# BAB 3
# METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimen kuantitatif dengan fokus pada implementasi GNN dalam mendeteksi pola ancaman internal berbasis perilaku pengguna. Dataset yang digunakan adalah CERT Insider Threat r6.2, yang merepresentasikan aktivitas pengguna dalam lingkungan organisasi. Model utama yang digunakan adalah GraphSAGE, yang diimplementasikan pada struktur graf heterogen untuk menghasilkan representasi vektor tiap entitas. GraphSVX digunakan sebagai metode XAI untuk meningkatkan interpretabilitas hasil deteksi ancaman. Dataset mencakup log aktivitas pengguna seperti login/logoff, akses file, koneksi perangkat eksternal, dan kunjungan web, yang dikonversi ke dalam struktur graf guna mengungkap pola interaksi yang menyimpang.

Struktur graf dibangun dengan tiga jenis node utama, yaitu user, pc, dan url, yang diperoleh melalui proses agregasi log dari beberapa file. Setiap node memiliki fitur yang mencerminkan perilaku agregat pengguna atau entitas tersebut dalam kurun waktu pengamatan. Tabel 3.1 berikut merangkum fitur pada masing-masing node:

**Tabel 3.1** Fitur node dalam struktur graf

| Node | Fitur utama |
|---|---|
| user | role_encoded, department_encoded, total_logon_events, total_file_events, total_device_events, total_http_events |
| pc | unique_users_count, avg_daily_logons, total_file_operations, total_device_connections |
| url | domain_category, total_visits, unique_visitors |

Struktur edge dalam graf merepresentasikan tiga jenis relasi utama:

1. user → pc, berdasarkan aktivitas autentikasi, akses file, dan koneksi perangkat,
2. user → url, berdasarkan aktivitas *browsing*,

3. user → user, dibentuk jika dua pengguna mengakses pc yang sama, sebagai bentuk interaksi perilaku secara implisit antar pengguna.

Selain fitur pada node, penelitian ini juga memperhatikan relasi antar entitas yang direpresentasikan dalam bentuk edge pada struktur graf. Setiap relasi diberi bobot berdasarkan frekuensi atau intensitas interaksi yang teramati. Tabel 3.2 merinci fitur pada masing-masing edge:

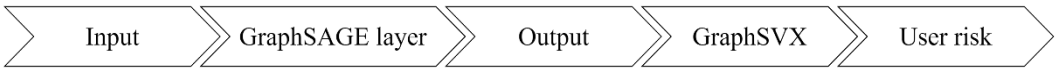
**Tabel 3.2** Fitur edge dalam struktur graf

| Edge Tipe | Fitur utama |
|---|---|
| user → pc | logon_count, logoff_count, file_open_count, file_write_count, file_copy_count, file_delete_count, device_connect_count, device_disconnect_count, after_hours_logon, weekend_logon |
| user → url | visit_frequency, unique_visit_days, after_hours_browsing, cloud_service_visits, job_site_visits |
| user → user | dipilih berdasarkan jumlah pc yang sama digunakan bersama |

Pemodelan struktur graf dalam penelitian ini didasarkan pada asumsi bahwa perilaku pengguna dapat direpresentasikan melalui interaksi multi-entitas yang tercatat dalam log sistem, seperti frekuensi akses file, aktivitas login di luar jam kerja, dan kunjungan ke situs penyimpanan atau pencarian kerja. Node "user" menjadi fokus utama klasifikasi karena mewakili aktor dalam sistem, sedangkan node "pc" dan "url" menjadi bagian dari konteks perilaku yang dikaitkan melalui edge dengan bobot interaksi spesifik. Pemilihan fitur pada edge, seperti after_hours_logon, device_connect_count, atau job_site_visits, merepresentasikan indikator perilaku menyimpang yang relevan dengan potensi ancaman internal. Penelitian ini hanya melakukan pengamatan pada skenario 2 dan 5 dalam dataset CERT Insider Threat, yang merepresentasikan pola-pola nyata dari ancaman internal berbasis perubahan perilaku sebelum terjadinya insiden. Label *ground truth* ditentukan berdasarkan informasi eksplisit dari dokumentasi resmi dataset CERT Insider Threat, di mana pengguna yang terlibat dalam skenario 2 dan 5 secara langsung dikategorikan sebagai pelaku ancaman internal. Dengan pendekatan ini, struktur graf tidak hanya merepresentasikan hubungan teknis antar entitas, tetapi

juga mengkodekan jejak perilaku yang digunakan untuk mendeteksi risiko secara lebih kontekstual dan dapat dijelaskan.

