat kepentingannya melalui *graph* attention layer sehingga diperoleh

attention coefficient. Nilai attention tersebut digunakan dalam proses feature aggregation untuk menggabungkan informasi dari *node-node* yang saling terhubung. Selanjutnya hasil agregasi diproses menggunakan *graph* convolution dan diklasifikasikan menggunakan *Softmax classification*. Selama proses pelatihan model digunakan *Weighted* Cross Entropy Loss yang dikombinasikan dengan *Class Weight* untuk mengatasi ketidakseimbangan jumlah data antara transaksi legal dan transaksi ilegal. Seluruh hasil dari model pembanding maupun metode yang diusulkan kemudian dievaluasi menggunakan beberapa metrik, yaitu *Accuracy*, *Precision*, *Recall*, *F1-score*, ROC-AUC, dan *Confusion Matrix*. Evaluasi ini bertujuan untuk mengetahui model yang memiliki kemampuan terbaik dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*.

Tahap terakhir adalah digital forensik *analysis*. Hasil klasifikasi yang diperoleh dimanfaatkan untuk membantu proses identifikasi transaksi yang terindikasi mencurigakan, melakukan penelusuran aliran transaksi (*transaction tracing*), serta mendukung penyediaan bukti digital (*evidence support*) yang dapat digunakan dalam proses investigasi *digital forensik*.

3.2.2 Jadwal Penelitian

Jadwal penelitian disusun sebagai pedoman pelaksanaan setiap tahapan penelitian mulai dari perbaikan proposal, implementasi model, hingga penyusunan laporan tesis. Penyusunan jadwal ini bertujuan agar seluruh kegiatan penelitian dapat dilaksanakan secara sistematis sesuai target waktu yang telah ditetapkan. Adapun jadwal pelaksanaan penelitian ditunjukkan pada Tabel 3.1

Tabel 3.1 Jadwal Penelitian

No	Kegiatan	April 2026				Mei 2026				Juni 2026				Juli 2026			
		M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4	M1	M2	M3	M4
1	Perbaikan Proposal dan Pembahasan Dataset																
2	Seminar Proposal																
3	Revisi Judul dan Kerangka Teori																
4	Saran Metodologi dan Implementasi																

5	Implementasi Menggunakan SVM dan GCN															
6	Implementasi GAT dan GAT + *Class Weight*															
7	Penyusunan Bab IV Hasil Implementasi															
8	Revisi Laporan Hasil Eksperimen															
9	Revisi Bab I–V Keseluruhan															
10	Finalisasi Tesis															

3.3 Sumber dan Pengumpulan Data

Penelitian ini menggunakan dataset Elliptic yang merupakan dataset transaksi *Bitcoin* dan telah banyak digunakan dalam penelitian deteksi transaksi ilegal pada *Blockchain*, sebagaimana telah diuraikan pada Bab II. Dataset ini dipilih karena memiliki struktur data yang lengkap, yaitu mencakup fitur transaksi, hubungan antar transaksi, serta label kelas yang menunjukkan transaksi legal dan ilegal.

Dataset Elliptic terdiri dari tiga komponen utama, yaitu:

1. **Dataset Fitur Transaksi (elliptic_txs_features)**
   Dataset ini berisi fitur numerik yang menggambarkan karakteristik transaksi, seperti informasi statistik dan pola aktivitas transaksi.
2. **Dataset Hubungan Transaksi (elliptic_txs_*edgelist*)**
   Dataset ini merepresentasikan hubungan antartransaksi dalam bentuk *graph*. Setiap baris menunjukkan keterkaitan antara satu transaksi dengan transaksi lainnya, yang nantinya digunakan untuk membangun struktur *graph*.
3. **Dataset Label Transaksi (elliptic_txs_classes)**
   Dataset ini berisi label yang menunjukkan apakah suatu transaksi termasuk kategori legal atau ilegal. Data yang tidak memiliki label tidak digunakan dalam proses pelatihan model.

Dalam penelitian ini, ketiga dataset tersebut digabungkan berdasarkan identitas transaksi untuk membentuk satu kesatuan data yang utuh. Selanjutnya, dilakukan proses praproses data yang meliputi pembersihan data, normalisasi fitur, serta penghapusan data yang tidak memiliki label.

Selain itu, penelitian ini juga memperhatikan kondisi ketidakseimbangan data (*imbalance*d data), di mana jumlah transaksi ilegal lebih sedikit dibandingkan transaksi legal. Penanganan terhadap kondisi ini dilakukan sesuai dengan kebutuhan eksperimen agar tidak mempengaruhi kinerja model secara signifikan.

Dengan memanfaatkan ketiga komponen dataset tersebut, penelitian ini diharapkan mampu merepresentasikan karakteristik transaksi *Bitcoin* secara lebih komprehensif untuk mendukung proses deteksi transaksi ilegal. Implementasi model *Graph Attention Network (GAT)* pada penelitian ini dilakukan menggunakan *Google Colaboratory (Google Colab)* sebagai lingkungan pengembangan dan komputasi. *Google Colab* dipilih karena menyediakan lingkungan pemrograman berbasis *Python* yang mudah digunakan serta mendukung berbagai *library* yang dibutuhkan dalam pengembangan model *Machine learning* dan *Graph Neural Network*. Selain itu, *Google Colab* memungkinkan proses pelatihan model dilakukan secara lebih efisien tanpa memerlukan spesifikasi perangkat keras yang tinggi pada komputer pengguna.

Seluruh proses penelitian, mulai dari pembacaan dataset, prapemrosesan data, pembentukan struktur *graph*, pelatihan model GAT dan GCN, hingga evaluasi hasil klasifikasi dilakukan menggunakan bahasa pemrograman *Python*. Beberapa library digunakan untuk mendukung setiap tahapan penelitian, seperti *Pandas* untuk mengolah dataset, *NumPy* untuk komputasi numerik, *PyTorch* dan *PyTorch Geometric* untuk membangun model *Graph Convolutional Network*, NetworkX untuk mengelola struktur *graph*, *Scikit-learn* untuk proses evaluasi model dan implementasi metode pembanding *Support Vector Machine (SVM)*, serta *Matplotlib* untuk menampilkan visualisasi hasil penelitian. Walaupun proses pelatihan model dilakukan menggunakan *Google Colab*, penelitian ini tetap menggunakan sebuah laptop sebagai media untuk mengembangkan program,

mengelola dataset, serta mendokumentasikan hasil penelitian. Spesifikasi perangkat yang digunakan selama penelitian dapat dilihat pada Tabel 3.1.

Tabel 3.2 Spesifikasi Perangkat yang Digunakan

Komponen	**Spesifikasi**
Prosesor	AMD Ryzen 5 Processor with AMD Radeon *Graphics*
Sistem Operasi	Windows 11 64-bit
Lingkungan Komputasi	Google Colaboratory (Google Colab)
Bahasa Pemrograman	*Python*
GPU *(Google Colab)*	NVIDIA Tesla T4, 16 GB VRAM
RAM Runtime *(Google Colab)*	± 12–13 GB

Sumber: Diolah oleh penulis (2026)

Adapun perangkat lunak dan library yang digunakan selama proses implementasi ditunjukkan pada Tabel 3.3.

Tabel 3.3 Perangkat Lunak dan Library

Perangkat Lunak / Library	**Fungsi**
Google Colaboratory	Menjalankan program dan proses pelatihan model
Python	Bahasa pemrograman utama
Pandas	Pengolahan dan manipulasi dataset
NumPy	Perhitungan numerik
PyTorch	*Framework deep learning*
PyTorch Geometric	Implementasi *Graph Attention Network*
Scikit-learn	Evaluasi model dan implementasi SVM
NetworkX	Pembentukan dan analisis *graph*
Matplotlib	Visualisasi grafik hasil penelitian

Sumber: Diolah oleh penulis (2026)

Konfigurasi *hyperparameter* ditetapkan tersendiri untuk tiap model agar sesuai dengan karakteristik arsitekturnya masing-masing, dengan mengacu pada konfigurasi yang umum digunakan pada penelitian sejenis. Rincian konfigurasi *hyperparameter* kelima model yang dibandingkan pada penelitian ini disajikan pada Tabel 3.4.

Tabel 3.4 Konfigurasi *Hyperparameter* Kelima Model

Parameter	SVM	GCN	GCN + Class Weight	GAT	GAT + Class Weight
Kernel	RBF	–	–	–	–
Input Feature	165	165	165	165	165
Hidden Channel	–	64	64	64	64
Attention Head	–	–	–	4 → 1	4 → 1
Output Class	2	2	2	2	2
Optimizer	–	Adam	Adam	Adam	Adam
Learning Rate	–	0,01	0,01	0,01	0,01
Epoch	–	200	200	200	200
Activation Function	–	ReLU	ReLU	ELU	ELU
Dropout	–	–	–	0.2	0.2
Loss function	–	CrossEntropyLoss	CrossEntropy Loss	CrossEntropyLoss	CrossEntropy Loss
Class Weight (Balanced)	–	No	Yes	No	Yes
Train : Test Split	80 : 20	80 : 20	80 : 20	80 : 20	80 : 20
Weight Decay	–	–	–	–	–
Random Seed	42	42	42	42	42
Device	CPU	CPU/GPU	CPU/GPU	CPU/GPU	CPU/GPU

Sumber: Diolah oleh penulis (2026), disesuaikan berdasarkan hasil verifikasi kode implementasi

Berdasarkan Tabel 3.4, model SVM tidak memanfaatkan struktur *graph* sehingga hanya dikonfigurasi dengan kernel RBF, sedangkan keempat model berbasis *graph* menggunakan 165 fitur input, 64 hidden channel, dan 2 kelas output (licit dan illicit), serta dilatih selama 200 *epoch* menggunakan optimizer Adam dengan learning rate yang sama, yaitu 0,01. GCN dan GCN dengan *Class Weight*

menggunakan fungsi aktivasi ReLU, sedangkan GAT dan GAT dengan *Class Weight* menggunakan fungsi aktivasi ELU karena mekanisme multi-head attention memungkinkan model mempelajari representasi hubungan antartransaksi dari berbagai perspektif sehingga informasi struktural *graph* dapat dimanfaatkan secara lebih optimal. Model GAT menggunakan 4 attention head pada layer pertama dan 1 attention head pada layer kedua. Dengan hidden dimension sebesar 64, keluaran layer pertama memiliki dimensi 256 (64 × 4). Model juga menggunakan dropout sebesar 0,2 sebagai regularisasi untuk membantu mengurangi risiko overfitting selama proses pelatihan. Pendekatan ini dipilih karena selama eksperimen tidak ditemukan indikasi *overfitting* yang memerlukan regularisasi tambahan seperti *dropout* maupun *weight decay*.

Selain menetapkan konfigurasi *hyperparameter* utama pada Tabel 3.4, penelitian ini juga melakukan pengujian sensitivitas terhadap salah satu *hyperparameter* penting pada model berbasis *graph*, yaitu *hidden dimension*. Pengujian dilakukan dengan mencoba beberapa nilai *hidden dimension*, yaitu 16, 32, 64, dan 128, pada model GCN dan GAT untuk mengetahui pengaruhnya terhadap performa klasifikasi yang diukur menggunakan metrik *F1-score*. Nilai *hidden dimension* yang menghasilkan performa terbaik kemudian dipilih sebagai konfigurasi akhir yang digunakan pada seluruh model dalam penelitian ini, sebagaimana telah ditetapkan sebesar 64 pada Tabel 3.4. Hasil lengkap dari pengujian sensitivitas *hidden dimension* ini disajikan dan dibahas lebih lanjut pada Bab IV subbab 4.3.1.

Untuk model dengan *Class Weight* (GCN dan GAT), bobot kelas dihitung menggunakan skema *balanced*, yaitu $w_c = n / (k \times n_c)$, dengan n adalah jumlah total data latih, k adalah jumlah kelas, dan n_c adalah jumlah data pada kelas c. Bobot ini diikutsertakan ke dalam fungsi *CrossEntropyLoss* sehingga kesalahan klasifikasi pada kelas illicit, yang jumlahnya jauh lebih sedikit dibanding kelas licit, diberi penalti yang lebih besar selama proses pelatihan. Pembagian data latih dan data uji (80:20) dilakukan secara *stratified* dengan random_state tetap agar proporsi kelas terjaga dan hasil eksperimen dapat direproduksi.

3.3.1 Alur Penelitian

Alur penelitian pada tahap implementasi disusun secara sistematis untuk memastikan bahwa setiap tahapan dilaksanakan sesuai dengan metodologi penelitian yang telah dijelaskan pada Bab III. Proses dimulai dari penggunaan Elliptic *Bitcoin* Dataset sebagai sumber data utama, dilanjutkan dengan eksplorasi dataset untuk memahami karakteristik data, kemudian dilakukan data *preprocessing*, pembentukan *graph*, implementasi dan pelatihan *Graph Attention Network* (GAT), evaluasi model, analisis hasil deteksi transaksi *fraud*, serta pembahasan hasil dalam konteks *digital forensik*.

Tabel 3.5 Ringkasan Tahapan Penelitian

Tahapan Penelitian	**Uraian**
Eksplorasi Dataset	Menganalisis karakteristik Elliptic *Bitcoin* Dataset, meliputi distribusi kelas, struktur *graph*, korelasi fitur, dan distribusi transaksi berdasarkan *time step*.
Data *Preprocessing*	Melakukan penggabungan dataset, pemeriksaan kualitas data, *label encoding*, serta normalisasi fitur sebelum data digunakan oleh model.
Graph Construction	Membentuk struktur *graph* dari dataset dengan merepresentasikan setiap transaksi sebagai *node* dan hubungan antartransaksi sebagai *edge* sehingga dapat diproses oleh *Graph Attention Network* (GAT).
Implementasi Model GAT	Membangun arsitektur model *Graph Attention Network (GAT)* beserta konfigurasi parameter pelatihan.

Pelatihan (*Training*) Model	Melatih model menggunakan data *training* agar mampu mengenali pola transaksi legal dan ilegal.
Evaluasi Model	Mengevaluasi performa model menggunakan *Accuracy*, *Precision*, *Recall*, *F1-score*, *Confusion Matrix*, ROC Curve, AUC, dan *Classification Report*.
Analisis Deteksi Transaksi *Fraud*	Menganalisis hasil deteksi transaksi ilegal melalui *Neighbor Risk Analysis*, distribusi risiko, *False Positive*, *False Negative*, dan pola penyebaran transaksi *fraud*.
Analisis *Digital forensik*	Menginterpretasikan hasil deteksi transaksi *fraud* untuk mendukung identifikasi serta investigasi transaksi mencurigakan pada jaringan *Bitcoin*.

Sumber: Diolah oleh Penulis (2026)

Secara keseluruhan, dataset yang digunakan terdiri dari 203.769 transaksi, yang terbagi menjadi 157.205 transaksi berlabel *Unknown***,** 42.019 transaksi berlabel licit**,** dan 4.545 transaksi berlabel illicit. Komposisi tersebut menunjukkan bahwa jumlah transaksi ilegal jauh lebih sedikit dibandingkan transaksi legal sehingga dataset memiliki karakteristik *class imbalance*. Kondisi ini menjadi salah satu tantangan dalam proses klasifikasi karena model harus mampu mengenali transaksi ilegal yang jumlahnya relatif sedikit tanpa menurunkan performa klasifikasi secara keseluruhan.

Selain memiliki informasi label, Elliptic Dataset juga menyimpan hubungan antartransaksi yang membentuk struktur *graph*. Dalam penelitian ini, setiap transaksi direpresentasikan sebagai *node*, sedangkan hubungan antartransaksi direpresentasikan sebagai *edge*. Representasi tersebut memungkinkan *Graph Attention Network (GAT)* mempelajari karakteristik suatu transaksi berdasarkan

atribut yang dimilikinya sekaligus mempertimbangkan informasi dari transaksi lain yang saling terhubung dalam jaringan *Blockchain*. Pendekatan ini menjadi salah satu keunggulan GAT dibandingkan metode klasifikasi konvensional yang hanya memanfaatkan fitur transaksi secara individual.

Ringkasan karakteristik Elliptic *Bitcoin* Dataset yang digunakan dalam penelitian ini disajikan pada Tabel 3.6**.**

Tabel 3.6 Karakteristik Elliptic *Bitcoin* Dataset

Komponen	Keterangan
Nama Dataset	Elliptic *Bitcoin* Dataset
Jumlah *Node* (Transaksi)	203.769 transaksi
Jumlah *Edge* (Hubungan Transaksi)	234.355 *edge*
Jumlah Fitur	165 fitur
Jumlah *Time step*	49 periode
Jumlah Label	3 (Licit, Illicit, *Unknown*)
Dataset Fitur	*elliptic_txs_features.csv*
Dataset Hubungan	elliptic_txs_*edgelist*.csv
Dataset Label	*elliptic_txs_classes.csv*

Sumber: Diolah oleh penulis berdasarkan Elliptic *Bitcoin* Dataset (2026)

Berdasarkan Tabel 3.6, Elliptic *Bitcoin* Dataset memiliki struktur *graph* yang terdiri atas 203.769 *node* yang merepresentasikan transaksi *Bitcoin* dan 234.355 *edge* yang menggambarkan hubungan antar transaksi. Setiap transaksi memiliki 165 fitur sebagai atribut masukan model serta tersebar dalam 49 periode waktu (*time step*)**.** Selain itu, dataset terdiri atas tiga kategori label, yaitu *Licit*, *Illicit*, dan *Unknown*. Karakteristik tersebut menunjukkan bahwa dataset tidak hanya menyimpan informasi atribut transaksi, tetapi juga struktur hubungan antar transaksi, sehingga sangat sesuai untuk diterapkan pada metode *Graph Attention Network (GAT)* yang memanfaatkan informasi fitur dan topologi *graph* secara bersamaan. Elliptic *Bitcoin* Dataset terdiri atas tiga file utama yang saling melengkapi dalam proses pembentukan dataset penelitian. Setiap file memiliki fungsi yang berbeda, mulai dari penyimpanan fitur transaksi, label transaksi, hingga hubungan antartransaksi

yang membentuk struktur *graph*. Ketiga file tersebut digunakan secara bersamaan pada tahap *preprocessing* sebelum diimplementasikan ke dalam model *Graph Attention Network* (GAT). Rincian fungsi masing-masing file disajikan pada Tabel 3.7.

Tabel 3.7 File Elliptic *Bitcoin* Dataset

Nama File	Fungsi
elliptic_txs_features.csv	Menyimpan 165 fitur numerik yang merepresentasikan karakteristik setiap transaksi *Bitcoin* sebagai masukan model GAT.
elliptic_txs_classes.csv	Menyimpan label setiap transaksi, yaitu Licit, Illicit, dan *Unknown* yang digunakan sebagai target klasifikasi.
elliptic_txs_*edgelist*.csv	Menyimpan hubungan antartransaksi (*edge*) yang digunakan untuk membentuk struktur *graph* pada jaringan *Bitcoin*.

Sumber: Diolah oleh penulis berdasarkan Elliptic *Bitcoin* Dataset (2026)

Berdasarkan Tabel 3.7, setiap file pada Elliptic *Bitcoin* Dataset memiliki peran yang berbeda namun saling berkaitan. File fitur menyediakan informasi karakteristik setiap transaksi, file label digunakan sebagai acuan dalam proses klasifikasi, sedangkan file hubungan transaksi digunakan untuk membentuk struktur *graph* yang menjadi keunggulan utama *Graph Attention Network* (GAT). Dengan menggabungkan ketiga file tersebut, model dapat mempelajari tidak hanya atribut transaksi, tetapi juga hubungan antartransaksi pada jaringan *Blockchain Bitcoin*.

3.3.2 Distribusi Label

Setelah karakteristik umum dataset dipahami, tahap berikutnya adalah menganalisis distribusi label transaksi pada Elliptic *Bitcoin* Dataset. Analisis ini bertujuan untuk mengetahui komposisi data yang akan digunakan dalam proses pelatihan dan pengujian model *Graph Attention Network* (GAT). Selain itu, distribusi label juga

memberikan gambaran mengenai keseimbangan kelas (*class balance*) yang dapat memengaruhi performa model klasifikasi.

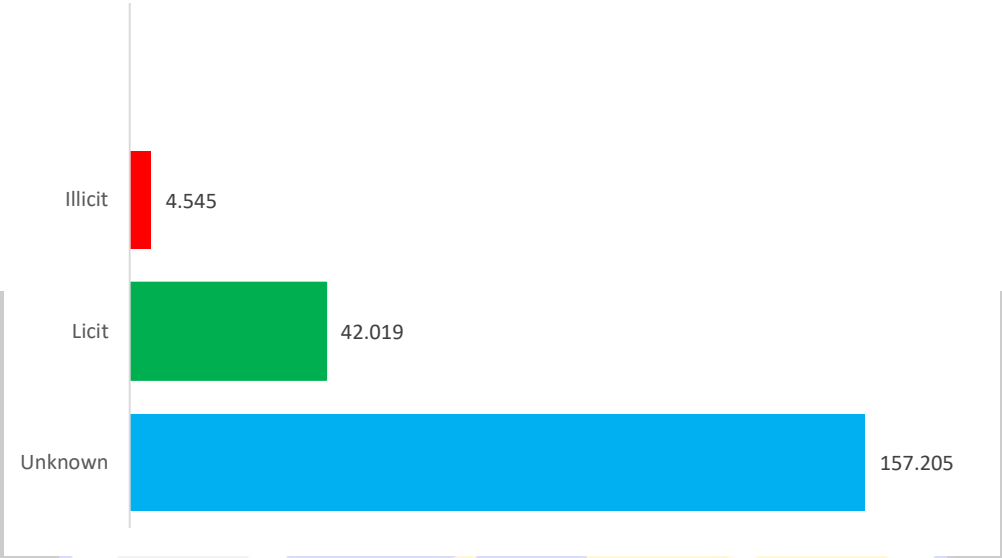
Dataset Elliptic terdiri atas tiga kategori label, yaitu *Unknown*, Licit, dan Illicit. Label *Unknown* menunjukkan transaksi yang belum memiliki kategori yang jelas sehingga tidak digunakan dalam proses pelatihan model. Sementara itu, label Licit merepresentasikan transaksi legal dan label Illicit merepresentasikan transaksi ilegal atau *fraud*. Komposisi masing-masing label ditunjukkan pada Tabel 3.8

Tabel 3.8 Distribusi Label Elliptic *Bitcoin* Dataset

Label	Jumlah	Persentase
Unknown	157.205	77,15%
Licit	42.019	20,62%
Illicit	4.545	2,23%

Sumber: Diolah oleh Penulis (2026)

Berdasarkan Tabel 3.8, sebagian besar transaksi pada Elliptic *Bitcoin* Dataset berlabel *Unknown*, yaitu sebanyak 157.205 transaksi atau 77,15% dari keseluruhan data. Transaksi berlabel Licit berjumlah 42.019 transaksi (20,62%), sedangkan transaksi berlabel Illicit hanya sebanyak 4.545 transaksi (2,23%). Perbedaan jumlah yang cukup besar antara transaksi legal dan ilegal menunjukkan bahwa dataset memiliki karakteristik *class imbalance*. Kondisi ini menjadi tantangan dalam proses klasifikasi karena model cenderung lebih mudah mempelajari kelas yang jumlahnya lebih banyak. Oleh karena itu, penanganan terhadap ketidakseimbangan kelas menjadi salah satu aspek penting dalam pengembangan model GAT pada penelitian ini. Untuk memberikan gambaran yang lebih jelas mengenai komposisi setiap kategori transaksi pada Elliptic *Bitcoin* Dataset, distribusi label divisualisasikan menggunakan diagram batang *(bar chart)*. Visualisasi ini memudahkan pembaca dalam membandingkan jumlah transaksi pada masing-masing kategori sehingga karakteristik dataset dapat dipahami dengan lebih baik. Distribusi label transaksi pada Elliptic *Bitcoin* Dataset ditunjukkan pada Gambar 3.3.



Sumber: Diolah oleh penulis berdasarkan Elliptic *Bitcoin* Dataset (2026)

Gambar 3.3 Distribusi Label Transaksi pada Elliptic *Bitcoin* Dataset

Berdasarkan Gambar 3.3 terlihat bahwa kategori *Unknown* memiliki jumlah transaksi paling banyak, yaitu 157.205 transaksi. Kategori Licit berjumlah 42.019 transaksi, sedangkan kategori Illicit/*fraud* hanya terdiri atas 4.545 transaksi. Perbedaan yang cukup besar tersebut menunjukkan bahwa dataset memiliki karakteristik *class imbalance* sehingga menjadi salah satu tantangan dalam proses pelatihan model *Graph Attention Network* (GAT).

3.3.3 Ringkasan Hasil Eksplorasi Dataset

Berdasarkan hasil eksplorasi yang telah dilakukan, diperoleh berbagai informasi mengenai karakteristik Elliptic *Bitcoin* Dataset yang menjadi dasar dalam proses data *preprocessing* dan pembangunan model *Graph Attention Network* (GAT). Tahap eksplorasi memberikan gambaran mengenai distribusi data, struktur *graph*, hubungan antar fitur, serta pola transaksi berdasarkan *time step* sehingga proses implementasi model dapat dilakukan secara lebih terarah.

Tabel 3.9 Ringkasan Hasil Eksplorasi Dataset

No	Hasil Eksplorasi	Temuan
1	Karakteristik Dataset	Dataset terdiri atas 203.769 transaksi, 234.355 *edge*, 165 fitur, dan 49 *time step*.
2	Distribusi Label	157.205 transaksi *Unknown*, 42.019 transaksi licit, dan 4.545 transaksi illicit sehingga dataset bersifat tidak seimbang (*class imbalance*).
3	Struktur *Graph*	Setiap transaksi direpresentasikan sebagai *node* dan hubungan antartransaksi direpresentasikan sebagai *edge* sehingga membentuk struktur *graph*.
4	Visualisasi *Graph*	Visualisasi menunjukkan adanya hubungan antartransaksi serta kelompok (*cluster*) transaksi berisiko tinggi yang saling terhubung.
5	Korelasi Fitur	Sebagian besar fitur memiliki korelasi rendah hingga sedang, namun terdapat beberapa pasangan fitur dengan korelasi sangat tinggi (>0,99).
6	Analisis *Time step*	Distribusi transaksi dan transaksi *fraud* tidak merata pada 49 *time step* sehingga menunjukkan pola aktivitas yang dinamis.

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset (2026)

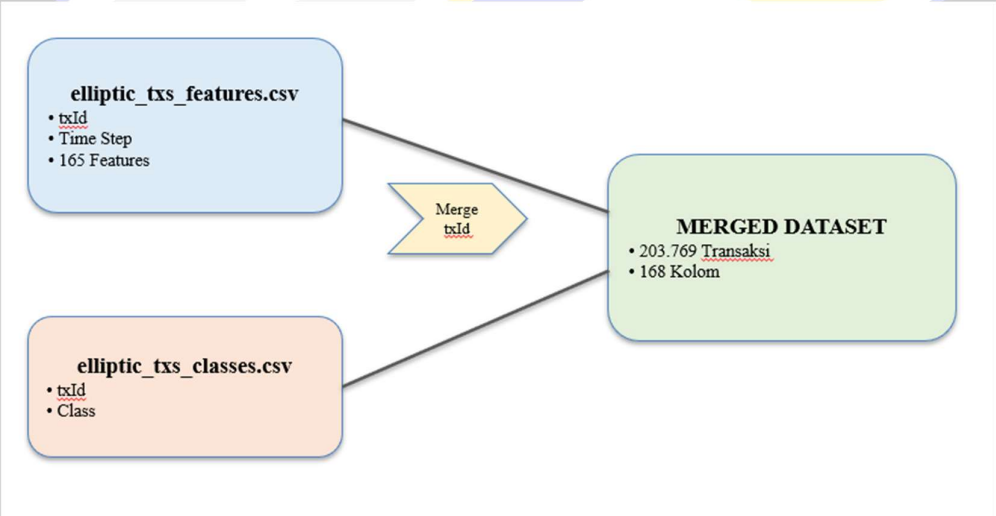
Berdasarkan Tabel 3.9 dapat disimpulkan bahwa Elliptic *Bitcoin* Dataset memiliki karakteristik yang kompleks karena memadukan atribut transaksi dengan hubungan antartransaksi dalam bentuk *graph*. Selain memiliki ketidakseimbangan kelas, dataset juga menunjukkan pola transaksi yang berubah pada setiap *time step* serta hubungan antar fitur yang bervariasi. Karakteristik tersebut menunjukkan bahwa pendekatan berbasis *graph* lebih sesuai digunakan untuk mendeteksi transaksi ilegal dibandingkan metode yang hanya memanfaatkan atribut transaksi.

