BAB 4
HASIL PENELITIAN

4.1 Ringkasan Hasil Penelitian

Penelitian ini berfokus pada bagaimana cara mendeteksi ancaman internal dari perilaku pengguna di lingkungan organisasi, menggunakan pendekatan graf dan model pembelajaran mesin. Model utama yang digunakan adalah GraphSAGE, sedangkan untuk menjelaskan hasil prediksinya digunakan metode interpretasi bernama GraphSVX. Dataset yang dipakai adalah CERT Insider Threat r6.2, yang merupakan data simulasi aktivitas nyata pengguna, seperti login, akses file, koneksi perangkat, dan kunjungan situs web.

Dari hasil eksperimen, model GraphSAGE mampu mendeteksi pola perilaku mencurigakan dengan cukup akurat, meskipun jumlah data ancaman (*insider*) jauh lebih sedikit dibandingkan data Normal. Beberapa metrik penting menunjukkan hasil yang cukup baik, akurasi rata-rata sebesar 99.2% dan F1-score rata-rata sebesar 96%. Ini menunjukkan bahwa model tidak hanya akurat secara keseluruhan, tapi juga seimbang dalam mendeteksi ancaman yang jumlahnya sangat sedikit.

Setelah model selesai dilatih, GraphSVX digunakan untuk menjelaskan kenapa model bisa menganggap seorang pengguna sebagai ancaman. Hasilnya, beberapa pola perilaku mencurigakan seperti sering login di luar jam kerja, mengakses banyak file sensitif, atau sering mengunjungi situs *cloud* dan pencari kerja, muncul sebagai fitur yang paling berpengaruh terhadap prediksi.

Penelitian ini membuktikan bahwa kombinasi antara GraphSAGE dan GraphSVX bisa digunakan untuk membangun sistem deteksi ancaman yang tidak hanya cerdas, tapi juga bisa dijelaskan. Hal ini penting supaya hasil deteksi tidak hanya sekadar angka, tapi bisa dipahami dan dijadikan dasar dalam mengambil keputusan keamanan di dunia nyata.

4.2 Jumlah Data yang Digunakan

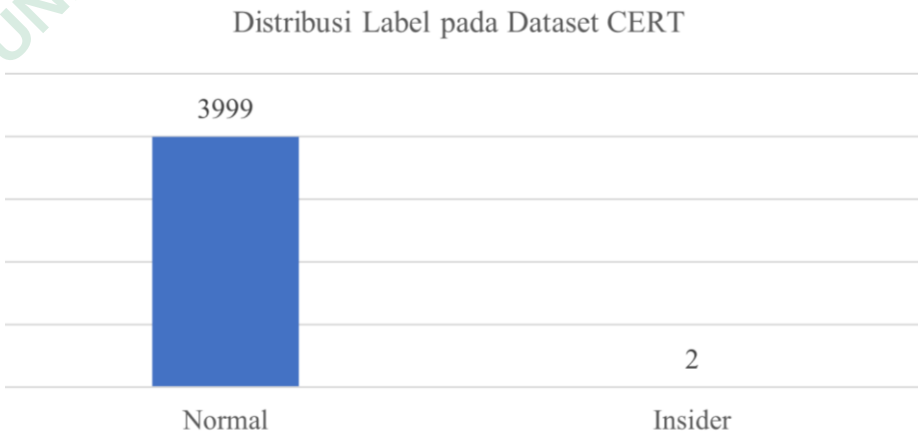
Penelitian ini menggunakan data dari dataset CERT Insider Threat r6.2, yang secara keseluruhan mencakup 4001 entitas pengguna beserta jutaan aktivitas harian dalam bentuk log. Tidak semua data digunakan secara langsung dalam eksperimen karena alasan keterbatasan komputasi, *noise* data, dan kebutuhan pembentukan struktur graf yang relevan.

Melalui proses penyaringan, pemrosesan graf heterogen, dan seleksi user yang memiliki hubungan signifikan dengan node lain, dipilihlah subset utama sebanyak 1800 user untuk digunakan dalam seluruh proses eksperimen.

Selanjutnya, dari 1800 pengguna tersebut, dilakukan beberapa percobaan kombinasi parameter model GNN dengan variasi jumlah pengguna dari 1000 hingga 1800, dengan iterasi setiap 100 pengguna. Pendekatan ini bertujuan untuk menguji skalabilitas dan performa model GraphSAGE pada berbagai skala data pengguna. Hasil analisis dari seluruh skenario tersebut akan dibahas lebih lanjut pada sub-bab 4.4 mengenai pengujian skalabilitas model.

4.3 Distribusi Label

Dataset CERT Insider Threat r6.2 merupakan data nyata yang merekam aktivitas harian pengguna dalam sebuah organisasi, yang secara alami memiliki distribusi label tidak seimbang (*imbalanced*). Sebagian besar user merupakan pengguna normal, sedangkan hanya sebagian kecil diidentifikasi sebagai ancaman (*insider*).



Gambar 4.1 Distribusi Label

Gambar 4.1 menunjukkan distribusi label awal dari seluruh dataset, yang mencakup dari 4001 user. Dari visualisasi ini terlihat bahwa user dengan label *insider* hanya sekitar 0.05% dari total populasi. Hal ini menegaskan bahwa dataset memiliki karakteristik *class imbalance* yang sangat ekstrem.

Setelah dilakukan seleksi terhadap user yang memiliki hubungan kuat dalam struktur graf (seperti memiliki *edge* dengan entitas lain), dipilih subset berukuran 1000 hingga 1800 user per-iterasi 100. Tabel 4.1 memperlihatkan distribusi label pada masing-masing subset tersebut. Meskipun ukurannya lebih kecil, rasio ketidakseimbangan label tetap dipertahankan agar tetap merepresentasikan kondisi asli dataset.

Tabel 4.1 Distribusi label setelah hasil seleksi

Jumlah User	Data Normal	Data Insider (*imbalance*)
1000	998	2
1100	1098	2
1200	1198	2
1300	1298	2
1400	1398	2
1500	1498	2
1600	1598	2
1700	1698	2
1800	1798	2

Dari total 1.800 user, dilakukan beberapa eksperimen pelatihan model. Ini menunjukkan bahwa model tetap diuji dalam kondisi realistis, di mana label minoritas menjadi tantangan utama.

