BAB 2
TINJAUAN PUSTAKA

2.1 Metode Tinjauan Pustaka

Tinjauan pustaka merupakan metode penelitian yang melibatkan pengumpulan, seleksi, dan analisis informasi yang ada dari sumber-sumber terpercaya seperti jurnal ilmiah, buku, dan dokumen resmi. Tujuannya adalah untuk memperoleh gambaran umum dan memahami perkembangan dan hasil penelitian terkait topik yang diteliti. Metode ini biasanya digunakan sebagai langkah awal dalam suatu penelitian dan membantu dalam pembuatan hipotesis dan perencanaan penelitian lebih lanjut [15].

Pada penulisan proposal laporan tugas akhir ini akan menggunakan metode tinjauan pustaka yang populer digunakan oleh para peneliti yaitu SLR *(Systematic Literature Review)* yang merupakan metode tinjauan pustaka untuk memastikan objektivitas dan validitas hasilnya secara sistematis dan terstruktur. Metode ini melibatkan langkah-langkah yang terdefinisi, seperti pemilihan kriteria inklusi dan eksklusi, pemilihan sumber-sumber yang valid, dan analisis data secara sistematis. Tujuannya adalah untuk menghasilkan tinjauan yang terpercaya dan akurat dari literature yang ada pada topik yang diteliti, sehingga membantu dalam membuat kesimpulan dan rekomendasi untuk penelitian selanjutnya [16].

2.2 *Systematic Literature Review*

Pada tahap ini akan dilakukan analisis data secara sistematis dan terstruktur terhadap penelitian – penelitian terdahulu pada sumber yang valid dengan tujuan untuk untuk memastikan objektivitas dan validitas hasilnya terlebih untuk mengetahui sudah sampai mana tingkat kemajuan penelitian yang dilakukan terhadap parameter dan kelemahan atau kendala apa saja yang dialami sehingga akan dapat dilakukan optimasi dan perbaikan.

Penelitian tentang deteksi serangan DDoS menggunakan machine learning telah menjadi topik yang semakin relevan dalam lingkup keamanan jaringan.

Tantangan yang dihadapi oleh penyedia layanan cloud dan sistem jaringan dalam menghadapi serangan DDoS semakin meningkat seiring dengan perkembangan teknologi. Beberapa penelitian terkini telah berfokus pada penggunaan teknik machine learning untuk mendeteksi serangan DDoS dengan tingkat akurasi yang mengesankan. Di antara penelitian-penelitian tersebut adalah penelitian oleh Wani et al. (2019) yang menganalisis dan mendeteksi serangan DDoS pada lingkungan cloud computing menggunakan teknik-teknik machine learning dengan mencapai tingkat akurasi sebesar 98%. Penelitian lainnya, seperti yang dilakukan oleh Chen et al. (2019), berhasil mengimplementasikan metode Random Forest dalam mendeteksi serangan DDoS dengan nilai akurasi mencapai 99.41%. Selain itu, Alduailij et al. (2022) berhasil mencapai tingkat akurasi mencapai 99.7% dengan menggunakan metode Mutual Information dan teknik Random Forest dalam pendekatan deteksi serangan DDoS berbasis machine learning. Namun, walaupun tingkat akurasi pada penelitian-penelitian ini telah mencapai tingkat yang mengesankan, masih ada potensi untuk meningkatkan kinerja deteksi serangan DDoS dengan mempertimbangkan seleksi fitur (*feature selection*) yang lebih tepat dan efektif.

Dalam upaya untuk meningkatkan nilai akurasi deteksi serangan DDoS, penelitian ini akan menggunakan metode CFS (Correlation-based Feature Selection) untuk menyeleksi fitur yang tidak berguna dalam dataset yang digunakan. Metode CFS telah terbukti efektif dalam memilih fitur-fitur yang relevan dan saling berkorelasi dalam dataset, sehingga dapat membantu meningkatkan kinerja sistem deteksi serangan DDoS berbasis machine learning. Langkah-langkah sistematis dari systematic literature review akan digunakan untuk mencari dan menyaring artikel-artikel terkini yang membahas mengenai penggunaan metode CFS dalam deteksi serangan DDoS. Data dan hasil dari penelitian-penelitian terkait akan dianalisis dan dibandingkan untuk menyusun landasan teoritis yang kuat bagi penelitian ini. Selain itu, akan dilakukan eksperimen menggunakan dataset yang relevan dan perangkat lunak yang tepat untuk mengimplementasikan metode CFS dalam sistem deteksi serangan DDoS yang diusulkan.