Secara keseluruhan, hasil eksplorasi dataset memberikan dasar yang kuat untuk melanjutkan ke tahap data *preprocessing*. Informasi mengenai distribusi label, struktur *graph*, korelasi fitur, dan pola *time step* menjadi acuan dalam proses penggabungan data, pembersihan data, transformasi label, normalisasi fitur, serta pembentukan *graph* yang selanjutnya digunakan sebagai masukan bagi model *Graph Attention Network* (GAT).

3.4 Data *Preprocessing*

3.4.1 Penggabungan Dataset (*Merge* Dataset)

Pada tahap ini dilakukan proses penggabungan (*merge*) antara dataset fitur transaksi (*elliptic_txs_features.csv*) dengan dataset label transaksi (*elliptic_txs_classes.csv*). Penggabungan dilakukan menggunakan atribut txId sebagai kunci utama (*primary key*) sehingga setiap transaksi memiliki informasi fitur dan label kelas dalam satu dataset yang terintegrasi. Tahap ini bertujuan untuk mempersiapkan data sebelum dilakukan proses *preprocessing* lebih lanjut seperti pemeriksaan *missing value*, normalisasi fitur, dan pembentukan struktur *graph*.



Sumber: Diolah oleh penulis berdasarkan Elliptic *Bitcoin* Dataset (2026)

Gambar 3.4 Proses Penggabungan Dataset (*Merge* Dataset)

Berdasarkan Gambar 3.4, proses penggabungan dilakukan dengan mengintegrasikan file *elliptic_txs_features.csv* dan *elliptic_txs_classes.csv* berdasarkan atribut txId. Hasil proses tersebut menghasilkan dataset yang terdiri

atas 203.769 transaksi dengan 168 kolom, yang mencakup informasi identitas transaksi, *time step*, fitur transaksi, dan label kelas. Dataset hasil penggabungan selanjutnya digunakan sebagai dasar pada tahapan *preprocessing* berikutnya sebelum dilakukan pembentukan struktur *graph* dan pelatihan model *Graph Attention Network* (GAT).

Tabel 3.10 Hasil Penggabungan Dataset

Komponen	Jumlah
Jumlah Transaksi (Baris)	203.769
Jumlah Kolom	168
Kolom Identitas (txId)	1
Kolom *Time step*	1
Jumlah Fitur Transaksi	165
Kolom Label	1

Sumber: Diolah oleh Penulis (2026)

```
df = features.merge(
    classes,
    on='txId'
)

print(df.shape)
```

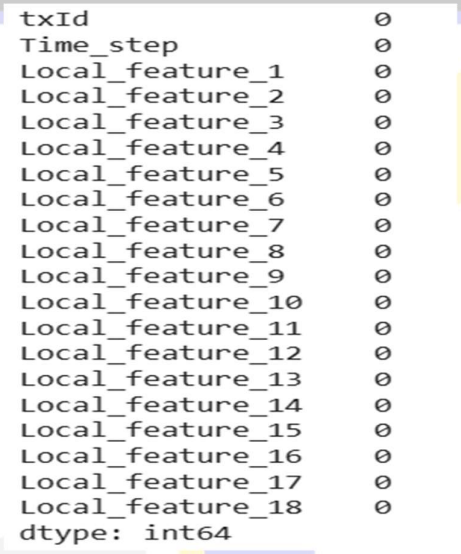
```
(203769, 168)
```

Gambar 3.5 Hasil Penggabungan Dataset

Berdasarkan Gambar 3.5 diketahui bahwa proses penggabungan dataset berhasil dilakukan tanpa mengalami kehilangan data. Dataset hasil penggabungan memiliki ukuran 203.769 baris (transaksi) dan 168 kolom. Jumlah tersebut terdiri atas 1 kolom txId**,** 1 kolom *time step***,** 165 fitur transaksi, dan 1 kolom label kelas. Dataset hasil penggabungan inilah yang selanjutnya digunakan pada tahap *preprocessing* berikutnya.

3.4.2 Pemeriksaan *Missing value*

Setelah proses penggabungan dataset selesai dilakukan, tahap berikutnya adalah melakukan pemeriksaan terhadap *missing value*. Pemeriksaan ini bertujuan untuk memastikan bahwa seluruh atribut pada dataset telah terisi dengan lengkap sehingga dapat digunakan dalam proses pelatihan model *Graph Attention Network (GAT)* tanpa memerlukan penanganan data yang hilang.



```
txId                0
Time_step           0
Local_feature_1     0
Local_feature_2     0
Local_feature_3     0
Local_feature_4     0
Local_feature_5     0
Local_feature_6     0
Local_feature_7     0
Local_feature_8     0
Local_feature_9     0
Local_feature_10    0
Local_feature_11    0
Local_feature_12    0
Local_feature_13    0
Local_feature_14    0
Local_feature_15    0
Local_feature_16    0
Local_feature_17    0
Local_feature_18    0
dtype: int64
```

Sumber: Diolah oleh penulis

Gambar 3.6 Hasil Pemeriksaan *Missing value* pada Dataset

Berdasarkan Gambar 3.6, hasil pemeriksaan menggunakan fungsi `isnull().sum()` menunjukkan bahwa seluruh atribut pada dataset memiliki nilai 0, yang berarti tidak ditemukan data yang hilang (*missing value*). Hal ini terlihat pada atribut txId**,** Time_step, maupun seluruh atribut fitur transaksi yang ditampilkan. Kondisi tersebut menunjukkan bahwa dataset memiliki kualitas data yang baik dan dapat langsung digunakan pada tahapan *preprocessing* berikutnya tanpa memerlukan proses imputasi ataupun penghapusan data.

Untuk memperjelas hasil pemeriksaan *missing value*, ringkasan jumlah data yang hilang pada setiap komponen dataset disajikan pada Tabel 3.11.

Tabel 3.11 Hasil Pemeriksaan Missing Value pada Elliptic Bitcoin Dataset

Komponen	Jumlah *Missing value*	Keterangan
txId	0	Tidak terdapat data yang hilang
Time step	0	Tidak terdapat data yang hilang
165 Fitur Transaksi	0	Seluruh fitur lengkap
Label Transaksi	0	Seluruh label tersedia
Total *Missing value*	0	Dataset lengkap dan siap diproses

Sumber: Diolah oleh Penulis (2026)

Berdasarkan Tabel 3.11, seluruh komponen pada Elliptic *Bitcoin* Dataset memiliki jumlah *missing value* sebesar 0. Hal ini menunjukkan bahwa tidak terdapat data yang hilang baik pada kolom identitas transaksi **(**txId**),** *Time step***,** 165 fitur transaksi, maupun label transaksi. Dengan demikian, dataset memiliki kualitas yang baik dan tidak memerlukan proses penanganan *missing value* seperti imputasi ataupun penghapusan data. Dataset selanjutnya dapat digunakan pada tahapan normalisasi fitur, pemisahan transaksi Unknown dari data yang digunakan sebagai target pelatihan dan pengujian model, serta pembentukan struktur graph. Transaksi Unknown tetap dipertahankan untuk kebutuhan eksplorasi dan analisis lanjutan setelah model selesai dilatih.

3.4.3 Persiapan Data untuk Model Berbasis *Graph* (*Data Preparation*)

Setelah proses penggabungan dataset dan pemeriksaan *missing value* selesai dilakukan, tahap berikutnya adalah persiapan data (*Data Preparation*) sebelum proses pelatihan model berbasis *graph*, yaitu *Graph Attention Network (GAT)* sebagai model utama beserta *Graph Convolutional Network (GCN)* sebagai model pembanding. Tahapan ini bertujuan untuk memastikan bahwa data memiliki format yang sesuai dengan kebutuhan model. Pada penelitian ini, proses *Data Preparation* meliputi *label encoding*, *Feature Scaling*, dan *graph* construction sehingga dataset siap digunakan sebagai masukan *(input)* pada model GAT maupun GCN.

Pada proses *label encoding*, label transaksi yang semula berbentuk kategori diubah menjadi nilai numerik agar dapat diproses oleh model. Pada penelitian ini, transaksi licit direpresentasikan dengan nilai 0, sedangkan transaksi illicit direpresentasikan

dengan nilai 1. Transaksi dengan label *Unknown* tidak disertakan dalam proses pelatihan karena tidak memiliki kelas yang pasti.

Tabel 3.12 Hasil Transformasi Label / Label Encoding

Label Asli	Nilai Numerik	Keterangan
Licit	0	Transaksi legal (normal)
Illicit	1	Transaksi ilegal (*fraud*)

Sumber: Diolah oleh Penulis (2026)

Setelah label diubah menjadi bentuk numerik, dilakukan proses *Feature Scaling* terhadap seluruh fitur transaksi. Proses ini bertujuan untuk menyamakan rentang nilai antar fitur sehingga tidak terdapat atribut yang memiliki pengaruh lebih besar hanya karena memiliki skala numerik yang berbeda. Pada penelitian ini digunakan metode *StandardScaler*, yang mentransformasikan setiap fitur sehingga memiliki rata-rata mendekati nol dan simpangan baku mendekati satu. Proses normalisasi ini membantu meningkatkan stabilitas dan efektivitas proses pembelajaran model berbasis *graph*.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)

print(X_scaled.shape)
```

```
(46564, 165)
```

Sumber: Dataset Penelitian (2026)

Gambar 3.7 Hasil *Feature Scaling* Menggunakan *StandardScaler*

Berdasarkan hasil *preprocessing*, jumlah data dan jumlah fitur tetap sebanyak 46.564 transaksi dengan 165 fitur, sehingga proses normalisasi hanya mengubah skala nilai tanpa mengubah struktur dataset.

Tahap terakhir pada proses *Data Preparation* adalah *graph* construction, yaitu mengubah dataset transaksi menjadi struktur *graph* yang dapat diproses oleh model berbasis *graph* seperti GAT dan GCN. Pada tahap ini setiap transaksi

direpresentasikan sebagai sebuah *node*, sedangkan hubungan antartransaksi direpresentasikan sebagai *edge* berdasarkan file elliptic_txs_*edgelist*.csv. Selanjutnya, seluruh informasi *graph* disusun ke dalam objek Data pada *PyTorch Geometric* yang terdiri atas matriks fitur (**x**), hubungan antar*node* (*edge*_index), dan label transaksi (**y**). Struktur *graph* inilah yang digunakan sebagai masukan pada proses pelatihan model GAT maupun GCN.

```
from torch_geometric.data import Data

data = Data(
    x=x,
    edge_index=edge_index,
    y=y
)

print(data)
```

```
Data(x=[46564, 165], edge_index=[2, 36624], y=[46564])
```

Sumber: Dataset Penelitian (2026)

Gambar 3.8 Struktur *Graph* Hasil *Data Preparation*

Berdasarkan hasil *Data Preparation*, seluruh proses *preprocessing* telah berhasil dilakukan. Label transaksi telah dikonversi ke dalam bentuk numerik, fitur transaksi telah dinormalisasi menggunakan StandardScaler, serta struktur *graph* telah berhasil dibentuk sesuai kebutuhan model berbasis *graph* yang digunakan pada penelitian ini. Dengan demikian, dataset telah siap digunakan pada tahap pembagian data *training* dan testing sebelum dilakukan proses pelatihan model.

Setelah seluruh tahapan *data preprocessing* selesai dilakukan, langkah berikutnya adalah melakukan pembagian dataset menjadi data *training* dan data *testing*. Pembagian data bertujuan untuk memisahkan data yang digunakan dalam proses pelatihan model dengan data yang digunakan untuk proses pengujian. Dengan demikian, kinerja model dapat dievaluasi menggunakan data yang belum pernah digunakan selama proses pelatihan sehingga hasil evaluasi lebih objektif.

Pada penelitian ini, pembagian data dilakukan menggunakan fungsi train_test_split() dari pustaka *Scikit-learn* dengan rasio 80% data *training* dan 20%

data testing. Selain itu, parameter *stratify* digunakan untuk mempertahankan proporsi transaksi legal (*licit*) dan transaksi ilegal (*illicit*) pada kedua kelompok data sehingga distribusi kelas tetap seimbang. Selanjutnya, indeks hasil pembagian dikonversi menjadi train_mask dan test_mask agar dapat digunakan pada proses pelatihan menggunakan *framework PyTorch Geometric*.

```
data.train_mask = torch.zeros(
    len(y),
    dtype=torch.bool
)

data.test_mask = torch.zeros(
    len(y),
    dtype=torch.bool
)

