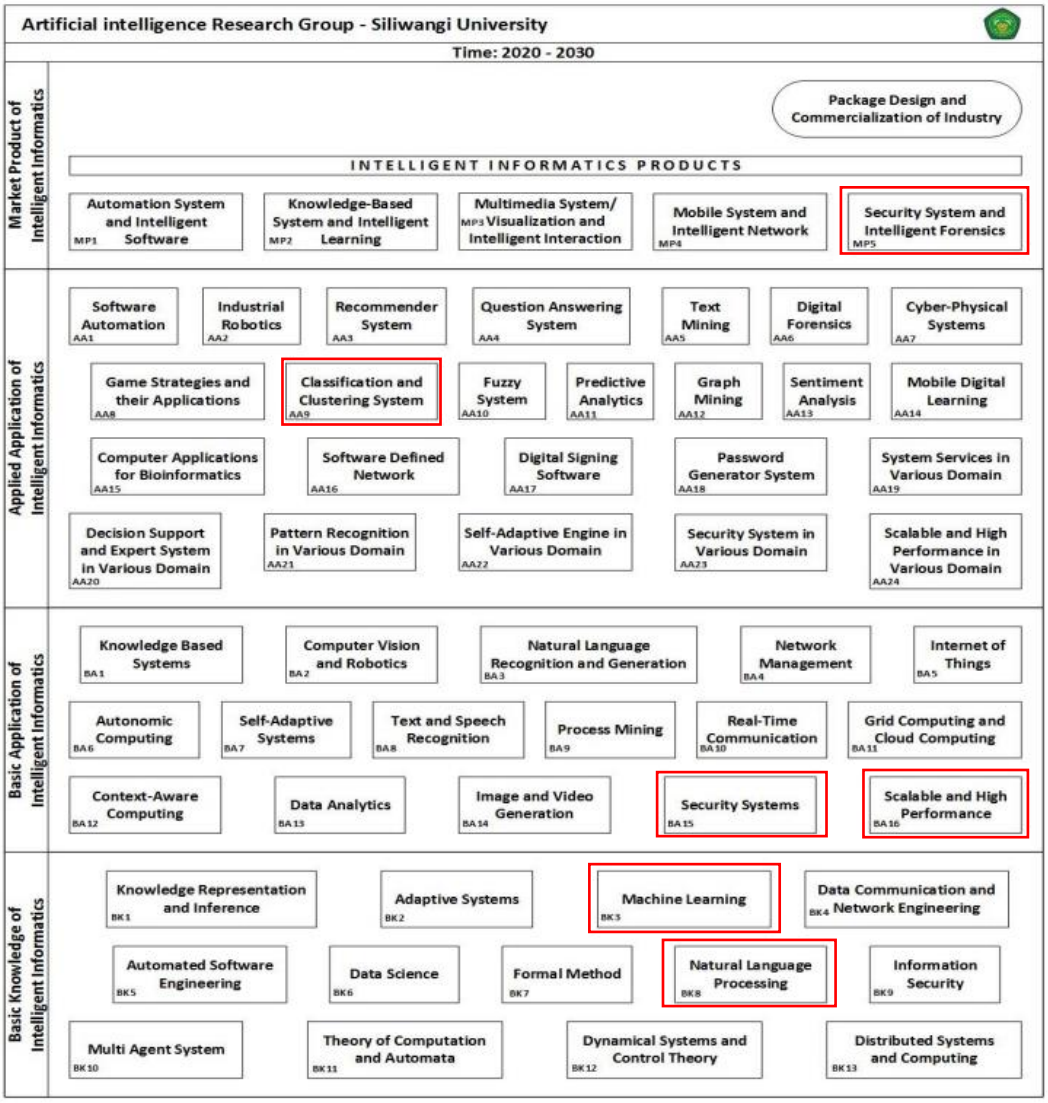
BAB III

METODOLOGI PENELITIAN

3.1 *Roadmap* Penelitian

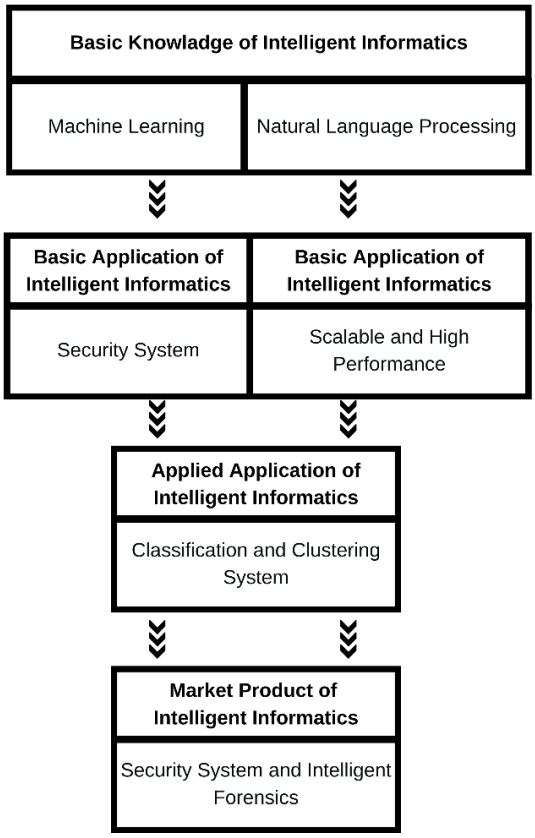


Gambar 3. 1 *Roadmap* penelitian (AIS Universitas Siliwangi, 2020)

Secara keseluruhan, arah penelitian yang dilakukan selaras dengan *roadmap* penelitian Universitas Siliwangi dalam sub-bidang *Artificial Intelligence (AI)* dalam penerapan model *deep learning* untuk keamanan jaringan. Peta jalan penelitian yang mendukung kajian ini dapat dilihat pada Gambar 3.1,

