Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

Deteksi Serangan DDoS *SYN Flood* Pada Jaringan *Internet of Things* (IoT) Menggunakan Metode *Deep Neural Network* (DNN)

Syifa Munawarah[1*], Kurniabudi[2], Eko Arip Winanto[3]

[1] Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: [1]syifamunawarah8@gmail.com [2]kurniabudi@unama.ac.id [3]ekoaripwinanto@unama.ac.id
Email Penulis Korespondensi: syifamunawarah8@gmail.com

Artikel Info :
Artikel History :
Submitted : 18-04-2024
Accepted : 25-04-2024
Published : 30-04-2024

Kata Kunci :
IoT, DdoS, Deep Learning, DNN, IDS

Abstrak− Sistem dan aplikasi Internet of Thing (IoT) mulai banyak diimplementasikan kedalam berbagai bidang, hal ini membuat IoT menjadi sasaran menarik untuk kejahata cyber, terutama serangan DDoS seperti SYN Flood, dimana serangan ini membuat ketersediaan layanan terganggu dan membanjiri server sehingga server kehilangan sumber daya. Salah satu cara untuk mendeteksi serangan DDoS dapat menggunakan Intrusion Detection System (IDS) dimana teknik baru dalam penerapan IDS adalah Deep Learning yaitu metode Deep Neural Network (DNN) yang mampu menemukan manipulasi matematis yang tepat untuk mengubah input menjadi output, maka pada penelitian ini mengusulkan penggunaan metode DNN dalam mendeteksi serangan DDoS SYN Flood pada jaringan IoT. Hasil pengujian pada penelitian yang menggunakan dataset CICIoT2023 dengan 14 file yang berisi dua label yaitu DDoS-SYN_Flood dan BenignTraffic memberikan hasil yang memuaskan. Pangujian menggunakan epoch dengan nilai 10, 50, dan 100, menunjukkan bahwa epoch 100 memberikan hasil performa tertinggi, dapat dilihat dari nilai rata-rata accuracy sebesar 99,36%, nilai precision sebesar 99,44%, nilai recall sebesar 99,75% dan nilai f1-score sebesar 99,59%.

Keywords :
IoT, DdoS, Deep Learning, DNN, IDS

Abstract− The implementation of Internet of Things (IoT) systems and applications is increasingly widespread across various fields. This makes IoT an attractive target for cyber crime, especially Distributed Denial of Service (DDoS) attacks such as SYN Flood. This type of attack disrupts service availability and floods servers, causing them to lose resources. One method for detecting DDoS attacks is through an Intrusion Detection System (IDS). A novel technique in IDS implementation is Deep Learning, specifically the Deep Neural Network (DNN) method, capable of identifying precise mathematical manipulations to transform input into output. Therefore, this research proposes the use of the DNN method to detect SYN Flood DDoS attacks in IoT networks. Testing results from the study, which utilized the CICIoT2023 dataset consisting of 14 files with two labels, DDoS-SYN_Flood and BenignTraffic, provided satisfactory outcomes. Evaluation using epochs with values of 10, 50, and 100 showed that epoch 100 yielded the highest performance. This is evident from the average accuracy rate of 99.36%, precision of 99.44%, recall of 99.75%, and an f1-score of 99.59%.

1. PENDAHULUAN

Internet of Things (IoT) telah membuat kemajuan yang signifikan dalam teknologi komunikasi dan informasi. Oleh karena itu, teknologi tersebut telah digunakan diberbagai industri penting untuk memberikan solusi yang hemat biaya, otomatis, berkelanjutan, dan cerdas [1]. Perangkat IoT memiliki kemampuan untuk berkomunikasi, bekerja sama, memproses, menganalisis dan mengirim data secara mandiri. Penerapan aplikasi dan sistem IoT mulai digunakan secara luas diberbagai bidang, seperti pemantauan kesehatan pribadi dan manajemen industri.

