**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Pada masa ini, perkembangan teknologi sangatlah cepat. Salah satu diantaranya adalah Iot. Perangkat Iot tidak memiliki antarmuka pengguna, sumber daya komputasi, serta media penyimpanan yang berfungsi untuk mengimplementasikan *firewall* dan alat diagnosis lainnya [1]. Oleh karena itu, perangkat Iot memiliki celah keamanan yang dapat dimanfaatkan oleh *attacker* untuk melemahkan proses komunikasi antara perangkat server Iot dengan pengguna [2].

Salah satu solusi mengatasi permasalahan dalam jaringan IoT adalah mengintegrasikan dengan arsitektur *Software Defined Network* (SDN). Penggabungan antara SDN dengan IoT dikenal dengan SD-IoT [3]. SDN berperan sebagai pengarah lalu lintas jaringan yang mampu mengelola sumber daya serta keamanan jaringan IoT [4]. Layer kontrol SDN berguna sebagai pusat manajemen trafik serta menjadi modul *Intrusion Detection System* (IDS) untuk menaaggulangi celah keamanan pada jaringan IoT.

Namun, sistem kontrol terpusat pada SDN memiliki ancaman utama yaitu serangan *Distributed Denial of Service* (DDoS) [5]. Serangan DDoS merupakan salah satu ancaman yang berbahaya. *Amazon Web Services* (AWS) melaporkan bahwa pada Q1 2020 menahan serangan DDos sebesar 2,3 Tbit/s [6]. DDoS bertujuan untuk melumpuhkan jaringan target dengan cara mengirimkan banyak paket secara bertubi-tubi dengan ukuran yang besar [7]. Serangan DDoS terdiri dari *Low-Rate* *Distributed Denial of Service* (LRDDoS) dan *High-Rate* *Distributed Denial of Service* (HRDDoS) [8]. HRDDoS memiliki karakteristik lalu lintas pengiriman paket data yang lebih dominan sehingga lebih mudah dideteksi, sedangkan LRDDoS sangat sulit dideteksi, karena serangan disembunyikan pada aliran data normal [9]. Oleh karena itu, LRDDoS menjadi perhatian utama dalam masalah keamanan SDN, sebab itu diperlukan sebuah sistem yang mampu mengidentifikasi dan mendeteksi serangan tersebut [10].

Pada penelitian terdahulu, telah diusulkan beberapa metode untuk deteksi serangan LRDDoS. Pada penelitian [11] telah dibuat sistem deteksi LRDDoS dengan metode *machine learning* menggunakan algoritma *K-Nearest Neighbors* (KNN). Jumlah dataset yang digunakan pada penelitian ini berjumlah 160.007 data training dan 39.995 data test. Hasil eksperimen untuk 10 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 50.51% dan *prediction loss* sebesar 1.682%. Hasil eksperimen untuk 20 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 92.71% dan *prediction loss* sebesar 82,67%. Hasil eksperimen untuk 50 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 13.21% dan *prediction loss* sebesar 82,48%. Hasil eksperimen untuk 100 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 58.47% dan *prediction loss* sebesar 99,27%. Hasil eksperimen untuk 200 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 62.28% dan *prediction loss* sebesar 99,27%.

Pada penelitian [12] telah dibuat sistem deteksi LRDDoS dengan metode *machine learning* menggunakan algoritma *Random Forest* dengan *Logistic Regression Coefficient*. Jumlah dataset yang digunakan pada penelitian ini adalah 160.006 data training dan 39.994 data test. Hasil eksperimen untuk 50 *Packet Sending Rate* (pps) ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 92.3% dan *prediction loss* sebesar 98,5%. Hasil eksperimen untuk 100 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 98.2% dan *prediction loss* sebesar 98,8%. Hasil eksperimen untuk 200 pps ini mendapatkan *accuracy, precision, recall, f1-score* sebesar 98.7% dan *prediction loss* sebesar 99,1%.