menggambarkan kontribusi penelitian dalam mendukung inovasi dan kemajuan teknologi di bidang kecerdasan buatan.

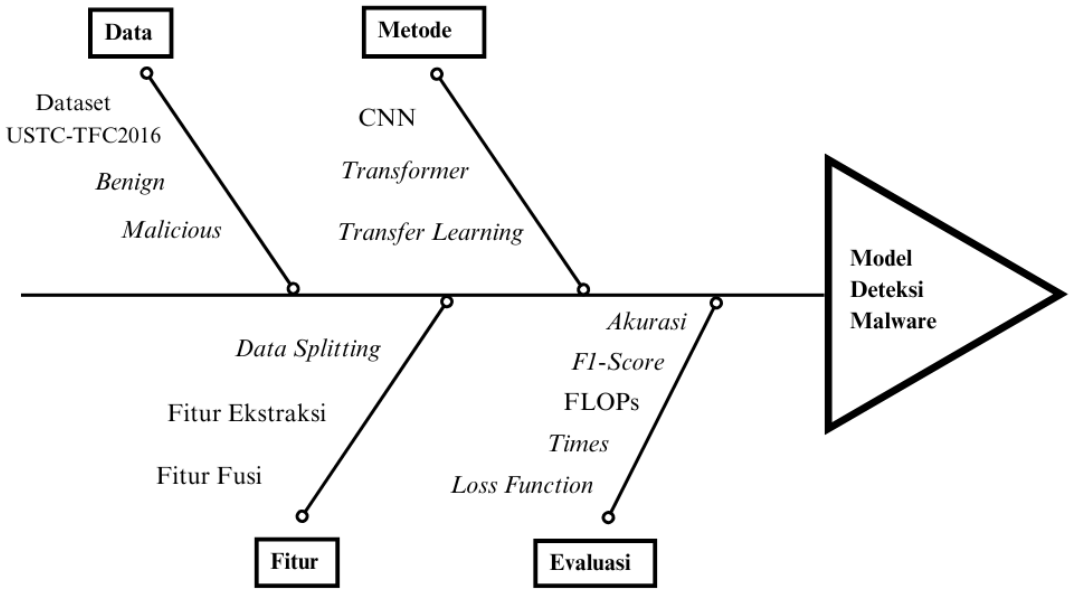
Topik utama pada penelitian yang dilakukan adalah mengenai *CNN* dan *Transformer* yang merupakan bagian dari *machine learning* serta *Natural Language Processsing* (*NLP*) sebagai induk dari *Transformer*. Adapun spesifikasi peta jalan penelitian direpresentasikan pada Gambar 3.2.