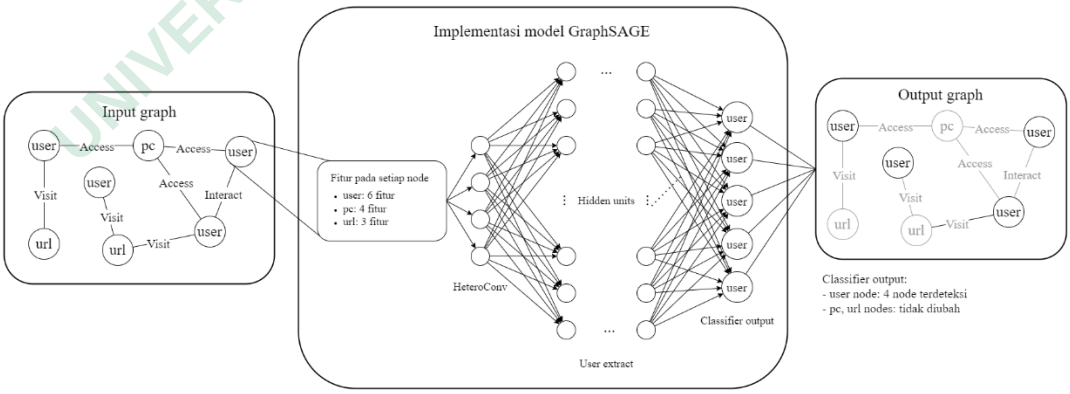
Tahap awal dalam penelitian ini adalah mengubah dataset CERT Insider Threat r6.2 menjadi struktur graf. Kemudian data yang terdiri dari berbagai aktivitas pengguna direpresentasikan sebagai graf berarah (*directed graph*).




**Gambar 3.1** Visualisasi alur proses

Gambar 3.1 menunjukkan alur proses utama dalam sistem yang dibangun, dimulai dari input berupa data graf heterogen hasil ekstraksi fitur, yang kemudian diproses melalui model GraphSAGE untuk menghasilkan prediksi klasifikasi ancaman. Output dari model kemudian dianalisis menggunakan GraphSVX untuk memberikan interpretasi terhadap keputusan model. Hasil akhirnya adalah penilaian risiko pengguna (*user risk*) berdasarkan kontribusi fitur perilaku terhadap klasifikasi yang dilakukan.

Model GraphSAGE digunakan untuk menghasilkan *embedding vektor* dari setiap entitas dengan cara mengagregasi informasi dari tetangga-tetangganya. *Embedding* ini kemudian digunakan untuk memprediksi apakah suatu entitas, khususnya user, menunjukkan indikasi sebagai ancaman internal. Implementasi model GraphSAGE ditujukan pada Gambar 3.2.