Tabel 2.1 Penelitian Terdahulu

Penulis	Masalah	Dataset	Metode	Tahun Jurnal	Akurasi Model
M. Alduailij, Q.W., Khan, M. Tahir, M. Sardaraz, M. Alduailij	Waktu latih yang diperlukan untuk melatih model sangat lama.	CICIDS2017 dan CICIDS2018	*Random Forest With Feature Importance Method*	2022	99.7%
Yini Chen, Jun Hou, Qianmu Li, dan Huaqiu Long	Model *Random Forest* membutuhkan lebih banyak waktu dan sumber daya untuk melatih dan memprediksi.	*LLDoS1.0, LLDoS2.0.1, UDP Flood Attack* dan *ICMP Flood Attack*	*Random Forest Classification*	2020	99.41%
Vimal Gaur, Rainessh Kumar	Akurasi deteksi Masih belum Maksimal	*CICDDOS_ 2019*	*Random Forest Classification*	2022	99%
A.R., Wani, Q.P. Rana, U. Sexana, Nitin Pandey	*Random Forest* dapat mengalami *overfitting* dan menghasilkan model yang sangat kompleks.	IDS SNORT	*Random Forest Classification*	2019	98%

2.3 Keamanan Siber

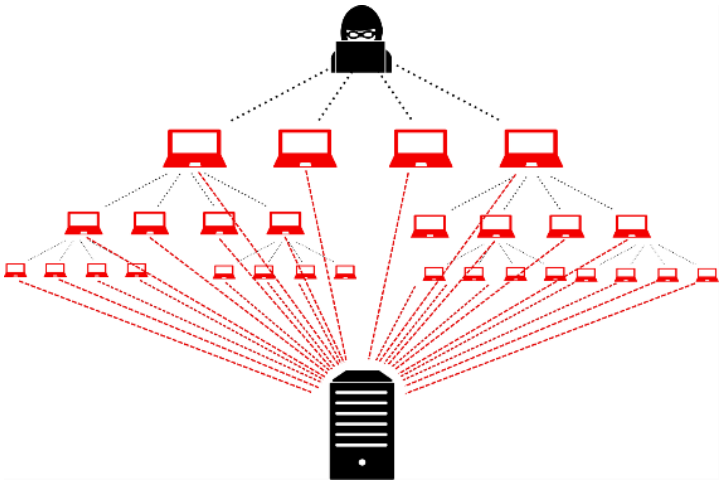
Keamanan siber adalah teknik untuk melindungi sistem, jaringan, dan data dari serangan atau akses yang tidak sah dari pihak-pihak yang tidak berwenang. Ini termasuk melindungi komputer, jaringan, aplikasi, dan data dari serangan *malware, phishing, hacking*, dan lainnya. *Cybersecurity* juga meliputi aspek-aspek teknis seperti enkripsi, autentikasi, dan pemantauan aktivitas jaringan [17].

2.4 Ancaman Jaringan

Ancaman serangan jaringan adalah tindakan yang dilakukan oleh seseorang atau kelompok yang bertujuan untuk merusak, mencuri, atau mengontrol jaringan komputer tanpa izin melalui jaringan komputer, baik jaringan lokal maupun jaringan internet [18].

2.5 DDoS *(Distributed Denial of Service)*

DDoS *(Distributed Denial of Service)* merupakan teknik penyerangan yang sering terjadi pada server web, pada serangan DDoS tidak memiliki tujuan untuk mencuri atau membocorkan data melainkan untuk merusak sistem dengan cara membanjiri *traffic* palsu pada server web target dan membuat server menjadi sibuk dengan banyaknya permintaan layanan sehingga dapat menurunkan performa bahkan membuat server web menjadi *down* [19].



Gambar 2.1 Ilustrasi Serangan DDoS

2.6 *Machine Learning*

2.6.1 Pengertian

Machine learning adalah satu program komputer yang dikatakan telah melakukan pembelajaran dari pengamalan E (*Experience*) terhadap tugas T (*Task*) dan mengukur peningkatan kinerja P (*Perfomance Measure*), jika kinerja Tugas T diukur oleh kinerja P, maka meningkatkan pengalaman E. Dari definisi ini dapat dikatakan sebuah aplikasi *machine learning* memiliki 3 komponen yaitu *Task* T, *Performance Measure* P, dan *Experience* E. Oleh karena itu, untuk membangun sebuah aplikasi ML maka komponen T,P dan E harus dapat diidentifikasi [20].

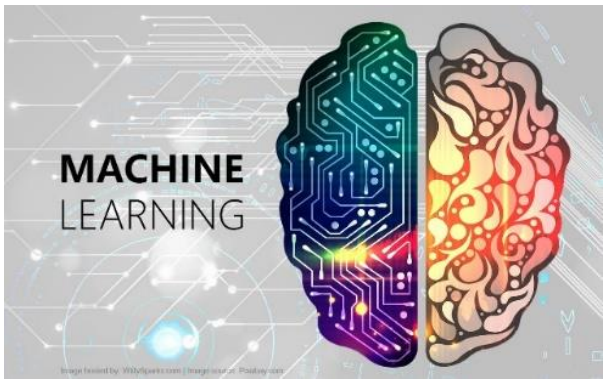
Prinsip cara kerja *machine learning* masih sama, meliputi pengumpulan data, eksplorasi data, pemilihan model atau teknik, memberikan pelatihan terhadap model yang dipilih dan mengevaluasi hasil dari *machine learning*. *Machine Learning* mempunya dua tipe yaitu:

1. *Supervised Learning*

   Supervised Learning adalah metode yang digunakan saat semua data yang dimiliki sudah mempunyai label dan algoritmanya belajar memprediksi output dari input. *Supervised Learning* meliputi: Teknik *Regression* dan *Classification*.