Gambar 3. 2 Spesifikasi *roadmap* penelitian

Basis pengetahuan *machine learning* dan *Natural Language Processing (NLP)* pada penelitian ini berfokus pada *Scalable and High Performance* dan *System Security* yang mencerminkan kebutuhan akan sistem keamanan yang optimal agar mampu menangani volume data besar dengan performa tinggi. Hal ini relevan untuk model deteksi *malware* pada klasifikasi lalu lintas jaringan yang

optimal. Kemudian, *Classification and Clustering System* menjadi fokus dalam pengelompokan dan klasifikasi data menggunakan model *AI* seperti *CNN* dan *Transformer* yang sesuai dengan penelitian klasifikasi lalu lintas jaringan berbasis *malware*. Pada alur terakhir yaitu *market product* mencakup *Security System and Intelligent Forensics* yang menitikberatkan pada keamanan siber dan forensik digital, mencakup deteksi dan analisis ancaman *malware*. Dengan demikian, penelitian ini mengoptimalkan model deteksi aktivitas *malware* dengan berbasis *Transformer* dan *CNN* dalam klasifikasi lalu lintas jaringan. Fokus penelitian ini adalah *Classification and Clustering System, Security System and Intelligent Forensics*, dan aspek *Scalable and High Performance* sebagai faktor utama dalam implementasi model yang optimal. Untuk melengkapi *roadmap* penelitian, disajikan diagram *fishbone* pada Gambar 3.3.



Gambar 3. 3 Diagram *fishbone* penelitian

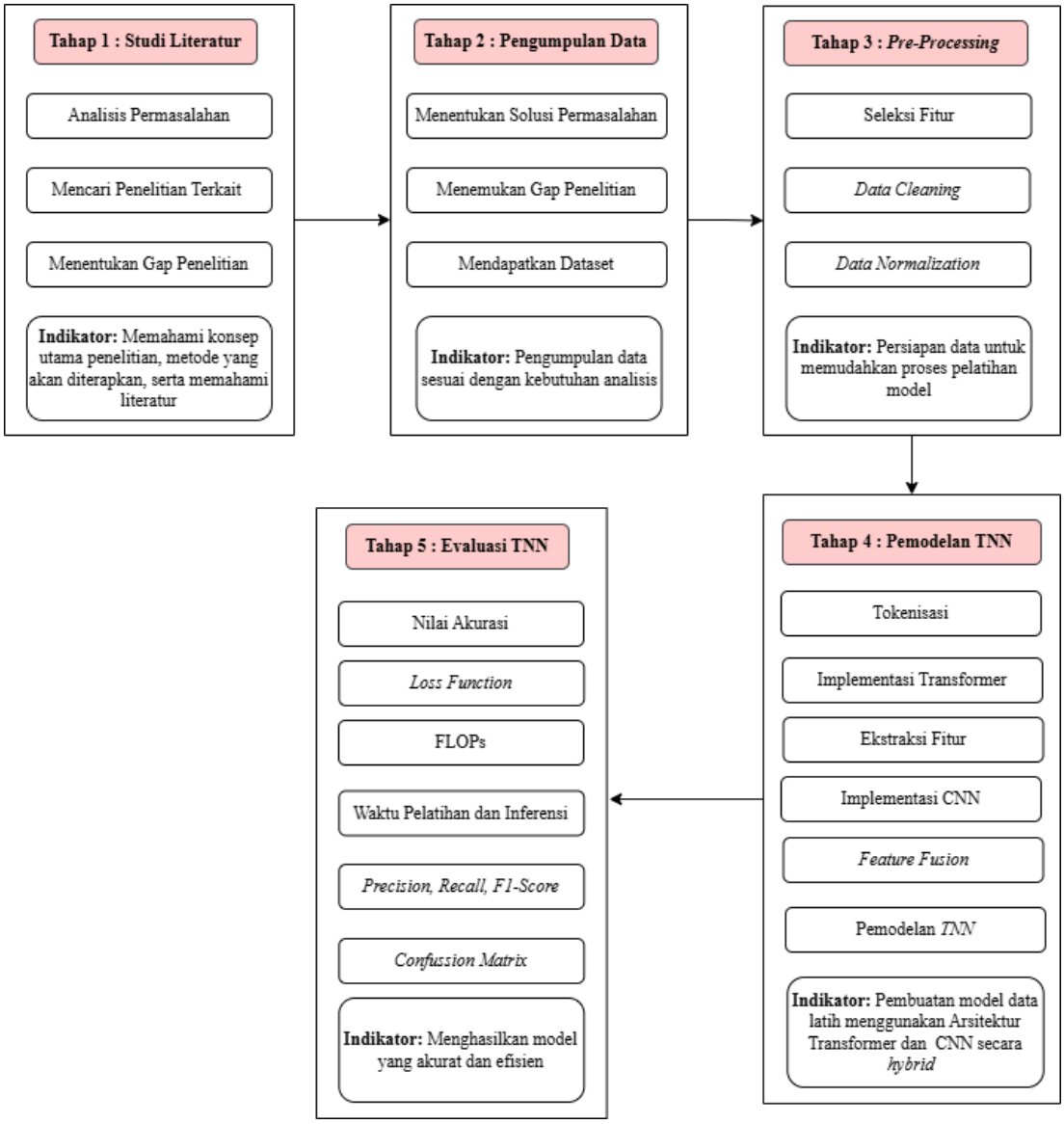
Diagram *fishbone* tersebut dibuat untuk memudahkan dalam mengidentifikasi, mengorganisir, dan menganalisis unsur-unsur potensial dari