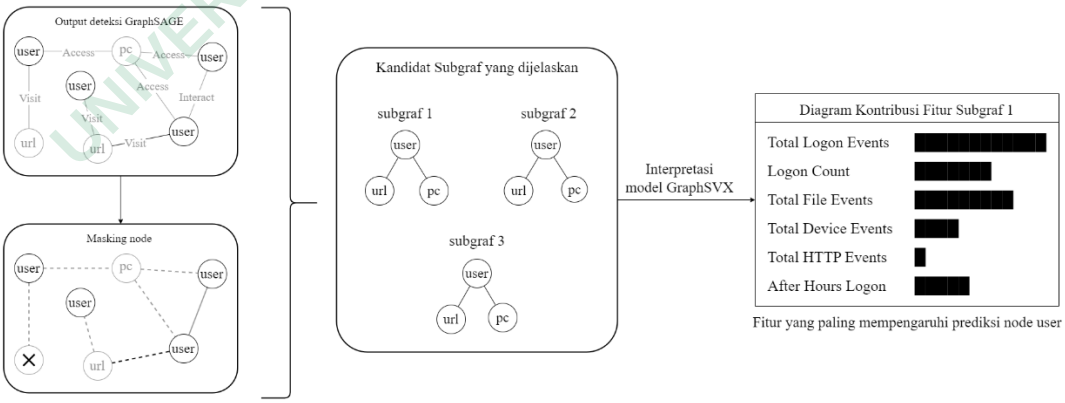

**Gambar 3.2** Model GraphSAGE

Model GraphSAGE yang dilakukan dalam penelitian ini didesain untuk memproses struktur graf heterogen, dengan tiga tipe node utama dan berbagai jenis

relasi antar node. Pada tahap pelatihan, semua node dan edge digunakan untuk membentuk representasi perilaku dalam konteks graf. Proses agregasi dilakukan melalui modul HeteroConv, yang memungkinkan model untuk menangkap konteks relasi antar tipe entitas berbeda. Setiap node menggabungkan informasi dari tetangganya sesuai dengan jenis relasi yang ada, kemudian menghasilkan embedding melalui lapisan GraphSAGE secara bertingkat.

Meskipun struktur graf terdiri dari node user, pc, dan url, output akhir dari model difokuskan hanya pada node bertipe user, sejalan dengan tujuan utama penelitian ini, yaitu mendeteksi potensi ancaman internal berbasis perilaku pengguna. Oleh karena itu, hasil prediksi klasifikasi disajikan dalam bentuk tensor embedding dari node user, sementara informasi dari node pc dan url tetap dimanfaatkan selama proses pembelajaran representasi untuk memperkaya konteks perilaku pengguna. Strategi ini tidak hanya meningkatkan efisiensi proses klasifikasi, tetapi juga mempertahankan relevansi pola interaksi antar entitas dalam organisasi. Tensor hasil klasifikasi tersebut kemudian digunakan sebagai masukan dalam tahap interpretasi model menggunakan GraphSVX.

Sebagai upaya untuk meningkatkan transparansi hasil prediksi, metode GraphSVX diterapkan sebagai pendekatan XAI. GraphSVX bekerja dengan mengekstraksi subgraf lokal yang memiliki kontribusi paling signifikan terhadap keputusan model. Implementasi metode GraphSVX ditunjukkan pada Gambar 3.3.




**Gambar 3.3** Model Penjelasan GraphSVX

Gambar diatas memperlihatkan proses interpretasi hasil klasifikasi node oleh GraphSAGE. Node utama yang dianalisis adalah user yang diprediksi sebagai

entitas mencurigakan. GraphSVX membangkitkan sejumlah kandidat subgraf dari lingkungan lokal node tersebut, lalu memilih satu subgraf yang paling signifikan terhadap keputusan model.

Untuk menyederhanakan interpretasi terhadap fitur dan atribut yang memengaruhi prediksi, ditampilkan juga visualisasi berupa diagram batang SHAP (SHAP *bar chart*). Diagram ini menggambarkan kontribusi relatif fitur-fitur seperti jumlah akses file, waktu login, dan jumlah perangkat yang digunakan. Visualisasi ini bertujuan untuk memberikan wawasan intuitif terhadap keputusan model dan mendukung analisis lebih lanjut terhadap pola perilaku mencurigakan.

Selanjutnya, proses interpretasi dilakukan dengan melakukan masking node secara selektif pada lingkungan graf di sekitar user target. Proses *masking* ini memungkinkan GraphSVX untuk mengevaluasi pengaruh struktur dan atribut lokal terhadap prediksi yang dihasilkan oleh model. Beberapa kandidat subgraf dihasilkan, masing-masing merepresentasikan kombinasi koneksi langsung dengan entitas url dan pc.