2. *Unsupervised Learning*

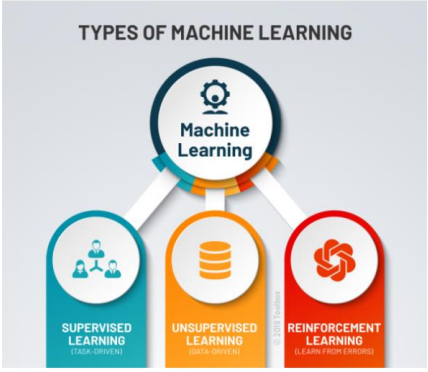
   Unsupervised Learning adalah metode yang digunakan saat semua data yang dipunya tidak mempunyai label dan algoritmanya mempelajari struktur yang melakat dari data tersebut. *Unsupervised Learning* meliput: Teknik *Clustering* dan *Association.*



Gambar 2.2 *Machine Learning*

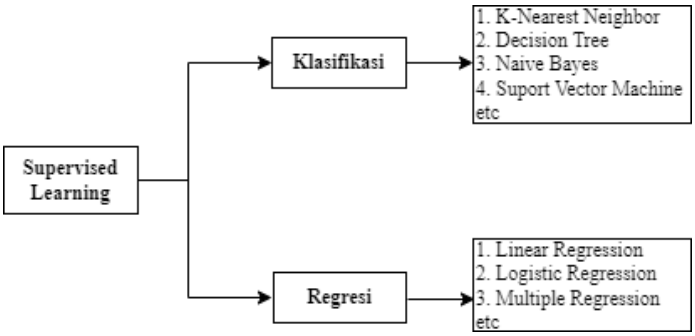
2.6.2 Penerapan Metode

Salah satu metode *machine learning* yang paling populer adalah *supervised learning*, yang merupakan jenis *machine learning* di mana model dilatih pada data berlabel, di mana keluaran yang diinginkan sudah diketahui. Tujuan dari model ini adalah untuk mempelajari pemetaan antara fitur masukan dan label keluaran, sehingga dapat menggeneralisasi dan membuat prediksi yang akurat pada data baru yang tidak terlihat. Contoh masalah *supervised learning* termasuk regresi dan klasifikasi.



Gambar 2.3 Metode *Machine Learning*

Ada dua hal utama dalam menerapkan *supervised learning*, yaitu klasifikasi dan regresi. Namun yang dibahas adalah metode klasifikasi yang merupakan proses pengelompokkan data berdasarkan pelatihan dataset berlabel. Algoritma yang digunakan dalam tugas akhir ini adalah *random forest* serta menggunakan metode *machine learning* dalam mendeteksi serangan DDoS. Penerapan metode *machine learning* sebagai metode pendukung untuk dilakukan sistem saat diprogram menggunakan Pycharm.



Gambar 2.4 Pembagian algoritma pada metode *supervised learning*

(sumber : https://www.ekrut.com/media/apa-itu-machine-learning)

Adapun tahapan membangun model pada *Machine Learning* yaitu sebagai berikut :

a. Memahami permasalahan, permasalahan yang dimaksud ialah bagaimana cara *machine learning* untuk mengetahui serangan DDoS.
b. Memilih metode yang digunakan untuk diimplementasikan yaitu *random forest* pada objek yang akan diteliti dalam hal ini serangan DDoS.

c. Mempersiapkan dataset untuk mendeteksi serangan DDoS yang didapatkan pada *platform* kaggle.
d. Implementasi, melakukan training dataset
e. Integrasi dan evaluasi, yaitu mengukur tingkat akurasi pada serangan DDoS.

2.7 Klasifikasi

Klasifikasi adalah proses pengelompokan item atau objek ke dalam kelompok-kelompok yang berbeda berdasarkan karakteristik tertentu yang dimilikinya. Ini sering digunakan dalam berbagai bidang, seperti data mining, analisis bisnis, dan pembelajaran mesin, untuk membuat keputusan yang didasarkan pada data. Klasifikasi dapat menggunakan teknik statistik, algoritma pembelajaran mesin, dan *rule-based systems* untuk membuat keputusan yang akurat [21].

Klasifikasi dokumen adalah proses mengelompokkan dokumen ke dalam kategori tertentu berdasarkan isi atau konten dokumen tersebut. Tujuannya adalah untuk mempermudah organisasi dalam melakukan pencarian informasi. Klasifikasi dokumen dapat dilakukan secara manual atau dengan menggunakan teknik pembelajaran mesin seperti klasifikasi *Naive Bayes, Support Vector Machine (SVM), atau Random Forest.*

Klasifikasi adalah salah satu peran utama dari *data mining*. Klasifikasi termasuk dalam *supervised learning* karena dalam proses klasifikasi ada proses belajar dengan data masa lalu. Proses ini digunakan oleh algoritma untuk mengenali pola dari data yang nantinya dapat diterapkan ke data baru yang kelompoknya belum diketahui.Teknik klasifikasi diterapkan secara luas di dunia nyata maupun di dunia kedokteran, pendidikan, teknik bangunan, jaringan komputer dan *cyber security.*

2.8 *Data Mining*

Data Mining merupakan serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Sehingga *data mining* menjadi alat yang semakin penting untuk mengubah data tersebut menjadi informasi. *Data mining* adalah teknik analisis data berbasis

padaaplikasi statistik yang bertujuan untuk mengekstrak informasi. Dengan *data mining* kumpulan data dalam jumlah besar dapat dijadikan informasi lain yang bermanfaat. *Data mining* dapat melakukan pekerjaan seperti memperkirakan, mengklasifikasikan sampai mengelompokkan data [22].