Fenomena ini menarik pihak-pihak yang tertarik untuk melakukan serangan *cyber* terhadap infrastruktur dan aplikasi IoT [2]. *Distributed Denial of Service* (DDoS) adalah salah satu dari berbagai teknik serangan yang digunakan oleh para peretas, yang mengambil keuntungan dari kelemahan dalam sebuah sistem. Salah satu kerentanannya adalah ketika peretas membanjiri sumber daya hingga mengakibatkan kesulitan bagi pengguna untuk mengakses layanan web service, serta membuat kinerja jaringan korban menjadi lambat [3]. Keamanan dalam implementasi IoT menjadi krusial untuk melindungi infrastruktur dan aplikasi dari serangan cyber.

Distributed Denial of Service (DDoS) merupakan serangan yang ditujukan untuk menggunakan sumber daya jaringan dan *bandwidth* yang tersedia hanya untuk mencegah akses pengguna asli kejaringan target menjadi terbatas [4]. Salah satu serangan DDoS adalah *SYN Flood*, penyerang akan membanjiri server dengan sejumlah besar paket SYN, memaksanya untuk berulangkali merespon dengan paket SYN ACK. Akibat serangan ini, server tidak dapat menangani permintaan baru, dan sumber dayanya akan terus bertambah [5].

Intrusion Detection System (IDS) dapat digunakan untuk memonitor aliran data jaringan dari penyerang ancaman untuk mengidentifikasi serangan DDoS [6]. Metode baru dalam menerapkan IDS adalah *Deep Learning*, sebuah pendekatan ilmu komputer yang menggunakan teknik statistik untuk memberikan kemampuan pada sistem

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

komputer untuk belajar dari data, salah satu jenis *Deep Learning* yang diterapkan sebagai IDS adalah *Deep Neural Network* (DNN). *Deep Neural Network* adalah jenis *Artificial Neural Network* yang terdiri dari beberapa lapisan yang memisahkan *input* dan *output* [7].

DNN sistem dapat menentukan operasi matematika yang sesuai untuk mengubah *input* menjadi *output*, baik itu berupa hubungan *linear* atau hubungan *non-linear* [7]. Kemampuan metode DNN yang mampu mengekstrak pola-pola yang kompleks dan menentukan operasi matematika yang tepat untuk mengubah *input* menjadi *output*, hal ini membuat metode DNN menjadi pilihan yang tepat dalam mendeteksi intrusi terhadap jaringan, termasuk untuk melindungi infrastruktur IoT dari serangan *cyber*. Dengan menggunakan metode DNN sebagai metode deteksi intrusi, perangkat IoT dapat lebih efektif dilindungi dari berbagai ancaman *cyber* yang terus berkembang, memastikan keberlangsungan dan keamanan sistem secara lebih optimal.

Pada penelitian [8], membahas mengenai penerapan *deep learning* dengan metode DNN untuk mendeteksi dan mengklasifikasi serangan DDoS pada jaringan *network*. Penelitian ini mengatakan bahwa penggunaan metode DNN lebih efektif untuk mendeteksi dan mengklasifikasikan serangan DDoS, pada dataset CICDDoS2019 yang di uji memperoleh hasil pengklasifikasian dengan tingkat akurasi sebesar 94,57%. Karena akurasi yang tinggi dalam analisis jaringan, penerapan metode DNN ke dalam sistem deteksi intrusi dan lapisan keamanan jaringan berbasis perangkat lunak seperti IoT merupakan pilihan yang tepat.