Dari kumpulan kandidat tersebut, dipilih satu subgraf yang memberikan kontribusi terbesar terhadap keputusan klasifikasi. Interpretasi akhir kemudian divisualisasikan dalam bentuk diagram kontribusi fitur terhadap subgraf terpilih. Panjang batang menunjukkan besar pengaruh fitur seperti Total Logon Events, Logon Count, hingga After Hours Logon terhadap prediksi. Dengan pendekatan ini, tidak hanya diperoleh keputusan klasifikasi, tetapi juga pemahaman mendalam mengenai alasan di balik keputusan tersebut.



## 3.1 Bahan dan Alat Penelitian

Dalam penelitian ini, dataset yang digunakan merupakan dataset sekunder dari CERT Insider Threat r6.2, yang dikembangkan oleh Carnegie Mellon University melalui CERT Coordination Center (CERT-CC). Dataset ini dirancang untuk mensimulasikan berbagai aktivitas pengguna dalam lingkungan organisasi guna mendukung penelitian di bidang deteksi ancaman internal. CERT Insider Threat r6.2 telah menjadi salah satu dataset standar dalam penelitian keamanan

siber, terutama dalam pengembangan model berbasis ML dan DL untuk deteksi aktivitas mencurigakan dalam sistem informasi perusahaan.

Dataset ini berisi berbagai jenis dataset yang mencerminkan interaksi pengguna dalam suatu organisasi, termasuk log aktivitas login dan logout, akses file, komunikasi email, penggunaan perangkat eksternal, serta aktivitas browsing. Dengan cakupan dataset yang luas, dataset ini memberikan kesempatan bagi peneliti untuk memodelkan pola perilaku pengguna serta mendeteksi aktivitas yang berpotensi menjadi ancaman. Pemilihan dataset CERT Insider Threat r6.2 dalam penelitian ini didasarkan pada beberapa faktor utama berikut:

1. Ketersediaan Dataset Perilaku yang Relevan

   Dataset ini mencakup berbagai interaksi pengguna dalam sistem organisasi, termasuk login, akses file, komunikasi melalui email, serta penggunaan perangkat eksternal. Informasi ini sangat penting untuk memahami pola perilaku pengguna dan mengidentifikasi aktivitas yang menyimpang dari kebiasaan normal.
2. Struktur Dataset yang Sesuai untuk Pemodelan Graf

   Struktur dataset ini dapat direpresentasikan dalam bentuk *graph-based structure*, di mana pengguna dapat dihubungkan dengan berbagai entitas lain, seperti file, perangkat, dan komunikasi antar pengguna. Struktur ini sangat sesuai dengan pendekatan GNN yang diterapkan dalam penelitian ini untuk mendeteksi pola ancaman internal.
3. Digunakan secara Luas dalam Penelitian Keamanan Siber

   Dataset ini telah digunakan dalam berbagai studi terkait ancaman internal, baik yang berbasis pembelajaran mesin maupun metode analitik lainnya.

Dataset CERT Insider Threat r6.2 terdiri dari berbagai file yang mencerminkan aktivitas pengguna dalam sistem organisasi. Tidak semua file digunakan dalam penelitian ini. Pemilihan file didasarkan pada relevansinya dengan deteksi pola perilaku pengguna yang mencurigakan. File yang digunakan serta fitur utama dalam penelitian ini dapat dilihat pada Tabel 3.3.

