BAB II
TINJAUAN PUSTAKA

1. Tinjauan Pustaka

Sebelum melakukan penelitian ini peneliti memiliki gambaran terhadap peneliti sebelumnya, beberapa penelitian tersebut yaitu terdapat pada Table 2.1 dibawah ini

Table 2.1 *Tinjauan Pustaka Penelitian Penulis*

No	Peneliti	Judul	*Training*	Metode	Hasil tertinggi Algoritma % akurasi
1	Mirza Maulana Azmi, dkk	*Low-Rate Attack* Detection on *SD-IoT* Using *SVM* Combined with *Feature Importance* Logistic Regression Coefficient (2022)	Skema *Training* dari Mirza Maulana	*SVM* Combined with *Feature Importance* Logistic Regression Coeffcient	Tanpa *SD-IOT SVM* LINEAR 100% dan Akurasi *SVM RBF* denggan *SD-IoT* adalah 74.3%. untuk rata-rata semua rate serangan dan tapa *SD-IoT* adalah 99%
2	Hui , dkk. (2017)	A *Machine Learning* approach for *DDoS* attack detection in *Internet of Things* systems	Skema *Training* dari Hui	Random forest	Hasil tertinggi terdapat pada Random forest senilai 98.62%
3	Wu , dkk (2018)	*IoT-DDoS* : Detecting *DDoS* attacks against *IoT* using *KNN* algorithm	In*SDN*	k-nearest neighbor (kNN)	Hasil tertinggi terdapat pada *KNN* senilai 99.22%
4	Singh , dkk (2019)	*DDoS* attack detection in *IoT* using *decision tree*	1999 DARPA and Skema *Training* dari Singh	*Decision tree*	Hasil tertinggi terdapat pada *Decision tree* senilai 95.48%
5	Zhang , dkk (2019)	Deep learning based *DDoS* attack detection in	In*SDN* and Skema *Training* dari	Deep learning	Hasil tertinggi terdapat pada Deep learning senilai 99.65%

		IoT network	Zhan		
6	Chang , dkk (2020)	*DDoS* attack detection in *IoT* networks using random forest	In*SDN*	Random forest	Hasil tertinggi terdapat pada Random forest senilai 99.2%, memiliki rata rata akurasi 98.11%
7	Bao , dkk (2020)	A feature selection method based on Fisher score for *DDoS* attack detection in *IoT*	Skema *Training* dari Bao	Fisher score dan *decision tree*	Hasil tertinggi terdapat pada *Decision tree* senilai 97.1%.
8	Siddiqui , dkk (2021)	A hybrid machine learning-based *DDoS* detection scheme for *IoT* networks	Skema *Training* dari dan Siddiqui	*Support Vector Machine* (SVM) dan *decision tree*	Hasil tertinggi terdapat pada *SVM* senilai 99.4%, memiliki rata rata akurasi 99.2%
9	Li , dkk (2021)	A real-time *DDoS* attack detection method for *IoT* based on machine learning	Skema *Training* dari Li	Long short-term memory (LSTM) neural network	Hasil tertinggi terdapat pada LSTM senilai 98.5%.

2. Jaringan *SD-IoT*

Jaringan *SD-IoT* adalah sebuah jaringan yang terdiri dari berbagai perangkat *IoT (Internet of Things)* yang saling terhubung dan berkomunikasi satu sama lain. Jaringan ini digunakan untuk menghubungkan berbagai perangkat di rumah, gedung, atau kota yang menggunakan teknologi *IoT* seperti sensor, aktuator, dan perangkat cerdas lainnya.[10]

3. *LrDDoS Attack*

LrDDoS (Low-Rate Distributed Denial of Service) Attack adalah sebuah serangan *DDoS* (Distributed Denial of Service) yang dilakukan dengan cara mengirimkan lalu lintas jaringan yang sangat kecil atau rendah secara terus-menerus ke target. Serangan ini bertujuan untuk membuat target terlambat dalam memproses permintaan atau merespons pengguna, sehingga mengganggu layanan yang disediakan oleh target.