Teknik-teknik, metode- metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses *Knowledge Discoveryin Database (KDD)* secara keseluruhan [23].

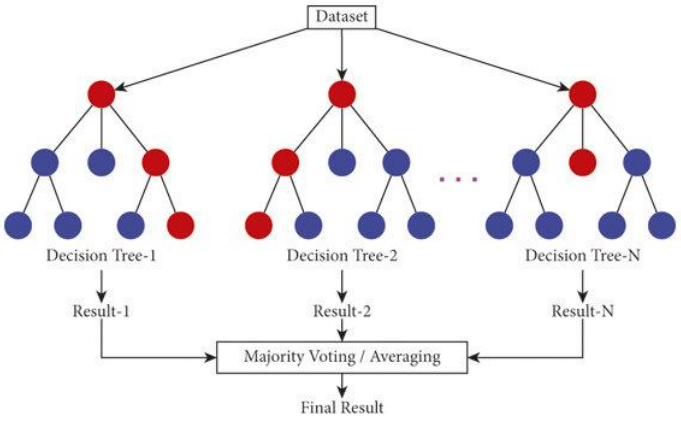
2.9 *Random Forest*

Random forest adalah metode *machine learning* yang menggunakan teknik *ensamble* (gabungan) beberapa *decision tree* (pohon keputusan), setiap algoritma *Decision Tree* dilakukan pelatihan menggunakan sampel individu yang dikelompokkan ulang secara acak. Pada proses klasifikasi, individunya didasarkan pada *vote* dari suara terbanyak pada kumpulan populasi *tree*. *Random Forest* yang dihasilkan memiliki banyak *tree*, dan setiap *tree* dibangun dengan cara yang sama. *Tree* dengan variabel x akan dibangun sejauh mungkin dengan *tree* dengan variabel y. Dan dalam perkembangannya, sejalan dengan bertambahnya *dataset*, maka *tree* pun ikut berkembang. Penempatan *tree* yang saling berjauhan membuat apabila terdapat *tree* disekitar *tree* x berarti pohon tersebut merupakan perkembangan dari *tree* x [24].

Metode *Random Forest* dapat meningkatkan hasil akurasi karena pada Metode *Random Forest* memperbaiki akurasi dengan melakukan *ensamble* atau penggabungan dari banyak *decision tree*. Dengan melakukan ensamble ini, *Random Forest* mampu menangani masalah *overfitting* dan memperbaiki akurasi model secara keseluruhan. *Decision tree* masing-masing membuat prediksi dan hasil akhir dari *Random Forest* adalah hasil dari voting atau rata-rata dari hasil prediksi dari masing-masing *decision tree*. Oleh karena itu, *random forest* dapat meningkatkan akurasi dengan memanfaatkan kelebihan dari banyak *decision tree*.

Pada algoritma *random forest* terdapat aspek – aspek penting yang perlu untuk diperhatikan antara lain :

1. *Ensemble of Decision Trees*: *Random forest* menggabungkan beberapa pohon keputusan untuk membuat prediksi yang lebih baik dan stabil.
2. *Bagging (Bootstrapped Aggregating)*: *Random forest* menggunakan teknik bagging untuk mengurangi overfitting dan memperkuat model.
3. *Feature Importance*: *Random forest* dapat menentukan tingkat penting suatu fitur dalam membuat prediksi, membantu dalam pemilihan fitur dan interpretasi model.
4. *Non-Parametric*: *Random forest* tidak memerlukan asumsi tentang distribusi data atau bentuk dari hubungan antara fitur dan target, sehingga cocok untuk berbagai jenis data.
5. *Scalable*: *Random forest* dapat digunakan untuk data besar dan dapat membuat prediksi secara cepat, membuatnya cocok untuk aplikasi *real-time*.



Gambar 2.5 Alur Kerja *Random Forest*

2.10 *Confusion Matrix*

Confusion matrix merupakan tabel yang digunakan untuk mengevaluasi akurasi dari suatu model klasifikasi. Tabel ini menunjukkan jumlah prediksi benar dan salah yang dilakukan oleh model terhadap data uji. Setiap baris pada tabel mewakili klasifikasi aktual, dan setiap kolom mewakili prediksi model. Elemen pada diagonal utama adalah prediksi benar, sedangkan elemen di luar diagonal utama adalah prediksi salah. Untuk menghitung nilai akurasi dari *confusion matrix* didapatkan dari nilai TP, TN, FP, FN [25].

- *False Positive* (FP) : jumlah catatan yang normal tetapi diklasifikasikan sebagai serangan.
- *False Negative* (FN) : jumlah serangan yang benar tetapi diklasifikasikan sebagai catatan yang normal.
- *True Positive* (TP) : jumlah catatan yang serangan dan diklasifikasikan sebagai catatan yang serangan.
- *True Negative* (TN) : jumlah normal yang diklasifikasikan sebagai normal.