**Tabel 3.3** Fitur pada dataset CERT Insider Threat r6.2

| No. | Nama File | Fitur | Deskripsi |
|---|---|---|---|
| 1 | *logon.csv* | user, pc, date, activity | Merekam aktivitas login dan *logout* pengguna, termasuk waktu dan perangkat yang digunakan. |
| 2 | *file.csv* | user, filename, activity, to_removable_media | Mencatat aktivitas akses file oleh pengguna, termasuk pemindahan file ke perangkat eksternal. |
| 3 | *device.csv* | user, device_type, activity | Berisi catatan penggunaan perangkat eksternal seperti USB, *hard drive eksternal*, dan perangkat lainnya. |
| 4 | *http.csv* | user, url, activity, content | Merekam aktivitas browsing pengguna, termasuk situs yang dikunjungi dan aktivitas yang dilakukan. |
| 5 | *users.csv* | user_id, role, department | Berisi informasi tentang pengguna, termasuk jabatan, departemen, dan supervisor. |



Dengan menggunakan dataset CERT Insider Threat r6.2, penelitian ini dapat mengembangkan model berbasis GNN yang mampu mendeteksi pola perilaku mencurigakan dalam suatu organisasi. Selain itu, pendekatan XAI yang

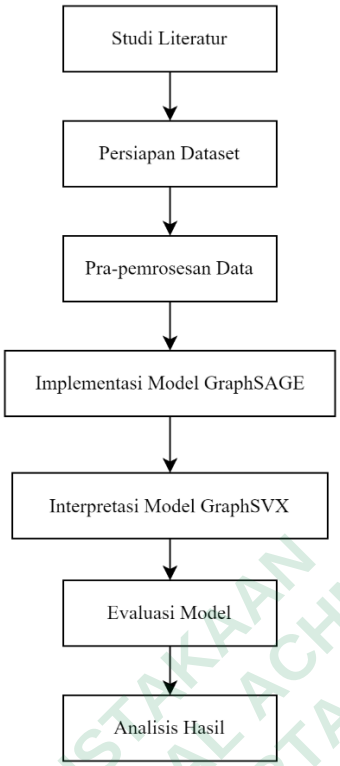
diterapkan dalam penelitian ini bertujuan untuk meningkatkan transparansi dalam proses deteksi ancaman, sehingga memungkinkan interpretasi hasil yang lebih jelas bagi pengguna sistem keamanan siber.

Untuk mendukung pelaksanaan penelitian ini, diperlukan perangkat komputer dengan spesifikasi yang memadai digunakan untuk menjalankan sistem operasi serta perangkat lunak yang mendukung analisis dataset dan konektivitas internet. Sistem Operasi dan program-program aplikasi yang dipergunakan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 11 Home (versi 24H2)
2. Prosesor: 11th Gen Intel® Core™ i5-1155G7 @ 2.50 GHz
3. GPU: Intel(R) Iris(R) Xe Graphics
4. Memori RAM: 8 GB DDR4
5. Penyimpanan Internal: 512 GB SSD-NVMe
6. Software: Visual Studio Code, Google Colab
7. Pustaka: Pytorch Geometric, Scikit-learn, Pandas, Numpy, Matplotlib, Seaborn

## 3.2 Jalan Penelitian

Jalan penelitian ini menggambarkan tahapan sistematis yang ditempuh dalam mengembangkan, mengimplementasikan, serta mengevaluasi model deteksi ancaman internal berbasis GNN dan teknik interpretasi model (XAI). Penelitian ini dilakukan secara bertahap dan sistematis, sebagaimana divisualisasikan pada Gambar 3.5 dan dijelaskan berikut ini.




**Gambar 3.4** Jalan penelitian

Agar mempermudah pemahaman, tiap langkah dalam proses penelitian dijelaskan melalui sub-bab berikut:

### 3.2.1 Studi Literatur

Penelitian dimulai dengan eksplorasi literatur sebagai pondasi teoritis. Tahap ini mencakup peninjauan terhadap berbagai karya ilmiah dan publikasi yang membahas pemanfaatan GNN dan metode XAI dalam konteks keamanan siber, khususnya deteksi ancaman internal. Temuan dari studi literatur ini menjadi referensi utama dalam menentukan pendekatan yang paling relevan dan aplikatif untuk kasus yang diangkat.