4. Machine Learning

Machine Learning adalah sebuah teknik atau metode pengolahan data

yang memungkinkan komputer untuk belajar dan meningkatkan kinerjanya dalam melakukan tugas tertentu, tanpa harus secara eksplisit diprogram. Dalam penelitian ini, teknik *Machine Learning* digunakan untuk mengidentifikasi dan mendeteksi serangan *LrDDoS* pada jaringan *SD-IoT*.[12]

5. *Feature Importance*

Feature Importance adalah sebuah metode dalam *Machine Learning* yang digunakan untuk menentukan fitur atau atribut mana yang paling berpengaruh dalam mempengaruhi hasil atau output dari model. Metode ini sangat berguna dalam pengambilan keputusan dan pemilihan fitur dalam machine learning, terutama dalam kasus yang melibatkan data yang sangat kompleks.[13,14,15]

6. *Random Forest Classification*

Random Forest Classifier adalah sebuah algoritma *Machine Learning* yang digunakan untuk melakukan klasifikasi pada data. Algoritma ini bekerja dengan membangun banyak pohon keputusan yang acak pada dataset, kemudian melakukan voting pada setiap pohon keputusan untuk menentukan kelas atau label pada data yang diberikan. *Random Forest Classifier* memiliki keunggulan dalam menangani data yang tidak seimbang dan memiliki performa yang cukup baik pada berbagai jenis dataset. Dalam penelitian ini, algoritma *Random Forest Classifier* digunakan untuk mendeteksi serangan *LrDDoS* pada jaringan *SD-IoT*.[16]

7. Algoritma

Dalam penelitian ini, ke-8 algoritma di bawah digunakan untuk membandingkan performa deteksi *LrDDoS* Attack pada jaringan *SD-IoT* dengan menggunakan metode *Machine Learning* dengan *Feature Importance* dan metode *Random Forest Classifier*.

a. *SVM (Support Vector Machine)*: *SVM* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. *SVM* mencari batas keputusan yang optimal yang memisahkan dua kelas dengan margin terbesar. [17-18]

b. *RBF* (*Radial Basis Function*): *RBF* adalah salah satu jenis jaringan saraf tiruan yang paling umum digunakan dalam pembelajaran mesin. *RBF* menghitung jarak antara input dan pusat kluster, kemudian menghitung outputnya berdasarkan fungsi aktivasi.[19-20]

c. *RFC* (*Random Forest Classifier*): *RFC* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi. *RFC* menggunakan sejumlah besar pohon keputusan untuk membuat prediksi. Setiap pohon memilih prediksi kelas, dan kelas dengan mayoritas pemilihan menjadi hasil prediksi.[21-22]

d. DTC (*Decision tree* Classifier): DTC adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi. DTC membangun pohon keputusan yang menggambarkan serangkaian keputusan yang diambil berdasarkan fitur input, hingga mencapai label kelas yang diinginkan.[23-27]

e. *MLP* (Multilayer Perceptron): *MLP* adalah jaringan saraf tiruan dengan banyak lapisan. *MLP* memproses input melalui beberapa lapisan, yang masing-masing memiliki sejumlah neuron. Setiap neuron menghitung bobot tertentu dari input dan menghasilkan keluaran yang diteruskan ke neuron di lapisan berikutnya.[28-31]

f. *GNB* (*Gaussian Naive Bayes*): *GNB* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi. *GNB* mengasumsikan bahwa setiap fitur input memiliki distribusi normal, dan memperkirakan probabilitas kelas dengan menerapkan aturan Bayes.[32-35]

g. *ADB* (*AdaBoost*): *ADB* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi. *ADB* menggabungkan beberapa model pembelajaran mesin lemah untuk membuat model yang lebih kuat. Setiap model lemah diberi bobot sesuai dengan tingkat kesalahan prediksi dan digunakan untuk memperbarui bobot sampel pelatihan.[36-37]

h. *KNN* (*K-Nearest Neighbors*): *KNN* adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. *KNN* memperkirakan label kelas atau nilai target dari data uji dengan mencari tetangga terdekat dari data pelatihan. Data pelatihan dikelompokkan ke dalam kelompok berdasarkan kesamaan fitur, dan tetangga terdekat dihitung dengan menghitung jarak antara data pelatihan dan data uji.[38-42]

8. *OpenFlow*

OpenFlow adalah sebuah protokol komunikasi yang digunakan untuk mengatur lalu lintas jaringan dalam arsitektur *SDN* (*Software-Defined*