permasalahan yang diangkat dalam penelitian deteksi aktivitas *malware* pada lalu lintas jaringan (IBIKKG, 2024). Pada penelitian yang dilakukan berfokus pada aspek data yang digunakan, yaitu dataset *USTC-TFC2016* yang telah diklasifikasikan menjadi kategori *benign* dan *malicious* dengan struktur data berbentuk teks dalam format .csv, aspek metodologi yang diterapkan berupa arsitektur *Transformer* untuk ekstraksi fitur sekuensial dan *Convolutional Neural Network (CNN)* untuk ekstraksi fitur spasial yang diintegrasikan melalui pendekatan *transfer learning* sebagai teknik penggabungan utama.

Aspek operasi fitur yang mencakup tahapan *data splitting* untuk pembagian data latih dan uji, ekstraksi fitur untuk memperoleh representasi yang relevan dari data masukan, serta fitur fusi untuk menggabungkan representasi fitur dari kedua arsitektur, dan aspek evaluasi model yang menggunakan metrik komprehensif meliputi akurasi untuk mengukur ketepatan klasifikasi secara keseluruhan, *F1-Score* untuk mengevaluasi keseimbangan antara *precision* dan *recall*, *Giga Floating Point Operations per Second (GFLOPs)* untuk menganalisis kompleksitas komputasi, serta waktu eksekusi dan inferensi untuk mengevaluasi efisiensi temporal dan dilengkapi *loss function* untuk memantau konvergensi dan kinerja pembelajaran model. Melalui diagram *fishbone* tersebut, membantu dalam proses perancangan sistematis dan realisasi penelitian dengan mengidentifikasi faktor-faktor kritis yang berpengaruh terhadap keberhasilan implementasi model *Trans Neural Network (TNN)* sebagai solusi *hybrid* untuk deteksi aktivitas *malware*.

3.2 Tahapan Penelitian

Inovasi dalam penggabungan arsitektur *CNN* dan *Transformer* serta peningkatan akurasi model menjadi target utama penelitian ini. Adapun tahapan penelitian yang akan dilakukan terdapat pada Gambar 3.4.



Gambar 3. 4 Tahapan penelitian

Berdasarkan Gambar 3.4, dapat dianalisis bahwa tahapan penelitian yang akan dilakukan meliputi 5 tahap sebagai berikut.