### 3.2.2 Persiapan Dataset

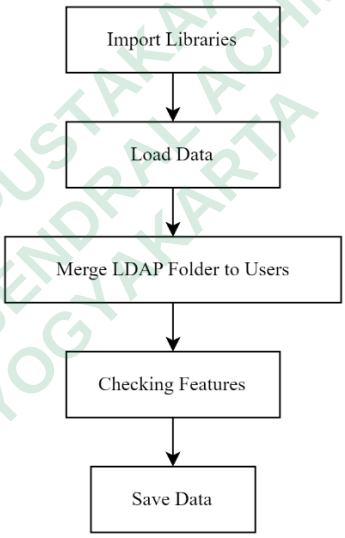
Setelah memahami landasan teoritis, langkah selanjutnya adalah menentukan dan mengunduh dataset yang sesuai. Dataset yang digunakan adalah CERT Insider Threat r6.2, yang merupakan data simulasi dari aktivitas pengguna dalam organisasi. Dataset ini banyak dipakai dalam penelitian keamanan siber karena menggambarkan skenario realistis dari perilaku ancaman internal.

### 3.2.3 Pra-pemrosesan Data

Pra-pemrosesan data dalam penelitian ini terdiri atas dua tahapan utama, yaitu tahap *pre-load* dan tahap *pre-processing*. Tahap ini krusial untuk memastikan bahwa data yang digunakan dalam pembangunan model telah bersih, terstruktur, dan kompatibel dengan format graf heterogen yang dibutuhkan oleh GraphSAGE.

a. Tahap *Pre-load*

Sebelum membentuk struktur graf, dilakukan proses *pre-load* untuk menggabungkan berbagai sumber informasi yang terkait dengan pengguna. Tujuan utamanya adalah menyiapkan data awal dengan mengintegrasikan atribut pengguna, seperti *department* dan *role*, dari direktori LDAP ke data aktivitas. Alur proses ini digambarkan pada Gambar 3.5.




**Gambar 3.5** Alur *Pre-load*



Langkah-langkahnya adalah sebagai berikut:

- *Import Libraries*

  Mengimpor pustaka yang diperlukan seperti pandas, numpy, dan library tambahan untuk manipulasi data.

- *Load Data*

  Memuat data mentah dari file log aktivitas dan data kepegawaian yang relevan.

- *Merge LDAP Folder to Users*

Mengintegrasikan data pengguna dengan struktur organisasi berdasarkan direktori LDAP. Ini penting untuk mendapatkan atribut seperti department dan role yang akan digunakan sebagai fitur.

- *Checking Features*

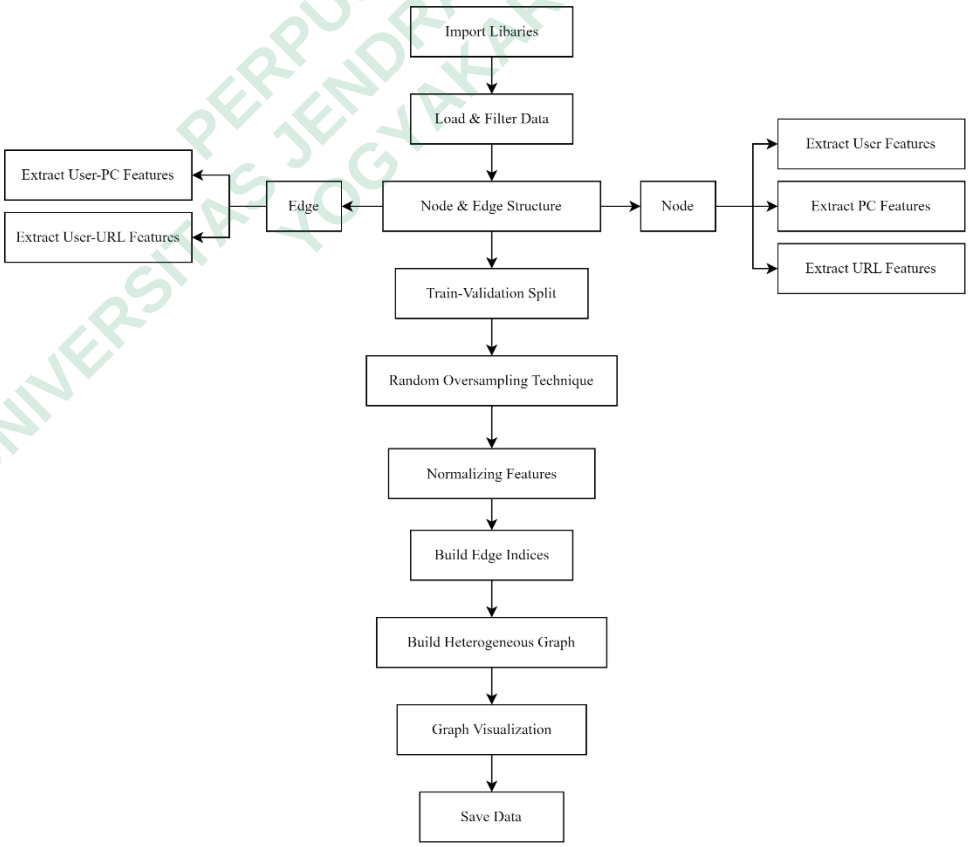
  Melakukan pengecekan dan validasi terhadap atribut pengguna yang akan diekstrak, memastikan tidak ada nilai kosong atau tidak valid.