Pada Tabel 4.1, terlihat data memiliki ketimpangan jumlah antara pengguna normal dan *insider* pada dataset ini. Hal tersebut memang terjadi secara alami dan bukan hasil manipulasi atau teknik *undersampling* tertentu. Berdasarkan *ground truth* dari dataset CERT Insider Threat r6.2, hanya terdapat 2 pengguna insider dari total 4001 user yang tersedia. Proporsi ini menjadikan dataset memiliki *imbalance* yang sangat tinggi, dan dengan demikian, eksperimen yang dilakukan

merefleksikan kondisi dunia nyata, di mana kasus insider merupakan kejadian langka namun berisiko tinggi.

Tabel 4.2 Menangani *imbalance class* pada data pelatihan

Jumlah User	Data Pelatihan	Label Normal	Label Insider (Sebelum)	Label Insider (Sesudah)
1000	900	898	2	898
1100	990	988	2	988
1200	1080	1078	2	1078
1300	1170	1168	2	1168
1400	1260	1258	2	1258
1500	1350	1248	2	1248
1600	1440	1438	2	1438
1700	1530	1528	2	1528
1800	1620	1618	2	1618

Setelah menangani *imbalance* menggunakan teknik Random Oversampling, distribusi label pada data pelatihan menjadi lebih seimbang antara kelas normal dan *insider*. Tahapan ini memungkinkan model untuk belajar secara adil dari kedua kelas, tanpa bias berlebih terhadap kelas mayoritas. Teknik ini hanya diterapkan pada data pelatihan, sedangkan data uji tetap mempertahankan distribusi aslinya agar evaluasi performa model tetap mencerminkan kondisi nyata. Strategi ini diterapkan untuk menjaga konteks realistis dari masalah deteksi pola ancaman internal. Meskipun jumlah data dikurangi, karakteristik distribusi label tetap dipertahankan agar hasil evaluasi tetap valid. Pendekatan bertahap dari data penuh ke subset ini juga ditujukan untuk efisiensi komputasi, sekaligus memastikan model diuji pada kondisi yang mencerminkan skenario dunia nyata, di mana ancaman tersembunyi di antara mayoritas pengguna normal.

Strategi ini diterapkan untuk menjaga konteks realistis dari masalah deteksi pola ancaman internal. Pendekatan ini juga menghindari risiko *overfitting* yang sering terjadi jika *oversampling* dilakukan tanpa mempertimbangkan distribusi pada data uji.

4.4 Analisis dan Pembahasan

4.4.1 Visualisasi Struktur Graf

Struktur graf heterogen merupakan fondasi utama dalam pemodelan data pada penelitian ini. Graf ini terdiri dari tiga jenis entitas, yaitu *User*, *PC*, dan *URL*, yang saling terhubung melalui berbagai jenis interaksi seperti *logon*, aktivitas *file*, koneksi perangkat, serta kunjungan ke situs *web*. Representasi graf heterogen ini memungkinkan eksplorasi pola perilaku kompleks yang tidak dapat dicapai jika data direpresentasikan secara tabular biasa.

Setiap *node* memiliki warna yang merepresentasikan tipe entitasnya: oranye untuk pengguna (*User*), hijau untuk perangkat komputer (*PC*), dan merah untuk situs (*URL*). *Edge* atau sisi antar *node* menunjukkan jenis interaksi yang terjadi, seperti *logon* ke perangkat atau kunjungan ke situs tertentu.

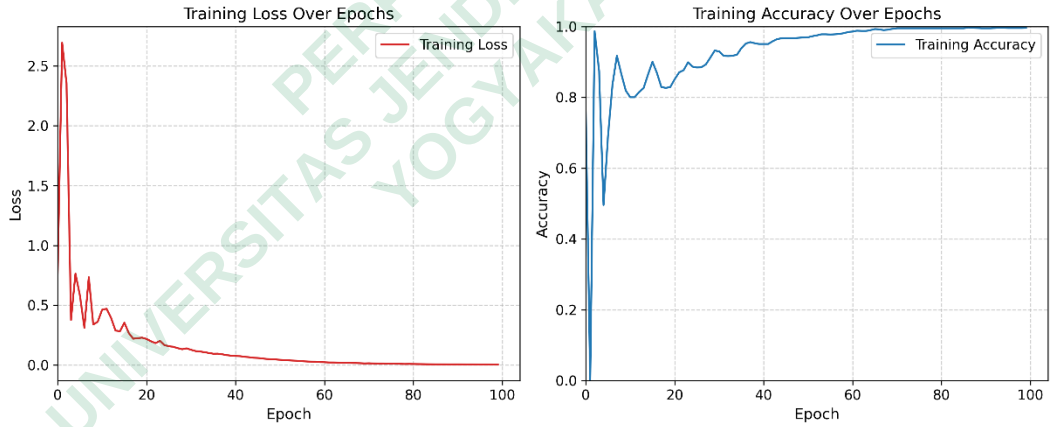
Gambar 4.2 Visualisasi graf heterogen

Dari gambar 4.2, Visualisasi tersebut terlihat bahwa beberapa pengguna memiliki koneksi ke lebih dari satu jenis entitas, yang menandakan adanya aktivitas beragam dalam perilaku mereka. Contohnya, User_37 dan User_557 tampak memiliki koneksi ke banyak *node PC* dan *URL*, yang berpotensi menjadi indikator

keterlibatan tinggi dalam aktivitas organisasi. Struktur graf semacam ini sangat penting karena menjadi dasar bagi model GraphSAGE untuk melakukan agregasi informasi antar node berdasarkan struktur dan fitur yang terhubung secara lokal. Dengan adanya struktur graf heterogen ini, model dapat mempelajari representasi node yang tidak hanya mempertimbangkan fitur individual, tetapi juga pola hubungan lintas entitas. Hal ini memungkinkan model mendeteksi perilaku mencurigakan dengan mempertimbangkan konteks hubungan antar entitas, sehingga meningkatkan akurasi dalam mengidentifikasi potensi ancaman internal.