data.train_mask[train_idx] = True
data.test_mask[test_idx] = True

print(data.train_mask.sum())
print(data.test_mask.sum())
```

```
tensor(37251)
tensor(9313)
```

Sumber: Dataset Penelitian (2026)

Gambar 3.9 Hasil Pembagian Data *Training* dan Testing

Berdasarkan Gambar 3.9, proses pembagian dataset berhasil dilakukan dengan membentuk train mask dan test mask. Hasil keluaran menunjukkan bahwa terdapat 37.251 *node* yang digunakan sebagai data *training*, sedangkan 9.313 *node* digunakan sebagai data testing. Kedua mask tersebut direpresentasikan dalam bentuk tensor bertipe boolean, di mana nilai True menunjukkan *node* yang termasuk ke dalam kelompok data *training* maupun testing. Penggunaan train mask dan test mask merupakan mekanisme standar pada *PyTorch Geometric* untuk menentukan *node* mana yang digunakan selama proses pelatihan dan evaluasi model GAT maupun GCN.

Pembagian dataset menjadi data *training* dan data testing dilakukan untuk memastikan bahwa proses evaluasi model dilakukan secara objektif menggunakan data yang tidak digunakan selama pelatihan. Dengan demikian, performa model berbasis *graph* dapat diukur secara lebih akurat dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Dataset yang telah dibagi ini selanjutnya digunakan pada tahap pelatihan model GAT maupun GCN.

Tabel 3.13 Pembagian Data *Training* dan Testing

Dataset	Jumlah *Node*	Persentase
Training	37.251	80%
Testing	9.313	20%
Total	**46.564**	**100%**

Sumber: Diolah oleh Penulis (2026)

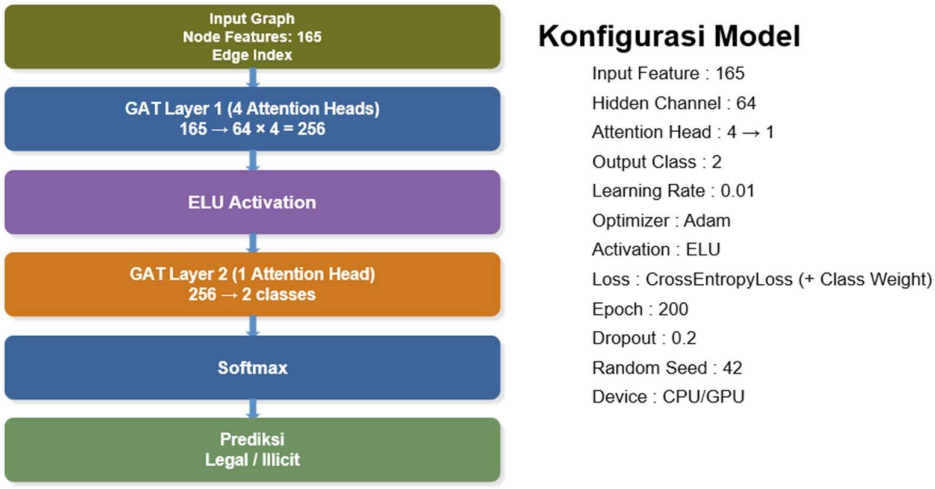
Berdasarkan Tabel 3.13, dataset penelitian berhasil dibagi menjadi 37.251 *node* (80%) sebagai data *training* dan 9.313 *node* (20%) sebagai data testing, sehingga total data yang digunakan dalam penelitian berjumlah 46.564 *node*. Komposisi pembagian data ini mengikuti rasio 80:20, yang umum digunakan dalam penelitian *Machine learning* karena mampu memberikan keseimbangan antara jumlah data untuk proses pembelajaran model dan jumlah data untuk proses evaluasi.

Data *training* digunakan untuk melatih model *Graph Attention Network (GAT)* agar mampu mempelajari pola hubungan antartransaksi *Bitcoin*, sedangkan data testing digunakan untuk mengukur kemampuan model dalam mengklasifikasikan transaksi yang belum pernah dipelajari sebelumnya. Dengan pembagian data tersebut, proses evaluasi dapat dilakukan secara lebih objektif sehingga hasil pengujian model dapat menggambarkan kemampuan generalisasi model terhadap data baru.

3.5 Implementasi Model *Graph* Attention Network (GAT)

3.5.1 Implementasi dan Konfigurasi Model GAT

Pada tahap ini dilakukan implementasi model *Graph Attention Network (GAT)* untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Model dibangun menggunakan *PyTorch Geometric* karena *framework* ini mampu mengolah data yang berbentuk *graph*. Berbeda dengan metode klasifikasi biasa yang hanya melihat karakteristik setiap transaksi secara terpisah, GAT juga mempertimbangkan hubungan antartransaksi melalui mekanisme attention, sehingga model dapat memberi bobot berbeda pada tiap hubungan sesuai tingkat kepentingannya dan informasi yang diperoleh menjadi lebih lengkap.



Sumber: Dataset Penelitian (2026)

Gambar 3.10 Arsitektur Model *Graph* Attention Network (GAT)

Gambar 3.10 menunjukkan arsitektur model *Graph Attention Network (GAT)* dengan penerapan *Class Weight* yang digunakan dalam penelitian ini untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Model ini memanfaatkan informasi atribut setiap *node* dan hubungan antar*node* dalam *graph* sehingga mampu mempelajari pola transaksi secara lebih komprehensif dibandingkan metode klasifikasi konvensional.

Proses diawali dengan input *graph* yang terdiri atas 165 fitur (*node* features) dan *edge* index. *Node* features merepresentasikan karakteristik setiap transaksi *Bitcoin*, sedangkan *edge* index menggambarkan hubungan antartransaksi dalam jaringan *Blockchain*. Kedua komponen tersebut menjadi masukan utama bagi model untuk mempelajari struktur *graph* dan hubungan antar transaksi. Selanjutnya, data diproses pada GAT Layer 1 yang menggunakan 4 attention head dengan hidden channel sebesar 64. Karena hasil dari setiap attention head digabungkan menggunakan operasi concatenation, keluaran layer pertama memiliki dimensi 256 (64 × 4). Representasi tersebut kemudian diproses menggunakan fungsi aktivasi ELU dan dropout sebesar 0,2. Pada GAT Layer 2 digunakan 1 attention head dengan concat=False untuk menghasilkan 2 nilai keluaran yang merepresentasikan kelas Licit dan Illicit.Representasi tersebut kemudian diproses

menggunakan fungsi aktivasi *Exponential Linear Unit (ELU).* Fungsi aktivasi ini digunakan untuk memberikan transformasi *nonlinier* sehingga model mampu mempelajari pola hubungan yang lebih kompleks antar*node* serta membantu meningkatkan stabilitas proses pelatihan. Tahap berikutnya adalah GAT Layer 2 yang menggunakan 1 *attention head*. Pada lapisan ini, representasi fitur berdimensi 512 diproyeksikan menjadi 2 *neuron* keluaran yang merepresentasikan dua kelas, yaitu transaksi legal (licit) dan transaksi ilegal (illicit).

Selanjutnya, fungsi *Softmax* digunakan untuk menghitung probabilitas masing-masing kelas sehingga setiap transaksi dapat diklasifikasikan berdasarkan probabilitas tertinggi. Hasil akhir dari proses ini adalah prediksi transaksi ke dalam kelas legal atau illicit. Konfigurasi model terdiri atas 165 input feature, hidden channel sebesar 64, 4 attention head pada layer pertama, 1 attention head pada layer kedua, dropout sebesar 0,2, dan 2 output class. Model dilatih selama 200 epoch menggunakan optimizer Adam dengan learning rate 0,01 dan random seed 42 Fungsi aktivasi yang digunakan adalah ELU, sedangkan fungsi kerugian menggunakan *CrossEntropyLoss* yang dipadukan dengan *Class Weight* untuk mengatasi ketidakseimbangan jumlah data antarkelas. Pada implementasi ini dropout tidak digunakan, nilai *random seed* ditetapkan sebesar 42 untuk menjaga konsistensi hasil pelatihan, dan proses komputasi dijalankan pada perangkat CPU maupun GPU.

Berdasarkan konfigurasi tersebut, model GAT dengan *Class Weight* diharapkan mampu memanfaatkan informasi struktur *graph* dan mekanisme *attention* secara optimal sehingga meningkatkan kemampuan dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*, terutama pada kondisi distribusi data yang tidak seimbang.Sebuah *graph* memiliki himpunan *node* $h = \{\vec{h}1, \vec{h}2, ..., \vec{h}N\}$, dengan $\vec{h}i \in \mathbb{R}F$ merupakan vektor fitur *node* ke-i dan F adalah jumlah fitur input (dalam penelitian ini F = 165).

1. Transformasi Linear Fitur *Node*

Setiap fitur *node* ditransformasikan menggunakan matriks bobot yang dipelajari (learnable weight matrix) W:

$$z_i = W\vec{h}_i \quad (3.1)$$

Setiap fitur *node* $\vec{h}_i$ ditransformasikan ke ruang fitur baru menggunakan matriks bobot $W \in \mathbb{R}^{F' \times F}$ yang dipelajari selama proses pelatihan, menghasilkan representasi zi.

2. Perhitungan *Attention coefficient* (mentah)

$$e_{ij} = \text{LeakyReLU}(\vec{a}^T[z_i \,\|\, z_j]) \quad (3.2)$$

Nilai eij menunjukkan tingkat kepentingan (importance) *node* j terhadap *node* i, dihitung dengan menggabungkan (concatenate, dilambangkan ‖) hasil transformasi zi dan zj, kemudian dikalikan dengan vektor bobot attention $\vec{a}$ yang dipelajari, dan dilewatkan pada fungsi aktivasi LeakyReLU.

3. Normalisasi *Attention coefficient* (*Softmax*)

$$\alpha_{ij} = \exp(e_{ij}) / \Sigma_{k \in \mathcal{N}i} \exp(e_{ik}) \quad (3.3)$$

Nilai eij dinormalisasi menggunakan fungsi *Softmax* terhadap seluruh *node* tetangga $\mathcal{N}$i dari *node* i, sehingga diperoleh koefisien attention αij yang menunjukkan bobot relatif *node* j terhadap *node* i, dengan total bobot seluruh tetangga bernilai 1.

4. Agregasi Fitur (Multi-Head Attention, Layer 1)

Karena layer pertama menggunakan 4 *attention head* yang digabungkan (concatenate):

$$h_{i'} = \|_{k=1}^{K} \sigma(\Sigma_{j \in \mathcal{N}i} \alpha_{ijk} W^k h_j) \quad (3.4)$$

Pada layer pertama, digunakan mekanisme multi-head attention dengan K = 4 head. Hasil dari tiap head digabungkan (concatenate, ‖) untuk menghasilkan representasi *node* yang lebih kaya, dengan σ merupakan fungsi aktivasi ELU.

5. Agregasi Fitur (Layer 2, Output)

Karena layer kedua hanya menggunakan 1 *attention head* untuk klasifikasi akhir:

$h_{i''} = \sigma(\Sigma_{j \in \mathcal{N}i} \alpha_{ij} W h_{j'})$……(3.5)

Pada layer kedua, digunakan satu *attention head* untuk menghasilkan representasi akhir *node* yang kemudian digunakan pada proses klasifikasi.

6. Klasifikasi (*Softmax*)

$\hat{y}_i = Softmax(h_{i''})$……(3.6)

Representasi akhir tiap *node* diubah menjadi probabilitas kelas menggunakan fungsi *Softmax*, menghasilkan prediksi ŷi berupa probabilitas transaksi termasuk kelas licit atau illicit.

7. *Weighted* Cross Entropy Loss

$\mathcal{L} = -\Sigma_{i=1N} w_{yi} y_i \log(\hat{y}_i)$……(3.7)

Fungsi loss yang digunakan adalah *Weighted* Cross Entropy Loss, di mana setiap kelas diberi bobot berdasarkan skema balanced (lihat Persamaan 3.8).

8. Perhitungan *Class Weight* (Skema Balanced)

$w_c = n / (k \times n_c)$……(3.8)

dengan n adalah jumlah total data latih, k adalah jumlah kelas, dan nc adalah jumlah data pada kelas c. Pemberian bobot ini bertujuan agar kesalahan klasifikasi pada kelas illicit, yang jumlahnya jauh lebih sedikit dibanding kelas licit, diberi penalti yang lebih besar selama proses pelatihan[21].

Untuk proses *training*-nya, model ini pakai optimizer Adam dengan learning rate 0.01.Pada varian GAT + *Class Weight*, fungsi *CrossEntropyLoss* diberikan bobot kelas berdasarkan distribusi data training agar kesalahan klasifikasi pada kelas Illicit memperoleh penalti lebih besar. Melalui mekanisme attention, model dapat mempelajari kontribusi yang berbeda dari setiap node tetangga dalam proses pembentukan representasi transaksi.

3.6 Model Klasifikasi

Penelitian ini membandingkan lima model klasifikasi untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*, yaitu *Support Vector Machine (SVM)* dan *Graph Convolutional Network (GCN)* beserta varian *Class Weight*-nya sebagai model pembanding (*baseline*), serta *Graph Attention Network (GAT)* beserta varian *Class Weight*-nya sebagai model yang diusulkan (proposed model) dalam penelitian ini.

3.6.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) digunakan sebagai model pembanding paling dasar dalam penelitian ini. SVM bekerja dengan mencari *hyperplane* pemisah terbaik antarkelas pada ruang fitur, menggunakan *kernel Radial Basis Function (RBF)* untuk menangani data yang tidak terpisah secara linear. Berbeda dari model berbasis *graph*, SVM hanya memanfaatkan fitur numerik dari masing-masing transaksi secara independen, tanpa mempertimbangkan hubungan antartransaksi dalam jaringan. Model ini dipakai untuk melihat seberapa besar peningkatan yang bisa diperoleh ketika struktur *graph* turut dimanfaatkan oleh GCN maupun GAT.

3.6.2 *Graph Convolutional Network* (GCN) dan GCN dengan *Class Weight*

Graph Convolutional Network (GCN) digunakan sebagai model pembanding kedua yang memanfaatkan struktur *graph* transaksi. GCN mempelajari representasi tiap *node* dengan meratakan fitur dirinya sendiri beserta fitur seluruh *node* tetangga langsungnya secara ternormalisasi, sehingga model bisa mengenali pola hubungan antartransaksi yang tidak dapat ditangkap oleh SVM. Karena distribusi kelas pada dataset Elliptic sangat tidak seimbang, GCN juga dilatih dalam dua varian, yaitu GCN tanpa *Class Weight* dan GCN dengan *Class Weight*, untuk melihat sejauh mana penanganan ketidakseimbangan kelas memengaruhi kemampuan model dalam mendeteksi transaksi illicit.

3.6.3 *Graph Attention Network (GAT)* dan GAT dengan *Class Weight* sebagai Model yang Diusulkan

Graph Attention Network (GAT) merupakan model utama yang diusulkan dalam penelitian ini. Berbeda dari GCN yang memberi bobot tetangga berdasarkan normalisasi derajat *node* sehingga tiap tetangga diperlakukan relatif setara, GAT menerapkan mekanisme attention yang memungkinkan model mempelajari bobot kepentingan berbeda untuk setiap tetangga secara adaptif selama pelatihan. Dengan begitu, GAT berpotensi lebih menonjolkan transaksi tetangga yang lebih relevan atau mencurigakan dibanding transaksi tetangga yang kurang relevan. Sama seperti GCN, GAT juga dilatih dalam dua varian, yaitu GAT tanpa *Class Weight* dan GAT dengan *Class Weight*, dengan GAT + *Class Weight* menjadi konfigurasi akhir yang diusulkan penelitian ini. Detail arsitektur dan konfigurasi lengkap model GAT dijelaskan lebih lanjut pada subbab 3.5.1.

3.6.4 Perbandingan Model

Kelima model, yaitu SVM, GCN, GCN dengan *Class Weight*, GAT, dan GAT dengan *Class Weight*, dilatih dan diuji menggunakan pembagian data latih dan data uji yang identik agar perbandingan performa antar model tetap adil. Performa kelima model kemudian dibandingkan berdasarkan Performa kelima model kemudian dibandingkan berdasarkan metrik evaluasi yang dijelaskan pada Subbab 3.8, guna mengetahui apakah pendekatan berbasis *graph*, khususnya GAT dengan mekanisme attention dan *Class Weight*, mampu memberikan performa deteksi transaksi ilegal yang lebih baik dibandingkan GCN maupun metode *Machine learning* konvensional seperti SVM.

3.7 Pelatihan dan Pengujian Model

Pada penelitian ini, proses pelatihan dan pengujian model dilakukan secara terstruktur untuk memastikan hasil yang diperoleh dapat dibandingkan secara objektif antara model yang diusulkan (GAT dan GAT dengan *Class Weight*) dan model pembanding (SVM, GCN, dan GCN dengan *Class Weight*). Tahap awal dimulai dengan menyiapkan data yang telah melalui proses praproses, yaitu data fitur transaksi, struktur *graph*, serta label kelas (legal dan ilegal). Data kemudian

dibagi menjadi dua bagian, yaitu data pelatihan (*training* set) dan data pengujian *(testing set)*.

Pada model berbasis *graph*, yaitu GCN, GCN dengan *Class Weight*, GAT, dan GAT dengan *Class Weight*, proses pelatihan dilakukan dengan menggunakan dua komponen utama, yaitu fitur *node* dan struktur *graph*. Setiap model mempelajari representasi tiap *node* dengan memanfaatkan informasi dari *node* tetangganya melalui proses agregasi, dengan GAT menerapkan mekanisme *attention* agar bobot tiap tetangga dapat dipelajari secara adaptif. Proses pelatihan dilakukan secara *iteratif* hingga model mampu mengenali pola transaksi ilegal dalam data. Sementara itu, pada model *Support Vector Machine (SVM)* sebagai *baseline*, proses pelatihan dilakukan hanya menggunakan fitur numerik dari setiap transaksi tanpa melibatkan struktur *graph*. Model dilatih untuk membedakan transaksi legal dan ilegal berdasarkan pola fitur yang ada.

Proses pelatihan model berbasis *graph* pada penelitian ini diimplementasikan menggunakan pustaka *PyTorch Geometric*, yaitu pustaka yang dirancang khusus untuk pembelajaran representasi pada data berstruktur *graph* dengan memanfaatkan akselerasi GPU dan penanganan mini-batch yang efisien [26]. Pembaruan bobot model dilakukan menggunakan algoritma optimisasi Adam (*Adaptive Moment Estimation*), yang menghitung laju pembelajaran adaptif untuk setiap parameter berdasarkan estimasi momen pertama dan kedua dari *gradien*, sehingga proses konvergensi model cenderung lebih cepat dan lebih stabil dibandingkan algoritma *gradient descent konvensional* [27].

Setelah proses pelatihan selesai, kelima model diuji menggunakan data pengujian yang belum pernah digunakan sebelumnya. Pengujian ini bertujuan untuk mengukur kemampuan model dalam melakukan generalisasi terhadap data baru.

Seluruh model menghasilkan output berupa prediksi kelas transaksi, yaitu legal atau ilegal. Hasil prediksi tersebut kemudian digunakan dalam tahap evaluasi untuk mengukur kinerja masing-masing model. Dengan menggunakan data pelatihan dan pengujian yang sama pada kelima model, perbandingan performa

antara GAT, GAT dengan *Class Weight*, GCN, GCN dengan *Class Weight*, dan SVM dapat dilakukan secara adil. Hasil dari proses ini akan digunakan untuk mengetahui apakah pendekatan berbasis *graph*, khususnya GAT dengan mekanisme attention dan *Class Weight*, mampu meningkatkan kemampuan deteksi transaksi ilegal dibandingkan GCN maupun metode konvensional seperti SVM.

Pada penelitian ini, Class Weight diterapkan pada proses pelatihan model GCN + Class Weight dan GAT + Class Weight untuk menangani ketidakseimbangan jumlah data antara kelas Licit dan Illicit. Nilai Class Weight dihitung berdasarkan distribusi kelas pada data training. Data testing tidak digunakan dalam perhitungan bobot karena Class Weight merupakan bagian dari proses pelatihan model.

Berdasarkan hasil pembagian data, jumlah data training sebanyak 37.251 transaksi yang terdiri atas 33.615 transaksi Licit dan 3.636 transaksi Illicit. Bobot masing-masing kelas dihitung menggunakan persamaan berikut:

$$w_c = \frac{N}{K \times N_c} \quad \text{.....3.9}$$

$$w_{\text{Licit}} = \frac{37251}{2 \times 33615} = 0.5541 \quad \text{.....3.10}$$

$$w_{\text{Illicit}} = \frac{37251}{2 \times 3636} = 5.1226 \quad \text{.....3.11}$$

wc = bobot untuk kelas ke-c
N = jumlah seluruh data training
K = jumlah kelas (Licit dan Illicit), yaitu 2
Nc = jumlah data training pada kelas ke-c

Tabel 3.14 Hasil Perhitungan Class Weight pada Data Training

Kelas	Jumlah Data Training	Class Weight
Licit	33615	0,5541
Illicit	3636	5,1225
Total	37251	-

Sumber: Diolah oleh Penulis (2026)

Berdasarkan hasil perhitungan tersebut, kelas Illicit memperoleh bobot sebesar 5,1226, sedangkan kelas Licit memperoleh bobot sebesar 0,5541. Bobot kelas Illicit lebih besar karena jumlah transaksi Illicit pada data training lebih sedikit dibandingkan transaksi Licit. Nilai bobot tersebut selanjutnya digunakan pada fungsi CrossEntropyLoss dalam proses pelatihan model GCN + Class Weight dan GAT + Class Weight. Dengan pemberian bobot yang lebih besar pada kelas Illicit, kesalahan prediksi terhadap kelas tersebut memberikan penalti *loss* yang lebih besar sehingga model diharapkan lebih mampu mengenali transaksi Illicit sebagai kelas minoritas.

Secara umum, penerapan *Class Weight* pada fungsi *Cross-Entropy Loss* dapat dinyatakan sebagai:

$$L = -\frac{1}{N}\sum_{i=1}^{N} w_{y_i} \log\left(p_{i,y_i}\right) \quad \text{.....3.12}$$

di mana w_{y_i} merupakan bobot kelas dari label sebenarnya pada sampel ke- i, sedangkan p_{i,y_i} merupakan probabilitas yang diprediksi model terhadap kelas sebenarnya. Dengan demikian, sampel dari kelas *Illicit* memiliki pengaruh yang lebih besar terhadap pembaruan parameter model ketika terjadi kesalahan prediksi.

Konsep penggunaan weighted loss untuk membuat kesalahan pada kelas minoritas lebih mahal juga digunakan dalam penelitian *fraud detection* dan GNN yang menghadapi *class imbalance*. Misalnya, penelitian GAT-COBO menggunakan pendekatan *cost-sensitive* agar model lebih fokus pada kelas minoritas dalam graph yang tidak seimbang

3.8 Metrik Evaluasi

Untuk mengukur kinerja model dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*, penelitian ini menggunakan beberapa metrik evaluasi klasifikasi. Pemilihan metrik ini disesuaikan dengan karakteristik data yang tidak seimbang (*imbalance*d), di mana jumlah transaksi ilegal lebih sedikit dibandingkan transaksi legal.

Metrik evaluasi yang digunakan dalam penelitian ini meliputi:

1. **Akurasi (*Accuracy*)**

   Akurasi digunakan untuk mengukur persentase keseluruhan prediksi yang benar, baik untuk transaksi legal maupun ilegal. Meskipun akurasi memberikan gambaran umum terhadap kinerja model, metrik ini kurang representatif pada kondisi data yang tidak seimbang.
2. ***Precision***

   Precision mengukur tingkat ketepatan model dalam memprediksi transaksi ilegal. Nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar transaksi yang diprediksi sebagai ilegal benar-benar merupakan transaksi ilegal.
3. ***Recall***

   Recall mengukur kemampuan model dalam mendeteksi seluruh transaksi ilegal yang ada dalam data. Metrik ini sangat penting dalam konteks *digital forensik*, karena kegagalan mendeteksi transaksi ilegal dapat berdampak pada terlewatnya aktivitas kriminal.
4. ***F1-Score***

   F1-score merupakan kombinasi antara *Precision* dan *recall* yang memberikan gambaran keseimbangan antara keduanya. Metrik ini digunakan untuk menilai performa model secara lebih menyeluruh, terutama pada data yang tidak seimbang.

Dalam penelitian ini, perhatian utama diberikan pada nilai *recall* dan *F1-score* untuk kelas transaksi ilegal, karena tujuan utama penelitian adalah mendeteksi sebanyak mungkin transaksi ilegal secara akurat.

Selain keempat metrik tersebut, penelitian ini juga menggunakan kurva *Receiver Operating Characteristic (ROC)* beserta nilai *Area Under Curve (AUC)* sebagai metrik pelengkap untuk melihat kemampuan model dalam membedakan transaksi legal dan ilegal pada berbagai nilai ambang (threshold).

Selain menghitung metrik evaluasi tersebut secara umum, penelitian ini juga menyajikan hasil evaluasi menggunakan pendekatan *Macro Average* dan *Micro*

Average. *Macro Average* menghitung nilai *Precision*, *recall*, dan *F1-score* pada setiap kelas secara terpisah kemudian merata-ratakannya tanpa mempertimbangkan jumlah data pada masing-masing kelas, sehingga lebih mencerminkan kemampuan model dalam menangani kelas minoritas secara adil. Sebaliknya, *Micro Average* menghitung seluruh *True Positive*, *False Positive*, dan *False Negative* dari semua kelas secara bersama-sama sebelum menghasilkan nilai *Precision*, *recall*, dan *F1-score*, sehingga hasilnya lebih dipengaruhi oleh kelas mayoritas[54]. Kedua pendekatan ini digunakan untuk memberikan gambaran evaluasi yang lebih menyeluruh pada dataset yang memiliki distribusi kelas tidak seimbang.

Hasil evaluasi dari model *Graph Attention Network (GAT)* kemudian dibandingkan dengan model *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, dan GCN dengan *Class Weight (GCN + Class Weight)* sebagai *baseline* untuk melihat peningkatan performa yang dihasilkan.

Pada penelitian ini, Recall menjadi salah satu metrik yang lebih diprioritaskan karena tujuan utama penelitian adalah mendeteksi sebanyak mungkin transaksi Illicit yang terdapat pada dataset. Recall menunjukkan kemampuan model dalam mengenali transaksi Illicit yang sebenarnya dari seluruh transaksi Illicit yang ada. Semakin tinggi nilai Recall, maka semakin sedikit transaksi Illicit yang salah diprediksi sebagai transaksi Licit atau False Negative (FN).

Dalam konteks digital forensik, False Negative menjadi perhatian penting karena kondisi tersebut menunjukkan adanya transaksi Illicit yang tidak berhasil terdeteksi oleh model. Transaksi yang tidak terdeteksi tersebut berpotensi tidak masuk ke tahap analisis atau pemeriksaan lebih lanjut. Sebaliknya, False Positive menunjukkan transaksi Licit yang diprediksi sebagai Illicit. Meskipun kondisi ini dapat menambah jumlah transaksi yang perlu diperiksa, transaksi tersebut masih dapat diverifikasi kembali pada tahap analisis oleh investigator. Oleh karena itu, penelitian ini lebih memprioritaskan peningkatan Recall dan penurunan False Negative, tanpa mengabaikan Precision, F1-score, Accuracy, dan ROC-AUC sebagai metrik evaluasi lainnya.

3.9 Visualisasi Representasi Model menggunakan t-SNE

Selain diukur secara kuantitatif melalui metrik evaluasi pada sub-bab 3.6, representasi (*embedding*) yang dipelajari oleh model GAT dan GCN juga divisualisasikan secara visual untuk mengamati apakah model benar-benar mampu memisahkan transaksi legal dan ilegal dalam ruang representasinya. Persoalannya, *embedding* yang dihasilkan model-model tersebut umumnya berbentuk vektor berdimensi tinggi, sehingga tidak dapat langsung digambarkan pada bidang dua dimensi.