Networking). Pada arsitektur *SDN*, perangkat jaringan seperti switch dan router diatur dan dikonfigurasi secara sentral melalui kontroler *SDN*. *OpenFlow* memungkinkan kontroler untuk mengirimkan instruksi ke perangkat jaringan untuk mengatur arus lalu lintas.[43-45]

API (Application Programming Interface) *OpenFlow* adalah kumpulan dari fungsi dan protokol yang digunakan untuk berinteraksi dengan perangkat jaringan yang mendukung protokol *OpenFlow*. API *OpenFlow* memungkinkan pengembang untuk membuat aplikasi dan layanan *SDN* yang dapat mengontrol lalu lintas jaringan secara fleksibel melalui kontroler *SDN*. Dalam hal ini, API *OpenFlow* memungkinkan pengembang untuk mengakses dan mengatur alur lalu lintas pada jaringan *SDN*, serta mengambil data lalu lintas yang berkaitan dengan alur lalu lintas tersebut.[46-47]

9. *Mininet-IoT*

Mininet-IoT adalah sebuah emulator jaringan yang dirancang khusus untuk menguji aplikasi jaringan *Internet of Things* (*IoT*) dengan topologi jaringan yang dapat disesuaikan. *Mininet-IoT* berbasis pada *Mininet*, yang merupakan sebuah emulator jaringan berbasis Linux. [48]

Mininet-IoT menyediakan banyak fitur untuk menguji aplikasi *IoT*, termasuk kemampuan untuk menambahkan berbagai jenis perangkat *IoT*, seperti sensor dan aktuator, dan kemudian menghubungkannya ke jaringan yang dihasilkan. Selain itu, *Mininet-IoT* juga mendukung berbagai protokol *IoT*, seperti *MQTT* dan *CoAP*.[48-50]

Untuk menghubungkan *Mininet-IoT* dengan *OpenFlow*, diperlukan perangkat lunak pengontrol yang sesuai. *RYU Controller* adalah salah satu pilihan yang populer untuk pengontrol *OpenFlow*. *RYU Controller* adalah platform pengontrol jaringan berbasis *Python* yang dapat digunakan untuk mengontrol perangkat jaringan yang mendukung protokol *OpenFlow*.[51,53]

Dalam penggunaannya, *Mininet-IoT* dapat dihubungkan dengan *RYU Controller* melalui protokol *OpenFlow*, sehingga kontrol jaringan dapat dilakukan dengan menggunakan *RYU Controller*. Dengan demikian, *Mininet-IoT* dan *RYU Controller* dapat bekerja sama untuk menguji aplikasi jaringan *IoT* dengan menggunakan protokol *OpenFlow*[51]

10. *RYU Controller*

RYU adalah salah satu contoh open-source *Controller SDN* yang dapat digunakan untuk membangun jaringan *SDN*. *RYU* dikembangkan menggunakan bahasa pemrograman *Python* dan mendukung protokol *OpenFlow* dan *NETCONF*. [51]

RYU Controller berfungsi sebagai pengontrol arus lalu lintas dalam jaringan *SDN*, dimana setiap switch *OpenFlow* diatur oleh *RYU Controller* melalui pesan *OpenFlow*. *RYU Controller* memiliki banyak fitur dan fungsi yang dapat dikustomisasi, seperti pemrosesan kebijakan dan keamanan jaringan, pengaturan QoS, dan manajemen sumber daya jaringan.[51,52]

Dalam kaitannya dengan penghubungan dengan *OpenFlow*, *RYU Controller* dapat berfungsi sebagai kontroler pusat yang mengelola seluruh switch *OpenFlow* di dalam jaringan. Setiap switch *OpenFlow* terhubung ke *RYU Controller* dan berkomunikasi dengan *Controller* untuk menerima perintah dan memberikan informasi tentang status jaringan.[51,52,55]

RYU Controller dapat digunakan dengan *Mininet-IoT* untuk menguji jaringan *SDN* yang dibangun dengan protokol *OpenFlow*. Kedua alat ini dapat digunakan bersama untuk membangun topologi jaringan *SDN*, mengonfigurasi switch *OpenFlow*, dan melakukan pengujian keamanan jaringan.[53]