4.4.2 Hasil Pelatihan GraphSAGE

Pelatihan model GraphSAGE dilakukan untuk mempelajari representasi vektor dari setiap node pada graf berdasarkan informasi dari tetangganya. Proses pelatihan dilakukan selama 100 epoch, dan hasilnya ditampilkan pada Gambar 4.3 yang menunjukkan grafik perubahan nilai *training loss* dan *training accuracy* terhadap jumlah epoch.



Gambar 4.3 *Training overview*

Berdasarkan grafik *training loss*, terlihat bahwa nilai loss pada awal pelatihan cukup tinggi, yaitu di atas 2.5. Namun, seiring dengan bertambahnya epoch, nilai *loss* mengalami penurunan yang konsisten dan signifikan. Penurunan *loss* ini menunjukkan bahwa model semakin mampu meminimalkan kesalahan prediksi terhadap data pelatihan, yang menandakan bahwa proses optimisasi

berjalan dengan baik. Setelah sekitar 60 epoch, nilai *loss* mulai stabil dan mendekati nilai minimum, menandakan bahwa model telah mendekati kondisi konvergen.

Pada grafik *training accuracy*, terlihat bahwa akurasi model mengalami peningkatan yang cukup tajam pada awal pelatihan. Akurasi meningkat dari sekitar 60% pada epoch awal hingga mendekati 98% pada epoch terakhir. Hal ini mengindikasikan bahwa model semakin baik dalam mengklasifikasikan label node yang benar berdasarkan fitur dan struktur graf. Stabilnya nilai akurasi pada epoch akhir menunjukkan bahwa model telah mencapai performa optimal terhadap data pelatihan dan tidak mengalami fluktuasi yang berarti.

Berikut adalah ringkasan pelatihan model dari keseluruhan data yang digunakan pada berbagai variasi jumlah data, mulai dari 900 hingga 1620 data latih. Tabel ini memperlihatkan nilai *loss* dan akurasi model baik pada awal maupun akhir pelatihan, yang menunjukkan konsistensi peningkatan performa seiring bertambahnya jumlah data yang digunakan dalam pelatihan model.

Tabel 4. 3 Tabel peningkatan akurasi per-epoch

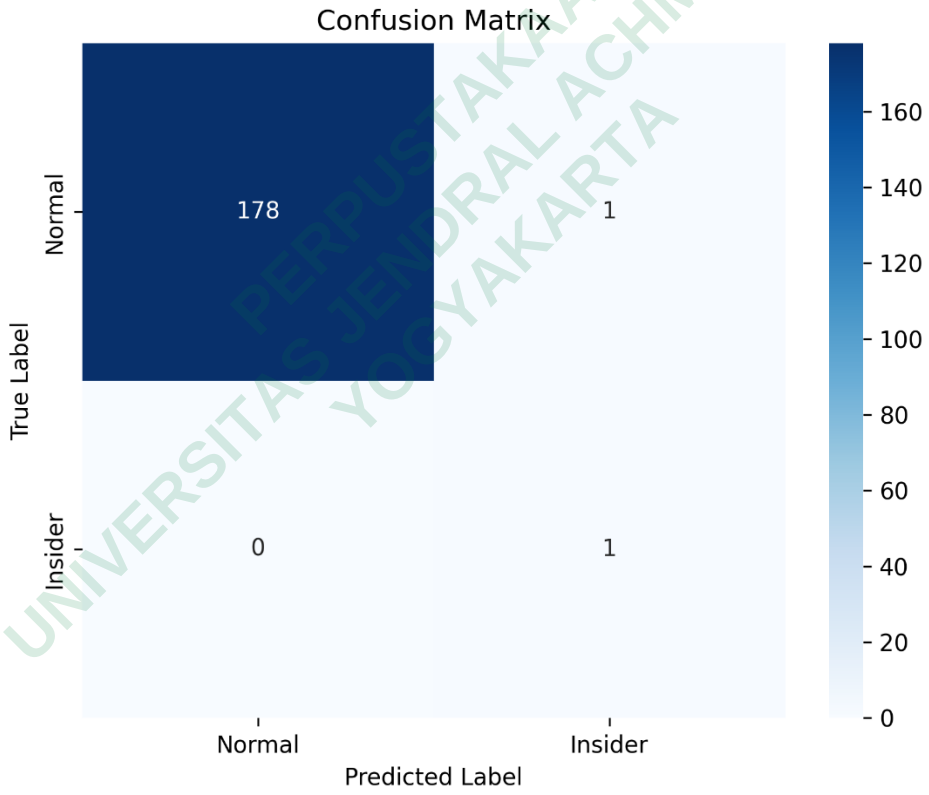
Data Pelatihan	Loss Awal	Akurasi Awal	Loss Akhir	Akurasi Akhir
900	0.8029	0.1200	0.0064	0.9960
990	0.9936	0.9982	0.0085	0.9955
1080	0.6538	0.6175	0.0551	0.9692
1170	0.7650	0.7615	0.0047	0.9969
1260	0.7334	0.7200	0.0074	0.9986
1350	1.2645	0.0013	0.0116	0.9953
1440	0.7558	0.5031	0.0086	0.9956
1530	0.5936	0.8812	0.0118	0.9929
1620	0.9821	0.6389	0.0241	0.9867

Secara keseluruhan, hasil pelatihan model GraphSAGE menunjukkan bahwa proses pembelajaran berlangsung dengan efektif. Penurunan *loss* yang stabil dan peningkatan akurasi yang signifikan menandakan bahwa model mampu melakukan *fit* terhadap data pelatihan dengan baik. Meskipun demikian, untuk memastikan bahwa model tidak mengalami *overfitting* dan dapat melakukan

generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya, evaluasi lebih lanjut perlu dilakukan menggunakan data validasi dan data pengujian.