Untuk mengatasi hal tersebut, penelitian ini menggunakan *t-distributed Stochastic Neighbor Embedding (t-SNE),* yaitu teknik reduksi dimensi non-linear yang bekerja dengan memodelkan kemiripan antar titik data pada ruang berdimensi tinggi sebagai distribusi probabilitas, kemudian mencari representasi pada ruang berdimensi rendah yang distribusi probabilitasnya semirip mungkin dengan distribusi aslinya [24]. Dengan cara ini, titik-titik data yang saling berdekatan pada ruang berdimensi tinggi cenderung digambarkan berdekatan pula pada ruang berdimensi rendah, sehingga pola pengelompokan (*clustering*) transaksi legal dan ilegal dapat diamati secara visual.

Visualisasi t-SNE pada penelitian ini digunakan sebagai pelengkap terhadap metrik evaluasi kuantitatif, guna memberikan pemahaman yang lebih menyeluruh mengenai kemampuan model GAT dan GCN dalam memisahkan pola transaksi ilegal dari transaksi legal pada jaringan *Bitcoin*.

3.10 Analisis Risiko *Neighbor* (*Neighbor* Risk *Analysis*)

Selain mengevaluasi performa model secara langsung, penelitian ini juga melakukan analisis risiko terhadap transaksi yang secara struktural terhubung (*neighbor*) dengan transaksi yang telah diketahui berlabel ilegal (illicit). Analisis ini bertujuan untuk melihat sejauh mana kedekatan suatu transaksi dengan transaksi ilegal dapat mengindikasikan tingkat risiko, sebagai informasi pelengkap di luar hasil klasifikasi model. Proses analisis risiko *neighbor* dilakukan melalui beberapa tahap. Pertama, seluruh transaksi yang berlabel illicit pada dataset ditetapkan

sebagai *fraud nodes*. Kedua, seluruh *edge* yang menghubungkan *fraud node*s dengan transaksi lain diidentifikasi sebagai *fraud edges*. Ketiga, *node* yang terhubung langsung dengan *fraud edge*s — tidak termasuk *fraud node*s itu sendiri — ditetapkan sebagai kelompok *neighbor*, yaitu transaksi legal maupun *Unknown* yang memiliki keterkaitan langsung dengan transaksi ilegal.

Selanjutnya, setiap *node* pada kelompok *neighbor* dihitung skor risikonya menggunakan persamaan berikut:

$$\text{risk_score} = \frac{\text{(jumlah koneksi } \textit{node} \text{ ke } \textit{fraud node}\text{s)}}{\text{(jumlah total koneksi } \textit{node}\text{)}} \quad \text{.....(3.13)}$$

Nilai *risk_score* menunjukkan proporsi koneksi suatu *neighbor* yang mengarah ke transaksi ilegal dibandingkan seluruh koneksi yang dimilikinya. Berdasarkan nilai tersebut, setiap *neighbor* dikelompokkan ke dalam empat kategori risiko, yaitu *Safe* apabila *risk_score* bernilai 0, *Low* apabila $0 < \text{risk_score} \leq 0{,}25$, *Medium* apabila $0{,}25 < \text{risk_score} \leq 0{,}50$, dan *High* apabila $\text{risk_score} > 0{,}50$. Kategori ini digunakan untuk mengetahui proporsi transaksi *neighbor* yang berpotensi memiliki keterkaitan tinggi dengan aktivitas ilegal, sehingga dapat memberikan konteks tambahan bagi proses investigasi forensik digital di luar hasil klasifikasi model.

3.11 Kerangka Investigasi *Digital forensik*

Sebagai bagian akhir dari metodologi penelitian ini, hasil deteksi model GAT diposisikan dalam kerangka investigasi *digital forensik*, mengingat tujuan akhir penelitian ini adalah mendukung kebutuhan *digital forensik* dalam mengidentifikasi transaksi ilegal pada jaringan *Bitcoin*. Kerangka kerja forensik digital yang diadopsi mengacu pada *Guide to Integrating Forensic Techniques into Incident Response* yang diterbitkan oleh *National Institute of Standards and Technology (NIST),* yang membagi proses forensik digital ke dalam empat tahap utama, yaitu *collection*, *examination*, *analysis*, dan reporting [2].

Pada tahap *collection*, data transaksi *Bitcoin* dari Elliptic *Bitcoin* Dataset dikumpulkan sebagai representasi bukti digital yang akan dianalisis, sebagaimana

dijelaskan pada sub-bab 3.3. Tahap *examination* dilakukan melalui proses praproses data dan pembentukan struktur *graph*, sehingga data siap diperiksa lebih lanjut secara teknis. Tahap *analysis* dilakukan melalui proses pelatihan, pengujian, dan evaluasi model GAT, dengan SVM dan GCN sebagai pembanding (*baseline*), sebagaimana diuraikan pada sub-bab 3.5 dan 3.6, yang menghasilkan klasifikasi transaksi ke dalam kategori legal maupun ilegal beserta metrik kinerjanya.



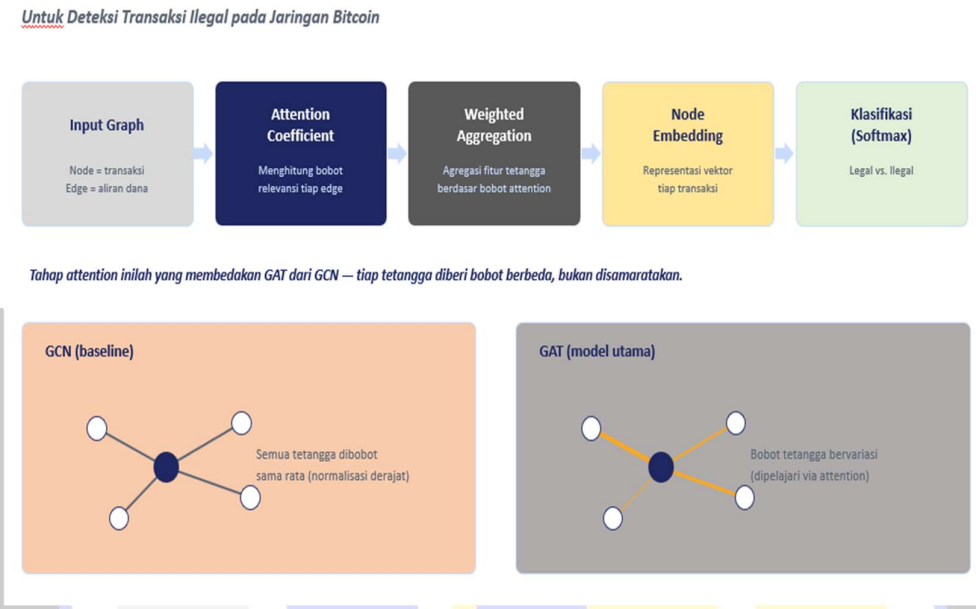
Sumber: diadaptasi dari NIST Guide to Integrating Forensic Techniques into Incident Response; diolah oleh penulis (2026)

Gambar 3.11 Kerangka Investigasi *Digital forensik*

Tahap reporting merupakan tahap penyajian hasil analisis, yaitu daftar transaksi yang terdeteksi sebagai ilegal beserta tingkat kepercayaan model, yang selanjutnya dapat digunakan oleh investigator sebagai dasar untuk penelusuran lebih lanjut, misalnya identifikasi alamat dompet terkait maupun pola aliran dana yang mencurigakan.

Dengan memetakan tahapan teknis penelitian ke dalam kerangka kerja forensik digital yang baku, hasil penelitian ini diharapkan tidak hanya bersifat teknis, tetapi juga dengan memetakan tahapan teknis penelitian ke dalam kerangka kerja forensik digital, hasil penelitian ini diharapkan dapat memberikan gambaran mengenai bagaimana output model dapat digunakan sebagai informasi pendukung dalam proses investigasi digital forensik, khususnya untuk membantu penyaringan dan penentuan prioritas transaksi yang perlu diperiksa lebih lanjut, khususnya dalam penelusuran transaksi ilegal pada jaringan *Bitcoin*.

UNIVERSITAS NUSA MANDIRI



Sumber: Diolah oleh penulis berdasarkan Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). *Graph Attention Networks: A Comprehensive Review of Methods and Applications*.

Gambar 3.12 Alur *Graph Attention Network (GAT)* untuk Deteksi Transaksi Ilegal pada Jaringan *Bitcoin*

Diagram berikut mengilustrasikan alur kerja *Graph Attention Network (GAT)* dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Transaksi *Bitcoin* direpresentasikan sebagai graf, dengan setiap transaksi menjadi *node* dan aliran dana antar-transaksi menjadi *edge*. GAT kemudian menghitung koefisien attention untuk setiap pasangan *node* yang terhubung, lalu melakukan agregasi fitur dari *node-node* tetangga secara berbobot (*weighted*) berdasarkan koefisien tersebut, untuk menghasilkan representasi (*embedding*) tiap *node* yang selanjutnya diklasifikasikan sebagai transaksi legal atau ilegal untuk keperluan investigasi *digital forensic.*

BAB IV
HASIL DAN PEMBAHASAN

4.1 Gambaran Umum Implementasi Penelitian

Bab ini menyajikan hasil implementasi dan pembahasan terhadap lima konfigurasi model, yaitu SVM, GCN, GCN + Class Weight, GAT, dan GAT + Class Weight, dalam mendeteksi transaksi ilegal pada jaringan Bitcoin menggunakan Elliptic Bitcoin Dataset. Tahapan yang dilakukan pada bab ini mengikuti metodologi penelitian yang telah dijelaskan pada Bab III, dimulai dari eksplorasi dataset, prapemrosesan data, pembentukan *graph*, pelatihan model, hingga evaluasi hasil klasifikasi.

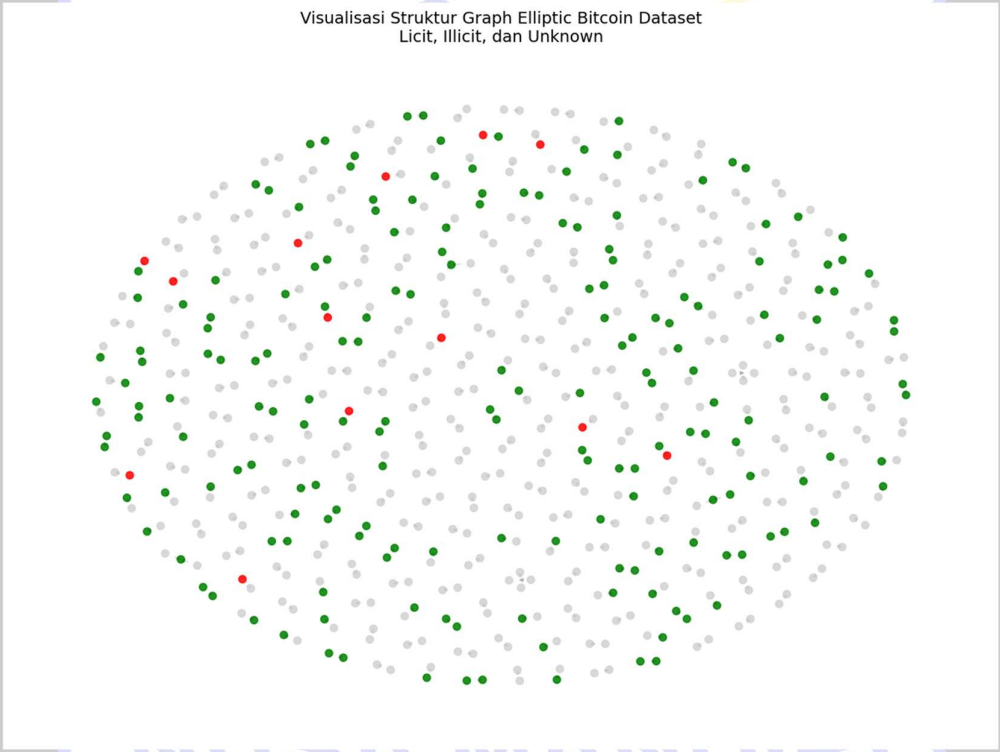
Pada penelitian ini, setiap transaksi *Bitcoin* direpresentasikan sebagai *node*, sedangkan hubungan antartransaksidirepresentasikan sebagai *edge* sehingga membentuk sebuah *graph*. Representasi tersebut memungkinkan model berbasis *graph*, khususnya *Graph Attention Network (GAT),* memanfaatkan tidak hanya informasi dari masing-masing transaksi, tetapi juga hubungan antartransaksi yang terdapat pada jaringan *Bitcoin*. Pendekatan ini diharapkan mampu mengenali pola transaksi ilegal dengan lebih baik dibandingkan metode klasifikasi yang hanya menggunakan fitur transaksi secara individual. Selain menyajikan hasil implementasi model, bab ini juga membahas karakteristik dataset yang digunakan, proses pembentukan *graph*, hasil pelatihan model, serta evaluasi performa menggunakan beberapa metrik klasifikasi. Selanjutnya dilakukan analisis terhadap transaksi *fraud* yang berhasil maupun yang belum berhasil dideteksi oleh model, sehingga dapat diketahui bagaimana struktur *graph* berkontribusi dalam proses deteksi transaksi ilegal.

Sebagai bagian dari pembahasan, hasil penelitian juga dianalisis dalam konteks *digital forensik* untuk melihat bagaimana model *Graph Attention Network (GAT)* dapat membantu proses identifikasi transaksi yang mencurigakan serta

memberikan informasi pendukung bagi proses investigasi pada jaringan *Blockchain*.

4.1.1 Visualisasi Struktur *Graph*

Untuk memperoleh gambaran mengenai karakteristik jaringan transaksi *Bitcoin*, dilakukan visualisasi struktur *graph* berdasarkan hubungan antartransaksi pada Elliptic *Bitcoin* Dataset. Visualisasi ini bertujuan untuk memperlihatkan pola keterhubungan antar*node* sehingga karakteristik jaringan dapat diamati sebelum dilakukan proses pelatihan model berbasis graph, yaitu GCN dan GAT



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.1 Visualisasi Struktur *Graph* pada Elliptic *Bitcoin* Dataset

Berdasarkan Gambar 4.1, struktur graph pada Elliptic Bitcoin Dataset divisualisasikan dengan tetap mempertahankan tiga kategori transaksi, yaitu Licit, Illicit, dan Unknown. Setiap node merepresentasikan satu transaksi Bitcoin, sedangkan edge menunjukkan adanya hubungan antartransaksi. Dataset secara keseluruhan terdiri atas 203.769 node dan 234.355 edge. Namun, karena jumlah

node dan edge cukup besar, visualisasi dilakukan menggunakan sampel sebanyak 300 edge agar pola hubungan antartransaksi dapat terlihat dengan lebih jelas.

Hasil visualisasi sampel menghasilkan 595 node yang terdiri atas 399 transaksi Unknown, 183 transaksi Licit, dan 13 transaksi Illicit. Jumlah node Unknown terlihat lebih dominan dibandingkan dua kategori lainnya. Hal ini sesuai dengan distribusi dataset, karena sebagian besar transaksi pada Elliptic Bitcoin Dataset memang memiliki label Unknown. Pada tahap pelatihan dan evaluasi model dalam penelitian ini, transaksi Unknown tidak digunakan sebagai target dalam proses pelatihan model karena tidak memiliki label pasti sebagai Licit atau Illicit. Namun, pada tahap analisis struktur graph, transaksi Unknown tetap penting untuk dipertahankan karena transaksi tersebut masih memiliki hubungan dengan node lain di dalam jaringan. Dengan demikian, transaksi Unknown tidak hanya dipandang sebagai data tanpa label, tetapi juga sebagai bagian dari struktur jaringan yang dapat memberikan informasi tambahan mengenai pola keterhubungan transaksi Bitcoin

Tabel 4.1 Ringkasan Hasil Visualisasi Struktur *Graph*

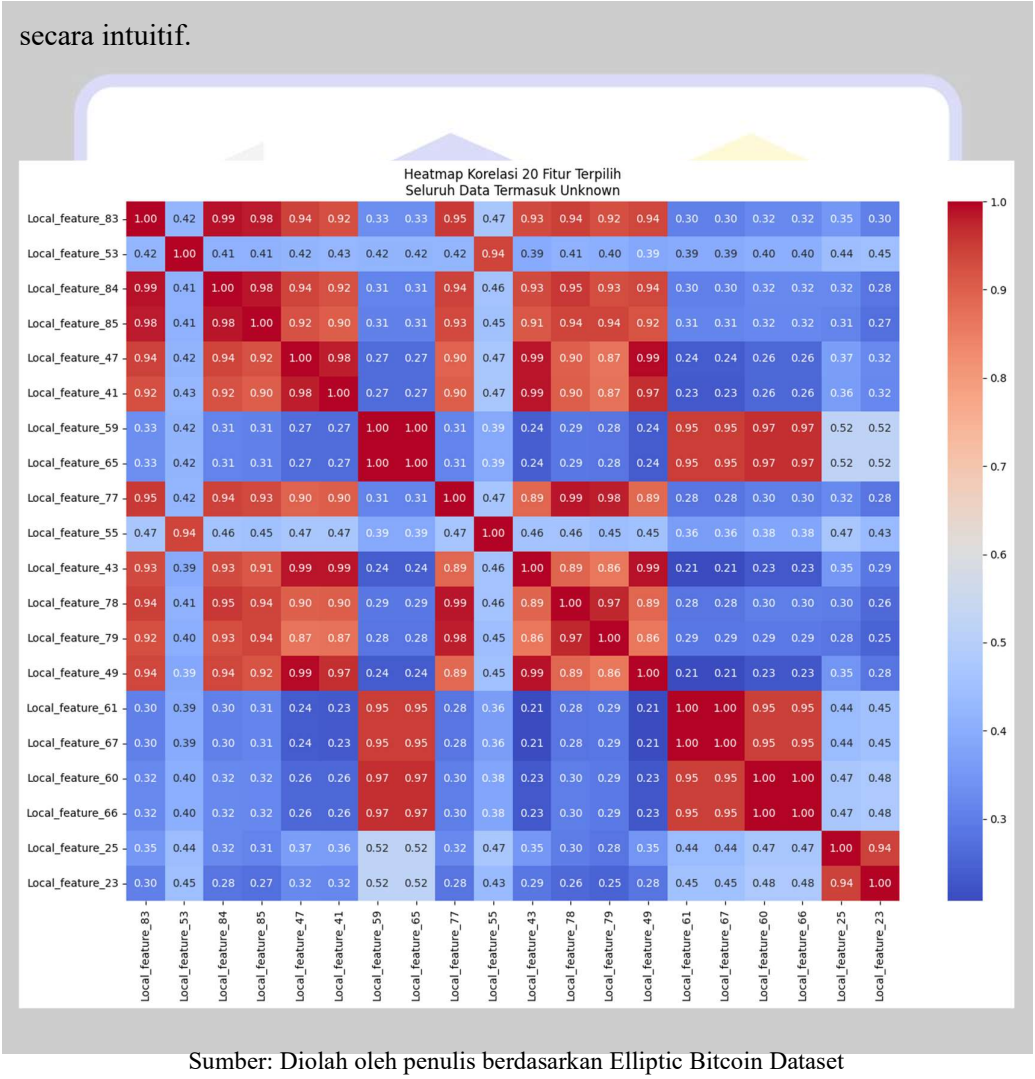
Parameter	Nilai
Jumlah *Node* yang divisualisasikan	595
Jumlah *Edge* yang divisualisasikan	300
Metode Visualisasi	*NetworkX*
Bentuk Data	*Graph*
Jenis Visualisasi	Sampel Sub*graph*

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Berdasarkan Tabel 4.1, visualisasi *graph* menggunakan 595 *node* dan 300 *edge* sebagai representasi dari struktur *graph* pada Elliptic *Bitcoin* Dataset. Jumlah tersebut merupakan sampel (sub*graph*) dari keseluruhan dataset yang terdiri atas 203.769 *node* dan 234.355 *edge*. Penggunaan sampel dilakukan agar pola hubungan antartransaksi dapat diamati dengan jelas, namun tetap mampu memberikan gambaran mengenai karakteristik struktur graph yang digunakan dalam implementasi model berbasis graph, yaitu GCN dan GAT.

4.1.2 Analisis Korelasi Fitur

Analisis korelasi fitur dilakukan untuk mengetahui tingkat hubungan antar fitur numerik yang terdapat pada Elliptic *Bitcoin* Dataset. Mengingat dataset memiliki 165 fitur, analisis difokuskan pada 20 fitur yang memiliki nilai korelasi rata-rata tertinggi agar pola hubungan antar fitur dapat diamati dengan lebih jelas. Visualisasi dilakukan menggunakan heatmap sehingga tingkat korelasi antar fitur dapat terlihat secara intuitif.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.2 Heatmap Korelasi Fitur

Berdasarkan Gambar 4.2, analisis korelasi dilakukan terhadap 165 fitur yang terdapat pada seluruh 203.769 transaksi di dalam Elliptic Bitcoin Dataset. Pada tahap ini, transaksi Unknown tetap disertakan karena analisis korelasi tidak

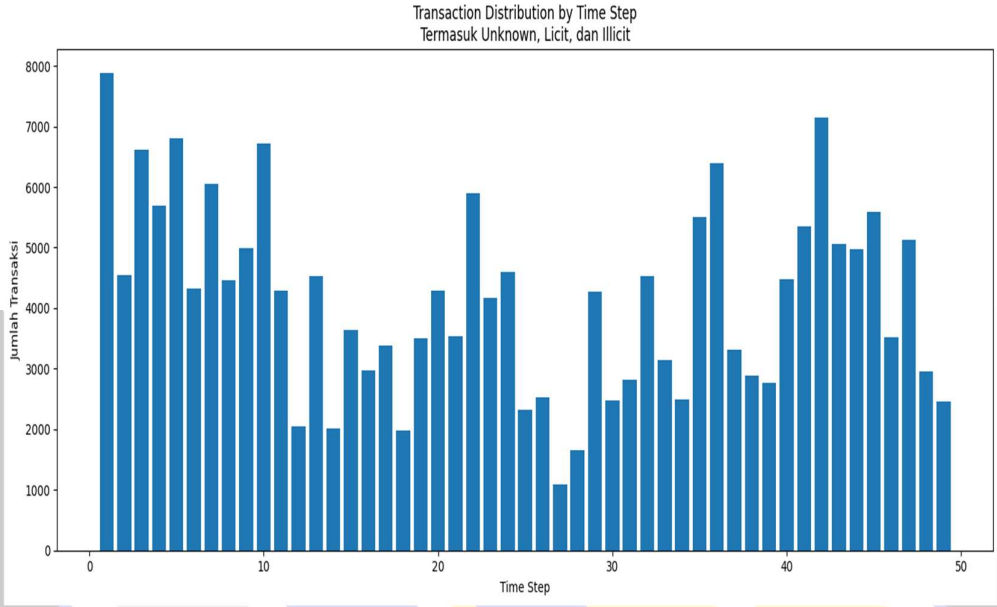
membutuhkan label kelas sebagai target, melainkan hanya menggunakan nilai numerik dari masing-masing fitur transaksi.

Dari keseluruhan 165 fitur, dipilih 20 fitur yang memiliki rata-rata korelasi absolut tertinggi terhadap fitur lainnya. Beberapa fitur yang berada pada urutan tertinggi antara lain Local_feature_83, Local_feature_53, Local_feature_84, Local_feature_85, Local_feature_47, dan Local_feature_41. Pemilihan 20 fitur dilakukan agar hubungan antarfitur dapat ditampilkan lebih jelas dalam bentuk heatmap dan tidak terlalu padat apabila seluruh 165 fitur ditampilkan sekaligus.

Hasil heatmap menunjukkan bahwa terdapat beberapa kelompok fitur yang memiliki hubungan cukup kuat, tetapi tidak seluruh fitur mempunyai korelasi yang tinggi. Kondisi ini menunjukkan bahwa informasi pada dataset tidak hanya bergantung pada satu kelompok fitur tertentu, tetapi berasal dari kombinasi berbagai karakteristik transaksi. Dengan memasukkan transaksi Unknown dalam analisis ini, pola fitur yang dianalisis menjadi lebih mewakili keseluruhan dataset dan tidak hanya terbatas pada transaksi yang telah memiliki label Licit dan Illicit.

4.1.3 Analisis *Time step*

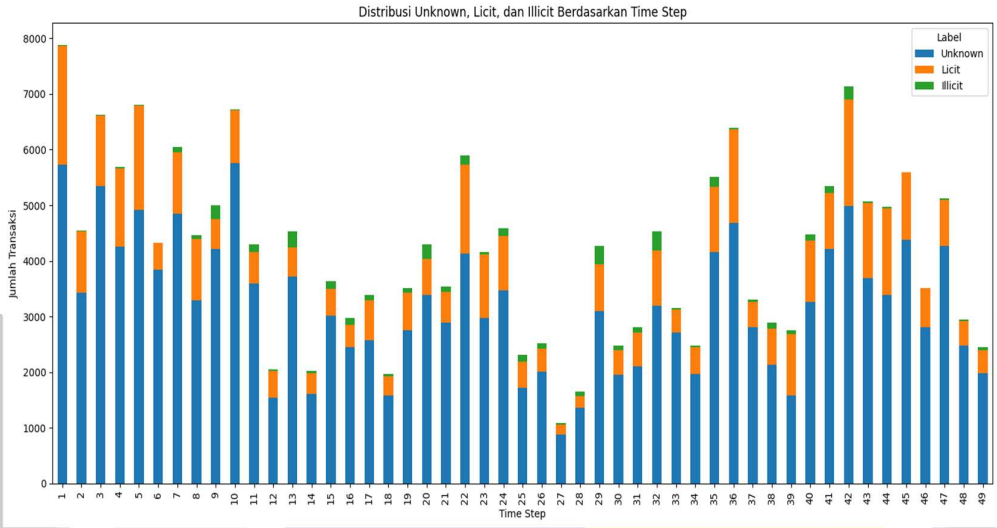
Analisis *time step* dilakukan untuk memahami bagaimana distribusi transaksi *Bitcoin* pada setiap periode pengamatan dalam Elliptic *Bitcoin* Dataset. Dataset ini membagi seluruh transaksi ke dalam 49 *time step* yang merepresentasikan urutan waktu transaksi pada jaringan *Blockchain*. Analisis ini bertujuan untuk mengetahui pola distribusi transaksi secara temporal serta mengidentifikasi apakah terdapat periode tertentu yang memiliki aktivitas transaksi lebih tinggi dibandingkan periode lainnya.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.3 Transaction Distribution by *Time step*

Berdasarkan Gambar 4.3, distribusi seluruh transaksi pada Elliptic Bitcoin Dataset tersebar dalam 49 time step. Jumlah transaksi pada setiap time step terlihat berbeda-beda, sehingga aktivitas transaksi pada jaringan Bitcoin tidak terjadi secara merata pada setiap periode pengamatan. Pada analisis terbaru ini, transaksi Unknown tetap disertakan bersama transaksi Licit dan Illicit. Hal tersebut dilakukan karena transaksi *Unknown* juga merupakan bagian dari aktivitas transaksi pada setiap time step, walaupun belum mempunyai label pasti. Dengan memasukkan transaksi *Unknown*, distribusi temporal yang diperoleh dapat menggambarkan keseluruhan aktivitas transaksi yang terdapat pada dataset, bukan hanya transaksi yang telah mempunyai label. Hasil analisis time step menunjukkan bahwa distribusi transaksi, termasuk transaksi Illicit, berubah pada setiap periode pengamatan. Analisis ini digunakan untuk memahami karakteristik temporal dataset, tetapi model GCN dan GAT pada penelitian ini tidak secara khusus memodelkan dependensi temporal. Model utama tetap memanfaatkan fitur transaksi dan struktur hubungan antarnode pada graph

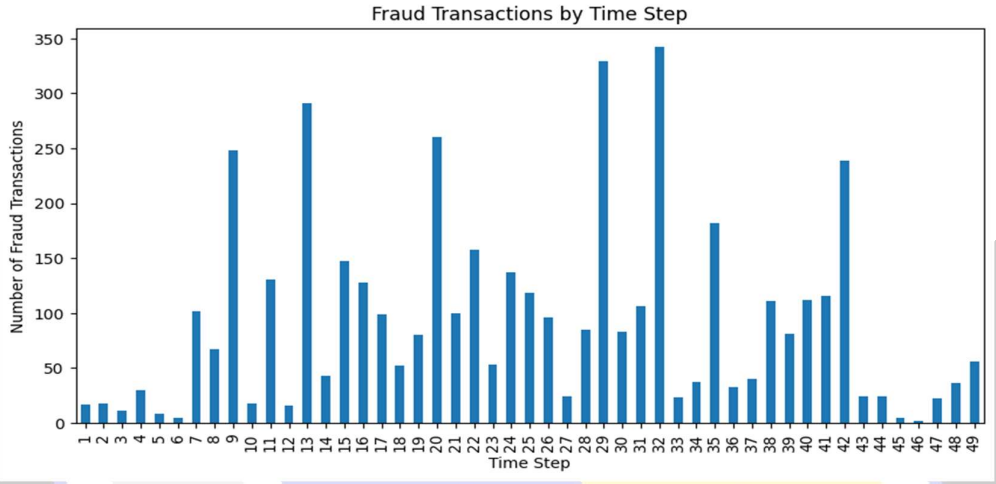


Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.4 Distribusi Label Unknown, Licit, dan Illicit Berdasarkan Time Step

Berdasarkan Gambar 4.4, transaksi Unknown terlihat mendominasi hampir seluruh time step dibandingkan transaksi Licit maupun Illicit. Sebagai contoh, pada time step 1 terdapat 5.733 transaksi Unknown, 2.130 transaksi Licit, dan 17 transaksi Illicit. Pada time step 32 terdapat 3.202 transaksi Unknown, 981 transaksi Licit, dan 342 transaksi Illicit. Dominasi transaksi Unknown menunjukkan bahwa sebagian besar transaksi pada dataset belum mempunyai ground truth yang pasti. Meskipun demikian, transaksi tersebut tetap memiliki fitur dan hubungan dengan transaksi lain pada struktur graph.

Oleh karena itu, transaksi Unknown perlu dianalisis lebih lanjut untuk mengetahui apakah pola fitur dan keterhubungannya lebih mendekati transaksi Licit atau Illicit. Distribusi ini juga menunjukkan bahwa jumlah transaksi Illicit tidak sama pada setiap periode. Pada beberapa time step jumlah transaksi Illicit meningkat, sedangkan pada periode lainnya relatif rendah. Hal tersebut menunjukkan bahwa aktivitas transaksi ilegal pada jaringan Bitcoin memiliki pola temporal yang berubah-ubah



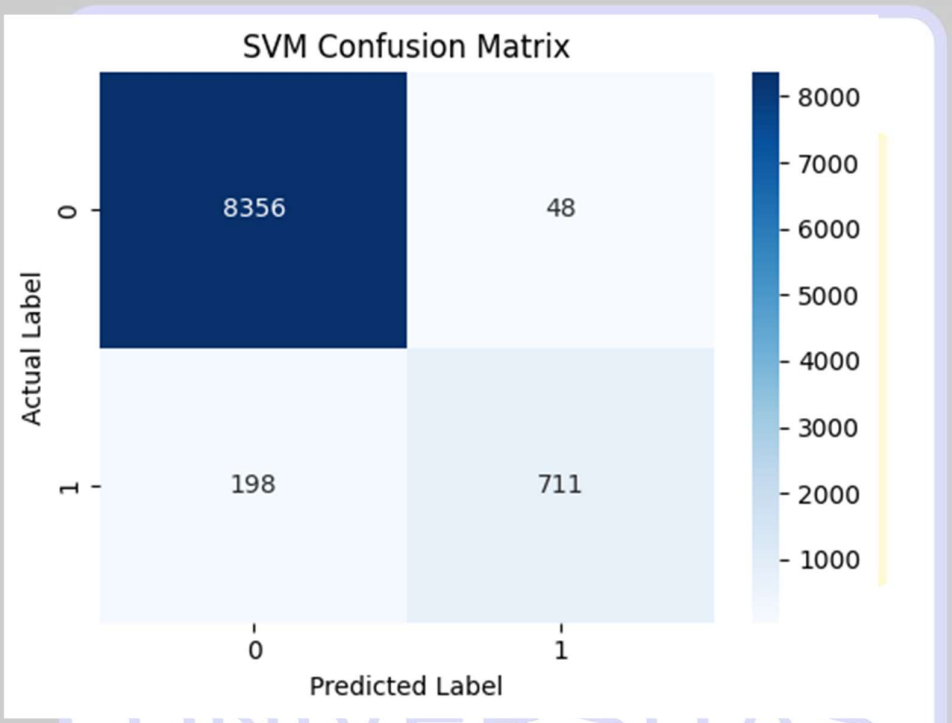
Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.5 *Fraud* Transaction by *Time step*

Berdasarkan Gambar 4.5, distribusi transaksi *fraud* juga tidak merata pada seluruh *time step*. Terlihat bahwa beberapa periode memiliki jumlah transaksi ilegal yang lebih tinggi dibandingkan periode lainnya, sedangkan pada beberapa periode awal maupun akhir jumlah transaksi *fraud* relatif lebih sedikit. Hal ini menunjukkan bahwa aktivitas transaksi ilegal tidak terjadi secara konstan, melainkan cenderung terkonsentrasi pada periode-periode tertentu.

Hasil analisis *time step* menunjukkan bahwa baik distribusi transaksi secara keseluruhan maupun distribusi transaksi *fraud* memiliki pola yang bersifat dinamis. Variasi jumlah transaksi pada setiap periode menunjukkan bahwa karakteristik jaringan *Bitcoin* berubah seiring waktu. Selain itu, adanya konsentrasi transaksi *fraud* pada beberapa *time step* mengindikasikan bahwa aktivitas ilegal memiliki kecenderungan muncul pada periode tertentu. Informasi temporal tersebut menjadi salah satu alasan penggunaan *Graph Convolutional Network (GCN)*, karena model tidak hanya memanfaatkan atribut transaksi, tetapi juga hubungan antartransaksi dalam struktur *graph* sehingga mampu menangkap pola yang sulit dikenali oleh metode klasifikasi berbasis fitur saja.

Setelah itu tahap berikutnya adalah *Confusion Matrix* yang digunakan untuk melihat kemampuan model dalam mengklasifikasikan transaksi *Bitcoin* ke dalam kelas licit dan illicit**.** Dari hasil pengujian dapat diketahui jumlah prediksi yang benar (*True Positive* dan *True Negative*) maupun prediksi yang salah (*False Positive* dan *False Negative*). Semakin tinggi nilai *True Positive* dan *True Negative*, serta semakin rendah nilai *False Positive* dan *False Negative*, maka semakin baik kemampuan model dalam melakukan klasifikasi



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.6 *Confusion Matrix* SVM

Berdasarkan Gambar 4.6, hasil pengujian model *Support Vector Machine (SVM)* menunjukkan bahwa model mampu mengklasifikasikan sebagian besar transaksi *Bitcoin* dengan baik. Dari hasil *Confusion Matrix* terlihat bahwa terdapat 8.356 transaksi legal yang berhasil diprediksi dengan benar sebagai transaksi legal **(***True Negative*/TN). Hasil ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengenali transaksi normal sehingga hanya sedikit transaksi legal yang mengalami kesalahan klasifikasi.

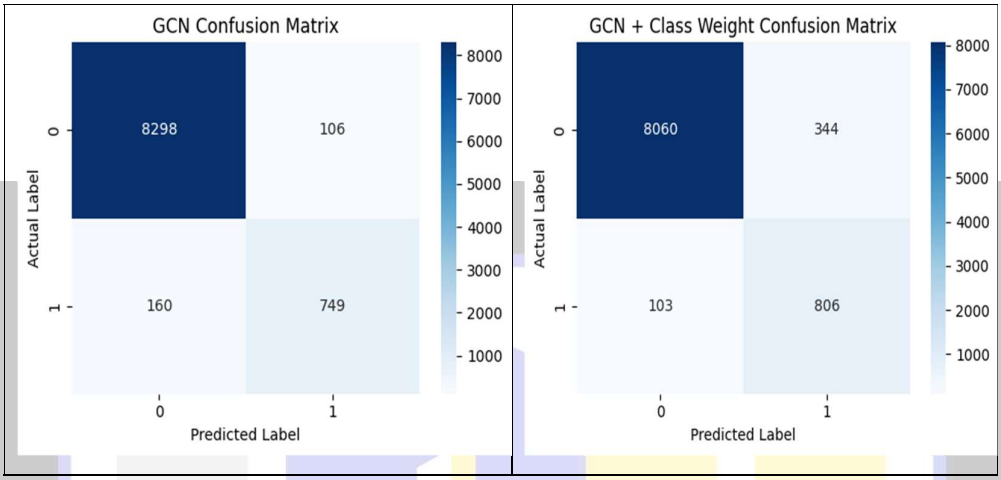
Selain itu, model juga berhasil mendeteksi 711 transaksi ilegal dengan benar (*True Positive*/TP). Hal ini menunjukkan bahwa SVM mampu mengenali pola pada sebagian besar transaksi ilegal yang terdapat pada dataset. Namun demikian, masih terdapat 198 transaksi ilegal yang diprediksi sebagai transaksi legal (*False Negative*/FN). Kesalahan ini menjadi perhatian utama karena transaksi yang seharusnya teridentifikasi justru tidak terdeteksi oleh model.

Di sisi lain, terdapat 48 transaksi legal yang diprediksi sebagai transaksi ilegal (*False Positive*/FP). Jumlah tersebut relatif kecil jika dibandingkan dengan jumlah transaksi legal yang berhasil diklasifikasikan dengan benar. Kesalahan seperti ini dapat menyebabkan adanya transaksi yang sebenarnya tidak bermasalah ikut diperiksa lebih lanjut. Meskipun demikian, dampaknya masih lebih ringan dibandingkan apabila transaksi ilegal tidak berhasil terdeteksi.

Apabila dikaitkan dengan hasil evaluasi model, nilai *Accuracy* sebesar 97,36% menunjukkan bahwa secara keseluruhan SVM memiliki tingkat klasifikasi yang tinggi. Akan tetapi, nilai *Recall* sebesar 78,22% mengindikasikan bahwa model masih belum mampu mendeteksi seluruh transaksi ilegal yang ada pada data pengujian. Hal ini terlihat dari masih adanya 198 transaksi ilegal yang lolos dari proses klasifikasi.

Dalam penelitian ini, *False Negative* merupakan jenis kesalahan yang paling penting untuk diperhatikan. Hal tersebut karena tujuan penelitian adalah mendeteksi transaksi ilegal pada jaringan *Bitcoin* untuk mendukung kebutuhan *digital forensik*. Semakin sedikit nilai *False Negative*, semakin kecil kemungkinan transaksi ilegal lolos dari proses identifikasi sehingga proses investigasi dapat dilakukan dengan lebih efektif. Oleh karena itu, meskipun model SVM menghasilkan akurasi yang tinggi, kemampuannya dalam mendeteksi seluruh transaksi ilegal masih perlu ditingkatkan. Kondisi inilah yang menjadi alasan penggunaan *Graph Attention Network (GAT)* sebagai metode utama dalam penelitian ini, karena GAT mampu memanfaatkan hubungan antartransaksi pada struktur *graph* melalui mekanisme attention sehingga diharapkan dapat

meningkatkan kemampuan deteksi terhadap transaksi ilegal dibandingkan model berbasis *graph* konvensional seperti GCN.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.7 *Confusion Matrix* GCN dan GCN + Class Weight

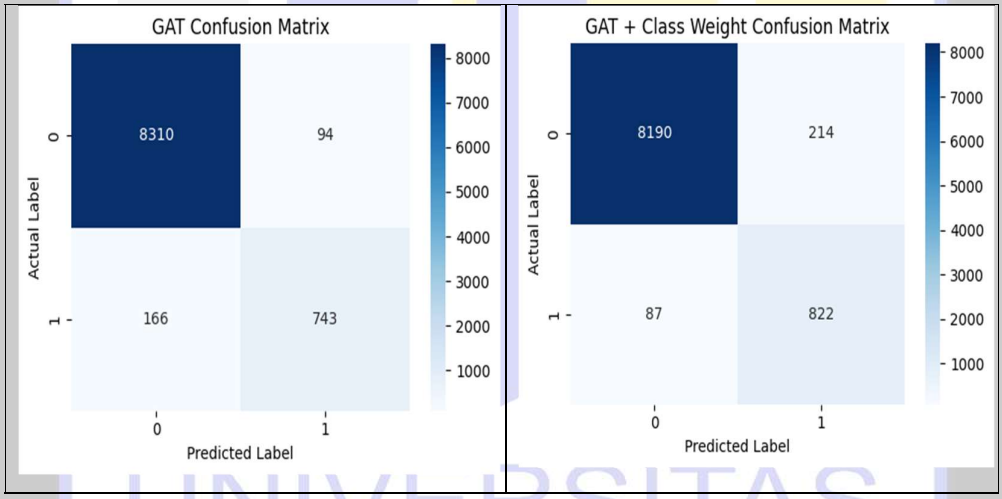
Gambar 4.7 menunjukkan hasil *Confusion Matrix* dari model *Graph Convolutional Network (GCN)* dan GCN + *Class Weight* yang digunakan untuk mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Perbandingan kedua model ini bertujuan untuk mengetahui pengaruh penerapan *Class Weight* dalam mengatasi ketidakseimbangan kelas (*class imbalance*) yang terdapat pada dataset Elliptic, di mana jumlah transaksi ilegal jauh lebih sedikit dibandingkan transaksi legal.

Berdasarkan hasil pengujian, model GCN menghasilkan *True Negative* (TN) sebanyak 8.298, yang berarti sebanyak 8.298 transaksi legal berhasil diklasifikasikan dengan benar sebagai transaksi legal. Selain itu, model juga mampu mendeteksi 749 transaksi ilegal dengan benar atau *True Positive* (TP). Namun, masih terdapat 160 transaksi ilegal yang diprediksi sebagai transaksi legal (*False Negative*/FN) dan 106 transaksi legal yang diprediksi sebagai transaksi ilegal (*False Positive*/FP). Nilai *False Negative* tersebut menunjukkan bahwa masih ada sejumlah transaksi ilegal yang tidak berhasil dikenali oleh model sehingga berpotensi lolos dari proses identifikasi.

Setelah diterapkan *Class Weight*, terlihat adanya perubahan pada hasil klasifikasi. Model GCN + *Class Weight* berhasil meningkatkan jumlah *True Positive* menjadi 806 transaksi, sementara jumlah *False Negative* menurun menjadi 103 transaksi. Hasil ini menunjukkan bahwa pemberian bobot pada kelas minoritas membuat model menjadi lebih sensitif dalam mengenali transaksi ilegal. Dengan demikian, jumlah transaksi ilegal yang berhasil dideteksi menjadi lebih banyak dibandingkan model GCN tanpa *Class Weight*. Di sisi lain, penerapan *Class Weight* juga menyebabkan jumlah *False Positive* meningkat dari 106 menjadi 344 transaksi. Hal ini menunjukkan bahwa model menjadi lebih sering mengklasifikasikan transaksi legal sebagai transaksi ilegal. Kondisi tersebut merupakan konsekuensi dari pemberian bobot yang lebih besar pada kelas minoritas sehingga model cenderung lebih berhati-hati dalam melakukan klasifikasi terhadap transaksi yang memiliki karakteristik mirip dengan transaksi ilegal.