Tabel 2.2 *Confusion Matrix*

		Actual Values	
		Positive	*Negative*
Predicted Values	*Positive*	TP	FP
	Negative	FN	TN

Rumus parameter berikut digunakan untuk mengevaluasi modelnya yaitu:

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN}$$

Akurasi yang dihasilkan dihitung berdasarkan *confusion matrix*. Perhitungan pada confusion matrix dihitung sesuai dengan prediksi positif yang benar (*True Positif*), prediksi positif yang salah (*False Positif*), prediksi negatif yang benar (*True Negatif*) dan prediksi negatif yang salah (*False Negatif*). Semakin tinggi nilai akurasi yang didapat maka semakin baik pula metode yang dihasilkan.

2.11 *Ensamble Learning*

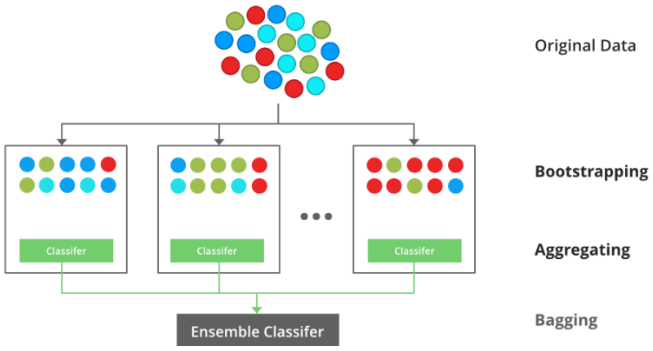
Ensemble learning adalah teknik pembelajaran mesin yang menggabungkan prediksi dari beberapa model untuk meningkatkan akurasi dan ketahanan secara keseluruhan dari prediksi. Ide dasarnya adalah menggunakan beberapa model

dengan pandangan dan bias yang beragam untuk memperoleh prediksi yang lebih akurat dan dapat diandalkan dibandingkan dengan yang dapat dicapai oleh satu model saja. Metode ensemble sering digunakan dalam berbagai domain termasuk computer vision, natural language processing, dan tugas prediksi. Beberapa algoritma ensemble learning yang populer meliputi Random Forest, Gradient Boosting, Bagging, dll [26].

2.12 Bagging (*Bootstrapped Aggregating)*

Metode *Bootstrapped Aggregating* merupakan sebuah teknik pembelajaran mesin yang menggunakan teknik *bootstrapping* (sampel ulang) untuk membuat beberapa model paralel yang dihasilkan dari sampel data yang diambil secara acak dan berulang dari populasi data asli. Hasil dari setiap model paralel ini kemudian digabungkan menjadi satu model akhir dengan memanfaatkan teknik seperti voting, rata-rata, atau varians. Tujuannya adalah untuk mengurangi varians dan *overfitting* dari model tunggal dan meningkatkan akurasi model [27].

Metode *bagging* ini akan diterapkan pada algoritma *random forest* dengan tujuan untuk mengurangi data *noise*, varians dan *overfitting* pada dataset sehingga diharapkan mampu untuk meningkatkan akurasi pada model.



Gambar 2.6 Metode Bagging

2.13 CFS *(Correlation-based Feature Selection)*

Correlation-based Feature Selection adalah suatu metode pemilihan fitur dalam machine learning yang menentukan kepentingan fitur berdasarkan tingkat korelasi antara fitur dan target. Ini berguna untuk mengurangi dimensi data dan

menghilangkan fitur yang tidak berguna atau berkorelasi dengan fitur lain, yang dapat memperbaiki performa model dan mempermudah interpretasi hasil [28].

2.14 Python

2.14.1 Pengertian

Python merupakan bahasa pemrograman tingkat tinggi dan interpretatif, yang digunakan untuk berbagai jenis aplikasi, seperti pengembangan web, analisis data, pembuatan aplikasi desktop dan mobile, dan banyak lagi. Python memiliki sintaks yang mudah dipahami dan memiliki beragam library dan framework yang membuat pengembangan aplikasi menjadi lebih efisien. Python juga memiliki komunitas yang aktif dan terbuka, yang membuat dokumentasi dan sumber daya tersedia secara luas [29].

Python memiliki beragam library dan framework untuk pembelajaran mesin, seperti scikit-learn, TensorFlow, PyTorch, dan banyak lagi. Library-library ini memudahkan pengembangan model pembelajaran mesin melalui penerapan metode-metode yang sudah terbukti dan memiliki performa tinggi, seperti regression, klasifikasi, clustering, dan deep learning. Library-library ini juga menyediakan fitur-fitur seperti preprocessing data, evaluasi model, dan visualisasi hasil. Oleh karena itu, Python menjadi pilihan yang populer bagi data scientist dan engineer machine learning.

Gambar 2.7 Bahasa Pemograman Python

2.14.2 Fitur Bahasa Pemrograman Python

Sisi utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Bagi para programmer yang tidak terbiasa menggunakan python akan dibingungkan dengan aturan indentasi, tipe data, tuple, dan dictionary. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai python. Selain itu python merupakan salah satu produk yang open source, gratis, dan multiplatform. Beberapa fitur dan kelebihan yang dimiliki Python adalah [29] :

1. Memiliki koleksi kepustakaan yang banyak. Artinya, telah tersedia modul-modul siap pakai untuk berbagai keperluan, seperti pembuatan game hingga artificial intelligence (misal: Tensor Flow).
2. Memiliki struktur bahasa yang jelas, sederhana, dan mudah dipelajari.
3. Berorientasi prosedural dan objek sekaligus (multi-paradigma)
4. Memiliki sistem pengelolaan memori otomatis (garbage collection) seperti halnya Java. 5. Bersifat modular sehingga mudah dikembangkan dengan menciptakan moodul-modul baru, baik dibangun dengan bahasa Python maupun C/C++