Pada penelitian [13] telah dibuat sistem deteksi LRDDoS dengan metode *machine learning*. Pada penelitian ini, Jumlah dataset yang digunakan ada 2 yaitu 420.000 data training serta 18.000 data test dan 420.000 data training serta 36.000 data test. Hasil eksperimen dengan data test 18.000 dengan algoritma SVM (RBF) mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 100%. Hasil eksperimen menggunakan algoritma SVM (LIN) mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 100%. Hasil eksperimen menggunakan algoritma KNN mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 87.8%, 93.1%, 88.2%, dan 86.5%. Hasil eksperimen menggunakan algoritma DTC mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 100%. Hasil eksperimen menggunakan algoritma RFC mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 84.2%, 86.4%, 84.6%, dan 83.1%. Hasil eksperimen menggunakan algoritma MLP mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 34.5%, 32.5%, 36.3%, dan 23%. Hasil eksperimen menggunakan algoritma GNB mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 93.9%, 95%, 94.1%, dan 93.5%. Sedangkan hasil eksperimen dengan data test 36.000 dengan algoritma SVM (RBF) mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 100%. Hasil eksperimen menggunakan algoritma SVM (LIN) mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 100%. Hasil eksperimen menggunakan algoritma KNN mendapatkan *accuracy, precision, recall,* dan *f1-score* sebesar 95.6%, 89.2%, 92.9%, dan 87.5%. Hasil eksperimen menggunakan algoritma DTC mendapatkan *accuracy, precision, recall, dan f1-score* sebesar 89.7.6%, 78.4.2%, 83.3%, dan 80.4%. Hasil eksperimen menggunakan algoritma RFC mendapatkan *accuracy, precision, recall, dan f1-score* sebesar 80.7%, 88.4%, 84.4%, dan 82.5%. Hasil eksperimen menggunakan algoritma MLP mendapatkan *accuracy, precision, recall, dan f1-score* sebesar 65.3%, 52.4%, 66.6%, dan 54.2%. Hasil eksperimen menggunakan algoritma GNB mendapatkan *accuracy, precision, recall, dan f1-score* sebesar 59.1%, 68.6%, 83.1%, dan 70%.

Berdasarkan pada penelitian [13], dalam penelitian ini diusulkan deteksi serangan menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost* pada jaringan SD-Iot dengan dataset yang sama pada penelitian [13] yang didapatkan dari website Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,UDP) [14]. Diharapkan metode yang diusulkan dapat meningkatkan akurasi pada penelitian [13].

* 1. **Rumusan Masalah**

Berdasarkan latar belakang diatas, rumusan masalah pada penelitian ini adalah

1. Bagaimana cara mendeteksi serangan LRDDoS pada jaringan SD-IoT menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost* ?
2. Apakah deteksi serangan LRDDoS menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost* ini efektif, jika dinilai menggunakan variabel *accuracy, precission, recal, f1-score dan predict loss* ?
   1. **Tujuan Penelitian**

Tujuan dari penelitian ini sebagai berikut :

1. Mengetahui hasil deteksi serangan LRDDoS pada jaringan SD-IoT.
2. Mengetahui hasil *accuracy, precission, recal, f1-score dan predict loss* menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost.*
   1. **Batasan Masalah**

Batasan masalah dari penelitian ini sebagai berikut :

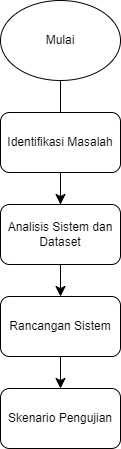
1. Dataset yang digunakan pada penelitian ini didapatkan dari website Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,UDP) [14].
2. Menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost* untuk mendeteksi serangan.
3. Menggunakan mininet untuk emulasi jaringan pada SDN.
4. Menggunakan bahasa pemrograman python.
5. Menggunakan RYU sebagai controller pada jaringan SDN.
6. Menggunakan sistem operasi Kali Linux.

**BAB II**

**METODOLOGI PENELITIAN**

* 1. **Alur Penelitian**

Pada bab ini akan membahas beberapa tahapan alur penelitian yang akan digunakan seperti indentifikasi masalah, analisis sistem dan dataset, rancangan sistem dan skenario pengujian.