Apabila kedua model dibandingkan secara keseluruhan, terlihat bahwa penggunaan *Class Weight* memberikan peningkatan yang cukup baik terhadap kemampuan model dalam mendeteksi transaksi ilegal. Hal tersebut ditunjukkan dengan bertambahnya jumlah *True Positive* dari 749 menjadi 806 transaksi, serta menurunnya jumlah *False Negative* dari 160 menjadi 103 transaksi. Meskipun jumlah *False Positive* meningkat, perubahan tersebut masih dapat diterima karena tujuan utama penelitian ini adalah meminimalkan jumlah transaksi ilegal yang tidak terdeteksi. Dalam konteks *digital forensik*, *False Negative* merupakan jenis kesalahan yang paling perlu diperhatikan. Transaksi ilegal yang tidak berhasil diidentifikasi berpotensi menyebabkan aktivitas kriminal, seperti pencucian uang atau pendanaan kegiatan ilegal, lolos dari proses investigasi. Sebaliknya, *False Positive* hanya menyebabkan transaksi legal ikut ditandai sebagai transaksi yang mencurigakan dan masih dapat diverifikasi kembali oleh penyidik melalui proses pemeriksaan lanjutan. Oleh karena itu, mengurangi jumlah *False Negative* menjadi prioritas yang lebih penting dibandingkan mempertahankan jumlah *False Positive* yang rendah.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa penerapan *Class Weight* mampu meningkatkan kemampuan model GCN dalam mendeteksi transaksi ilegal pada dataset yang memiliki distribusi kelas tidak seimbang. Meskipun jumlah *False Positive* mengalami peningkatan, penurunan *False Negative* menunjukkan bahwa model menjadi lebih efektif dalam mengenali transaksi ilegal. Hasil ini sejalan dengan tujuan penelitian, yaitu membangun model yang mampu mendeteksi transaksi ilegal pada jaringan *Bitcoin* secara lebih akurat untuk mendukung kebutuhan *digital forensik*. Oleh karena itu, model GCN + *Class Weight* dapat dianggap lebih sesuai digunakan dibandingkan model GCN tanpa *Class Weight* karena mampu mengidentifikasi lebih banyak transaksi ilegal yang berpotensi menjadi objek investigasi.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.8 *Confusion Matrix* GAT dan GAT + Class Weight

Gambar 4.8 menunjukkan hasil *Confusion Matrix* dari model *Graph Attention Network (GAT)* dan *GAT + Class Weight* dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Perbandingan kedua model ini dilakukan untuk mengetahui pengaruh penerapan *Class Weight* terhadap kemampuan model dalam mengatasi ketidakseimbangan kelas (*class imbalance*) pada dataset Elliptic. Karena jumlah transaksi ilegal jauh lebih sedikit dibandingkan transaksi legal, pemberian bobot pada kelas minoritas diharapkan dapat meningkatkan kemampuan model dalam mengenali transaksi ilegal.

Berdasarkan hasil pengujian, model GAT menghasilkan *True Negative* (TN) sebanyak 8.310, yang berarti sebanyak 8.310 transaksi legal berhasil diprediksi dengan benar sebagai transaksi legal. Selain itu, model juga mampu mendeteksi 743 transaksi ilegal dengan benar atau *True Positive* (TP). Namun demikian, masih terdapat 166 transaksi ilegal yang diprediksi sebagai transaksi legal (*False Negative*/FN) serta 94 transaksi legal yang diprediksi sebagai transaksi ilegal (*False Positive*/FP). Hasil ini menunjukkan bahwa model GAT sudah memiliki kemampuan klasifikasi yang cukup baik, tetapi masih terdapat sejumlah transaksi ilegal yang belum berhasil diidentifikasi. Setelah diterapkan *Class Weight*, hasil klasifikasi mengalami perubahan. Model GAT + *Class Weight* mampu meningkatkan jumlah *True Positive* menjadi 822 transaksi, sedangkan jumlah *False Negative* menurun menjadi 87 transaksi. Penurunan *False Negative* ini menunjukkan bahwa pemberian bobot pada kelas minoritas membuat model lebih efektif dalam mengenali transaksi ilegal. Dengan demikian, jumlah transaksi ilegal yang berhasil dideteksi meningkat dibandingkan model GAT tanpa *Class Weight*. Di sisi lain, penerapan *Class Weight* juga menyebabkan jumlah *False Positive* meningkat dari 94 menjadi 214 transaksi. Hal ini menunjukkan bahwa model menjadi lebih sensitif terhadap pola transaksi yang dianggap mencurigakan sehingga lebih banyak transaksi legal yang diprediksi sebagai transaksi ilegal. Kondisi tersebut merupakan konsekuensi dari penerapan *Class Weight*, di mana model lebih memprioritaskan pendeteksian kelas minoritas sehingga sebagian transaksi legal ikut terklasifikasi sebagai transaksi yang perlu mendapat perhatian.

Jika kedua model dibandingkan, terlihat bahwa penerapan *Class Weight* memberikan peningkatan yang cukup signifikan terhadap kemampuan model dalam mendeteksi transaksi ilegal. Hal ini ditunjukkan dengan meningkatnya jumlah *True Positive* dari 743 menjadi 822 transaksi, serta berkurangnya jumlah *False Negative* dari 166 menjadi 87 transaksi. Meskipun jumlah *False Positive* mengalami peningkatan, hasil tersebut masih dapat diterima karena tujuan utama penelitian ini adalah mengurangi kemungkinan transaksi ilegal tidak terdeteksi.

Dalam konteks *digital forensik*, *False Negative* merupakan jenis kesalahan yang paling perlu diperhatikan karena transaksi ilegal yang tidak berhasil diidentifikasi

dapat menyebabkan aktivitas kriminal, seperti pencucian uang, pendanaan ilegal, atau tindak kejahatan lainnya, lolos dari proses investigasi. Sebaliknya, *False Positive* hanya menyebabkan transaksi legal ikut ditandai sebagai transaksi yang mencurigakan dan masih dapat diverifikasi kembali melalui proses investigasi lanjutan. Oleh karena itu, penurunan jumlah *False Negative* menjadi indikator yang lebih penting dibandingkan peningkatan *False Positive*.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa penerapan *Class Weight* berhasil meningkatkan kemampuan model GAT dalam mendeteksi transaksi ilegal pada dataset yang memiliki distribusi kelas tidak seimbang. Meskipun jumlah *False Positive* meningkat, penurunan *False Negative* menunjukkan bahwa model menjadi lebih efektif dalam mengenali transaksi ilegal dibandingkan model GAT tanpa *Class Weight*. Hasil ini menunjukkan bahwa penggunaan *Class Weight* mampu meningkatkan sensitivitas model terhadap kelas minoritas sehingga lebih sesuai digunakan untuk mendukung kebutuhan *digital forensik*, di mana kemampuan mendeteksi transaksi ilegal menjadi aspek yang lebih penting daripada meminimalkan kesalahan pada transaksi legal.

Berdasarkan hasil yang diperoleh, model GAT + *Class Weight* bahkan menunjukkan peningkatan yang lebih besar dibandingkan GCN + *Class Weight*, yang terlihat dari penurunan *False Negative* pada GAT dari 166 menjadi 87 transaksi setelah penerapan *Class Weight*, sementara pada GCN penurunannya hanya dari 160 menjadi 103 transaksi. Hal ini menunjukkan bahwa mekanisme attention pada GAT mampu memanfaatkan informasi hubungan antartransaksi dengan lebih baik ketika dikombinasikan dengan *Class Weight*. Dengan demikian, model ini memiliki potensi yang lebih tinggi dalam mendukung proses identifikasi transaksi ilegal pada jaringan *Bitcoin* untuk kebutuhan digital forensik.

Hasil pengujian menunjukkan bahwa penerapan *Class Weight* memberikan pengaruh terhadap kemampuan model dalam mengenali transaksi *Illicit*. Pada GAT + Class Weight, kelas *Illicit* diberikan bobot sebesar 5,1225, jauh lebih besar dibandingkan bobot kelas *Licit* sebesar 0,5541. Pemberian bobot tersebut menyebabkan kesalahan klasifikasi terhadap transaksi *Illicit* menghasilkan penalti

loss yang lebih besar selama proses pelatihan. Akibatnya, model menjadi lebih sensitif dalam mengenali kelas *Illicit*. Pengaruh tersebut terlihat dari peningkatan kemampuan deteksi kelas minoritas. Model GAT + Class Weight memperoleh *recall* sebesar 90,43%, lebih tinggi dibandingkan GAT tanpa Class Weight. Selain itu, GAT + Class Weight menghasilkan *Macro Recall* sebesar 0,9394. Hasil tersebut menunjukkan bahwa penerapan *Class Weight* membantu model mengurangi kecenderungan untuk terlalu dominan memprediksi kelas mayoritas.

Namun, peningkatan sensitivitas terhadap kelas *Illicit* juga menimbulkan *trade-off*. Peningkatan *recall* diikuti dengan penurunan *precision* karena lebih banyak transaksi *Licit* yang ikut diprediksi sebagai *Illicit*. Dalam konteks penelitian ini, kondisi tersebut masih dapat diterima karena tujuan utama model adalah mengurangi *False Negative*, yaitu transaksi ilegal yang tidak berhasil terdeteksi. Dalam proses digital forensik, transaksi yang terindikasi ilegal masih dapat diperiksa lebih lanjut oleh investigator, sedangkan transaksi ilegal yang tidak terdeteksi berpotensi tidak masuk ke dalam proses investigasi.

Tabel 4.2 Perbandingan Hasil *Confusion Matrix* Seluruh Model

Model	TN	FP	FN	TP
SVM	8356	48	198	711
GCN	8298	106	160	749
GCN + *Class Weight*	8060	344	103	806
GAT	8310	94	166	743
GAT + *Class Weight*	8190	214	**87**	**822**

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Tabel 4.2 menyajikan perbandingan hasil *Confusion Matrix* dari lima model yang digunakan dalam penelitian, yaitu *Support Vector Machine (SVM)*, *Graph Convolutional Network (GCN)*, GCN + *Class Weight*, *Graph Attention Network* (GAT), dan GAT + *Class Weight*. Perbandingan ini dilakukan untuk melihat kemampuan masing-masing model dalam mengklasifikasikan transaksi legal dan transaksi ilegal berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP).

Berdasarkan hasil pengujian, model SVM menghasilkan nilai *True Negative* sebanyak 8.356, *False Positive* sebanyak 48, *False Negative* sebanyak 198, dan *True Positive* sebanyak 711. Hasil tersebut menunjukkan bahwa SVM memiliki kemampuan yang sangat baik dalam mengenali transaksi legal karena menghasilkan nilai *False Positive* yang paling rendah. Namun, model ini masih memiliki kelemahan dalam mendeteksi transaksi ilegal, yang terlihat dari nilai *False Negative* sebesar 198. Artinya, masih terdapat cukup banyak transaksi ilegal yang tidak berhasil dikenali oleh model.

Model GCN menunjukkan peningkatan dibandingkan SVM. Jumlah *False Negative* berhasil diturunkan menjadi 160 transaksi, sedangkan jumlah *True Positive* meningkat menjadi 749 transaksi. Hasil ini menunjukkan bahwa pemanfaatan struktur *graph* mampu membantu model mengenali hubungan antartransaksi sehingga kemampuan mendeteksi transaksi ilegal menjadi lebih baik dibandingkan pendekatan berbasis data tabular seperti SVM.

Penerapan *Class Weight* pada model GCN memberikan perubahan yang cukup signifikan. Nilai *False Negative* kembali menurun menjadi 103 transaksi, sedangkan *True Positive* meningkat menjadi 806 transaksi. Hasil tersebut menunjukkan bahwa pemberian bobot pada kelas minoritas berhasil meningkatkan sensitivitas model terhadap transaksi ilegal. Akan tetapi, peningkatan tersebut juga menyebabkan nilai *False Positive* bertambah menjadi 344 transaksi. Hal ini menunjukkan bahwa model menjadi lebih sering mengklasifikasikan transaksi legal sebagai transaksi ilegal karena lebih memprioritaskan pendeteksian kelas minoritas.

Pada model GAT, diperoleh nilai *True Negative* sebesar 8.310, *False Positive* sebanyak 94, *False Negative* sebanyak 166, dan *True Positive* sebanyak 743. Secara umum, performa GAT hampir setara dengan GCN, meskipun jumlah transaksi ilegal yang berhasil dideteksi masih sedikit lebih rendah dibandingkan GCN. Hasil ini menunjukkan bahwa mekanisme attention pada GAT sudah mampu memanfaatkan hubungan antar*node*, tetapi belum memberikan peningkatan yang signifikan ketika belum dikombinasikan dengan teknik penanganan ketidakseimbangan kelas.

Performa terbaik diperoleh oleh model *GAT + Class Weight*. Model ini menghasilkan *True Positive* sebanyak 822 transaksi dan *False Negative* hanya sebesar 87 transaksi, yang merupakan nilai terendah di antara seluruh model yang diuji. Hal tersebut menunjukkan bahwa GAT + *Class Weight* memiliki kemampuan paling baik dalam mengenali transaksi ilegal. Walaupun nilai *False Positive* meningkat menjadi 214 transaksi, jumlah tersebut masih lebih rendah dibandingkan GCN + *Class Weight* yang menghasilkan 344 *False Positive*. Dengan demikian, model GAT + *Class Weight* mampu memberikan keseimbangan yang lebih baik antara peningkatan deteksi transaksi ilegal dan jumlah kesalahan klasifikasi pada transaksi legal.

Apabila dibandingkan secara keseluruhan, terlihat bahwa penggunaan metode berbasis *graph* memberikan hasil yang lebih baik dibandingkan model SVM. Selain itu, penerapan *Class Weight* pada model GCN maupun GAT terbukti mampu mengurangi jumlah *False Negative* secara signifikan. Penurunan *False Negative* merupakan hasil yang sangat penting karena menunjukkan semakin sedikit transaksi ilegal yang lolos dari proses klasifikasi.

Dalam konteks *digital forensik*, *False Negative* merupakan jenis kesalahan yang paling perlu diminimalkan. Transaksi ilegal yang tidak berhasil dideteksi berpotensi menyebabkan aktivitas kriminal, seperti pencucian uang, pendanaan ilegal, maupun tindak kejahatan lainnya, tidak teridentifikasi selama proses investigasi. Sebaliknya, *False Positive* hanya menyebabkan transaksi legal ikut ditandai sebagai transaksi yang mencurigakan dan masih dapat diverifikasi kembali melalui proses pemeriksaan lanjutan oleh investigator. Oleh karena itu, model dengan nilai *False Negative* yang lebih rendah dinilai lebih sesuai untuk mendukung kebutuhan *digital forensik*.

Berdasarkan hasil perbandingan pada Tabel 4.2, dapat disimpulkan bahwa model GAT + *Class Weight* merupakan model dengan performa terbaik dalam penelitian ini. Model tersebut berhasil menghasilkan jumlah *True Positive* tertinggi dan *False Negative* terendah dibandingkan model lainnya. Hasil ini menunjukkan bahwa kombinasi mekanisme *Graph Attention Network* dengan *Class Weight* mampu meningkatkan kemampuan model dalam mendeteksi transaksi ilegal pada

dataset yang memiliki distribusi kelas tidak seimbang. Dengan demikian, model GAT + *Class Weight* dinilai paling efektif untuk mendukung proses identifikasi transaksi ilegal pada jaringan *Bitcoin* dalam kebutuhan *digital forensik*.

Tabel 4.3 Hasil Evaluasi Model

Model	*Accuracy* (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	**97,36**	**93,68**	78,22	**85,25**
GCN	97,14	87,60	82,40	84,92
GCN + *Class Weight*	95,20	70,09	88,67	78,29
GAT	97,21	88,77	81,74	85,11
GAT + *Class Weight*	96,77	79,34	**90,43**	84,52

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Berdasarkan Tabel 4.3, dapat diketahui bahwa setiap model memiliki keunggulan pada metrik evaluasi yang berbeda. Hal ini menunjukkan bahwa tidak ada satu model yang unggul pada seluruh aspek, sehingga pemilihan model perlu disesuaikan dengan tujuan penelitian, yaitu mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Model SVM memperoleh akurasi tertinggi sebesar 97,36%, dengan *Precision* 93,68% dan *F1-score* 85,25%, yang juga merupakan nilai tertinggi dibandingkan model lainnya. Hasil ini menunjukkan bahwa SVM sangat baik dalam mengklasifikasikan data secara keseluruhan dan menghasilkan prediksi transaksi ilegal dengan tingkat ketepatan yang tinggi. Namun, nilai *recall* sebesar 78,22% merupakan yang terendah di antara seluruh model, sehingga masih terdapat cukup banyak transaksi ilegal yang tidak berhasil terdeteksi (*False Negative*).

Model GCN menghasilkan akurasi sebesar 97,14%, yang hanya sedikit lebih rendah dibandingkan SVM. Meskipun *Precision* GCN menurun menjadi 87,60%, nilai *recall* meningkat menjadi 82,40%. Peningkatan *recall* ini menunjukkan bahwa GCN mampu menemukan lebih banyak transaksi ilegal dibandingkan SVM. Hal tersebut mengindikasikan bahwa pemanfaatan informasi hubungan antartransaksi pada struktur *graph* memberikan kontribusi dalam meningkatkan kemampuan deteksi transaksi ilegal.

Pada model GCN + *Class Weight*, terjadi perubahan karakteristik performa. Nilai *recall* meningkat cukup signifikan menjadi 88,67%, namun diikuti penurunan akurasi menjadi 95,20%, *Precision* menjadi 70,09%, dan *F1-score* menjadi 78,29%. Kondisi ini menunjukkan bahwa pemberian bobot kelas berhasil meningkatkan kemampuan model dalam mendeteksi transaksi ilegal, tetapi juga menyebabkan jumlah prediksi positif yang salah (*False Positive*) menjadi lebih banyak. Model GAT memperoleh akurasi sebesar 97,21%, dengan *Precision* 88,77%, *recall* 81,74%, dan *F1-score* 85,11%. Secara keseluruhan, performa GAT cukup seimbang dan mendekati GCN. Mekanisme attention pada GAT mampu memberikan hasil klasifikasi yang baik, meskipun peningkatan *recall* dibandingkan GCN tidak terlalu signifikan.

Sementara itu, GAT + *Class Weight* menghasilkan *recall* tertinggi sebesar 90,43%, yang berarti model ini paling baik dalam mendeteksi transaksi ilegal dan paling sedikit melewatkan transaksi ilegal dibandingkan model lainnya. Meskipun demikian, peningkatan *recall* tersebut diikuti penurunan *Precision* menjadi 79,34% dan akurasi menjadi 96,77%. Hal ini menunjukkan adanya *trade-off* antara kemampuan mendeteksi transaksi ilegal dan meningkatnya jumlah *False Positive*.

Berdasarkan keseluruhan hasil evaluasi, apabila akurasi dan *Precision* dijadikan sebagai acuan utama, maka SVM merupakan model dengan performa terbaik. Namun, apabila penelitian lebih memprioritaskan kemampuan mendeteksi sebanyak mungkin transaksi ilegal, maka GAT + *Class Weight* menjadi model yang lebih sesuai karena memiliki nilai *recall* tertinggi sebesar 90,43%. Mengingat penelitian ini berfokus pada deteksi transaksi ilegal dalam konteks *digital forensik*, metrik *recall* menjadi lebih penting dibandingkan akurasi, karena kegagalan mendeteksi transaksi ilegal (*False Negative*) dapat menyebabkan aktivitas kriminal pada jaringan *Blockchain* tidak teridentifikasi. Dengan demikian, dapat disimpulkan bahwa penambahan *Class Weight* pada model berbasis *Graph Neural Network* mampu meningkatkan kemampuan model dalam mendeteksi transaksi ilegal yang ditunjukkan oleh peningkatan nilai *recall*. Meskipun terjadi penurunan pada nilai akurasi dan *Precision*, peningkatan *recall* menjadi aspek yang lebih penting dalam penelitian ini karena tujuan utama adalah meminimalkan jumlah

False Negative, sehingga semakin sedikit transaksi ilegal yang lolos dari proses deteksi.

Pemilihan Recall sebagai metrik yang lebih diprioritaskan juga berkaitan dengan tujuan penggunaan model dalam mendukung digital forensik. Pada hasil pengujian, GAT + Class Weight memperoleh Recall tertinggi sebesar 90,43%. Nilai tersebut menunjukkan bahwa model mampu mendeteksi sebagian besar transaksi Illicit yang terdapat pada data pengujian. Meskipun Precision yang dihasilkan lebih rendah dibandingkan beberapa model lainnya, peningkatan Recall menunjukkan bahwa jumlah transaksi Illicit yang terlewat atau False Negative dapat ditekan. Dalam proses investigasi, kondisi ini lebih diutamakan karena transaksi yang terdeteksi sebagai mencurigakan masih dapat diperiksa kembali oleh investigator, sedangkan transaksi Illicit yang tidak terdeteksi berpotensi tidak masuk ke proses pemeriksaan lebih lanjut. Pada penelitian ini, Recall menjadi metrik yang lebih diprioritaskan karena tujuan utama penelitian adalah mendeteksi sebanyak mungkin transaksi Illicit. Recall menunjukkan kemampuan model dalam menemukan transaksi Illicit dari seluruh transaksi Illicit yang sebenarnya terdapat pada data pengujian. Nilai Recall yang tinggi menunjukkan bahwa jumlah transaksi Illicit yang tidak berhasil terdeteksi atau False Negative (FN) semakin sedikit. Dalam konteks digital forensik, False Negative menjadi perhatian karena transaksi Illicit yang tidak terdeteksi berpotensi tidak masuk ke tahap pemeriksaan lebih lanjut. Sementara itu, False Positive masih dapat dilakukan verifikasi kembali pada tahap analisis oleh investigator. Oleh karena itu, penelitian ini lebih memprioritaskan Recall untuk menekan False Negative, tetapi tetap mempertimbangkan Precision, F1-score, Accuracy, dan AUC dalam menilai performa model secara keseluruhan.

Tabel 4.4 Hasil Evaluasi Model Menggunakan *Macro Average* dan *Micro Average*

Model	Macro Precision	Macro Recall	Macro F1	Micro Precision	Micro Recall	Micro F1
SVM	0,9568	0,8882	0,9190	0,9736	0,9736	0,9736
GCN	0,9286	0,9057	0,9167	0,9714	0,9714	0,9714
GCN + Class Weight	0,8441	0,9229	0,8780	0,9520	0,9520	0,9520
GAT	0,9341	0,9031	0,9178	0,9721	0,9721	0,9721
GAT + Class Weight	0,8915	0,9394	0,9136	0,9677	0,9677	0,9677

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Tabel 4.4 menyajikan hasil evaluasi kelima model menggunakan pendekatan *Macro Average* dan *Micro Average*. Evaluasi ini digunakan untuk memberikan gambaran performa model secara lebih menyeluruh, terutama karena dataset yang digunakan memiliki distribusi kelas yang tidak seimbang antara transaksi *Licit* dan *Illicit*. *Macro Average* menghitung nilai *Precision*, *Recall*, dan *F1-score* pada setiap kelas secara terpisah, kemudian menghitung rata-ratanya tanpa mempertimbangkan jumlah data pada masing-masing kelas. Oleh karena itu, *Macro Average* dapat memberikan gambaran mengenai kemampuan model dalam mengenali kedua kelas secara lebih seimbang, termasuk kelas minoritas yaitu transaksi *Illicit*.

Sementara itu, *Micro Average* menghitung seluruh hasil klasifikasi dari semua kelas secara bersama-sama sebelum menghitung nilai *Precision*, *Recall*, dan *F1-score*. Pada klasifikasi *binary single-label* yang digunakan dalam penelitian ini, nilai *Micro Precision*, *Micro Recall*, dan *Micro F1-score* memiliki nilai yang sama dengan *Accuracy*. Oleh karena itu, nilai *Micro Average* lebih menggambarkan performa klasifikasi secara keseluruhan dan dapat lebih dipengaruhi oleh jumlah data pada kelas mayoritas.Berdasarkan hasil evaluasi *Macro Average*, SVM memperoleh nilai *Macro Precision* tertinggi sebesar 0,9568 dan *Macro F1-score* tertinggi sebesar 0,9190. Hasil tersebut menunjukkan bahwa SVM memiliki keseimbangan yang baik antara *Precision* dan *Recall* ketika performa kedua kelas dihitung secara rata-rata. Model GAT memperoleh *Macro Precision* sebesar 0,9341, *Macro Recall* sebesar 0,9031, dan *Macro F1-score* sebesar 0,9178, sehingga memiliki nilai *Macro F1-score* yang mendekati SVM. Sementara itu,

GCN memperoleh *Macro Precision* sebesar 0,9286, *Macro Recall* sebesar 0,9057, dan *Macro F1-score* sebesar 0,9167.

Penerapan *Class Weight* memberikan perubahan terhadap karakteristik performa model. Pada GCN + *Class Weight*, nilai *Macro Recall* meningkat menjadi 0,9229, sedangkan *Macro Precision* menjadi 0,8441 dan *Macro F1-score* menjadi 0,8780. Pada GAT + *Class Weight*, nilai *Macro Recall* meningkat menjadi **0,9394**, yang merupakan nilai tertinggi di antara seluruh model. Namun, *Macro Precision* model tersebut menjadi 0,8915 dengan *Macro F1-score* sebesar 0,9136. Hasil ini menunjukkan adanya *trade-off*, yaitu peningkatan kemampuan model dalam mengenali kelas secara lebih sensitif diikuti dengan penurunan ketepatan prediksi pada sebagian kasus.

Pada evaluasi *Micro Average*, SVM memperoleh nilai tertinggi sebesar 0,9736, diikuti oleh GAT sebesar 0,9721, GCN sebesar 0,9714, GAT + *Class Weight* sebesar 0,9677, dan GCN + *Class Weight* sebesar 0,9520. Nilai tersebut menunjukkan bahwa SVM memiliki performa klasifikasi keseluruhan tertinggi berdasarkan *Micro Average*. Meskipun demikian, *Micro Average* tidak menjadi satu-satunya dasar dalam menentukan model utama karena penelitian ini berfokus pada deteksi transaksi *Illicit* yang merupakan kelas minoritas.

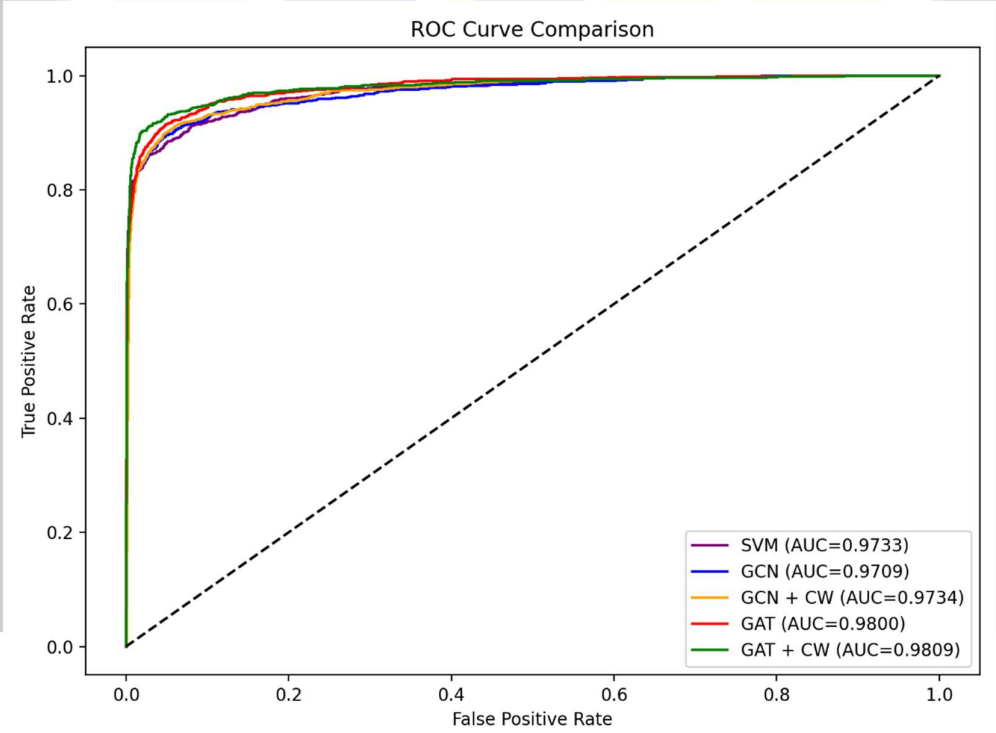
Secara keseluruhan, hasil evaluasi menunjukkan bahwa setiap model memiliki keunggulan pada metrik yang berbeda. SVM memperoleh nilai *Macro Precision*, *Macro F1-score*, dan *Micro F1-score* tertinggi, sedangkan GAT + *Class Weight* memperoleh *Macro Recall* tertinggi sebesar 0,9394**.** Hasil ini menunjukkan bahwa penggunaan *Class Weight* meningkatkan sensitivitas model terhadap kelas minoritas, meskipun diikuti oleh penurunan beberapa metrik lainnya. Dalam konteks penelitian ini, GAT + *Class Weight* tetap dipilih sebagai model utama karena tujuan penelitian lebih memprioritaskan kemampuan mendeteksi transaksi *Illicit* dan mengurangi jumlah *False Negative*, bukan hanya memperoleh nilai akurasi atau *Micro Average* tertinggi.

Tabel 4.5 Perkembangan Nilai *Training* Loss per *Epoch*

Model	Epoch 0	Epoch 50	Epoch 100	Epoch 150
GCN	0,5638	0,1126	0,0891	0,0748
GCN + Class Weight	0,7824	0,1926	0,1290	0,0943
GAT	0,8777	0,1369	0,0965	0,0706
GAT + Class Weight	0,7985	0,2460	0,1074	0,0581

Sumber: Diolah oleh Penulis (2026), berdasarkan output eksekusi kode pelatihan model pada Lampiran tesis (variabel losses, dicetak tiap 50 epoch).