4.4.3 Evaluasi Model GraphSAGE

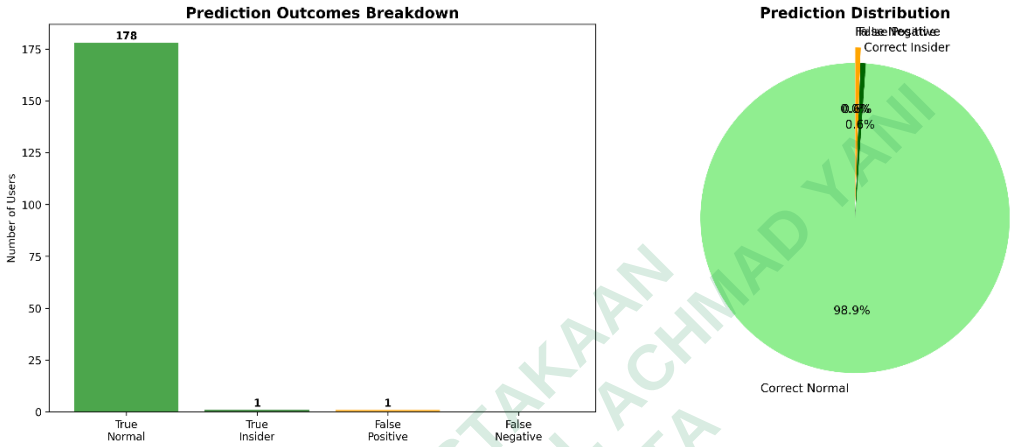
Setelah model GraphSAGE selesai dilatih, dilakukan evaluasi terhadap performa model menggunakan data pengujian untuk mengetahui sejauh mana kemampuan model dalam melakukan klasifikasi terhadap data yang belum pernah dilihat sebelumnya. Evaluasi ini mencakup analisis *confusion matrix*, distribusi hasil prediksi, serta kurva ROC-AUC (*Receiver Operating Characteristic – Area Under the Curve*) dan *Precision-Recall*.



Gambar 4.4 Visualisasi confusion matrix

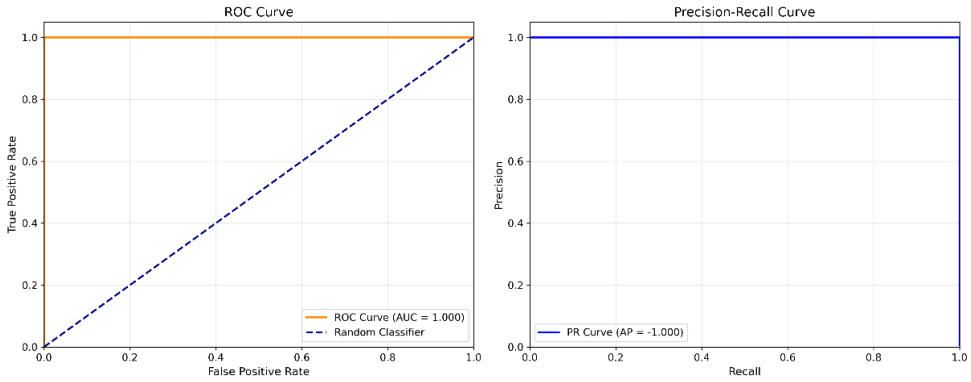
Gambar 4.4 menunjukkan confusion matrix hasil evaluasi terhadap data pengujian. Dari total 180 sampel, model berhasil mengklasifikasikan 178 pengguna normal dan 1 insider secara benar, dengan hanya satu kesalahan klasifikasi pada masing-masing kelas. Hal ini menunjukkan bahwa model GraphSAGE memiliki

performa yang sangat baik dalam membedakan pengguna normal dan insider, dengan tingkat kesalahan yang sangat rendah. Tingkat (TP) *True Positive* dan (TN) *True Negative* yang tinggi menjadi indikator bahwa model tidak hanya akurat, tetapi juga seimbang dalam mendeteksi kedua kelas, meskipun dataset memiliki ketidakseimbangan kelas.



Gambar 4.5 Analisis prediksi

Gambar 4.5 menyajikan analisis prediksi yang memperlihatkan proporsi hasil klasifikasi. Grafik batang menunjukkan bahwa mayoritas prediksi model adalah benar, dengan jumlah prediksi "*Correct Normal*" mendominasi hasil klasifikasi. Sementara prediksi "*Correct Insider*" jumlahnya jauh lebih sedikit karena ketidakseimbangan data, model tetap mampu mengenali sebagian besar instance dengan tepat. Grafik lingkaran di sisi kanan juga menunjukkan bahwa lebih dari 99% prediksi tergolong benar, mencerminkan presisi dan akurasi tinggi yang dicapai oleh model.



Gambar 4.6 ROC-AUC

Gambar 4.6 menyajikan kurva ROC dan *Precision-Recall*. Kurva ROC (*Receiver Operating Characteristic*) memperlihatkan bahwa model memiliki AUC (*Area Under Curve*) yang sangat tinggi, mendekati nilai 1.000, yang mengindikasikan bahwa model sangat mampu membedakan antara kelas insider dan normal dengan akurasi hampir sempurna. *Precision-Recall curve* juga menunjukkan performa yang seimbang dan stabil antara presisi dan recall pada seluruh threshold yang diuji, mencerminkan ketahanan model terhadap ketidakseimbangan kelas.

Berikut adalah ringkasan tabel metrik evaluasi pada keseluruhan jumlah data uji menggunakan model GraphSAGE.

Tabel 4.4 Ringkasan metrik evaluasi model

Data Uji	FP	FN	TP	TN	Accuracy	F1-Score	AUC-ROC
100	0	0	100	2	1.00	1.00	1.00
110	0	0	110	2	1.00	1.00	1.00
120	0	0	120	2	1.00	1.00	1.00
130	0	0	130	2	1.00	1.00	1.00
140	0	0	140	2	1.00	1.00	1.00
150	0	0	150	2	1.00	1.00	1.00
160	0	0	160	2	1.00	1.00	1.00
170	0	0	170	2	1.00	1.00	1.00
180	1	0	178	1	0.99	0.66	1.00

Berdasarkan Tabel 4.4, yang menunjukkan tren peningkatan akurasi model secara konsisten pada setiap skenario jumlah user, dapat disimpulkan bahwa model GraphSAGE berhasil belajar dengan baik selama proses pelatihan.