*Gambar 2.1 Alur Penelitian*

* 1. **Penelitian Terdahulu**

Tabel 2.1 merupakan beberapa penelitian terdahulu yang digunakan sebagai referensi dan acuan dasar dalam melakukan penelitian :

*Tabel 2.1 Penelitian Terdahulu*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Penulis | Judul | Dataset | Metode | Hasil |
| 1 | Achmad Irfani Nur Iman, Fauzi Dwi Sumadi, Zamah Sari | Low Rate DDOS Attack Detection Using KNN On SD-IOT | Dataset dibuat sendiri oleh penulis | *K-Nearest Neighbors (KNN)* | Hasil eksperimen untuk 10 pps mendapatkan *accuracy, precision, recall, f1-score* sebesar 50.51%. Hasil eksperimen untuk 20 pps sebesar 92.71%. Hasil eksperimen untuk 50 pps sebesar 13.21%. Hasil eksperimen untuk 100 pps sebesar 58.47%. Hasil eksperimen untuk 200 pps sebesar62.28%. |
| 2 | Wahyuli Dwiki Nanda, Fauzi Dwi Setiawan Sumadi | LRDDoS Attack Detection on SD-IoT Using Random Forest with Logistic Regression Coefficient | Mendeley Data dengan judul Low Rate DDoS (MQTT) | *Random Forest* dengan *Logistic Regression Coefficient* | Hasil untuk 50 Packet pps mendapatkan accuracy, precision, recall, f1-score sebesar 92.3%. Hasil untuk 100 pps sebesar 98.2%, 98.2%. Hasil untuk 200 pps sebesar 98.7%. |
| 3 | Fauzi Dwi Setiawan Sumadi, Christian Sri Kusuma Aditya | Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network | Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,  UDP) | SVM (RBF), SVM (LIN), KNN, DTC, RFC, MLP, GNB | Hasil untuk 420.000:18.000 menggunakan SVM (RBF) mendapatkan *accuracy, precision, recall, f1-score* sebesar 100%. Hasil untuk SVM (LIN) sebesar 100%. Hasil untuk KNN sebesar 87%, 93%, 88% dan 86%. Hasil untuk DTC sebesar 100%. Hasil untuk RFC sebesar 84%, 86%, 84% dan 83%. Hasil untuk MLP sebesar 34%, 32%, 36% dan 23%. Hasil untuk GNB sebesar 93%, 95%, 94% dan 93%.  Hasil untuk 420.000:36.000 menggunakan SVM (RBF) mendapatkan *accuracy, precision, recall, f1-score* sebesar 100%. Hasil untuk SVM (LIN) sebesar 100%. Hasil untuk KNN sebesar 95%, 89%, 92% dan 87%. Hasil untuk DTC sebesar 89%, 78%, 83% dan 80%. Hasil untuk RFC sebesar 80%, 88%, 84% dan 82%. Hasil untuk MLP sebesar 65%, 52%, 66% dan 54%. Hasil untuk GNB sebesar 59%, 68%, 83% dan 70%. |

Berdasarkan paparan penelitian diatas, dalam penelitian ini penulis akan menggunakan metode *Ensemble* dengan *Decision Tree, Random Forest*, serta *AdaBoost* dan dataset yang sama pada penelitian no.3 yang didapatkan dari website Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,UDP) [14]. Diharapkan metode yang diusulkan dapat meningkatkan akurasi pada data test 18.000 dan data test 36.000.

* 1. **Identifikasi Masalah**

Berdasarkan latar belakang yang sudah dijelaskan, penelitian ini akan melakukan percobaan untuk mendeteksi serangan LRDDoS menggunakan dataset yang diambil dari Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,UDP). Dalam penelitian ini, penulis akan mengidentifikasi dan menganalisis serangan LRDDoS jenis ICMP, TCP dan UDP pada jaringan SD-IoT dengan menggunakan *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost.*

* 1. **Analisis Sistem dan Dataset**

Pada sub bab ini akan menjelaskan tentang kebutuhan sumber daya perangkat, dan kebutuhan data yang akan digunakan.

* + 1. Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini adalah sistem operasi, emulator SDN, SDN controller dan beberapa perangkat lunak lainnya.

:

*Tabel 2.2 Spesifikasi Perangkat Lunak*

|  |  |
| --- | --- |
| Kebutuhan | Versi |
| Sistem Operasi | Kali Linux |
| SDN Controller | Ryu |
| Emulator SDN | Mininet |
| Southbound API | Openflow |
| IoT Application | CpAPthon 3 |

* + 1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini memiliki spesifikasi sebagai berikut :

*Tabel 2,3 Spesifikasi Perangkat Keras*

|  |  |
| --- | --- |
| Nama | Spesifikasi |
| Processor | Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.59 GHz |
| Memory (RAM) | 8 GB |
| SSD | 500GB |
| GPU | NVIDIA GeForce GTX 1650 |
| System Type | x64-based processor |

* + 1. Dataset

Dataset pada penelitian ini didapatkan dari Mendeley Data dengan judul SDN-DDOS (ICMP,TCP,UDP).