1. Studi literatur

Tahapan ini mengkaji dan menganalisis permasalahan yang diangkat disertai dengan referensi penelitian terkait untuk menentukan gap penelitian sehingga menghasilkan pemahaman mendalam mengenai konsep dan rancangan penelitian serta metode yang akan diterapkan (Snyder, 2019). Pada penelitian sebelumnya, terdapat beberapa kesenjangan signifikan dalam pengembangan model deteksi *malware*. Pada model *CNN* konvensional, meski unggul dalam ekstraksi fitur lokal dan pola spasial, tetapi masih memiliki keterbatasan dalam menangkap hubungan sekuensial kompleks dan dependensi jangka panjang. Sementara itu, arsitektur *Transformer* menunjukkan kemampuan superior dalam memahami konteks sekuensial dan hubungan jangka panjang, namun terkadang kurang presisi dalam ekstraksi fitur lokal yang detail. Penelitian awal pada model *hybrid* seperti *SeMalBERT* telah memperlihatkan hasil yang menjanjikan, tetapi masih terdapat ruang pengembangan untuk meningkatkan akurasi melalui desain *hybrid* yang lebih optimal (Liu dkk., 2024b). Maka, *Trans Neural Network (TNN)* dirancang sebagai alternatif solusi untuk meningkatkan akurasi deteksi aktivitas *malware* pada klasifikasi lalu lintas jaringan dengan menggabungkan keunggulan arsitektur *Transformer* dalam pemahaman kontekstual global serta hubungan data jangka panjang dengan kapabilitas *CNN* dalam ekstraksi fitur lokal secara efisien.

2. Pengumpulan data

Proses pengumpulan dataset lalu lintas jaringan yang memuat aktivitas *benign* (tidak berbahaya) dan *malicious* (berbahaya) sebagai data latih model dilakukan dengan merujuk pada penelitian terdahulu. Pada penelitian ini, digunakan dataset

USTC-TFC2016 (Wei Wang dkk., 2017) yang terdiri dari 6 kolom dan sebanyak 5.941.006 baris. Dataset tersebut digunakan dalam penelitian dengan jumlah nilai yang mumpuni agar mampu memberikan dasar yang kuat untuk pengembangan model awal, dan proses eksplorasi dalam meningkatkan kinerja model deteksi *malware*. Selain itu, dataset *USTC-TFC2016* dipilih karena menyediakan representasi yang komprehensif dari berbagai jenis aktivitas *malware* dalam lalu lintas jaringan, sehingga cocok untuk mendukung pengembangan model deteksi *malware* dan bidang terkait (Y. Zhang dkk., 2023).

3. *Pre-processing*

Tahapan *preprocessing* dilakukan pada pengembangan model yang bertujuan untuk mempersiapkan data mentah menjadi siap dimodelkan. Tahapan ini menjadi komponen penting dalam alur penelitian dan pengembangan model *Trans Neural Network (TNN)* untuk deteksi aktivitas *malware* pada klasifikasi lalu lintas jaringan yang berperan penting dalam memastikan kualitas dan kesesuaian data sebelum dimasukkan ke dalam arsitektur model *hybrid* yang menggabungkan arsitektur *Transformer* dan *Convolutional Neural Network* (*CNN*). Pada tahap ini juga diawali dengan melakukan *import library* yang dibutuhkan dalam proses perancangan model. Adapun library yang digunakan pada penelitian ini disajikan pada Tabel 3.1.

Tabel 3. 1 Library Python yang digunakan dalam penelitian.

Nama *Library*	**Fungsi**
Pandas	Manipulasi dan analisis data tabular, seperti pembacaan dan pengolahan dataset (Pandas Documentation, 2025).
Numpy	Operasi komputasi numerik dan manipulasi *array* (Numpy Documentation, 2025).
Matplotlib	Visualisasi data dan hasil evaluasi model dalam bentuk grafik atau plot (Matplotlib Documentation, 2021).