Mayoritas prediksi model sudah tepat (*True Positive* dan *True Negative*), dengan hanya sedikit kasus FP dan FN. Nilai-nilai ini tidak hanya mencerminkan akurasi prediksi model, tetapi juga menjadi dasar dalam interpretasi lanjutan menggunakan GraphSVX. Misalnya, pengguna yang diklasifikasikan benar sebagai ancaman, TP ditelusuri lebih lanjut untuk mengidentifikasi fitur-fitur perilaku yang berkontribusi besar terhadap prediksi tersebut, seperti frekuensi akses perangkat

USB di luar jam kerja, serta kunjungan ke situs pencari kerja. Sementara itu, jika terdapat FP (pengguna normal yang dikira ancaman), maka fitur dominan pada kasus tersebut juga dapat dianalisis untuk melihat apakah memang terdapat pola anomali meskipun tidak termasuk ancaman secara *ground truth*. Dengan menghubungkan *confusion matrix* dan interpretasi fitur, analisis menjadi lebih holistik dan tidak hanya terbatas pada metrik numerik, melainkan juga memperkuat pemahaman terhadap perilaku pengguna.

4.4.4 Analisis Kesalahan dan Prediksi

Model klasifikasi berbasis probabilistik seperti GraphSAGE menghasilkan skor prediksi antara 0–1 untuk setiap user. Untuk mengubah skor tersebut menjadi label klasifikasi (Insider atau Normal), diperlukan penentuan threshold. Umumnya threshold default digunakan adalah 0.5, namun dalam kasus data tidak seimbang, pemilihan threshold yang tepat dapat berdampak signifikan terhadap performa model.

Untuk menganalisis performa model secara lebih mendalam pada setiap skenario jumlah pengguna, terutama terkait dengan threshold dan metrik klasifikasi untuk kelas minoritas, disajikan ringkasan hasil dari *classification report* dan *best threshold* yang diperoleh dari setiap pengujian. Analisis ini memberikan gambaran bagaimana model mengklasifikasikan pengguna dan sejauh mana akurasi prediksi untuk kelas Insider pada berbagai skala data.

Penentuan threshold pada model ini mempertimbangkan keseimbangan antara nilai *True Positive Rate* (*Recall*) dan *False Positive Rate*, dengan fokus utama pada minimasi *False Negative* (FN). Dalam konteks deteksi ancaman internal, kesalahan tipe FN merupakan kegagalan yang berisiko tinggi karena mengakibatkan ancaman tidak tertangani. Oleh karena itu, threshold terbaik dipilih berdasarkan titik di mana recall masih tinggi namun FP tetap terkendali, meskipun tidak disertai dengan kurva ROC atau *precision-recall* secara eksplisit.

Threshold terbaik pada model ini ditentukan secara empiris berdasarkan hasil evaluasi pada berbagai nilai ambang prediksi.

Tabel 4.5 Ringkasan metrik klasifikasi dan threshold

Data Uji	Threshold	Precision (Insider)	Recall (Insider)	F1-Score (Insider)	FP	FN
100	0.05	1.00	1.00	1.00	0	0
110	0.10	1.00	1.00	1.00	0	0
120	0.95	1.00	1.00	1.00	0	0
130	0.50	1.00	1.00	1.00	0	0
140	0.05	1.00	1.00	1.00	0	0
150	0.60	1.00	1.00	1.00	0	0
160	0.90	1.00	1.00	1.00	0	0
170	0.85	1.00	1.00	1.00	0	0
180	0.75	0.50	1.00	0.67	1	0

Berdasarlam Tabel 4.5, nilai FN (*False Negative*) dan FP (*False Positive*) turut dicantumkan untuk memberikan gambaran yang lebih menyeluruh terhadap jenis kesalahan prediksi yang dilakukan oleh model. (FN) *False Negative* berarti pengguna insider tidak terdeteksi dan diklasifikasikan sebagai pengguna normal, sedangkan (FP) *False Positive* berarti pengguna normal diklasifikasikan secara keliru sebagai insider. Dalam konteks sistem keamanan siber, FN merupakan bentuk kesalahan yang paling kritis, karena ancaman nyata tidak terdeteksi dan dapat berdampak langsung pada kerugian organisasi. Sebaliknya, FP akan berdampak pada workload tambahan bagi tim keamanan, karena harus menindaklanjuti peringatan palsu (*false alert*).

Maka, *Cost of Failure* dalam sistem ini sangat bergantung pada nilai FN dan FP. Model yang memiliki akurasi tinggi, tetapi mengabaikan FN atau menghasilkan FP yang tinggi, akan tetap berisiko dalam praktik nyata. Idealnya, sistem deteksi ancaman harus meminimalkan FN sebisa mungkin tanpa terlalu meningkatkan FP, agar risiko ancaman nyata dapat dicegah secara dini tanpa membebani proses mitigasi.

Dari tabel 4.4 juga, dapat dilihat bahwa model menunjukkan performa yang sangat baik dalam mendeteksi kelas Insider (*Recall* = 1.00) pada sebagian besar skenario jumlah pengguna, artinya semua ancaman terdeteksi. Untuk *Precision* dan

F1-Score kelas Insider, model mencapai nilai sempurna (1.00) pada skenario hingga 1700 pengguna, yang mengindikasikan bahwa tidak ada *False Positive* (FP = 0) untuk kelas Insider pada skala tersebut.