1. Dekripsi

Data terdiri dari serangan LRDDoS di SDN yang meliputi ICMP, TCP, dan UDP. Serangan itu dihasilkan di Emulator Mininet menggunakan *Scapy* dan *TCPReplay*. Topologi yang digunakan adalah : sudo mn --topo tree, depth=3,fanout=2 --mac --controller remote,ip=0.0.0.0:6633 yang dilakukan oleh H1 ke H4.

1. Langkah-Langkah Pembuatan

* File \*.pcap dihasilkan dari proses pembuatan paket menggunakan library *scapy*. Setiap paket berbahaya terdiri dari alamat sumber IP yang dihasilkan secara acak yang ditargetkan ke H4.
* Pada tahap berikutnya, RYU controller diedit agar memiliki kemampuan untuk menyimpan informasi serangan dalam file \*.csv yang diekstraksi dari informasi header paket (IPv4, TCP, UDP, dan ICMP) ditambah dengan data statistik port berdasarkan informasi paket masuk.

1. Institusi

Universitas Muhammadiyah Malang

1. Lisensi

CC BY 4.0

1. Pembuat Dataset

Oxicusa Gugi Housman, Hafida Isnaini, Fauzi Dwi Setiawan Sumadi

Pada tabel 2.4 akan memaparkan rincian seluruh fitur yang pada dataset

*Tabel 2.4 List Fitur*

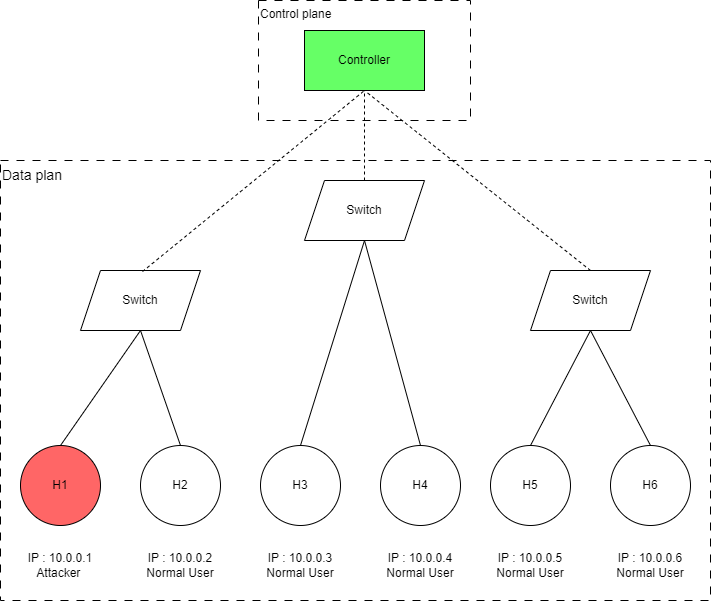
|  |  |
| --- | --- |
| No | NamaFitur |
| 1 | datapath\_id |
| 2 | version |
| 3 | header\_length |
| 4 | tos |
| 5 | total\_length |
| 6 | flags |
| 7 | offset |
| 8 | ttl |
| 9 | proto |
| 10 | csum |
| 11 | src\_ip |
| 12 | dst\_ip |
| 13 | src\_port |
| 14 | dst\_port |
| 15 | tcp\_flag |
| 16 | type\_icmp |
| 17 | code\_icmp |
| 18 | csum\_icmp |
| 19 | port\_no |
| 20 | rx\_bytes\_ave |
| 21 | rx\_error\_ave |
| 22 | rx\_dropped\_ave |
| 23 | tx\_bytes\_ave |
| 24 | tx\_error\_ave |
| 25 | tx\_dropped\_ave |

Seluruh fitur dalam tabel 2.4 digunakan untuk proses training model klasifikasi dan menjadi parameter input saat model diimplementasikan kedalam sistem deteksi. Jumlah data pada dataset ini berjumlah 420.000 data train dan 180.000 data test. Data test yang akan digunakan pada penelitian ini berjumah 18.000 dan 36.000. Untuk menyeleksi data test dari jumlah 180.000 menjadi 18.000 dan 36.000, penulis menggunakan 18000.py dan 36000.py.