Nama *Library*	Fungsi
Seaborn	Visualisasi data statistik, khususnya untuk plot seperti *heatmap* dan *confusion matrix* (Waskom, 2021).
Time	Mengukur waktu eksekusi proses tertentu, seperti pelatihan model (Python Software Foundation, 2025b).
Regex (re)	Pencocokan dan manipulasi *string* menggunakan *regular expressions* (Python Software Foundation, 2025a).
Chardet	Mendeteksi *encoding file* saat pembacaan data teks (Pilgrim dkk., 2015).
PyTorch	Pembangunan model, definisi arsitektur jaringan, *DataLoader*, dan menjalankan fungsi aktivasi (Linux Foundation, 2017).
Transformers	Implementasi model *pre-trained*, termasuk *tokenizer* dan konfigurasi model *Transformer* (Zalando, 2019).
AdamW	*Optimizer* berbasis Adam dengan *weight decay*, digunakan dalam pelatihan model BERT (The Linux Foundation, 2021).
Scikit-Learn Model Selection	Membagi dataset menjadi data latih dan uji secara proporsional (scikit-learn developers, 2024a).
Scikit-Learn Preprocessing	Mengubah label kategorikal menjadi numerik pada *LabelEncoder* (scikit-learn developers, 2024b) .
Scikit-Learn Metrics	Evaluasi model menggunakan metrik seperti akurasi, presisi, *recall*, *F1-score*, dan *confusion matrix* (scikit-learn developers, 2012).
Thop	Menghitung dan memformat jumlah operasi *GFLOPs* untuk mengukur efisiensi model (Python community, 2022).

Setelah dilakukan import dataset dan dimuat dalam lingkungan pemrograman, langkah selanjutnya adalah melakukan akuisisi atau pengumpulan secara keseluruhan dataset dalam satu *dataframe* yang sama melalui *concatenation* untuk masing-masing kategori *dataframe* yaitu *benign* dan *malicious.* Tahapan ini menerapkan prinsip *encoding* standar *UTF-8* dengan mekanisme penanganan kesalahan menggunakan library *'Chardet'.* Dilanjutkan dengan seleksi fitur untuk memilih variabel-variabel yang relevan dan menghilangkan fitur yang tidak diperlukan atau redundan, serta dilakukan *data cleaning* untuk membersihkan data dari *missing values.* Dalam model *hybrid* yang digunakan, *Transformer* memiliki

peran utama dalam menangkap keterkaitan global antar fitur dalam satu sampel teks, termasuk dependensi panjang yang tersebar di seluruh urutan data. Sementara itu, *CNN* melengkapi proses dengan mengekstraksi pola spasial lokal, yaitu relasi antar fitur yang saling berdekatan, sehingga mampu memperkuat representasi lokal yang relevan untuk deteksi *malware*. Tahapan ini juga meliputi *data normalization* untuk menyeragamkan skala data agar semua fitur memiliki rentang nilai yang sebanding, mencegah bias pada algoritma yang sensitif terhadap perbedaan skala.

Secara keseluruhan, tahapan *preprocessing* ini dirancang dengan menyesuaikan pada karakteristik khas dari data lalu lintas jaringan serta kebutuhan khusus dari model *hybrid* yang menggabungkan arsitektur *Transformer* dan *CNN*. Tahapan ini bertujuan untuk memastikan bahwa data yang akan digunakan dalam pelatihan dan evaluasi model berada dalam kondisi terbaik, sehingga model dapat belajar secara optimal dengan fokus utama untuk menjaga informasi penting yang dapat membedakan antara aktivitas normal dan aktivitas berbahaya, serta mengurangi gangguan seperti data yang tidak relevan atau menyimpang yang berpotensi menurunkan akurasi model dalam mendeteksi aktivitas *malware*.

4. Pemodelan *TNN*

Tahapan pemodelan *TNN* dalam penelitian ini terdiri dari enam sub-tahapan utama yang terintegrasi secara sistematis untuk membangun arsitektur *Trans Neural Network*. Proses dimulai dengan tahapan tokenisasi yang dimulai dengan pengambilan data mentah berupa teks yang belum terstruktur, kemudian melalui tahap pembersihan dan persiapan untuk diproses lebih lanjut. Data mentah ini, yang pada awalnya terdiri dari teks bebas, perlu dipecah menjadi token-token yang lebih

kecil agar dapat dipahami dan diproses oleh model pembelajaran mesin. Dalam tahap ini, setiap kata, tanda baca, atau sub-bagian dari kata diubah menjadi unit token yang lebih kecil berdasarkan kamus yang sudah ada dalam model *BERT*. Proses tokenisasi ini mengubah teks menjadi format numerik yang terstruktur, yang kemudian dapat diproses secara efisien.