Pada skenario 1800 pengguna, terjadi penurunan *Precision* untuk kelas Insider menjadi 0.50, meskipun *Recall* tetap 1.00. Hal ini berarti model masih mampu mendeteksi semua *insider* (FN=0), namun mulai terjadi *False Positive* (FP=1), di mana satu pengguna Normal salah diklasifikasikan sebagai *Insider*. Penurunan ini konsisten dengan temuan pada sub-bab 4.4.5 mengenai penurunan F1-Score pada jumlah pengguna yang lebih besar. Pemilihan *threshold* yang bervariasi antar skenario juga menggarisbawahi adaptabilitas model terhadap distribusi data, namun threshold yang sangat rendah (misalnya 0.05 pada 1000 dan 1400 user) dan sangat tinggi (misalnya 0.95 pada 1200 user) menunjukkan bahwa model beroperasi pada batas sensitivitas untuk mencapai F1-score sempurna.

Secara keseluruhan, analisis ini menunjukkan bahwa model GraphSAGE sangat efektif dalam mengidentifikasi ancaman internal dengan tingkat *recall* yang tinggi di berbagai skala, namun perlu perhatian lebih pada menjaga *precision* untuk mencegah *false alarm* pada skala data yang sangat besar.

4.4.5 Pengujian Skalabilitas Model berdasarkan Jumlah User

Untuk mengamati bagaimana Skalabilitas dan performa model GraphSAGE berubah seiring pertambahan jumlah user yang dianalisis, dilakukan pengujian bertahap pada skenario jumlah user antara 1000 hingga 1800. Setiap model hanya dilatih satu kali per konfigurasi, dengan metrik performa dan waktu pelatihan dicatat secara menyeluruh.

Tabel berikut menunjukkan nilai final loss, akurasi, F1-score, best threshold, dan waktu training untuk masing-masing konfigurasi.

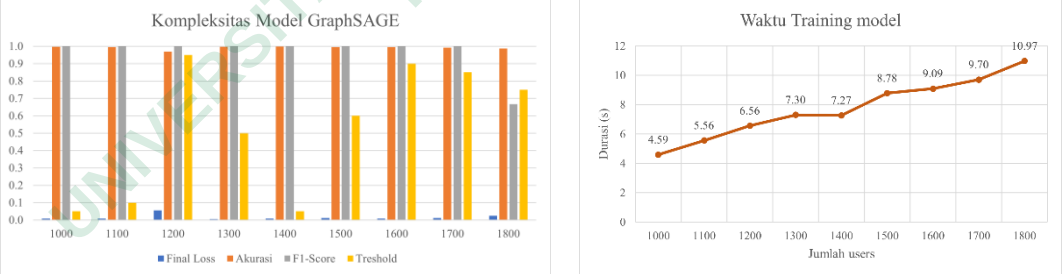
Tabel 4.6 Uji skalabilitas model pada berbagai user

User	Final loss	Akurasi	F1-score	Treshold	Durasi (s)
1000	0.009	0.997	1.00	0.05	4.59
1100	0.009	0.996	1.00	0.10	5.56
1200	0.055	0.969	1.00	0.95	6.56

User	Final loss	Akurasi	F1-score	Treshold	Durasi (s)
1300	0.005	0.997	1.00	0.50	7.30
1400	0.007	0.999	1.00	0.05	7.27
1500	0.012	0.995	1.00	0.60	8.78
1600	0.009	0.996	1.00	0.90	9.09
1700	0.012	0.993	1.00	0.85	9.70
1800	0.024	0.987	0.67	0.75	10.97

Berdasarkan Tabel 4.6 dapat diamati bahwa performa model relatif stabil dengan nilai akurasi dan F1-score mencapai 1.00 hingga skenario 1700 user. Namun, pada skenario 1800 user, terjadi penurunan F1-score secara signifikan menjadi 0.67, yang mengindikasikan bahwa model mulai mengalami kesulitan dalam generalisasi ketika jumlah user terlalu besar. Nilai final loss juga meningkat di beberapa konfigurasi, sejalan dengan naiknya best threshold menjadi lebih ekstrem pada user 1000 dan 1800, yakni 0.05 dan 0.75. Hal ini menunjukkan bahwa model perlu menyesuaikan ambang prediksi secara tidak wajar, yang bisa menjadi sinyal awal overfitting atau bias.

Sementara itu, waktu training menunjukkan kenaikan secara bertahap dan hampir linear, sebagaimana divisualisasikan dalam grafik berikut.



(1) Grafik batang multimetric (2) Grafik baris peningkatan waktu

Gambar 4.7 Visualisasi skalabilitas model

Gambar 4.7 memperlihatkan dua jenis visualisasi: (1) grafik batang multimetric untuk melihat tren final loss, akurasi, F1-score, dan threshold; serta (2) grafik garis yang menunjukkan peningkatan waktu training terhadap jumlah user. Dapat dilihat bahwa waktu training meningkat dari 5.59 detik (1000 user) menjadi

10.97 detik (1800 user), mengindikasikan bahwa kompleksitas model meningkat secara sejalan dengan skala data.

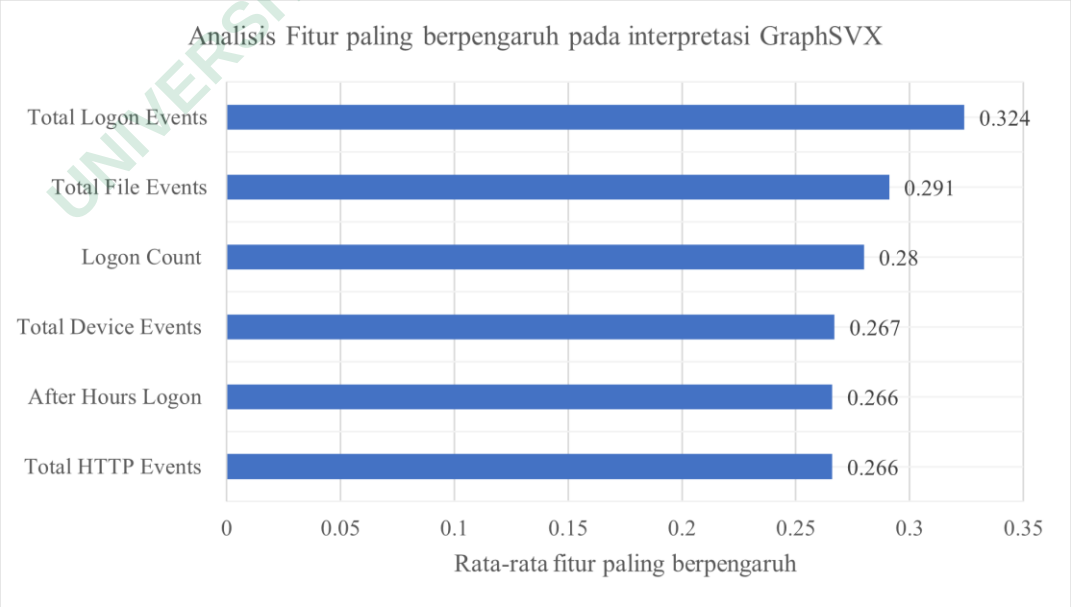
Untuk menentukan jumlah user terbaik, digunakan kriteria yang ditampilkan pada bagian bawah tabel, yaitu:

1. F1-Score dan Akurasi tinggi (idealnya 1.00)
2. Final Loss rendah (mendekati 0)
3. Threshold dalam rentang wajar (0.4–0.7)
4. Waktu training tidak berlebihan (idealnya ≤ 7 detik)

Berdasarkan kriteria tersebut, konfigurasi dengan 1300 user dinilai sebagai titik optimal karena menghasilkan performa sempurna (akurasi dan F1 = 1.00), final loss sangat rendah (0.005), threshold stabil (0.50), dan waktu training yang efisien (7.30 detik).

4.4.6 Interpretasi dengan GraphSVX

Interpretasi model merupakan langkah penting dalam memahami bagaimana suatu model membuat prediksi, serta untuk menjelaskan kontribusi setiap fitur terhadap keputusan model. Pada penelitian ini, interpretasi dilakukan menggunakan pendekatan GraphSVX, yaitu metode interpretabilitas berbasis GNN yang mampu menjelaskan kontribusi node dan fitur terhadap hasil prediksi.



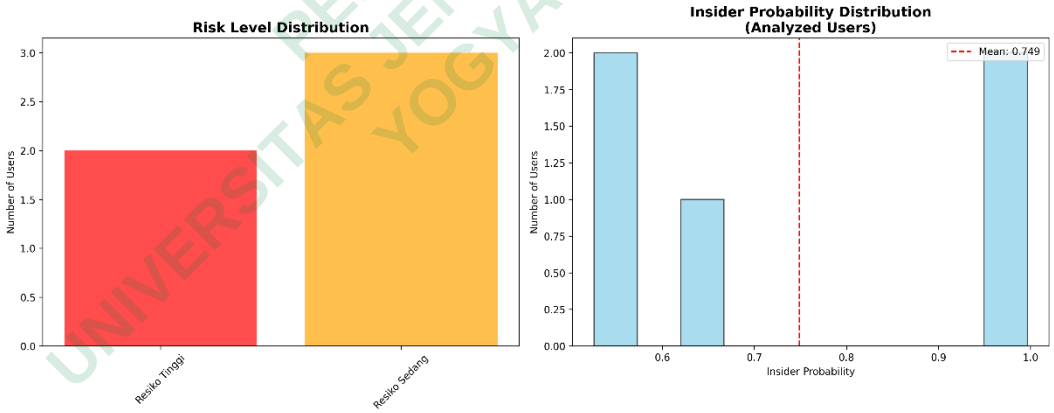
Gambar 4.8 Fitur paling berpengaruh

Pengguna dengan skor tinggi pada *Total Logon Events* dan *Total File Events* menunjukkan perilaku tidak biasa yang dalam konteks organisasi dapat dikaitkan dengan potensi *data exfiltration*.

Dengan demikian, interpretasi dari GraphSVX tidak hanya memberikan kejelasan terhadap keputusan model, tetapi juga memperkuat kepercayaan pengguna terhadap sistem, karena fitur-fitur yang diandalkan selaras dengan ekspektasi dan pengetahuan domain. Analisis ini juga memberikan insight bagi pengambilan keputusan keamanan lebih lanjut, seperti penentuan kebijakan pemantauan atau sistem peringatan dini.

4.4.7 Analisis Statistik Risiko

Dari hasil interpretasi model, dilakukan analisis statistik terhadap tingkat risiko pengguna berdasarkan probabilitas prediksi *insider* yang dihasilkan oleh model GraphSAGE. Visualisasi pada Gambar 4.9 memperlihatkan dua aspek utama dalam analisis risiko, yaitu distribusi tingkat risiko dan distribusi probabilitas klasifikasi *insider*.



Gambar 4.9 Analisis risiko

Grafik *Risk Level Distribution* menunjukkan bahwa sebagian besar pengguna yang dianalisis tergolong dalam kategori risiko sedang, diikuti oleh jumlah yang lebih sedikit pada kategori risiko tinggi. Hal ini menunjukkan bahwa meskipun secara umum aktivitas pengguna terklasifikasi aman, terdapat sebagian kecil pengguna yang menampilkan pola perilaku mencurigakan dengan probabilitas tinggi diklasifikasikan sebagai *insider*.