Selain menggunakan nilai *training loss*, performa model juga dievaluasi menggunakan *Receiver Operating Characteristic (ROC) Curve*. Kurva ROC digunakan untuk menggambarkan kemampuan model dalam membedakan transaksi legal dan transaksi ilegal pada berbagai nilai ambang (*threshold*). Perbandingan kurva ROC antar model memberikan gambaran mengenai model yang memiliki kemampuan diskriminasi paling baik.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.9 Perbandingan Kurva ROC

Berdasarkan Gambar 4.9, kurva ROC digunakan untuk membandingkan kemampuan masing-masing model dalam membedakan transaksi legal dan transaksi ilegal. Semakin mendekati sudut kiri atas, maka semakin baik kemampuan model dalam melakukan klasifikasi. Garis putus-putus diagonal menunjukkan performa klasifikasi secara acak (*random classifier*), sehingga kurva yang berada jauh di atas garis tersebut menunjukkan bahwa model memiliki kemampuan klasifikasi yang baik.

Hasil pengujian menunjukkan bahwa seluruh model memiliki nilai ROC-AUC di atas 0,97. Nilai ROC-AUC tertinggi diperoleh oleh GAT + Class Weight sebesar 0,9809, diikuti oleh GAT yang memiliki nilai sekitar 0,9800. Perbedaan keduanya relatif kecil, tetapi hasil ini menunjukkan bahwa penambahan Class Weight tetap mampu mempertahankan kemampuan diskriminasi model sekaligus meningkatkan Recall terhadap kelas Illicit.

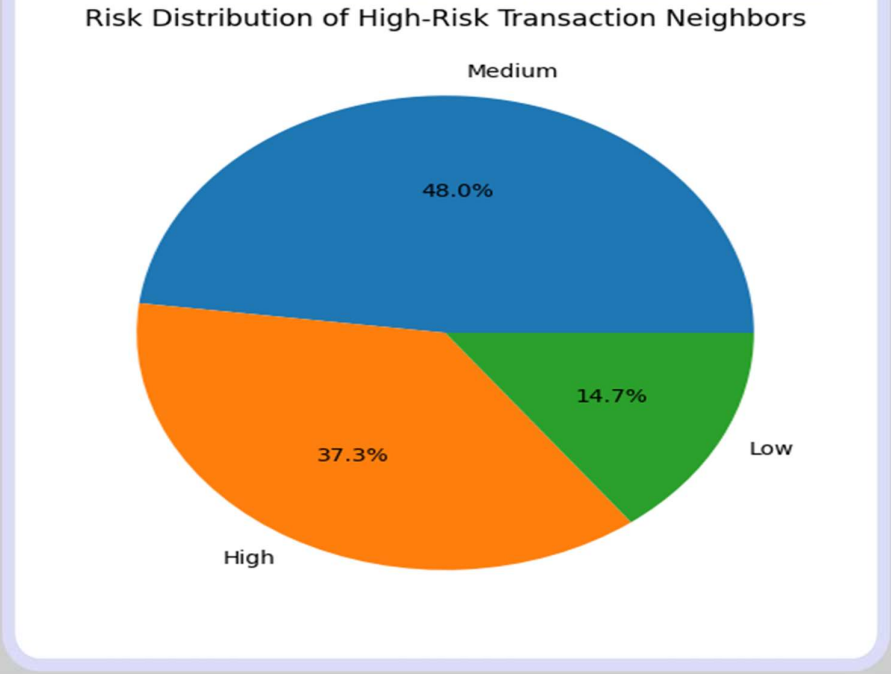
Sementara itu, model SVM memperoleh nilai AUC sebesar 0,9733, sedangkan GCN memperoleh nilai AUC sebesar 0,9709. Kedua model tersebut tetap menunjukkan performa yang sangat baik karena memiliki nilai AUC di atas 0,97. Namun, dibandingkan model berbasis *Graph Neural Network*, kemampuan keduanya dalam membedakan transaksi legal dan ilegal masih sedikit lebih rendah.

Secara keseluruhan, perbedaan nilai AUC antar model relatif kecil, sehingga seluruh model dapat dikatakan memiliki kemampuan klasifikasi yang sangat baik. Akan tetapi, model berbasis *Graph Neural Network*, khususnya GAT dan GAT + *Class Weight*, menunjukkan performa yang lebih unggul karena mampu memanfaatkan informasi struktur *graph* pada jaringan *Bitcoin* secara lebih efektif. Hasil ini mendukung temuan pada evaluasi sebelumnya bahwa penggunaan *Graph Neural Network*, terutama dengan penambahan *Class Weight*, tidak hanya memberikan kemampuan klasifikasi yang baik, tetapi juga meningkatkan keberhasilan dalam mendeteksi transaksi ilegal, yang merupakan tujuan utama penelitian ini.

4.2 Analisis Hasil Deteksi Transaksi *Illicit*

4.2.1 Analisis Karakteristik dan Pola Transaksi *Illicit*

Selain mengevaluasi performa model, penelitian ini juga menganalisis distribusi tingkat risiko *neighbor* pada transaksi *Bitcoin* menggunakan metode yang telah dijelaskan pada subbab 3.10. Analisis ini dilakukan terhadap 5.054 transaksi *neighbor* yang terhubung langsung dengan 4.545 transaksi illicit melalui 8.145 *edge*, untuk mengetahui proporsi transaksi yang termasuk dalam kategori risiko rendah (*Low*), sedang (*Medium*), dan tinggi (*High*). Informasi tersebut memberikan gambaran mengenai tingkat risiko transaksi berdasarkan hubungan dengan transaksi lain dalam jaringan *Bitcoin*. Hasil visualisasi distribusi tingkat risiko *neighbor* dapat dilihat pada Gambar 4.10.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.10 Distribusi Tingkat Risiko *Neighbor*

Gambar 4.10 menunjukkan distribusi tingkat risiko *neighbor* pada transaksi *Bitcoin* berdasarkan hasil analisis terhadap hubungan antar*node* dalam jaringan *Blockchain*. Setiap transaksi dikelompokkan ke dalam tiga kategori risiko, yaitu *High*, *Medium*, dan *Low*, berdasarkan karakteristik *neighbor* yang dimilikinya.

Berdasarkan hasil visualisasi, kategori *Medium* memiliki proporsi terbesar, yaitu sebesar 48,0%. Hasil ini menunjukkan bahwa sebagian besar transaksi berada pada tingkat risiko menengah, sehingga masih memerlukan perhatian lebih lanjut karena berpotensi memiliki hubungan dengan transaksi yang mencurigakan. Kategori *High* menempati urutan kedua dengan persentase 37,3%, yang menunjukkan bahwa lebih dari sepertiga transaksi memiliki tingkat risiko tinggi dan berpotensi terlibat dalam aktivitas ilegal. Sementara itu, kategori *Low* hanya sebesar 14,7%, yang mengindikasikan bahwa jumlah transaksi dengan risiko rendah relatif lebih sedikit dibandingkan dua kategori lainnya.

Distribusi tersebut menunjukkan bahwa sebagian besar transaksi dalam jaringan *Bitcoin* memiliki tingkat risiko menengah hingga tinggi. Kondisi ini mengindikasikan bahwa informasi hubungan antartransaksi (*neighbor*) dapat memberikan kontribusi penting dalam proses deteksi transaksi ilegal. Dengan memanfaatkan struktur *graph*, model tidak hanya menganalisis karakteristik setiap transaksi secara individual, tetapi juga mempertimbangkan keterkaitannya dengan transaksi lain yang berada di sekitarnya. Hasil ini mendukung penggunaan *Graph Neural Network* dalam penelitian ini, karena mampu memanfaatkan informasi struktur jaringan untuk meningkatkan efektivitas deteksi transaksi yang berpotensi bersifat *fraud*.

Tabel 4.6 Transaksi Unknown dengan Jumlah Hubungan Tertinggi terhadap Transaksi Illicit

txId	Time Step	Illicit → Unknown	Total Koneksi
355174807	32	68	68
339502544	13	63	63
139273997	28	59	59
355238405	32	58	58
73160850	42	49	49
246979389	15	36	36
246979279	15	29	29
179039621	9	27	27
73282397	42	25	25
94654151	7	22	22

Berdasarkan hasil analisis hubungan antartransaksi, ditemukan sebanyak 1.458 edge dengan arah dari transaksi Illicit menuju transaksi Unknown dan 3.993 edge dengan arah dari transaksi Unknown menuju transaksi Illicit. Setelah dilakukan penggabungan berdasarkan txId, diperoleh sebanyak 4.027 transaksi Unknown yang memiliki hubungan langsung dengan setidaknya satu transaksi Illicit.

Jumlah hubungan setiap transaksi Unknown dengan transaksi Illicit bervariasi. Hasil statistik menunjukkan bahwa rata-rata transaksi Unknown memiliki sekitar 1,35 hubungan dengan transaksi Illicit, dengan nilai median sebesar 1 hubungan. Hal ini menunjukkan bahwa sebagian besar transaksi Unknown hanya memiliki satu keterhubungan langsung dengan transaksi Illicit. Namun, terdapat beberapa transaksi yang memiliki jumlah hubungan jauh lebih tinggi, bahkan mencapai 68 hubungan. Sebagai contoh, transaksi Unknown dengan txId 355174807 pada time step 32 memiliki 68 hubungan langsung dengan transaksi Illicit. Selain itu, txId 339502544 memiliki 63 hubungan dan txId 139273997 memiliki 59 hubungan. Jumlah hubungan yang tinggi tidak dapat digunakan untuk langsung menyimpulkan bahwa transaksi Unknown tersebut merupakan transaksi ilegal. Namun, pola keterhubungan ini dapat digunakan sebagai indikator struktural untuk menentukan transaksi yang perlu mendapatkan perhatian lebih lanjut dalam proses investigasi digital forensik.

4.2.2 Ringkasan Hasil Deteksi *Fraud*

Setelah seluruh proses pelatihan, pengujian, dan analisis hasil dilakukan, diperoleh beberapa temuan utama yang menjadi ringkasan dari penelitian ini. Ringkasan tersebut disusun berdasarkan hasil evaluasi model menggunakan berbagai metrik, seperti *Accuracy*, *Recall*, *ROC-AUC*, serta hasil *Confusion Matrix*. Ringkasan hasil penelitian disajikan pada Tabel 4.7.

Tabel 4.7 Ringkasan Hasil Deteksi *Fraud*

Aspek Analisis	Hasil
Model Terbaik berdasarkan *Recall*	GAT + *Class Weight*
Model Terbaik berdasarkan *Accuracy*	SVM
Nilai *Recall* Tertinggi	90,43%
Nilai AUC Tertinggi	0,9809
False Negative Terendah	87
True Positive Tertinggi	822

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Berdasarkan Tabel 4.7, dapat disimpulkan bahwa setiap model memiliki keunggulan pada aspek evaluasi yang berbeda. Model SVM memperoleh nilai *Accuracy* tertinggi sebesar 97,36%, yang menunjukkan bahwa model ini memiliki kemampuan terbaik dalam mengklasifikasikan transaksi secara keseluruhan. Namun, nilai *Accuracy* yang tinggi tidak selalu menunjukkan kemampuan terbaik dalam mendeteksi transaksi ilegal, terutama pada dataset yang memiliki distribusi kelas tidak seimbang. Model GAT + *Class Weight* menunjukkan performa terbaik dalam mendeteksi transaksi ilegal dengan memperoleh nilai *recall* tertinggi sebesar 90,43% dan nilai AUC tertinggi sebesar 0,9809. Selain itu, model ini juga menghasilkan jumlah *False Negative* terendah, yaitu sebanyak 87 transaksi, serta *True Positive* tertinggi, yaitu sebanyak 822 transaksi. Hasil tersebut menunjukkan bahwa model GAT + *Class Weight* mampu mengidentifikasi lebih banyak transaksi ilegal dan meminimalkan jumlah transaksi ilegal yang tidak terdeteksi dibandingkan model lainnya.

Secara keseluruhan, hasil ringkasan ini menunjukkan bahwa penggunaan *Graph Attention Network* yang dikombinasikan dengan *Class Weight* memberikan peningkatan kemampuan dalam mendeteksi transaksi *fraud* pada jaringan *Bitcoin*. Meskipun model SVM memiliki akurasi yang sedikit lebih tinggi, model GAT + *Class Weight* lebih sesuai dengan tujuan penelitian karena mampu meningkatkan keberhasilan deteksi transaksi ilegal sekaligus mengurangi jumlah *False Negative*. Oleh karena itu, *GAT + Class Weight* dipilih sebagai model utama dalam penelitian ini karena paling sesuai dengan tujuan penelitian, yaitu meningkatkan kemampuan

deteksi transaksi Illicit. Pemilihan tersebut didasarkan terutama pada Recall tertinggi dan jumlah *False Negative* terendah, bukan karena model tersebut unggul pada seluruh metrik evaluasi..

Tabel 4.8 Hasil Prediksi GAT + Class Weight terhadap Transaksi Unknown

Prediksi	Jumlah	Persentase
Licit	119.100	75,76%
Illicit	38.105	24,24%
Total	157.205	100%

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Setelah model GAT + Class Weight selesai dilatih menggunakan transaksi yang memiliki label Licit dan Illicit, model selanjutnya digunakan untuk melakukan inferensi terhadap 157.205 transaksi Unknown. Berdasarkan hasil prediksi, sebanyak 119.100 transaksi atau 75,76% memiliki pola yang lebih mendekati kelas Licit. Sementara itu, sebanyak 38.105 transaksi atau 24,24% diprediksi memiliki pola yang lebih mendekati kelas Illicit.

Hasil ini menunjukkan bahwa sebagian besar transaksi Unknown memiliki karakteristik yang lebih mendekati transaksi Licit. Namun, terdapat sekitar seperempat dari keseluruhan transaksi Unknown yang memperoleh probabilitas lebih tinggi pada kelas Illicit. Kelompok ini dapat digunakan sebagai hasil penyaringan awal untuk menentukan transaksi yang perlu diperiksa lebih lanjut. Perlu ditegaskan bahwa transaksi Unknown yang diprediksi sebagai Illicit tidak dapat langsung dinyatakan sebagai transaksi ilegal, karena data Unknown tidak memiliki ground truth yang dapat digunakan untuk memastikan label sebenarnya. Prediksi model hanya menunjukkan kecenderungan pola berdasarkan fitur dan hubungan graph yang telah dipelajari dari transaksi Licit dan Illicit.

Tabel 4.9 Ringkasan Analisis Transaksi Unknown untuk Digital Forensik

Hasil Analisis	Jumlah
Total Unknown	157.205
Diprediksi Illicit	38.105
Terhubung langsung dengan Illicit	4.027
Diprediksi Illicit dan terhubung dengan Illicit	**1.942**

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Hasil prediksi model kemudian dikombinasikan dengan analisis struktur graph. Dari 157.205 transaksi Unknown, ditemukan sebanyak 4.027 transaksi yang memiliki hubungan langsung dengan transaksi Illicit. Ketika informasi tersebut digabungkan dengan hasil prediksi GAT + Class Weight, diperoleh sebanyak 1.942 transaksi Unknown yang diprediksi sebagai Illicit dan pada saat yang sama memiliki hubungan langsung dengan transaksi Illicit.

Kelompok sebanyak 1.942 transaksi tersebut dapat dipertimbangkan sebagai kandidat yang memiliki tingkat prioritas lebih tinggi untuk diperiksa lebih lanjut. Hal ini disebabkan karena transaksi tersebut tidak hanya memiliki pola fitur yang menurut model lebih mendekati kelas Illicit, tetapi juga mempunyai keterhubungan langsung dengan transaksi yang telah memiliki label Illicit di dalam graph.

Meskipun demikian, hasil ini tetap bersifat sebagai indikator awal dan bukan bukti final bahwa transaksi tersebut merupakan transaksi ilegal. Dalam konteks digital forensik, hasil model dapat digunakan untuk mempersempit jumlah transaksi yang perlu dianalisis oleh investigator sehingga pemeriksaan dapat dilakukan secara lebih terarah.

Tabel 4.10 Transaksi Unknown dengan Probabilitas Illicit Tertinggi

No.	txId	Time Step	Prob. Licit	Prob. Illicit	Prediksi
1	100832039	35	8,814314E-10	1,000000	Illicit
2	99997463	42	3,701203E-09	1,000000	Illicit
3	386161499	31	5,057577E-08	1,000000	Illicit
4	66665998	38	1,733060E-08	1,000000	Illicit
5	386134758	31	1,105979E-08	1,000000	Illicit
6	277797469	24	1,186307E-08	1,000000	Illicit
7	139227905	28	1,396797E-09	1,000000	Illicit
8	379046510	23	5,008599E-08	1,000000	Illicit
9	139227935	28	5,384553E-08	1,000000	Illicit
10	166521556	29	8,140619E-08	1,000000	Illicit

Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Berdasarkan Tabel 4.10, sepuluh transaksi Unknown dengan nilai probabilitas Illicit tertinggi seluruhnya memperoleh nilai probabilitas yang sangat mendekati 1. Nilai tersebut menunjukkan bahwa berdasarkan pola fitur transaksi yang telah dipelajari oleh model GAT + Class Weight, transaksi-transaksi tersebut memiliki karakteristik yang sangat mendekati pola transaksi Illicit. Sebaliknya, nilai probabilitas pada kelas Licit sangat kecil, yaitu berada pada kisaran 10^{-10} sampai 10^{-8}

Sebagai contoh, transaksi dengan txId 100832039 pada time step 35 memperoleh probabilitas Licit sebesar $8{,}814314 \times 10^{-10}$ dan probabilitas Illicit sebesar 1, sehingga model memprediksi transaksi tersebut sebagai Illicit. Pola yang sama juga terlihat pada sembilan transaksi lainnya yang memiliki probabilitas Illicit sangat tinggi. Meskipun nilai probabilitas model sangat tinggi, hasil tersebut tidak dapat langsung digunakan untuk menyatakan bahwa transaksi Unknown tersebut benar-benar merupakan transaksi ilegal. Hal ini karena transaksi Unknown tidak memiliki label aktual atau ground truth yang dapat digunakan untuk memastikan kelas sebenarnya. Oleh karena itu, hasil prediksi ini lebih tepat digunakan sebagai indikator awal atau skor prioritas untuk membantu menentukan transaksi yang perlu diperiksa lebih lanjut dalam proses digital forensik.

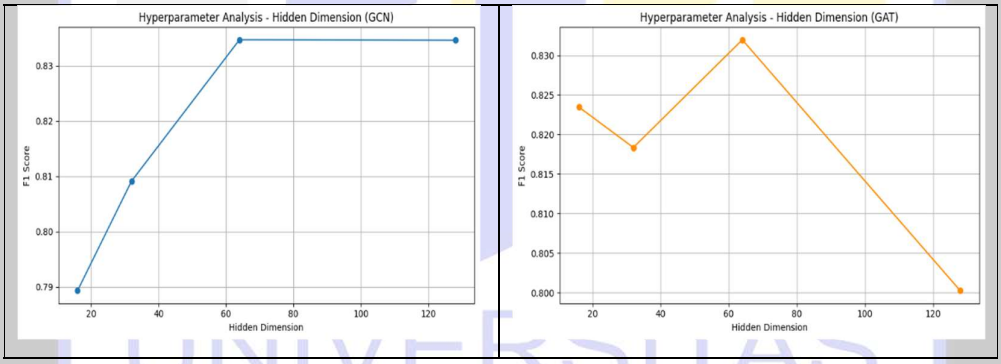
Dalam penelitian ini, hasil probabilitas tersebut kemudian dapat dikombinasikan dengan informasi struktur graph, khususnya hubungan transaksi Unknown dengan

transaksi yang telah berlabel Illicit. Dengan demikian, transaksi yang memiliki probabilitas Illicit tinggi sekaligus mempunyai keterhubungan langsung dengan transaksi Illicit dapat diprioritaskan untuk proses analisis lebih lanjut oleh investigator.

4.3 Analisis Struktur *Graph*

4.3.1 Analisis *Hyperparameter* dan *Embedding* GCN & GAT

Selain mengevaluasi performa model menggunakan berbagai metrik evaluasi, penelitian ini juga melakukan analisis terhadap salah satu *hyperparameter* penting pada model berbasis *graph*, yaitu *hidden dimension*, yang diuji pada model *Graph Convolutional Network (GCN)* dan *Graph Attention Network* (GAT). Analisis ini bertujuan untuk mengetahui pengaruh jumlah *hidden dimension* terhadap kemampuan model dalam mendeteksi transaksi ilegal pada jaringan *Bitcoin*. Hasil analisis ditunjukkan pada Gambar 4.11.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

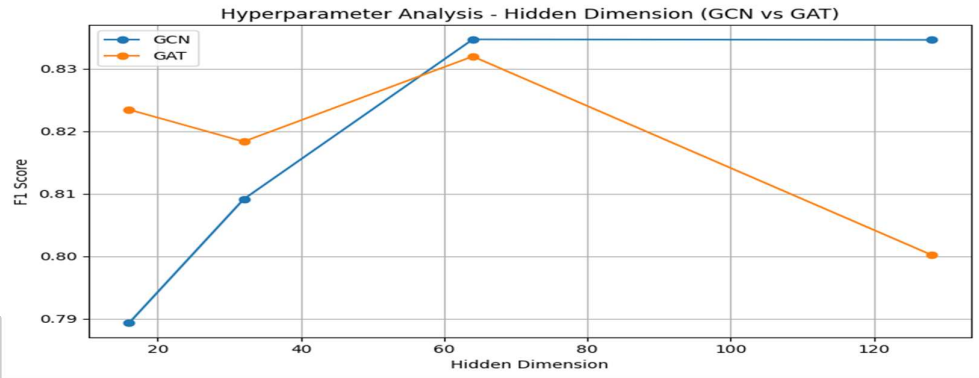
Gambar 4.11 Analisis Pengaruh *Hidden dimension* GCN dan GAT

Berdasarkan Gambar 4.11, dilakukan analisis pengaruh jumlah *hidden dimension* terhadap performa model GCN dan GAT menggunakan metrik *F1-score*. Pengujian dilakukan pada beberapa nilai *hidden dimension*, yaitu 16, 32, 64, dan 128, untuk mengetahui konfigurasi yang memberikan hasil terbaik pada proses deteksi transaksi *fraud*. Pada model GCN, terlihat bahwa nilai *F1-score* mengalami peningkatan seiring bertambahnya jumlah *hidden dimension*. Saat *hidden dimension* sebesar 16, model menghasilkan *F1-score* sekitar 0,79. Nilai tersebut kemudian meningkat menjadi sekitar 0,81 pada *hidden dimension* 32 dan mencapai

sekitar 0,835 pada *hidden dimension* 64. Ketika *hidden dimension* ditingkatkan menjadi 128, nilai *F1-score* tidak mengalami peningkatan yang berarti dan cenderung tetap. Hal ini menunjukkan bahwa penggunaan *hidden dimension* sebesar 64 sudah mampu menghasilkan representasi fitur yang optimal sehingga penambahan dimensi yang lebih besar tidak memberikan peningkatan performa yang signifikan. Sementara itu, pada model GAT terlihat pola yang sedikit berbeda. Nilai *F1-score* mengalami penurunan dari *hidden dimension* 16 ke 32, kemudian meningkat dan mencapai nilai tertinggi sekitar 0,832 pada *hidden dimension* 64. Namun, ketika *hidden dimension* dinaikkan menjadi 128, nilai *F1-score* justru mengalami penurunan yang cukup besar hingga mendekati 0,80. Kondisi ini menunjukkan bahwa penggunaan *hidden dimension* yang terlalu besar pada GAT dapat menyebabkan model menjadi kurang optimal dalam melakukan generalisasi terhadap data pengujian.

Secara keseluruhan, kedua model menunjukkan bahwa *hidden dimension* sebesar 64 merupakan konfigurasi yang memberikan performa terbaik. Hasil ini mengindikasikan bahwa jumlah *hidden dimension* yang terlalu kecil belum mampu menangkap informasi penting dari struktur *graph*, sedangkan penggunaan *hidden dimension* yang terlalu besar tidak selalu meningkatkan performa model dan bahkan dapat menyebabkan penurunan akurasi klasifikasi akibat kompleksitas model yang semakin tinggi. Oleh karena itu, *hidden dimension* sebesar 64 dipilih sebagai konfigurasi terbaik dan digunakan pada proses pelatihan model dalam penelitian ini.

UNIVERSITAS NUSA MANDIRI



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.12 Analisis Pengaruh *Hidden dimension* GCN vs GAT

Berdasarkan Gambar 4.12, dilakukan analisis pengaruh nilai *hidden dimension* terhadap performa model GCN dan GAT menggunakan metrik *F1-score*. *Hidden dimension* merupakan salah satu *hyperparameter* yang menentukan jumlah neuron pada lapisan tersembunyi (hidden layer) dalam proses pembelajaran *Graph Neural Network*. Pengujian dilakukan menggunakan empat nilai *hidden dimension*, yaitu 16, 32, 64, dan 128, untuk mengetahui konfigurasi yang menghasilkan performa terbaik dalam mendeteksi transaksi *fraud* pada jaringan *Bitcoin*. Pada model GCN, terlihat bahwa nilai *F1-score* mengalami peningkatan seiring bertambahnya *hidden dimension*. Saat *hidden dimension* sebesar 16, model menghasilkan *F1-score* sekitar 0,79. Nilai tersebut meningkat menjadi sekitar 0,81 pada *hidden dimension* 32, kemudian mencapai nilai tertinggi sekitar 0,835 pada *hidden dimension* 64. Ketika *hidden dimension* dinaikkan menjadi 128, nilai *F1-score* tidak mengalami peningkatan lagi dan cenderung tetap. Hal ini menunjukkan bahwa *hidden dimension* sebesar 64 sudah mampu menghasilkan representasi fitur yang optimal sehingga penambahan dimensi yang lebih besar tidak memberikan peningkatan performa yang signifikan. Berbeda dengan GCN, model GAT menunjukkan pola yang lebih fluktuatif. Pada *hidden dimension* 16, model memperoleh *F1-score* sekitar 0,823. Nilai tersebut sedikit menurun pada *hidden dimension* 32, kemudian meningkat dan mencapai nilai terbaik sekitar 0,832 pada *hidden dimension* 64. Namun, ketika *hidden dimension* ditingkatkan menjadi 128, nilai *F1-score* mengalami penurunan hingga sekitar 0,80. Hasil ini menunjukkan

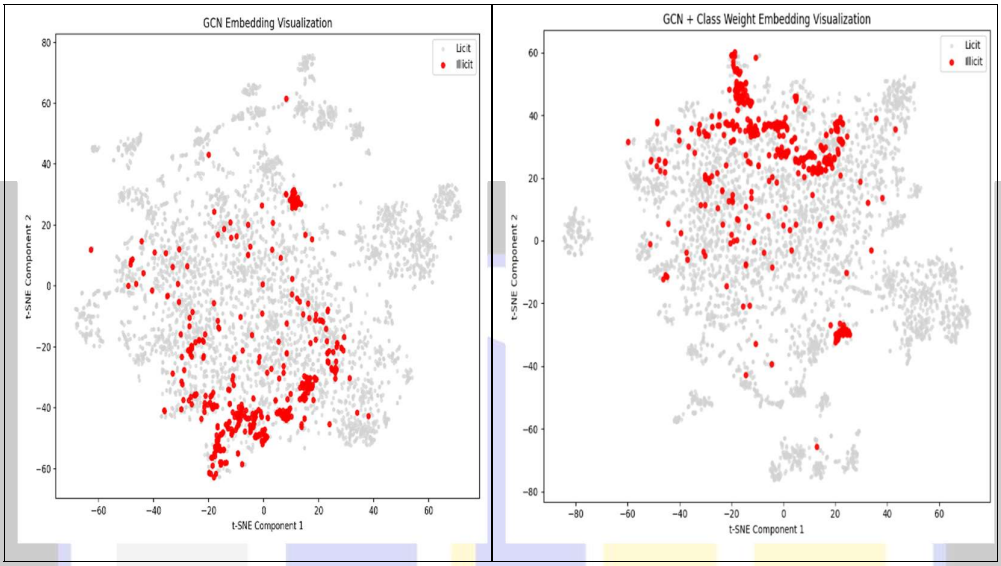
bahwa penggunaan *hidden dimension* yang terlalu besar pada GAT dapat menyebabkan penurunan kemampuan model dalam melakukan generalisasi terhadap data pengujian.

Jika dibandingkan secara keseluruhan, kedua model sama-sama mencapai performa terbaik pada *hidden dimension* 64. Namun, GCN menunjukkan performa yang lebih stabil setelah mencapai nilai tersebut, sedangkan GAT mengalami penurunan performa ketika *hidden dimension* diperbesar menjadi 128. Hal ini menunjukkan bahwa GCN lebih konsisten terhadap perubahan nilai *hidden dimension* dibandingkan GAT. Berdasarkan hasil analisis tersebut, dapat disimpulkan bahwa *hidden dimension* sebesar 64 merupakan konfigurasi yang paling optimal untuk kedua model. Nilai tersebut mampu menghasilkan keseimbangan antara kemampuan model dalam mempelajari representasi fitur pada *graph* dan kompleksitas model, sehingga memberikan performa deteksi transaksi *fraud* yang lebih baik dibandingkan konfigurasi *hidden dimension* lainnya. Oleh karena itu, *hidden dimension* sebesar 64 dipilih sebagai parameter yang digunakan pada proses pelatihan model dalam penelitian ini.

Analisis *hidden dimension* dilakukan pada model dasar GCN dan GAT, karena *Class Weight* hanya memengaruhi bobot pada fungsi loss dan tidak mengubah arsitektur maupun kapasitas representasi model, sehingga hasil *hidden dimension* optimal pada GCN dan GAT juga berlaku untuk varian GCN + *Class Weight* dan GAT + *Class Weight*.

Selain melakukan evaluasi menggunakan metrik *Accuracy*, *Precision*, *recall*, dan *F1-score*, penelitian ini juga melakukan visualisasi *embedding* untuk melihat bagaimana model memetakan representasi setiap transaksi ke dalam ruang fitur *(feature space).* Visualisasi *embedding* bertujuan untuk mengetahui sejauh mana model mampu membedakan transaksi legal (licit) dan transaksi ilegal (illicit) berdasarkan representasi yang dipelajari selama proses pelatihan. Pada visualisasi ini digunakan metode *t-Distributed Stochastic Neighbor Embedding (t-SNE)* untuk mereduksi dimensi fitur sehingga hasil pembelajaran model dapat ditampilkan

dalam bentuk dua dimensi. Perbandingan hasil visualisasi *embedding* antara model GCN dan GCN + *Class Weight* ditunjukkan pada Gambar 4.13.



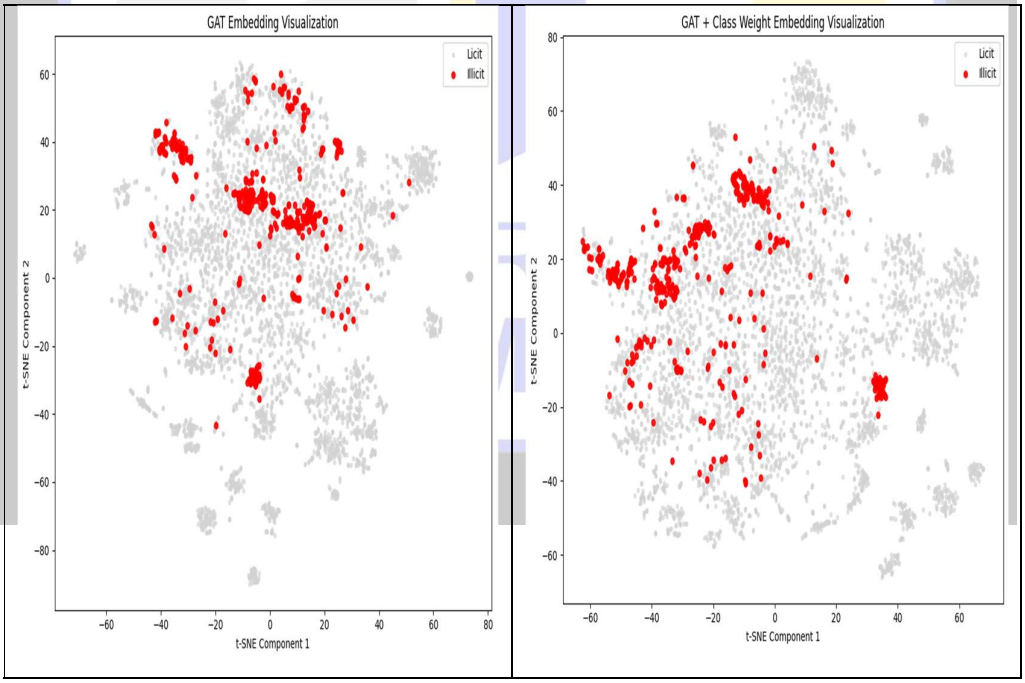
Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.13 Perbandingan Visualisasi *Embedding* Transaksi Menggunakan GCN dan GCN + Class Weight

Berdasarkan Gambar 4.13, terlihat bahwa model GCN dan GCN + *Class Weight* menghasilkan pola distribusi *embedding* yang berbeda. Pada model GCN, titik-titik yang merepresentasikan transaksi ilegal (ditunjukkan dengan warna merah) masih banyak tersebar di antara transaksi legal (warna abu-abu). Kondisi ini menunjukkan bahwa model GCN masih mengalami kesulitan dalam memisahkan karakteristik transaksi ilegal secara lebih jelas, sehingga masih terdapat kemungkinan terjadinya kesalahan klasifikasi. Sementara itu, pada model GCN + *Class Weight* terlihat bahwa transaksi ilegal membentuk kelompok (*cluster*) yang lebih jelas dibandingkan model GCN. Penyebaran titik berwarna merah menjadi lebih terfokus pada beberapa area tertentu, sehingga pemisahan antara transaksi legal dan transaksi ilegal menjadi lebih mudah diamati. Hasil ini menunjukkan bahwa penambahan *Class Weight* membantu model memberikan perhatian yang lebih besar terhadap kelas minoritas, sehingga representasi fitur yang dihasilkan menjadi lebih baik. Meskipun masih terdapat beberapa titik transaksi ilegal yang berada di sekitar kelompok transaksi legal, secara keseluruhan model GCN + *Class Weight*

mampu menghasilkan pemisahan kelas yang lebih baik dibandingkan GCN. Hasil visualisasi ini sejalan dengan hasil evaluasi sebelumnya yang menunjukkan bahwa penambahan *Class Weight* meningkatkan nilai *recall* dan mengurangi jumlah *False Negative*. Dengan demikian, representasi *embedding* yang lebih terstruktur menunjukkan bahwa model lebih mampu mengenali pola transaksi ilegal pada jaringan *Bitcoin*.

Setelah dilakukan analisis pada model GCN, penelitian ini juga membandingkan hasil visualisasi *embedding* yang dihasilkan oleh model GAT dan GAT + *Class Weight*. Visualisasi *embedding* digunakan untuk melihat bagaimana kedua model memetakan representasi setiap transaksi ke dalam ruang fitur setelah proses pelatihan selesai. Dengan menggunakan metode *t-Distributed Stochastic Neighbor Embedding* (t-SNE), representasi berdimensi tinggi direduksi menjadi dua dimensi sehingga pola penyebaran transaksi legal (licit) dan transaksi ilegal (illicit) dapat diamati secara lebih jelas. Perbandingan hasil visualisasi *embedding* pada kedua model ditunjukkan pada Gambar 4.14.