2.15 Anaconda

Anaconda adalah distribusi Python dan R yang dikhususkan untuk Data Science dan Machine Learning. Anaconda menyediakan lingkungan virtual yang membuat instalasi, pengaturan, dan manajemen paket Python dan R menjadi mudah dan efisien Anaconda menyediakan paket-paket populer seperti NumPy, pandas, Matplotlib, dan banyak paket lainnya yang dibutuhkan dalam bidang Data Science dan machine learning. Anaconda juga menyediakan akses ke berbagai distribusi dan pustaka machine learning seperti scikit-learn, TensorFlow, PyTorch, dll. Anaconda membuat proses pembelajaran dan implementasi teknologi Data Science dan Machine Learning menjadi lebih mudah dan efisien[30].

Gambar 2.8 Logo Anaconda

Anaconda diciptakan agar mempermudah pengguna memanajeman paket python. Dengan menggunakan Anaconda, maka versi dari paket yang ada, di manajemen oleh package management system conda.

2.16 *Jetbrains Pycharm*

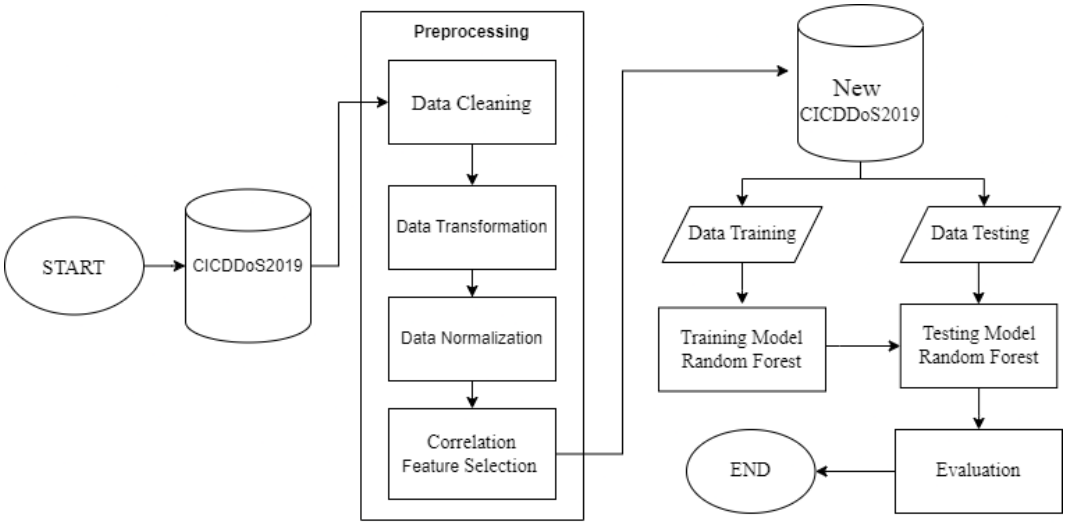
IDE yang cukup populer dikalangan developer Python adalah PyCharm. PyCharm sendiri memiliki dua versi yaitu *Professional Edition* dan *Community Edition*. PyCharm *Professional Edition* merupakan versi berbayar dari PyCharm dan *Community Edition* merupakan versi gratis yang tersedia bagi komunitas python dengan lisensi Apache 2. Pycharm digunakan sebagai tool dalam pensimulasian ini [31].

PyCharm merupakan text editor atau Integrated Development Environment (IDE). PyCharm Edu merupakan text editor dengan tampilan user interface yang mudah dipahami sehingga mudah digunakan dalam tujuan pembelajaran. File Python menggunakan format .py.

Gambar 2.9 Logo Pycharm

BAB 3
METODOLOGI PENELITIAN

Tahap metodologi penelitian ini akan disajikan dalam bentuk diagram komprehensif yang memberikan gambaran jelas tentang langkah-langkah penelitian. Diagram tersebut berfungsi sebagai representasi visual dari proses penelitian, yang memungkinkan para peneliti untuk memahami secara menyeluruh seluruh studi. Ini menawarkan panduan yang menguraikan urutan langkah-langkah secara logis dan terorganisir dalam penelitian.



Gambar 3.1 Metodologi Penelitian

3.1 Dataset

Penelitian ini akan menggunakan dataset CIC-DDoS2019 yang diperoleh dari sumber [12]. Dataset ini mengatasi keterbatasan dataset sebelumnya dan menawarkan kumpulan fitur aliran jaringan yang komprehensif untuk mendeteksi dan mengklasifikasikan serangan DDoS. Selain itu, dataset ini memperkenalkan taksonomi baru untuk mengategorikan serangan DDoS dan berbagi kumpulan fitur dengan dataset NIDS CIC lainnya, yaitu IDS2017, IDS2018, dan DoS2017. Dataset ini terdiri dari 431.371 baris dan mencakup 17 jenis serangan. Dataset ini mencakup 97.831 contoh lalu lintas benign (non-jahat) dan 212.172 contoh serangan DDoS. Informasi lebih rinci tentang jenis serangan dapat ditemukan dalam Tabel 3.1.