11. *CoAP* Server

Constrained Application Protocol (*CoAP*) adalah protokol komunikasi ringan yang dirancang untuk digunakan dalam lingkungan dengan sumber daya terbatas, seperti jaringan sensor nirkabel dan perangkat *Internet of Things* (*IoT*). *CoAP* memungkinkan perangkat untuk berkomunikasi melalui protokol *HTTP* yang lebih sederhana dan efisien.[56]

Dalam konteks jaringan *SD-IoT*, *CoAP* dapat digunakan sebagai protokol komunikasi antara perangkat *IoT* dan server. *CoAP* memungkinkan perangkat *IoT* untuk memperoleh atau memperbarui sumber daya dari server secara efisien, dengan overhead jaringan yang minimal.[56]

Untuk mengaitkan *CoAP* dengan jaringan *SD-IoT*, perlu ada pengaturan yang tepat pada perangkat *IoT* dan server untuk mendukung protokol *CoAP*. Selain itu, dapat diterapkan teknik keamanan untuk melindungi komunikasi

antara perangkat dan server.[56-57]

Dalam penelitian ini, *CoAP* server dapat membantu dalam merancang sistem deteksi *LrDDoS* attack pada jaringan *SD-IoT* menggunakan machine learning. *CoAP* server dapat digunakan sebagai protokol komunikasi antara perangkat *IoT* dan server yang diuji dengan serangan *LrDDoS* , dan hasilnya dapat dianalisis dengan algoritma machine learning.

12. *TcpReplay*

TcpReplay adalah sebuah *Tool* open-source yang digunakan untuk merekam dan memainkan kembali paket jaringan. *Tool* ini digunakan untuk mensimulasikan lalu lintas jaringan pada lingkungan yang berbeda-beda, dan sering digunakan dalam pengujian dan debugging jaringan.[58-59]

TcpReplay dapat merekam paket jaringan yang ditangkap oleh tcpdump, Wireshark atau *Tool* sejenisnya. Paket yang sudah direkam dapat disimpan dalam berbagai format file, seperti *PCAP* atau *PCAPNG*, dan kemudian dimainkan kembali menggunakan *TcpReplay*. Dalam penggunaannya, *TcpReplay* dapat mengirim paket jaringan ke alamat tujuan yang diinginkan, baik melalui interface jaringan yang sesungguhnya maupun melalui interface virtual yang dibuat menggunakan *Tool* seperti *Netmap* atau *VirtualBox*.

Dalam pengembangan *SD-IoT*, *TcpReplay* dapat digunakan untuk merekam dan memainkan kembali lalu lintas jaringan pada lingkungan simulasi *SDN* yang dibangun dengan menggunakan *Mininet*.[59] Hal ini memungkinkan pengujian sistem *SD-IoT* pada lingkungan yang lebih realistis, dan dapat membantu dalam mengidentifikasi dan menyelesaikan masalah yang mungkin terjadi pada jaringan *SD-IoT* yang sebenarnya.

13. Python

Python adalah bahasa pemrograman tingkat tinggi, yang terkenal dengan sintaksis yang mudah dibaca dan mudah dipahami. *Python* dirancang dengan filosofi yang menekankan keterbacaan kode, sehingga menghasilkan kode yang lebih bersih dan lebih mudah dipelihara. *Python* mendukung berbagai paradigma pemrograman seperti pemrograman prosedural, pemrograman berorientasi objek, dan pemrograman fungsional.[60]

Python juga memiliki banyak modul dan *library* yang memudahkan dalam mengembangkan berbagai jenis aplikasi, mulai dari aplikasi desktop hingga aplikasi web dan machine learning. Salah satu modul populer di *Python* adalah *NumPy, Pandas, Matplotlib*, dan *TensorFlow*.[60-62]

Python juga digunakan secara luas di bidang pengembangan web, pengembangan game, ilmu data, dan kecerdasan buatan (*AI*). Kelebihan *Python* sebagai bahasa pemrograman antara lain mudah dipelajari, memiliki sintaksis yang bersih dan mudah dibaca, dan memiliki banyak dukungan modul dan *library* yang memudahkan pengembangan aplikasi.[60-62]