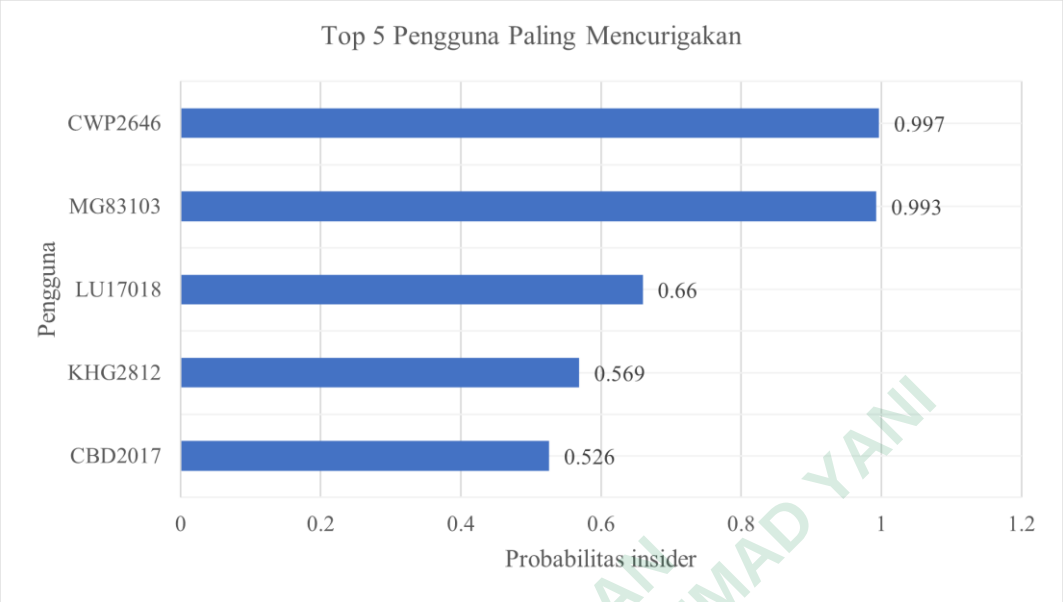
Selanjutnya, grafik *Insider Probability Distribution* menunjukkan distribusi probabilitas klasifikasi terhadap label *insider*. Terlihat bahwa sebagian besar probabilitas prediksi berada di atas ambang batas rata-rata (mean = 0.749), mengindikasikan adanya indikasi kuat terhadap aktivitas berisiko. Model berhasil membedakan pengguna berdasarkan skor probabilitas, sehingga memungkinkan identifikasi lebih awal terhadap potensi ancaman internal.

Berikut adalah tabel analisis statistik risiko pada keseluruhan jumlah data:

Jumlah User	Risiko Rendah	Risiko Sedang	Risiko Tinggi
1000	0	2	3
1100	0	1	4
1200	0	0	5
1300	0	3	2
1400	0	2	3
1500	0	0	5
1600	0	0	4
1700	0	0	5
1800	0	0	5

Dari tabel tersebut, terlihat bahwa semakin banyak jumlah user yang digunakan dalam pelatihan, semakin konsisten model dalam mengidentifikasi pengguna berisiko tinggi. Jumlah pengguna dengan risiko tinggi mencapai angka maksimum (5 pengguna) pada seluruh skenario dari 1200 hingga 1800 user, yang menunjukkan bahwa model telah mampu menggeneralisasi pola perilaku mencurigakan secara stabil di berbagai ukuran data.

Berikut adalah gambar top 5 pengguna paling mencurigakan berdasarkan nilai probabilitas tertinggi yang dihasilkan oleh model terhadap label *insider*. Nilai ini merepresentasikan keyakinan model terhadap kemungkinan seorang pengguna menjadi *insider*, di mana semakin mendekati 1, maka semakin tinggi tingkat kecurigaan terhadap pengguna tersebut.



Gambar 4.10 Visualisasi top 5 pengguna paling mencurigakan

Pada Gambar 4.10 ditampilkan visualisasi yang memperkuat hasil analisis risiko dengan menunjukkan identitas lima pengguna dengan skor probabilitas tertinggi sebagai Insider. Visualisasi ini menegaskan hasil pemodelan risiko sebelumnya, di mana pengguna-pengguna tersebut secara konsisten berada dalam kategori risiko tinggi pada berbagai skenario jumlah data. Keberadaan pengguna-pengguna ini dalam daftar teratas mencerminkan akurasi dan konsistensi model dalam mendeteksi pola perilaku yang menyimpang.

4.4.8 Perbandingan GraphSAGE dan GraphSAGE + GraphSVX

Untuk mengevaluasi dampak dari penambahan metode interpretasi GraphSVX terhadap model GraphSAGE, dilakukan perbandingan performa antara model GraphSAGE murni dan model GraphSAGE yang dilengkapi interpretasi hasil menggunakan GraphSVX. Tabel berikut menyajikan perbandingan metrik utama dari kedua pendekatan:

Tabel 4.7 Perbandingan performa model

Metrik Evaluasi	GraphSAGE	GraphSAGE + GraphSVX
Akurasi	Ada	Ada
Precision	Ada	Ada

Metrik Evaluasi	GraphSAGE	GraphSAGE + GraphSVX
Recall	Ada	Ada
F1-Score	Ada	Ada
ROC-AUC	Ada	Ada
Interpretabilitas	Tidak ada	Ada
Analisis Risiko	Tidak ada	Ada

Dari hasil tersebut dapat dilihat bahwa penambahan GraphSVX tidak memiliki perubahan metrik secara signifikan, namun memberikan keuntungan penting dari sisi transparansi dan interpretasi hasil prediksi Insider.

Pada Tabel 4.7 memperlihatkan perbandingan kinerja antara pendekatan GraphSAGE murni dan integrasi GraphSAGE dengan GraphSVX. Secara metrik klasifikasi (akurasi, precision, recall, dan F1-score), tidak terdapat perbedaan signifikan karena GraphSVX tidak mengubah arsitektur atau bobot model, melainkan hanya digunakan untuk menjelaskan hasil prediksi. Namun, pendekatan kedua memberikan keuntungan tambahan dari sisi interpretabilitas, yang sangat penting dalam konteks keamanan siber. Dengan GraphSVX, prediksi model tidak hanya menghasilkan label, tetapi juga menyertakan alasan di balik keputusan tersebut melalui kontribusi fitur.

Hal ini sangat krusial dalam konteks keamanan siber, di mana tidak hanya akurasi yang penting, tetapi juga pemahaman terhadap alasan di balik setiap prediksi. Dengan adanya interpretasi fitur melalui GraphSVX, model tidak hanya mampu mendeteksi potensi ancaman internal, tetapi juga menjelaskan faktor-faktor perilaku apa yang mendorong keputusan tersebut. Oleh karena itu, integrasi antara model GraphSAGE dan GraphSVX dapat dianggap sebagai pendekatan yang seimbang antara performa dan interpretabilitas dalam sistem deteksi ancaman internal.