Tabel 3.1 Lalu lintas Dataset CIC-DDoS2019

No.	Jenis Lalu Lintas Jaringan	Jumlah
1.	DrDos_NTP	121,368
2.	TFTP	98,917
3.	Benign	97,831
4.	Syn	49,373
5.	UDP	18,090
6.	DrDoS_UDP	10,420
7.	UDP-lag	8,872
8.	MSSQL	8,523
9.	DrDoS_MSSQL	6,212
10.	DrDoS_DNS	3,669
11.	DrDoS_SNMP	2,717
12.	LDAP	1,906
13.	DrDoS_LDAP	1,440
14.	Portmap	685
15.	NetBIOS	644
16.	DrDoS_NetBIOS	598
17.	UDPLag	55
18.	WebDDoS	51
	Total Traffic	**431,371**

3.2 Tahap Pra-Pemrosesan Data

3.2.1 Pembersihan Data

Pada tahap ini, dataset CIC-DDoS2019 mengalami proses pembersihan data yang memiliki peran penting dalam meningkatkan akurasi dan efektivitas pembelajaran mesin. Hal ini membantu menghilangkan noise, pencilan (*outliers*), dan kesalahan dalam data, sehingga mengurangi *overfitting*, dan meningkatkan efisiensi model [13]. Kami menghapus baris dan kolom yang memiliki nilai duplikat, NaN (*Not a Number*), *infinite* (tak terhingga), dan -*infinite* (negatif tak

terhingga). Selanjutnya, data dikelompokkan berdasarkan kolom 'Label' dan difilter agar hanya mencakup kelompok data dengan jumlah data lebih dari 10.000. Tujuan langkah ini adalah untuk mendapatkan subset data yang bermakna dengan jumlah yang mencukupi untuk analisis atau pemodelan lanjutan.

3.2.2 Transformasi Data

Transformasi data mengacu pada proses mengubah data dari bentuk aslinya menjadi bentuk yang berbeda, sehingga memungkinkan penggunaan yang lebih efektif atau sesuai untuk analisis atau aplikasi tertentu dengan tujuan meningkatkan kualitas data atau memenuhi persyaratan khusus dalam analisis data [14]. Pada tahap transformasi data, saya menjalankan konversi kolom int64 menjadi tipe data int32 dan kolom float64 menjadi tipe data float32. Konversi ini dilakukan untuk mengoptimalkan penggunaan memori dan meningkatkan kecepatan pemrosesan data.

3.2.3 Normalisasi Data

Normalisasi data melibatkan proses mengubah nilai-nilai dalam dataset menjadi rentang standar dengan tujuan untuk menghilangkan variasi skala di antara fitur atau variabel, memudahkan perbandingan dan analisis data [15]. Dalam proses ini akan menggunakan pendekatan *RandomUnderSampler* untuk menangani ketidakseimbangan kelas dalam dataset dengan mengurangi jumlah sampel dari kelas mayoritas, sehingga mencapai representasi yang seimbang dari kelas minoritas. Selanjutnya, variabel-variabel yang telah diundersample mengalami penskalaan Z-score, yang mengnormalisasi distribusi fitur dalam dataset dengan mengatur nilai-nilai tersebut di sekitar data set dengan mean (rerata) 0 dan simpangan baku (standard deviation) 1. Formula penskalaan Z-score adalah:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Di mana (z) adalah Z-score atau nilai yang telah diubahskala, (x) adalah nilai pada data yang akan diubahskala, (μ) adalah rerata data, dan (σ) adalah simpangan baku dari data.

3.2.4 *Correlation-based Feature Selection*

Metode *Correlation-based Feature Selection* adalah cara untuk mengevaluasi signifikansi fitur-fitur dengan mengukur tingkat korelasi antara fitur-fitur tersebut. Tujuannya adalah untuk mengurangi dimensionalitas data dan menghilangkan fitur-fitur yang tidak relevan atau berkorelasi, sehingga dapat meningkatkan kinerja model dan mempermudah interpretasi hasil [16]. Dalam metode ini, digunakan formula untuk menghitung korelasi antara fitur-fitur dalam dataset sebagai berikut :

$$Corr(Fitur1, Fitur2) = \frac{\sum_{i=1}^{n}(X_{1i} - \overline{X_1}).(X_{2i} - \overline{X_2})}{\sqrt{\sum_{i=1}^{n}(X_{1i} - \overline{X_1})^2 . \sum_{i=1}^{n}((X_{2i} - \overline{X_2})^2}}$$

Di mana X_{1i} adalah nilai yang mewakili fitur pertama pada sampel ke-i, $\overline{X_1}$adalah nilai rerata dari fitur pertama, X_{2i} adalah nilai yang sesuai dengan fitur kedua pada sampel ke-i, $\overline{X_2}$ adalah nilai rerata dari fitur kedua, dan n adalah jumlah sampel. Tujuan dari rumus ini adalah untuk mengukur tingkat korelasi atau hubungan linear antara dua fitur (fitur 1 dan fitur 2) dalam suatu dataset. Korelasi antara fitur-fitur ini dapat membantu dalam memilih fitur-fitur yang sangat terkait atau berkorelasi. Hasil dari rumus ini adalah nilai korelasi antara fitur 1 dan fitur 2, di mana koefisien korelasi berkisar dari -1 hingga 1, di mana nilai 1 mewakili hubungan linier positif sempurna antara fitur 1 dan fitur 2, nilai -1 menunjukkan hubungan linier negatif sempurna, dan nilai 0 menunjukkan tidak ada hubungan linier antara kedua fitur tersebut. Koefisien korelasi berkisar dari -1 hingga 1, semakin tinggi korelasi antara fitur-fitur tersebut [17].