Sumber: Diolah oleh penulis berdasarkan Elliptic Bitcoin Dataset

Gambar 4.14 Perbandingan Visualisasi *Embedding* Transaksi Menggunakan GAT dan GAT + Class Weight

Berdasarkan Gambar 4.14, terlihat bahwa model GAT dan GAT + *Class Weight* mampu menghasilkan representasi *embedding* yang cukup baik dalam membedakan transaksi legal dan transaksi ilegal. Pada model GAT, sebagian besar transaksi ilegal yang ditunjukkan dengan warna merah mulai membentuk beberapa kelompok (*cluster*), meskipun masih terdapat sejumlah titik yang tersebar di antara transaksi legal. Kondisi ini menunjukkan bahwa model telah mampu mengenali pola hubungan antar transaksi, namun pemisahan antarkelas belum sepenuhnya optimal. Sementara itu, pada model GAT + *Class Weight* terlihat bahwa transaksi ilegal membentuk kelompok yang lebih jelas dan penyebarannya lebih terpusat dibandingkan model GAT. Selain itu, jumlah transaksi ilegal yang berada di dalam kelompok transaksi legal juga tampak lebih sedikit. Hal ini menunjukkan bahwa penambahan *Class Weight* membantu model memberikan perhatian yang lebih besar terhadap kelas minoritas sehingga representasi fitur transaksi ilegal menjadi lebih mudah dibedakan dari transaksi legal. Secara keseluruhan, hasil visualisasi *embedding* menunjukkan bahwa model GAT + *Class Weight* menghasilkan pemisahan kelas yang lebih baik dibandingkan model GAT. Temuan ini sejalan dengan hasil evaluasi sebelumnya, di mana GAT + *Class Weight* memperoleh nilai *recall* tertinggi sebesar 90,43% dan jumlah *False Negative* terendah sebanyak 87 transaksi. Dengan demikian, visualisasi *embedding* ini memperkuat hasil evaluasi kuantitatif bahwa penambahan *Class Weight* mampu meningkatkan kemampuan model dalam mengenali pola transaksi *fraud* pada jaringan *Bitcoin*.

4.4 Pemetaan Hasil Penelitian ke dalam Kerangka Investigasi *Digital forensik*

Sebagaimana telah dijelaskan pada subbab 3.11, penelitian ini mengadopsi kerangka kerja forensik digital dari NIST yang terdiri atas empat tahap, yaitu *collection*, *examination*, *analysis*, dan *reporting*. Pada tahap collection, data yang digunakan berasal dari Elliptic Bitcoin Dataset yang terdiri atas 203.769 transaksi, termasuk 157.205 transaksi dengan label Unknown. Seluruh transaksi tetap dipertahankan pada tahap eksplorasi agar struktur jaringan yang dianalisis tetap menggambarkan kondisi dataset secara keseluruhan.

Pada tahap examination, dilakukan pemeriksaan struktur graph, distribusi time step, korelasi fitur, serta hubungan transaksi Unknown dengan transaksi Illicit. Hasil examination menunjukkan bahwa terdapat 4.027 transaksi Unknown yang memiliki hubungan langsung dengan transaksi Illicit. Pada tahap analysis, model GAT + Class Weight digunakan untuk melakukan inferensi terhadap transaksi Unknown. Hasil prediksi menunjukkan bahwa 119.100 transaksi Unknown diprediksi memiliki pola lebih mendekati Licit, sedangkan 38.105 transaksi diprediksi lebih mendekati Illicit. Hasil prediksi tersebut kemudian digabungkan dengan analisis hubungan graph. Pada tahap reporting, ditemukan sebanyak 1.942 transaksi Unknown yang memenuhi dua indikator, yaitu diprediksi sebagai Illicit oleh model dan memiliki hubungan langsung dengan transaksi Illicit. Transaksi tersebut dapat dimasukkan dalam daftar kandidat untuk pemeriksaan lanjutan oleh investigator. Namun, hasil ini tidak digunakan sebagai bukti bahwa transaksi tersebut pasti ilegal, melainkan sebagai informasi pendukung dalam proses penentuan prioritas investigasi.

UNIVERSITAS
NUSA MANDIRI

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil mengembangkan model deteksi transaksi ilegal pada jaringan Bitcoin menggunakan pendekatan berbasis Graph Neural Network dengan memanfaatkan Elliptic Bitcoin Dataset. Dataset yang digunakan terdiri atas 203.769 transaksi dan 234.355 hubungan antartransaksi (edge), dengan 42.019 transaksi berlabel Licit, 4.545 transaksi berlabel Illicit, dan 157.205 transaksi berlabel Unknown. Struktur graph memungkinkan hubungan antartransaksi ikut dianalisis sehingga informasi yang digunakan tidak hanya berasal dari fitur setiap transaksi secara individual.

2. Hasil perbandingan menunjukkan bahwa model berbasis graph memberikan peningkatan terutama pada kemampuan mendeteksi kelas Illicit. SVM memperoleh Accuracy tertinggi sebesar 97,36%, tetapi Recall SVM sebesar 78,22% merupakan yang terendah. Sebaliknya, beberapa model berbasis graph menghasilkan Recall yang lebih tinggi, yang menunjukkan bahwa pemanfaatan hubungan antartransaksi dapat membantu mengurangi jumlah transaksi Illicit yang tidak terdeteksi.

3. Penerapan Class Weight mampu meningkatkan kemampuan model dalam mendeteksi kelas minoritas, yaitu transaksi Illicit. Pada GCN maupun GAT, penggunaan Class Weight meningkatkan nilai Recall. Hal ini menunjukkan bahwa pemberian bobot yang lebih besar terhadap kelas minoritas dapat mengurangi kecenderungan model untuk lebih dominan mengenali kelas mayoritas dan membantu menurunkan jumlah transaksi Illicit yang tidak berhasil terdeteksi.

4. Berdasarkan hasil evaluasi, GAT + Class Weight menghasilkan nilai Recall tertinggi sebesar 90,43%, ROC-AUC sebesar 0,9809, False Negative sebanyak 87

transaksi, dan True Positive sebanyak 822 transaksi. Hasil tersebut menunjukkan bahwa GAT + Class Weight mampu mendeteksi lebih banyak transaksi Illicit dibandingkan model lainnya. Oleh karena itu, model ini dipilih sebagai model utama untuk melakukan analisis lebih lanjut terhadap transaksi Unknown.

5. Implementasi GAT + Class Weight terhadap 157.205 transaksi Unknown menghasilkan 119.100 transaksi atau 75,76% yang diprediksi memiliki pola lebih mendekati kelas Licit, sedangkan 38.105 transaksi atau 24,24% diprediksi memiliki pola lebih mendekati kelas Illicit. Hasil prediksi tersebut menunjukkan bahwa model dapat digunakan untuk melakukan penyaringan awal terhadap transaksi yang belum memiliki label. Namun, transaksi Unknown yang diprediksi sebagai Illicit tidak dapat langsung dinyatakan sebagai transaksi ilegal karena tidak memiliki ground truth yang dapat digunakan untuk memastikan kelas sebenarnya.

6. Berdasarkan analisis struktur graph, ditemukan sebanyak 4.027 transaksi Unknown yang mempunyai hubungan langsung dengan transaksi berlabel Illicit. Ketika hasil tersebut dikombinasikan dengan hasil prediksi GAT + Class Weight, diperoleh sebanyak 1.942 transaksi Unknown yang diprediksi sebagai Illicit sekaligus memiliki hubungan langsung dengan transaksi Illicit. Kelompok transaksi tersebut dapat digunakan sebagai kandidat dengan prioritas lebih tinggi untuk pemeriksaan lebih lanjut karena memenuhi dua indikator, yaitu indikator berdasarkan hasil prediksi model dan indikator berdasarkan keterhubungan pada struktur graph.

7. Dalam konteks digital forensik, hasil penelitian dapat digunakan sebagai alat bantu (decision support) untuk membantu investigator melakukan penyaringan dan menentukan prioritas transaksi yang perlu diperiksa lebih lanjut. Hasil penelitian telah dipetakan ke dalam tahapan collection, examination, analysis, dan reporting. Pada tahap analysis, hasil prediksi transaksi Unknown dikombinasikan dengan hubungan pada struktur graph, sedangkan pada tahap reporting, 1.942 transaksi yang memenuhi kedua indikator dapat dijadikan kandidat untuk pemeriksaan lanjutan. Hasil tersebut bukan merupakan bukti final bahwa transaksi tersebut

ilegal, tetapi merupakan informasi pendukung untuk membantu proses investigasi digital forensik.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan sebagai pengembangan pada penelitian selanjutnya:

1. Penelitian selanjutnya dapat menggunakan dataset Blockchain yang lebih besar dan lebih beragam untuk menguji kemampuan generalisasi model pada karakteristik transaksi dan jaringan yang berbeda.
2. Pengembangan model dapat dilakukan dengan membandingkan GAT dengan arsitektur Graph Neural Network lainnya, seperti GraphSAGE, Graph Isomorphism Network (GIN), atau Dynamic Graph Neural Network, untuk mengetahui metode yang paling sesuai dalam mendeteksi transaksi ilegal.
3. Optimasi hyperparameter dapat dikembangkan menggunakan metode yang lebih sistematis, seperti Grid Search, Random Search, atau Bayesian Optimization, sehingga konfigurasi model tidak hanya ditentukan berdasarkan beberapa percobaan parameter.
4. Penanganan ketidakseimbangan kelas dapat dikembangkan menggunakan metode lain seperti Focal Loss, oversampling, atau pendekatan cost-sensitive learning lainnya, kemudian dibandingkan dengan Class Weight untuk mengetahui pengaruhnya terhadap Recall, Precision, dan jumlah False Negative.
5. Analisis terhadap transaksi Unknown dapat dikembangkan lebih lanjut dengan menggabungkan probabilitas prediksi model, jumlah keterhubungan dengan transaksi Illicit, neighbor risk score, dan informasi temporal. Penggabungan beberapa indikator tersebut diharapkan dapat menghasilkan sistem penilaian risiko yang lebih terukur untuk menentukan prioritas transaksi yang perlu dianalisis lebih lanjut.

6. Transaksi Unknown yang diprediksi sebagai Illicit perlu divalidasi menggunakan sumber data atau informasi tambahan apabila tersedia. Hal ini penting karena transaksi Unknown tidak memiliki ground truth, sehingga hasil prediksi model pada

penelitian ini belum dapat digunakan sebagai bukti bahwa transaksi tersebut benar-benar merupakan transaksi ilegal.

7. Penelitian selanjutnya dapat mengembangkan model menjadi sistem deteksi atau decision support system yang dapat digunakan oleh investigator untuk menampilkan transaksi mencurigakan, probabilitas prediksi, hubungan antartransaksi, dan tingkat risiko dalam satu sistem analisis.

8. Pengembangan lebih lanjut dapat menggunakan data transaksi yang lebih dinamis atau mendekati kondisi real-time sehingga perubahan pola transaksi dari waktu ke waktu dapat dianalisis. Hal tersebut dapat memperluas penerapan model dari eksperimen menggunakan dataset statis menuju sistem pendukung investigasi yang lebih mendekati kondisi operasional.

UNIVERSITAS
NUSA MANDIRI

DAFTAR REFERENSI

[1] M. Venturini, D. García-Costa, E. Álvarez-García, F. Grimaldo, and F. Squazzoni, “Mapping network structures and dynamics of decentralised cryptocurrencies: The evolution of Bitcoin (2009–2023),” *PLoS One*, vol. 20, no. 8 August, Aug. 2025, doi: 10.1371/journal.pone.0329617.

[2] H. F. Atlam, N. Ekuri, M. A. Azad, and H. S. Lallie, “Blockchain Forensics: A Systematic Literature Review of Techniques, Applications, Challenges, and Future Directions,” Sep. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics13173568.

[3] Y. Elmougy and L. Liu, “Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2023, pp. 3979–3990. doi: 10.1145/3580305.3599803.

[4] A. Toha, P. Purwono, and W. Gata, “Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12–21, Apr. 2022, doi: 10.12928/biste.v4i1.6079.

[5] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, “A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions,” *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-023-00876-4.

[6] I. Alarab and S. Prakoonwit, “Robust recurrent graph convolutional network approach based sequential prediction of illicit transactions in cryptocurrencies,” *Multimed. Tools Appl.*, vol. 83, no. 20, pp. 58449–58464, Jun. 2024, doi: 10.1007/s11042-023-17323-4.

[7] B. Fu, Y. Wang, and T. Feng, “CT-GCN+: a high-performance cryptocurrency transaction graph convolutional model for phishing node classification,” *Cybersecurity*, vol. 7, no. 1, Dec. 2024, doi: 10.1186/s42400-023-00194-5.

[8] A. Asiri and K. Somasundaram, “Graph convolution network for fraud detection in bitcoin transactions,” *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-95672-w.

[9] C. Guo, S. Zhang, P. Zhang, M. Alkubati, and J. Song, “LB-GLAT: Long-Term Bi-Graph Layer Attention Convolutional Network for Anti-Money Laundering in Transactional Blockchain,” *Mathematics*, vol. 11, no. 18, Sep. 2023, doi: 10.3390/math11183927.

[10] H. Song, Y. Wei, Z. Qu, and W. Wang, “Unveiling Decentralization: A Comprehensive Review of Technologies, Comparison, Challenges in Bitcoin, Ethereum, and Solana Blockchain,” Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.04841

[11] J. Nicholls, A. Kuppa, and N. A. Le-Khac, “The next phase of identifying illicit activity in Bitcoin,” *International Journal of Network Management*, vol. 34, no. 5, Sep. 2024, doi: 10.1002/nem.2259.

[12] O. A. Ekle and W. Eberle, “Anomaly Detection in Dynamic Graphs: A Comprehensive Survey,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 8, Jul. 2024, doi: 10.1145/3669906.

[13] S. A. Alsaif, “Machine Learning-Based Ransomware Classification of Bitcoin Transactions,” *Applied Computational Intelligence and Soft Computing*, vol. 2023, 2023, doi: 10.1155/2023/6274260.

[14] N. Pocher, M. Zichichi, F. Merizzi, M. Z. Shafiq, and S. Ferretti, “Detecting anomalous cryptocurrency transactions: An AML/CFT application of machine learning-based forensics,” *Electronic Markets*, vol. 33, no. 1, Dec. 2023, doi: 10.1007/s12525-023-00654-3.

[15] I. Alarab and S. Prakoonwit, “Graph-Based LSTM for Anti-money Laundering: Experimenting Temporal Graph Convolutional Network with Bitcoin Data,” *Neural Process. Lett.*, vol. 55, no. 1, pp. 689–707, Feb. 2023, doi: 10.1007/s11063-022-10904-8.

[16] M. Li, L. Jia, and X. Q. Su, “Global-local graph attention with cyclic pseudo-labels for bitcoin anti-money laundering detection,” *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-08365-9.

[17] L. Garin and V. Gisin, “Machine learning in classifying bitcoin addresses,” *Journal of Finance and Data Science*, vol. 9, Nov. 2023, doi: 10.1016/j.jfds.2023.100109.

[18] V. T. Hoang, H. J. Jeon, E. S. You, Y. Yoon, S. Jung, and O. J. Lee, “Graph Representation Learning and Its Applications: A Survey,” Apr. 01, 2023, *MDPI*. doi: 10.3390/s23084168.

[19] F. Guan, T. Zhu, W. Zhou, and K. K. R. Choo, “Graph neural networks: a survey on the links between privacy and security,” *Artif. Intell. Rev.*, vol. 57, no. 2, Feb. 2024, doi: 10.1007/s10462-023-10656-4.

[20] Y. Pan, “Enhancing predictive models for illicit activities in the Bitcoin transaction network using advanced graph analytical techniques,” *Applied and Computational Engineering*, vol. 48, no. 1, pp. 78–86, Mar. 2024, doi: 10.54254/2755-2721/48/20241181.

[21] H. Adloori, V. Dasanapu, and A. C. Mergu, “Graph Network Models To Detect Illicit Transactions In Block Chain,” Sep. 2024, [Online]. Available: http://arxiv.org/abs/2410.07150

[22] J. M. H. Pinheiro *et al.*, “The Impact of Feature Scaling in Machine Learning: Effects on Regression and Classification Tasks,” *IEEE Access*, vol. 13, pp. 199903–199931, 2025, doi: 10.1109/ACCESS.2025.3635541.

[23] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, “Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance,” *Technologies (Basel).*, vol. 9, no. 3, Sep. 2021, doi: 10.3390/technologies9030052.

[24] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, “A survey on imbalanced learning: latest research, applications and future directions,” *Artif. Intell. Rev.*, vol. 57, no. 6, Jun. 2024, doi: 10.1007/s10462-024-10759-6.

[25] L. Xu, H. Zhu, and J. Chen, “Imbalanced graph learning via mixed entropy minimization,” *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-75999-6.

[26] M. Markwald and E. Demidova, “REFUEL: rule extraction for imbalanced neural node classification,” *Mach. Learn.*, vol. 113, no. 9, pp. 6227–6246, Sep. 2024, doi: 10.1007/s10994-024-06569-0.

[27] P. Gao, Z. Li, D. Zhou, and L. Zhang, “Reinforced Cost-Sensitive Graph Network for Detecting Fraud Leaders in Telecom Fraud,” *IEEE Access*, vol. 12, pp. 173638–173646, 2024, doi: 10.1109/ACCESS.2024.3448260.

[28] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, “Survey of Optimization Algorithms in Modern Neural Networks,” Jun. 01, 2023, *MDPI*. doi: 10.3390/math11112466.

[29] J. W. Shim, “Enhancing cross entropy with a linearly adaptive loss function for optimized classification performance,” *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-024-78858-6.

[30] R. Connor, A. Dearle, B. Claydon, and L. Vadicamo, “Correlations of Cross-Entropy Loss in Machine Learning,” *Entropy*, vol. 26, no. 6, Jun. 2024, doi: 10.3390/e26060491.

[31] M. Owusu-Adjei, J. Ben Hayfron-Acquah, T. Frimpong, and G. Abdul-Salaam, “Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems,” *PLOS Digital Health*, vol. 2, no. 11 November, Nov. 2023, doi: 10.1371/journal.pdig.0000290.

[32] J. Opitz, “A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice”, doi: 10.1162/tacl.

[33] D. Chicco and G. Jurman, “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification,” *BioData Min.*, vol. 16, no. 1, Dec. 2023, doi: 10.1186/s13040-023-00322-4.

[34] M. Salmi, D. Atif, D. Oliva, A. Abraham, and S. Ventura, “Handling imbalanced medical datasets: review of a decade of research,” *Artif. Intell. Rev.*, vol. 57, no. 10, Oct. 2024, doi: 10.1007/s10462-024-10884-2.

[35] M. Carpita and S. Golia, “Categorical classifiers in multiclass classification with imbalanced datasets,” *Stat. Anal. Data Min.*, vol. 16, no. 4, pp. 391–405, Aug. 2023, doi: 10.1002/sam.11624.

[36] M. I. Nugraha and A. Armansyah, “Penutupan Kompetensi Keahlian SMK dengan Pendekatan Klasifikasi Minat Siswa Menggunakan Jaringan Syaraf Tiruan,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 6, no. 3, pp. 269–275, Jul. 2023, doi: 10.32493/jtsi.v6i3.30143.

[37] J. Khatib Sulaiman, D. Safitri, and T. Arita Fitri, “Perbandingan Algoritma XGBoost dan SVM Dalam Analisis Opini Publik Pemilihan Presiden 2024,” *Indonesian Journal of Computer Science*.

[38] R. M. Dhodapkar, E. Li, K. Nwanyanwu, R. Adelman, S. Krishnaswamy, and J. C. Wang, “Deep learning for quality assessment of optical coherence tomography angiography images,”

Sci. Rep., vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-17709-8.

[39] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment," *IEEE Access*, vol. 9, pp. 78368–78381, 2021, doi: 10.1109/ACCESS.2021.3084050.

[40] P. Christen, D. J. Hand, and N. Kirielle, "A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives," Mar. 31, 2024, *Association for Computing Machinery*. doi: 10.1145/3606367.

[41] S. - and A. -, "Klasifikasi Status Kemiskinan Menggunakan Algoritma Random Forest," *EIGEN MATHEMATICS JOURNAL*, pp. 37–44, Jun. 2022, doi: 10.29303/emj.v5i1.133.

[42] S. Jung, T. Dagobert, J. M. Morel, and G. Facciolo, "A Review of t-SNE," *Image Processing On Line*, vol. 14, pp. 250–270, 2024, doi: 10.5201/ipol.2024.528.

[43] S. Ouyang, Q. Bai, H. Feng, and B. Hu, "Bitcoin Money Laundering Detection via Subgraph Contrastive Learning," *Entropy*, vol. 26, no. 3, Mar. 2024, doi: 10.3390/e26030211.

[44] Z. Li and E. He, "Graph Neural Network-Based Bitcoin Transaction Tracking Model," *IEEE Access*, vol. 11, pp. 62109–62120, 2023, doi: 10.1109/ACCESS.2023.3288026.

[45] D. Cheng, Y. Ye, S. Xiang, Z. Ma, Y. Zhang, and C. Jiang, "Anti-Money Laundering by Group-Aware Deep Graph Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12444–12457, Dec. 2023, doi: 10.1109/TKDE.2023.3272396.

[46] F. Wan and P. Li, "A Novel Money Laundering Prediction Model Based on a Dynamic Graph Convolutional Neural Network and Long Short-Term Memory," *Symmetry (Basel).*, vol. 16, no. 3, Mar. 2024, doi: 10.3390/sym16030378.

[47] W. W. Lo, G. K. Kulatilleke, M. Sarhan, S. Layeghy, and M. Portmann, "Inspection-L: Self-Supervised GNN Node Embeddings for Money Laundering Detection in Bitcoin," Oct. 2022, [Online]. Available: http://arxiv.org/abs/2203.10465

[48] J. Liu *et al.*, “Graph Embedding-Based Money Laundering Detection for Ethereum,” *Electronics (Switzerland)*, vol. 12, no. 14, Jul. 2023, doi: 10.3390/electronics12143180.

[49] Q. Wang, W. T. Tsai, and B. Du, “RMGANets: reinforcement learning-enhanced multi-relational attention graph-aware network for anti-money laundering detection,” *Complex and Intelligent Systems*, vol. 11, no. 1, Jan. 2025, doi: 10.1007/s40747-024-01615-9.

[50] Z. Lin *et al.*, “Detecting illicit transactions in bitcoin: a wavelet-temporal graph transformer approach for anti-money laundering,” *Sci. Rep.*, vol. 16, no. 1, Dec. 2026, doi: 10.1038/s41598-025-23901-3.

[51] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, “Graph Attention Networks: A Comprehensive Review of Methods and Applications,” Sep. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/fi16090318.

[52] K. Wang, Y. Ding, and S. C. Han, “Graph neural networks for text classification: a survey,” *Artif. Intell. Rev.*, vol. 57, no. 8, Aug. 2024, doi: 10.1007/s10462-024-10808-0.

[53] K. H. Almotairi, “Towards partial autonomy of operation and maintenance of unreliable equipment,” *Heliyon*, vol. 9, no. 9, Sep. 2023, doi: 10.1016/j.heliyon.2023.e20055.

[54] M. C. Hinojosa Lee, J. Braet, and J. Springael, “Performance Metrics for Multilabel Emotion Classification: Comparing Micro, Macro, and Weighted F1-Scores,” *Applied Sciences (Switzerland)*, vol. 14, no. 21, Nov. 2024, doi: 10.3390/app14219863.

UNIVERSITAS NUSA MANDIRI

Lampiran 1. Daftar Riwayat Hidup Peneliti

DAFTAR RIWAYAT HIDUP

Nama Lengkap	: Lusindah Erdian Nova
NIM	: 14240010
Program Studi	: Informatika
Fakultas	: Teknologi Informasi
Jenjang	: Strata Dua (S2)
Peminatan	: Artificial Intelligence dan *Blockchain*
Tempat Tinggal	: DKI Jakarta, Indonesia
Email	: 14240010@nusamandiri.ac.id

RIWAYAT PENDIDIKAN

1. D3 Universitas Bina Sarana Informatika, Program Studi System Informasi, 2003-2006
2. S1 Universitas Nusa Mandiri, Program Studi System Informasi, 2012-2013
3. S2 Universitas Persada Indonesia Y.A.I, Program Studi Magister Manajemen, 2014–2016.
4. S2 Universitas Nusa Mandiri, Program Studi Informatika, 2024–2026.

PENGALAMAN KERJA

1. Operation Area Manager, PT. Tunas Berkat Semesta, 2017–2025
2. Dosen International Bussines, STIE Swadaya, 2016-2017
3. Shift Leader, PT. Sari Melati Kencana, 2011–2015.
4. Store Manager, PT. Perintis Pelayanan Paripurna, 2008–2011.



Jakarta, Agustus 2026

Lusindah Erdian Nova

Lampiran 2. Hasil Cek Plagiarisme



GRAPH ATTENTIVE CONVOLUTIONAL NETWORKS DALAM DETEKSI TRANSAKSI ILEGAL PADA JARINGAN BITCOIN UNTUK KEBUTUHAN DIGITAL FORENSIK

ORIGINALITY REPORT

25%	19%	17%	17%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	123dok.com Internet Source	2%
2	repository.bsi.ac.id Internet Source	1%
3	Submitted to UPN Veteran Yogyakarta Student Paper	<1%
4	Submitted to Universitat Oberta de Catalunya Student Paper	<1%
5	repository.unissula.ac.id Internet Source	<1%
6	Submitted to Berlin School of Business and Innovation Student Paper	<1%
7	Submitted to Universitas Putera Batam Student Paper	<1%
8	repository.ubpkarawang.ac.id Internet Source	<1%
9	kc.umn.ac.id Internet Source	<1%
10	dokumen.tips Internet Source	<1%
11	repository.ub.ac.id Internet Source	<1%
12	arxiv.org Internet Source	<1%
13	Submitted to Universitas Islam Riau Student Paper	<1%
14	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1%
	Submitted to Universitas Sangga Buana YPKP	

Lampiran 3. Daftar Publikasi Yang Telah Dihasilkan

Jurnal Terakreditasi Sinta 4

1. "Real-Time Road Damage Detection on Mobile Devices using TensorF*Low* Lite and Teachable Machine",2025, MALCOM (Indonesian Journal of *Machine learning* and Computer Science), Juli 2025, Pekanbaru, DOI: https://doi.org/10.57152/malcom.v5i3.1933, ISSN 2775-8575 (Online), 2797-2313 (Print), Publisher Institut Riset dan Publikasi Indonesia (IRPI), url: https://journal.irpi.or.id/index.php/malcom

IEEE Xplore (ICITRI)

1. "Automatic CRUD Code Generation from UML Models for *Low*-Code Platforms: A Model-Driven Engineering Approach", 2025, ICITRI (International Conference on Information Technology Research and Innovation), September 2025, Jakarta, DOI: 10.1109/ICITRI67507.2025.11232843, Publisher IEEE, url: https://ieeexplore.ieee.org/document/11232843

Lampiran 4. Code Hasil Penelitian

A. Persiapan Library dan Upload Dataset