Dilanjutkan dengan implementasi *Transformer* untuk menangkap dependensi sekuensial dan pola temporal dalam data yang kemudian dilakukan ekstraksi fitur untuk mengidentifikasi karakteristik penting dari representasi data. Selanjutnya, melakukan proses implementasi *CNN* untuk mengekstraksi fitur spasial dan pola lokal dari data yang telah diproses, serta dilanjutkan dengan tahapan *feature fusion* yang menggabungkan berbagai representasi fitur Transformer dan CNN untuk menciptakan representasi yang lebih komprehensif. Setelah *fitur fussion* selesai, tahapan dilanjutkan pada pemodelan *Trans Neural Network* yang membangun arsitektur model deteksi *malware* secara *hybrid* dengan *Transformer* dan *CNN* menjadi satu model *unified* yang mampu melakukan klasifikasi lalu lintas jaringan dengan memanfaatkan kelebihan masing-masing arsitektur untuk mencapai performa deteksi *malware* yang optimal.

Penentuan jumlah *epoch* dalam proses pelatihan model didasarkan pada prinsip konvergensi dalam pembelajaran mesin. Konvergensi mengacu pada kondisi ketika nilai fungsi kerugian *(loss function)* menunjukkan kecenderungan stabil atau tidak mengalami penurunan kinerja secara signifikan, yang menandakan bahwa model telah belajar secara optimal dari data pelatihan (Song dkk., 2019). Dalam implementasinya, jumlah *epoch* dipilih berdasarkan praktik umum yang

mempertimbangkan titik stabilitas performa model guna menghindari *overfitting* maupun *underfitting*. Oleh karena itu, pada penelitian ini digunakan jumlah *epoch* sebanyak 35, dengan acuan dari eksperimen serupa yang dilakukan oleh (Rahman, dkk., 2023) yang menetapkan jumlah *epoch* tersebut dalam proses pelatihan model untuk mencapai konvergensi yang optimal. Selain itu, penggunaan *optimizer AdamW* dalam proses pelatihan model diterapkan untuk mendukung kontrol regularisasi yang lebih stabil dan membantu model mencapai titik konvergensi *loss function* yang optimal, serta menjaga performa generalisasi model terhadap data yang belum pernah dilihat.

5. Evaluasi

Tahapan evaluasi berfokus pada proses anali kinerja model menggunakan berbagai metrik seperti pengukuran nilai akurasi untuk menilai ketepatan klasifikasi secara keseluruhan, analisis *loss function* untuk memantau proses konvergensi dan stabilitas pelatihan model, pengukuran *GFLOPs* untuk mengevaluasi kompleksitas dan efisiensi arsitektur komputasi *hybrid*, dan pengukuran waktu inferensi untuk menilai aspek praktikalitas implementasi model dalam skenario, serta evaluasi kinerja klasifikasi yang lebih mendalam melalui pengukuran *precision*, *recall*, dan *F1-score* untuk menganalisis kemampuan model dalam mendeteksi *True Positive* dan meminimalkan *False Positive/Negative* yang kemudian diperkuat dengan analisis *confusion matrix* untuk memberikan Gambaran detail distribusi prediksi pada setiap kelas. Keseluruhan metrik evaluasi ini bertujuan untuk menghasilkan model yang optimal, tetapi juga praktis untuk diimplementasikan dalam sistem deteksi *malware* yang memerlukan respon cepat dan andal dalam

mengklasifikasikan lalu lintas jaringan sebagai *malicious* atau *benign*. Proses eksperiman yang dilakukan pada penelitian ini menggunakan *Framework PyTorch.*

Tahapan-tahapan yang dilakukan pada proses penelitian disusun secara sistematis agar mampu mengefektifkan setiap arsitektur dan memberikan hasil yang sesuai. Setiap langkah dirancang untuk memastikan proses pelatihan dan evaluasi model berjalan secara terstruktur dan dapat direproduksi. Dengan demikian, hasil yang diperoleh tidak hanya akurat, tetapi juga dapat dipertanggungjawabkan secara ilmiah.