* 1. **Rancangan Sistem**

Pada sub bab ini menjelaskan langkah-langkah serangan *low rate* dan deteksi menggunakan *Ensemble* dengan *Decision Tree, Random Forest*, serta *AdaBoost*.

* + 1. Topologi Jaringan



*Gambar 2.2 Rancangan Topologi*

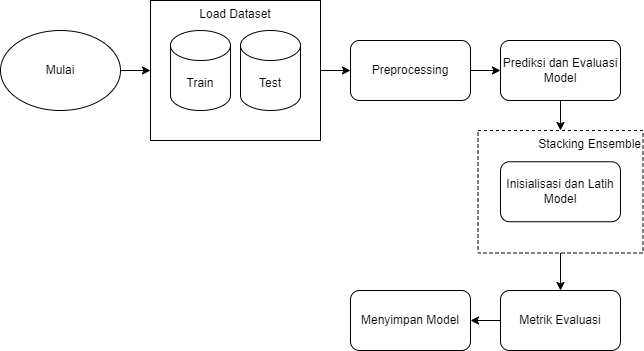
Pada gambar 2.2, Topologi yang digunakan pada penelitian ini yaitu topologi *tree* yang terdiri dari satu *controller*, tiga *switch* dan masing-masing *switch* memiliki dua *host*. *Host* satu (H1) bertindak sebagai *attacker* yang akan mengirimkan serangan. Serangan yang dikirimkan berupa LRDDoS melalui tools *tcpreplay* dengan lima paket serangan yang berbeda.

* + 1. Paket Serangan Low Rate

Paket serangan dibagi menjadi lima *packet rate per-second* (pps) yaitu 10pps, 20pps, 50pps, 100pps, 200pps. Pengiriman paket serangan dilakukan oleh H1 menuju *controller* yang memanfaatkan *tools* *tcpreplay*. *Controller* akan melakukan ekstraksi dengan menggunakan API *handler* melalui *statereplay* yang disimpan kedalam dataset serta model klasifikasi yang ada pada *controller*. *Controller* akan melakukan klasifikasi paket jika terdapat serangan *low rate* yang masuk kedalam jaringan SD-IoT maka akan terdeteksi.

* + 1. Model Klasifikasi

Beberapa tahapan yang dilakukan untuk membuat model klasifikasi *Ensemble* pada jaringan SD-IoT dijelaskan pada gambar dibawah ini.



*Gambar 2.3 Diagram pembuatan model*

Pada gambar 2.3, Penulis menggunakan *Ensemble* dengan *Decision Tree, Random Forest*, dan *AdaBoost*. Untuk *Decision Tree,* penulis menggunakan algoritma *entropy* dengan *max\_depth* sebanyak lima. Untuk *Random Forest,* penulis menggunakan *n\_estimators* sebanyak dua puluh. Sedangkan untuk *AdaBoost,* penulis menggunakan *n\_estimators* sebanyak sepuluh. Setelah itu model tersebut disimpan kedalam file berformat \*.sav.

* 1. **Skenario Pengujian**

Skenario pengujian dilakukan untuk mengetahui performa deteksi dari model klasifikasi *Ensemble* pada *controller* SD-IoT. Model klasifikasi *Ensemble* memiliki empat parameter hasil sebagai acuan, diantaranya : *accuracy, precision, recall dan f1-score*. Model klasifikasi akan diletakkan pada *controller* SDN, model akan melakukan klasifikasi pada paket uji yang masuk kedalam *controller*. Terdapat parameter tambahan yaitu *predict loss* digunakan untuk bahan *evaluasi* yang didapatkan dari *indexing* karena pada saat melakukan pengiriman paket uji memiliki kemungkinan ada paket yang tidak terdeteksi oleh model klasifikasi.

*Tabel 2.5 Hasil Evaluasi*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter (%) | *Decession Tree* | | | | |
| 10pps | 20pps | 50pps | 100pps | 200pps |
| *Accuracy* |  |  |  |  |  |
| *Precision* |  |  |  |  |  |
| *Recall* |  |  |  |  |  |
| *F-1 Score* |  |  |  |  |  |
| *Predict Loss* |  |  |  |  |  |