```
# Instalasi library utama
!pip install torch-geometric -q

import os
import zipfile
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.nn.functional as F

from google.colab import files
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.utils.class_weight import compute_class_weight
from sklearn.metrics import (
    confusion_matrix, accuracy_score, precision_score,
    recall_score, f1_score, roc_auc_score, roc_curve, auc
)
from torch_geometric.data import Data
from torch_geometric.nn import GCNConv, GATConv
```

```
SEED = 42
random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(SEED)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print('Device:', device)

# Upload tiga file Elliptic Bitcoin Dataset:
# 1) elliptic_txs_features.zip
# 2) elliptic_txs_classes.csv
# 3) elliptic_txs_edgelist.csv
uploaded = files.upload()

# Ekstrak file features jika masih berbentuk ZIP
zip_files = [f for f in os.listdir('/content') if f.lower().endswith('.zip')]
for z in zip_files:
    with zipfile.ZipFile(os.path.join('/content', z), 'r') as zip_ref:
        zip_ref.extractall('/content')

print('File pada /content:')
print(os.listdir('/content'))
```

B. Membaca Dataset dan Memberi Nama Kolom

```
# Sesuaikan nama file dengan file yang terunggah di Google Colab
all_files = os.listdir('/content')
```

```
FEATURE_PATH = next(os.path.join('/content', f) for f in all_files
                    if 'features' in f.lower() and f.lower().endswith('.csv'))
CLASS_PATH = next(os.path.join('/content', f) for f in all_files
                  if 'classes' in f.lower() and f.lower().endswith('.csv'))
EDGE_PATH = next(os.path.join('/content', f) for f in all_files
                 if 'edgelist' in f.lower() and f.lower().endswith('.csv'))
features = pd.read_csv(FEATURE_PATH, header=None)
classes = pd.read_csv(CLASS_PATH)
edges_all = pd.read_csv(EDGE_PATH)

# 167 kolom = txId + Time_step + 165 fitur
# 165 fitur terdiri atas 93 local features dan 72 aggregate features
column_names = (
    ['txId', 'Time_step']
    + [f'Local_feature_{i}' for i in range(1, 94)]
    + [f'Aggregate_feature_{i}' for i in range(1, 73)]
)
features.columns = column_names

# Standarisasi nama kolom edge
edges_all.columns = ['txId1', 'txId2']

# Normalisasi label agar konsisten
classes['class'] = classes['class'].astype(str).str.strip().str.lower()
classes['class'] = classes['class'].replace({'unknown':'unknown', '1':'1', '2':'2'})

df_all = features.merge(classes[['txId','class']], on='txId', how='left')
df_all['label_name'] = df_all['class'].map({
```

```
   '1':'Illicit', '2':'Licit', 'unknown':'Unknown'
})

print('Features :', features.shape)
print('Classes  :', classes.shape)
print('Edges    :', edges_all.shape)
print('Merged   :', df_all.shape)
print(df_all['label_name'].value_counts())
```

Output utama dataset

Jumlah transaksi : 203.769

Licit : 42.019

Illicit : 4.545

Unknown : 157.205

Jumlah edge : 234.355

Jumlah fitur : 165

C. Preprocessing Data Berlabel (Licit dan Illicit)

```
feature_cols = (
   [f'Local_feature_{i}' for i in range(1, 94)]
   + [f'Aggregate_feature_{i}' for i in range(1, 73)]
)

# Unknown tidak digunakan sebagai target training/evaluasi
# karena tidak memiliki ground truth.
df_known = df_all[df_all['class'] != 'unknown'].copy()
df_known['target'] = df_known['class'].map({'2':0, '1':1})
df_known.reset_index(drop=True, inplace=True)

indices = np.arange(len(df_known))
```

```
train_idx, test_idx = train_test_split(
    indices,
    test_size=0.20,
    random_state=SEED,
    stratify=df_known['target']
)

X_known = df_known[feature_cols].to_numpy(dtype=np.float32)
y_known = df_known['target'].to_numpy(dtype=np.int64)

# Scaling hanya di-fit pada data training untuk mencegah data leakage
scaler = StandardScaler()
scaler.fit(X_known[train_idx])
X_known_scaled = scaler.transform(X_known)

print('Total labeled :', len(df_known))
print('Training      :', len(train_idx))
print('Testing       :', len(test_idx))
print('Shape fitur   :', X_known_scaled.shape)
print(df_known['target'].value_counts())
```

Output preprocessing

Total data berlabel : 46.564

Training : 37.251

Testing : 9.313

Licit (0) : 42.019

Illicit (1) : 4.545

D. Baseline Support Vector Machine (SVM)

```
# SVM menggunakan fitur transaksi secara individual (tanpa edge graph)
```

```
X_train = X_known_scaled[train_idx]
X_test  = X_known_scaled[test_idx]
y_train = y_known[train_idx]
y_test  = y_known[test_idx]

svm = SVC(kernel='rbf', random_state=SEED, probability=True)
svm.fit(X_train, y_train)

svm_pred = svm.predict(X_test)
svm_prob = svm.predict_proba(X_test)[:, 1]

svm_cm = confusion_matrix(y_test, svm_pred, labels=[0,1])
print('Confusion Matrix SVM:')
print(svm_cm)
print('Accuracy :', accuracy_score(y_test, svm_pred))
print('Precision:', precision_score(y_test, svm_pred))
print('Recall   :', recall_score(y_test, svm_pred))
print('F1-score :', f1_score(y_test, svm_pred))
print('ROC-AUC  :', roc_auc_score(y_test, svm_prob))
```

Output SVM yang digunakan pada Bab IV

Accuracy : 97,36%

Recall : 78,22%

FN : 198

TP : 711

E. Pembentukan Graph untuk Model GCN dan GAT

```
known_mapping = {txid: idx for idx, txid in enumerate(df_known['txId'])}

edges_known = edges_all[
```

```
    edges_all['txId1'].isin(known_mapping)
    & edges_all['txId2'].isin(known_mapping)
].copy()

edges_known['source'] = edges_known['txId1'].map(known_mapping)
edges_known['target'] = edges_known['txId2'].map(known_mapping)

edge_index_known = torch.tensor(
    edges_known[['source','target']].values.T,
    dtype=torch.long
)

x_known_tensor = torch.tensor(X_known_scaled, dtype=torch.float)
y_known_tensor = torch.tensor(y_known, dtype=torch.long)

train_mask = torch.zeros(len(df_known), dtype=torch.bool)
test_mask = torch.zeros(len(df_known), dtype=torch.bool)
train_mask[train_idx] = True
test_mask[test_idx] = True

data_known = Data(
    x=x_known_tensor,
    edge_index=edge_index_known,
    y=y_known_tensor
)
data_known.train_mask = train_mask
data_known.test_mask = test_mask
data_known = data_known.to(device)
```

```
print('Known nodes:', len(df_known))
print('Known edges:', len(edges_known))
print(data_known)
```

Output graph data berlabel

Known nodes : 46.564

Known edges : 36.624

Data(x=[46564, 165], edge_index=[2, 36624], y=[46564])

F. Definisi Model GCN dan GAT

```
class GCNModel(torch.nn.Module):
    def __init__(self, input_dim, hidden_dim=64, output_dim=2):
        super().__init__()
        self.conv1 = GCNConv(input_dim, hidden_dim)
        self.conv2 = GCNConv(hidden_dim, output_dim)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index)
        x = F.relu(x)
        x = self.conv2(x, edge_index)
        return x
class GATModel(torch.nn.Module):
    def __init__(self, input_dim, hidden_dim=64, output_dim=2,
                 heads=4, dropout=0.2):
        super().__init__()
        self.gat1 = GATConv(
            input_dim, hidden_dim, heads=heads, dropout=dropout
        )
        self.gat2 = GATConv(
```

```
            hidden_dim * heads, output_dim,
            heads=1, concat=False, dropout=dropout
        )
        self.dropout = dropout

    def forward(self, x, edge_index):
        x = self.gat1(x, edge_index)
        x = F.elu(x)
        x = F.dropout(x, p=self.dropout, training=self.training)
        x = self.gat2(x, edge_index)
        return x
```

G. Fungsi Training dan Evaluasi GCN/GAT

```
def train_graph_model(model, criterion, epochs=200, lr=0.01):
    optimizer = torch.optim.Adam(model.parameters(), lr=lr)
    loss_history = []

    for epoch in range(1, epochs + 1):
        model.train()
        optimizer.zero_grad()
        output = model(data_known.x, data_known.edge_index)
        loss = criterion(
            output[data_known.train_mask],
            data_known.y[data_known.train_mask]
        )
        loss.backward()
        optimizer.step()
        loss_history.append(loss.item())
```

```
        if epoch == 1 or epoch % 20 == 0:
            print(f'Epoch {epoch:03d} | Loss: {loss.item():.6f}')
    model.eval()
    with torch.no_grad():
        logits = model(data_known.x, data_known.edge_index)
        prob = F.softmax(logits, dim=1)[:,1]
        pred = logits.argmax(dim=1)
    yt = data_known.y[data_known.test_mask].cpu().numpy()
    yp = pred[data_known.test_mask].cpu().numpy()
    ys = prob[data_known.test_mask].cpu().numpy()
    return model, loss_history, yt, yp, ys
def metric_row(name, yt, yp, ys):
    tn, fp, fn, tp = confusion_matrix(yt, yp, labels=[0,1]).ravel()
    return {
        'Model': name,
        'TN': tn, 'FP': fp, 'FN': fn, 'TP': tp,
        'Accuracy': accuracy_score(yt, yp),
        'Precision': precision_score(yt, yp),
        'Recall': recall_score(yt, yp),
        'F1-score': f1_score(yt, yp),
        'ROC-AUC': roc_auc_score(yt, ys)
    }
```

H. Training GCN dan GAT Tanpa Class Weight

```
criterion_plain = torch.nn.CrossEntropyLoss()
gcn_model = GCNModel(input_dim=len(feature_cols),
hidden_dim=64).to(device)
gcn_model, gcn_loss, y_gcn, p_gcn, s_gcn = train_graph_model(
    gcn_model, criterion_plain
```

```
)


gat_model = GATModel(
    input_dim=len(feature_cols), hidden_dim=64,
    output_dim=2, heads=4, dropout=0.2
).to(device)
gat_model, gat_loss, y_gat, p_gat, s_gat = train_graph_model(
    gat_model, criterion_plain
)
print(metric_row('GCN', y_gcn, p_gcn, s_gcn))
print(metric_row('GAT', y_gat, p_gat, s_gat))
```

Ringkasan hasil model graph pada Bab IV

GCN: kemampuan deteksi Illicit lebih baik daripada baseline SVM pada Recall.

GAT: mekanisme attention meningkatkan kemampuan model dalam mengenali kelas Illicit.

I. Perhitungan Class Weight

```
train_labels = y_known[train_idx]
weights = compute_class_weight(
    class_weight='balanced',
    classes=np.array([0,1]),
    y=train_labels
)
class_weights = torch.tensor(weights, dtype=torch.float).to(device)


print('Weight Licit   :', class_weights[0].item())
print('Weight Illicit :', class_weights[1].item())
```

Output Class Weight

Weight Licit   : 0,554083

Weight Illicit : 5,122525

J. Model Utama: GAT + Class Weight

```
gat_cw_model = GATModel(
    input_dim=len(feature_cols),
    hidden_dim=64,
    output_dim=2,
    heads=4,
    dropout=0.2
).to(device)
criterion_weighted = torch.nn.CrossEntropyLoss(weight=class_weights)


gat_cw_model, gat_cw_loss, y_gat_cw, p_gat_cw, s_gat_cw =
train_graph_model(
    gat_cw_model,
    criterion_weighted,
    epochs=200,
    lr=0.01
)


final_gat_cw = metric_row(
    'GAT + Class Weight',
    y_gat_cw,
    p_gat_cw,
    s_gat_cw
)
print(final_gat_cw)
```

Hasil final GAT + Class Weight yang digunakan pada Bab IV–V

Recall : 90,43%

ROC-AUC : 0,9809

False Negative (FN) : 87

True Positive (TP) : 822

K. Evaluasi Final dan Perbandingan Model

```
# Gabungkan seluruh hasil model ke satu tabel evaluasi
rows = []
rows.append(metric_row('SVM', y_test, svm_pred, svm_prob))
rows.append(metric_row('GCN', y_gcn, p_gcn, s_gcn))
rows.append(metric_row('GAT', y_gat, p_gat, s_gat))
rows.append(final_gat_cw)

result_final = pd.DataFrame(rows)
for c in ['Accuracy','Precision','Recall','F1-score']:
    result_final[c] = (result_final[c] * 100).round(2)
result_final['ROC-AUC'] = result_final['ROC-AUC'].round(4)

print(result_final)

# ROC Curve seluruh model
plt.figure(figsize=(8,6))
for name, yt, ys in [
    ('SVM', y_test, svm_prob),
    ('GCN', y_gcn, s_gcn),
    ('GAT', y_gat, s_gat),
    ('GAT + Class Weight', y_gat_cw, s_gat_cw)
]:
    fpr, tpr, _ = roc_curve(yt, ys)
    model_auc = auc(fpr, tpr)
```

```
    plt.plot(fpr, tpr, label=f'{name} (AUC={model_auc:.4f})')

plt.plot([0,1],[0,1], linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Perbandingan Kurva ROC')
plt.legend()
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```

L. Analisis Struktur Graph: Unknown yang Terhubung dengan Illicit

```
illicit_txids = set(
    df_all.loc[df_all['label_name'] == 'Illicit', 'txId']
)
unknown_txids = set(
    df_all.loc[df_all['label_name'] == 'Unknown', 'txId']
)

# Arah Illicit -> Unknown
illicit_to_unknown = edges_all[
    edges_all['txId1'].isin(illicit_txids)
    & edges_all['txId2'].isin(unknown_txids)
].copy()

# Arah Unknown -> Illicit
unknown_to_illicit = edges_all[
    edges_all['txId1'].isin(unknown_txids)
    & edges_all['txId2'].isin(illicit_txids)
```

```
].copy()

unknown_neighbor_ids = set(illicit_to_unknown['txId2']).union(
    set(unknown_to_illicit['txId1'])
)

unknown_neighbor_df = df_all[
    df_all['txId'].isin(unknown_neighbor_ids)
][['txId','Time_step','label_name']].copy()

outgoing_count = unknown_to_illicit.groupby('txId').size()
incoming_count = illicit_to_unknown.groupby('txId2').size()

unknown_neighbor_df['Unknown_to_Illicit'] = (
    unknown_neighbor_df['txId'].map(outgoing_count).fillna(0).astype(int)
)
unknown_neighbor_df['Illicit_to_Unknown'] = (
    unknown_neighbor_df['txId'].map(incoming_count).fillna(0).astype(int)
)
unknown_neighbor_df['Total_Illicit_Connections'] = (
    unknown_neighbor_df['Unknown_to_Illicit']
    + unknown_neighbor_df['Illicit_to_Unknown']
)

print('Unknown yang terhubung langsung dengan Illicit:',
len(unknown_neighbor_df))
```

Output Neighbor Risk Analysis

Unknown terhubung langsung dengan Illicit : 4.027 transaksi

M. Inferensi GAT + Class Weight terhadap Seluruh Transaksi Unknown

```
# Bentuk graph penuh: 203.769 node dan 234.355 edge
all_mapping = {txid: idx for idx, txid in enumerate(df_all['txId'])}
edges_full = edges_all.copy()
edges_full['source'] = edges_full['txId1'].map(all_mapping)
edges_full['target'] = edges_full['txId2'].map(all_mapping)

edge_index_full = torch.tensor(
    edges_full[['source','target']].values.T,
    dtype=torch.long
).to(device)

X_all = df_all[feature_cols].to_numpy(dtype=np.float32)
X_all_scaled = scaler.transform(X_all)
x_all = torch.tensor(X_all_scaled, dtype=torch.float).to(device)

# Inferensi menggunakan model utama GAT + Class Weight
gat_cw_model.eval()
with torch.no_grad():
    logits_all = gat_cw_model(x_all, edge_index_full)
    probabilities_all = F.softmax(logits_all, dim=1)
    predictions_all = probabilities_all.argmax(dim=1)

unknown_mask = (df_all['label_name'] == 'Unknown').to_numpy()
unknown_indices = np.where(unknown_mask)[0]

unknown_probabilities = probabilities_all[unknown_indices].cpu().numpy()
unknown_predictions = predictions_all[unknown_indices].cpu().numpy()
```

```
unknown_result = pd.DataFrame({
   'txId': df_all.loc[unknown_mask, 'txId'].values,
   'Time_step': df_all.loc[unknown_mask, 'Time_step'].values,
   'Prob_Licit': unknown_probabilities[:,0],
   'Prob_Illicit': unknown_probabilities[:,1],
   'Prediction_Code': unknown_predictions
})
unknown_result['Prediction'] = unknown_result['Prediction_Code'].map({
   0:'Licit', 1:'Illicit'
})

unknown_summary = (
   unknown_result.groupby('Prediction').size().reset_index(name='Jumlah')
)
unknown_summary['Persentase (%)'] = (
   unknown_summary['Jumlah'] / len(unknown_result) * 100
).round(2)

print('Total Unknown:', len(unknown_result))
print(unknown_summary)
```

Output prediksi Unknown

Total Unknown : 157.205

Diprediksi Licit : 119.100 (75,76%)

Diprediksi Illicit : 38.105 (24,24%)

N. Kombinasi Prediksi Model dan Hubungan Graph

```
unknown_forensic = unknown_result.merge(
   unknown_neighbor_df[[
```

```
        'txId',
        'Unknown_to_Illicit',
        'Illicit_to_Unknown',
        'Total_Illicit_Connections'
    ]],
    on='txId',
    how='left'
)

cols_conn = [
    'Unknown_to_Illicit',
    'Illicit_to_Unknown',
    'Total_Illicit_Connections'
]
unknown_forensic[cols_conn] = (
    unknown_forensic[cols_conn].fillna(0).astype(int)
)

# Kandidat prioritas: diprediksi Illicit DAN memiliki hubungan langsung
# dengan transaksi yang telah berlabel Illicit.
forensic_candidates = unknown_forensic[
    (unknown_forensic['Prediction'] == 'Illicit')
    & (unknown_forensic['Total_Illicit_Connections'] > 0)
].sort_values(
    ['Prob_Illicit','Total_Illicit_Connections'],
    ascending=[False,False]
)
```

```
print('=' * 70)
print('RINGKASAN ANALISIS UNKNOWN')
print('=' * 70)
print('Total Unknown:', len(unknown_result))
print('Diprediksi Illicit:', (unknown_result['Prediction']=='Illicit').sum())
print('Terhubung langsung dengan Illicit:', len(unknown_neighbor_df))
print('Diprediksi Illicit DAN terhubung dengan Illicit:', len(forensic_candidates))
```

Output final analisis Unknown untuk digital forensik

Total Unknown : 157.205

Diprediksi Illicit : 38.105

Terhubung langsung dengan Illicit : 4.027

Diprediksi Illicit + terhubung Illicit : 1.942

O. Menampilkan Kandidat Prioritas dan Menyimpan Hasil

```
# Tampilkan 20 kandidat prioritas tertinggi
print(forensic_candidates[[
    'txId','Time_step','Prob_Licit','Prob_Illicit','Prediction',
    'Unknown_to_Illicit','Illicit_to_Unknown',
    'Total_Illicit_Connections'
]].head(20))

# Simpan seluruh hasil analisis
unknown_result.to_csv(
    '/content/Hasil_Prediksi_Unknown.csv', index=False
)
unknown_neighbor_df.to_csv(
    '/content/Unknown_Neighbor_Illicit.csv', index=False
)
forensic_candidates.to_csv(
```

```
    '/content/Kandidat_Unknown_Digital_Forensik.csv', index=False
)
result_final.to_csv(
    '/content/Hasil_Evaluasi_Final_Model.csv', index=False
)

print('Semua file hasil penelitian berhasil disimpan.')

# Download bila diperlukan
files.download('/content/Hasil_Evaluasi_Final_Model.csv')
files.download('/content/Hasil_Prediksi_Unknown.csv')
files.download('/content/Unknown_Neighbor_Illicit.csv')
files.download('/content/Kandidat_Unknown_Digital_Forensik.csv')
```