- *Save Data*

  Data yang telah digabung dan divalidasi kemudian disimpan sebagai dataset dasar untuk proses selanjutnya.

b. Tahap *Pre-processing*

Tahap ini bertujuan membentuk struktur graf heterogen dari data yang telah dikonsolidasikan. Proses *pre-processing* mencakup ekstraksi fitur, pembuatan *edge*, hingga normalisasi data. Alur lengkap dari tahapan ini ditunjukkan pada Gambar 3.6.




**Gambar 3.6** Alur *Pre-processing*

Tahapannya sebagai berikut:

- *Import Libraries*

  Library tambahan untuk proses graf, seperti torch, torch_geometric, dan library visualisasi graf.

- *Load & Filter Data*

  Data yang telah disimpan dari tahap pre-load dimuat kembali. Filtering dilakukan untuk membatasi entitas dan interaksi sesuai ruang lingkup penelitian (misalnya, hanya *user, pc, url* dan aktivitas utama seperti *logon*, *file access*, dan *browsing*).

- *Node & Edge Structure*

  Menentukan struktur simpul (*node*) dan sisi (*edge*) berdasarkan jenis entitas serta relasi interaksi yang dianalisis.

- *Extract Node Features*
  - *User Features*: Ekstraksi atribut seperti *role*, *department*, dan statistik aktivitas pengguna.
  - *PC Features*: Meliputi jumlah pengguna unik, aktivitas file, dan frekuensi *logon*.
  - *URL Features*: Berisi informasi kategori domain, jumlah kunjungan, dan statistik lainnya.

- *Extract Edge Features*
  - *User-PC*: Mencakup interaksi seperti *logon/logoff*, akses file, dan koneksi perangkat.
  - *User-URL*: Mencakup frekuensi kunjungan, waktu akses (*after hours*), serta situs yang dikategorikan sebagai layanan cloud atau *job site*.

- *Train-Validation Split*

  Membagi dataset menjadi dua bagian, yaitu data latih (90%) dan data validasi (10%) untuk kebutuhan pelatihan dan evaluasi awal model.

- *Random Oversampling Technique*

Untuk menangani ketidakseimbangan data label ancaman, digunakan teknik oversampling terhadap kelas minoritas. Metode ini dilakukan dengan menggandakan secara acak node user yang termasuk kelas ancaman (*threat*), sehingga jumlahnya mendekati kelas mayoritas (*normal*). Oversampling dilakukan hanya pada data latih untuk mencegah *data leakage* ke data validasi.