3.3 Pisah Data Latih dan Data Uji

Dataset CIC-DDoS2019, yang telah melalui tahapan pra-pemrosesan, dibagi menjadi dua bagian terpisah: data pelatihan (training data) dan data uji (test data), dengan rasio 70% dan 30% masing-masing. Data pelatihan, yang terdiri dari 70% dari total dataset, digunakan untuk melatih model random forest, sementara data uji, yang mencakup 30% dari keseluruhan dataset, digunakan untuk menilai performa model random forest yang telah dilatih pada data sebelumnya yang belum pernah dilihat sebelumnya. Dengan membagi dataset ini, kita dapat memperoleh

pemahaman yang lebih realistis tentang seberapa baik model dapat berperforma dan mencegah terjadinya overfitting, yaitu kondisi di mana model menghafal dan memprediksi data latihan dengan baik tetapi gagal menggeneralisasi dengan baik pada data baru.

3.4 *Random Forest*

Random Forest adalah metode machine learning yang kuat yang menggunakan teknik ensemble dengan menggabungkan banyak pohon keputusan (decision tree). Setiap algoritma pohon keputusan dilatih menggunakan subset acak dari dataset. Selama proses klasifikasi, prediksi akhir diperoleh dengan mengumpulkan suara mayoritas dari seluruh himpunan pohon. Random Forest terdiri dari beberapa pohon, dan setiap pohon mengikuti pendekatan konstruksi yang sama. Pohon-pohon dengan variabel yang berbeda dibangun sesendirian mungkin, dengan tujuan meminimalkan redundansi. Seiring dengan bertambahnya dataset, pohon-pohon tersebut beradaptasi dan berkembang sesuai, menangkap pola-pola yang lebih kompleks dan meningkatkan kinerja. Pemisahan spasial dari pohon-pohon tersebut menandakan perkembangan mereka yang berbeda dan memberikan beragam perspektif untuk meningkatkan akurasi.

3.5 Evaluasi Model

Pada tahap evaluasi model, dilakukan analisis dan pengukuran kinerja dari model Random Forest yang telah dilatih dalam mengidentifikasi dan mendeteksi Serangan Distributed Denial of Service (DDoS). Evaluasi ini bertujuan untuk menilai sejauh mana model dapat mengenali dan mengklasifikasikan serangan DDoS dengan akurasi tinggi. Metrik evaluasi, termasuk akurasi, presisi, recall, dan F1-score, digunakan untuk menilai kinerja model dalam mendeteksi serangan DDoS secara akurat. Selain itu, matriks kebingungan (confusion matrix) digunakan untuk memberikan representasi visual dari hasil klasifikasi dan memperoleh informasi tentang serangan yang terdeteksi dengan benar, serangan yang terlewatkan, dan prediksi yang salah. Ukuran evaluasi ini membantu menilai efektivitas dan keandalan model Random Forest dalam mendeteksi serangan DDoS, memberikan wawasan tentang kelebihan dan area yang perlu diperbaiki.

Tabel 3.2 *Confusion Matrix*

Kelas Sebenarnya	Prediksi	
	Positive	*Negative*
Positive	*True Positive* (TP)	*False Negative* (FN)
Negative	*False Positive* (FP)	*True Negative* (TN)

Confusion Matrix adalah metode evaluasi untuk model klasifikasi yang membandingkan prediksi dengan nilai-nilai aktual untuk menghitung metrik evaluasi seperti akurasi (accuracy), presisi (precision), recall, dan F1-score. Matriks kebingungan menyediakan informasi detail tentang kinerja model dalam mengklasifikasikan data dengan benar atau salah [20].

Akurasi adalah metrik evaluasi yang mengukur persentase klasifikasi yang benar yang dibuat oleh sebuah model. Rumus dasar untuk akurasi adalah [20].Accuracy

$$= \frac{(TP + TN)}{(Total\ )} \text{x} 100\%$$

Presisi adalah metrik evaluasi yang mengukur sejauh mana prediksi positif yang dibuat oleh model klasifikasi benar. Rumus untuk menghitung presisi ditunjukkan dalam persamaan 4 [20].

$$\text{Precision} = \frac{TP}{TP+FP} \text{x} 100\%$$

Recall adalah metrik evaluasi yang mengukur sejauh mana model klasifikasi dapat menemukan atau mendeteksi total jumlah data positif yang benar. Rumus untuk menghitung recall ditunjukkan dalam persamaan 5 [20].

$$\text{Recall} = \frac{TP}{TP+FN} \text{x} 100\%$$

Skor F1 adalah metrik kinerja yang seimbang antara presisi dan recall, memberikan ukuran komprehensif tentang efektivitas model klasifikasi. Skor F1 dihitung dengan menggabungkan presisi dan recall menggunakan persamaan 6 [20].

$$\text{F1 Score} = 2 \times \frac{Precession \times Recall}{Precession + Recall} \text{x} 100\%$$