- *Normalizing Features*

  Fitur dinormalisasi agar skala antar fitur sebanding dan tidak bias terhadap model. Proses normalisasi dilakukan dengan pendekatan *Min-Max Scaling* ke rentang [0, 1], menggunakan nilai maksimum dan minimum pada masing-masing fitur secara global.
- *Build Edge Indices*

  Menyusun *edge index* dari relasi antar node sebagai input utama untuk model GNN.
- *Build Heterogeneous Graph*

  Menggabungkan seluruh informasi menjadi objek graf heterogen menggunakan library PyTorch Geometric.
- *Graph Visualization*

  Visualisasi graf dilakukan untuk validasi struktur serta mendapatkan gambaran umum dari interaksi antar entitas.
- *Save Data*

  Dataset akhir yang telah berbentuk graf disimpan dalam format .pt untuk digunakan pada tahap pelatihan model.



### 3.2.4 Implementasi Model GraphSAGE

Dengan dataset yang siap, proses utama pun dimulai: membangun model GraphSAGE menggunakan framework PyTorch Geometric. Model dilatih untuk mengidentifikasi pola mencurigakan pada node pengguna. Parameter pelatihan disesuaikan secara eksperimental untuk memperoleh hasil terbaik.

### 3.2.5 Interpretasi Model GraphSVX

Setelah model GraphSAGE selesai dilatih dan menghasilkan prediksi, tahap selanjutnya adalah memahami alasan di balik keputusan tersebut. Untuk itu, digunakan pendekatan interpretasi berbasis GraphSVX, yang membangkitkan sejumlah subgraf lokal dari node target. GraphSVX mengevaluasi kontribusi setiap fitur dan hubungan (*edge*) dalam subgraf terhadap keputusan model, sehingga memberikan pemahaman yang lebih transparan mengenai pola perilaku yang diklasifikasikan sebagai ancaman.

### 3.2.6 Evaluasi Model

Performa model dievaluasi menggunakan sejumlah metrik yang umum digunakan dalam klasifikasi biner, khususnya dalam konteks data yang tidak seimbang (*imbalanced class*). Evaluasi ini bertujuan untuk memberikan gambaran menyeluruh mengenai kekuatan dan kelemahan model dalam mendeteksi ancaman internal. Adapun metrik evaluasi yang digunakan adalah sebagai berikut:

- Accuracy

  Mengukur proporsi prediksi yang benar terhadap seluruh data:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

  Di mana:

  - TP (*True Positive*): jumlah ancaman yang terdeteksi dengan benar
  - TN (*True Negative*): jumlah non-ancaman yang diklasifikasi dengan benar
  - FP (*False Positive*): jumlah non-ancaman yang salah diklasifikasikan sebagai ancaman
  - FN (*False Negative*): jumlah ancaman yang gagal terdeteksi

- Precision

  Mengukur akurasi deteksi positif, yaitu seberapa banyak prediksi positif yang benar:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

- Recall

  Mengukur kemampuan model dalam mendeteksi ancaman aktual:

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

- F1-Score

  Merupakan harmonic mean antara precision dan recall, digunakan saat distribusi kelas tidak seimbang:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (7)$$

- AUC-ROC (*Area Under Curve – Receiver Operating Characteristic*)

  Mengukur kemampuan model dalam membedakan antara kelas positif dan negatif. Semakin mendekati 1, semakin baik performa klasifikasinya.

Evaluasi ini dilakukan pada data validasi dan digunakan untuk menilai kemampuan generalisasi model GraphSAGE dalam mengidentifikasi node user yang menunjukkan pola perilaku ancaman internal. Dengan metrik-metrik ini, model tidak hanya dinilai dari ketepatan klasifikasinya, tetapi juga dari sensitivitas dan keseimbangannya dalam mengenali kasus ancaman yang jumlahnya sangat sedikit.

### 3.2.7 Analisis Hasil

Setelah model dievaluasi dan diinterpretasi, tahap ini mengeksplorasi temuan yang muncul. Analisis difokuskan pada sejauh mana kombinasi GraphSAGE dan GraphSVX mampu memodelkan dan menjelaskan perilaku pengguna yang mencurigakan. Hasil ini menjadi bukti bahwa integrasi GNN dan XAI memiliki potensi kuat dalam mendeteksi ancaman internal secara lebih transparan.