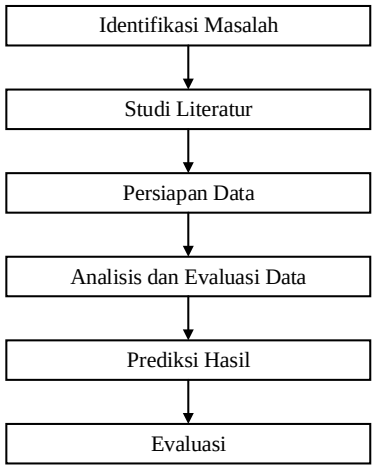
Selanjutnya pada penelitian [7], yang membahas ketidak pastian dalam menemukan jenis serangan dan meningkatkan kompleksitas serangan *cyber*, IDS memerlukan integritas DNN untuk memprediksi serangan pada *Network Intrusion Detection System* (N-IDS). Setelah melakukan uji coba pada dataset KDDCup-'99' dengan hasil *accuracy*, *precision*, *recall*, *dan f1-score* yang dibandingkan antar algoritma, didapat hasil bahwa DNN dengan 3 *layer* mengungguli dari semua algoritma *machine learning* yang lain. Hal ini dikarenakan kemampuan DNN untuk mengekstrak data dan fitur dengan abstraksi yang lebih tinggi dan non-linear menambah keunggulan dibandingan dengan algoritma lainnya.

Berdasarkan penjelasan diatas, penulis akan membahas bagaimana menerapkan *Deep Learning* dengan metode *Deep Neural Network* (DNN) untuk mendeteksi salah satu serangan *Distributed Denial of Service* (DDoS) yaitu *SYN Flood* pada jaringan IoT khususnya seranga *SYN Flood*.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Kerangka kerja penelitian (*framework*) diperlukan untuk membantu mendefinisikan langkah-langkah yang harus diambil untuk melaksanakan penelitian. Adapun kerangka kerja penelitian dapat dilihat pada gambar 1 dibawah ini:



Gambar 1. Kerangka Kerja Penelitian

Berdasarkan kerangka kerja penelitian yang dijabarkan pada gambar 1 maka setiap tahap dalam penelitian dijabarkan masing-masing sebagai berikut:

1. Identifikasi Masalah
   Identifikasi masalah pada penelitian ini adalah menerapkan teknik *deep learning* dalam mendeteksi serangan atau anomaly DDoS *SYN Flood* pada jaringan IoT menggunakan metode *Deep Neural Network* (DNN) serta mengetahui ciri-ciri dari serangan tersebut dan mengetahui akurasi atau kinerja dari metode DNN dalam mendeteksi serangan DDoS *SYN Flood*.
2. Studi Literatur

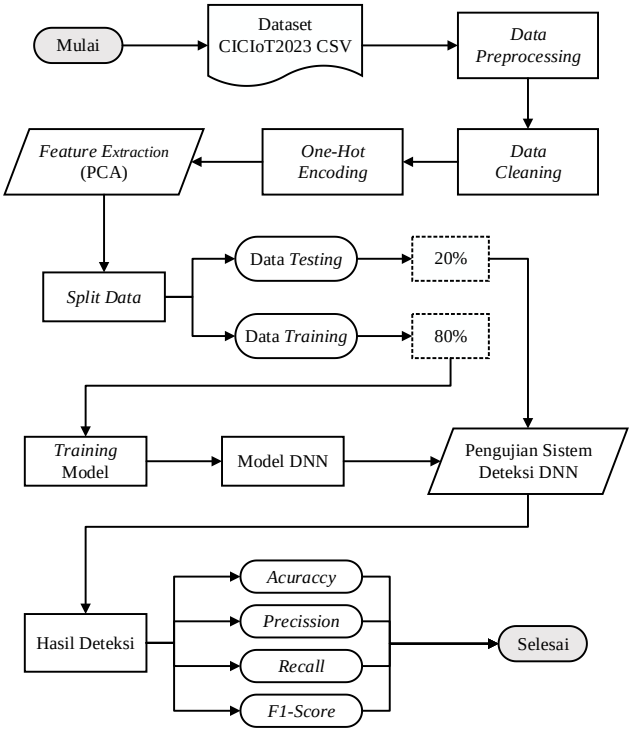
Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

Pada tahap ini, penulis melakukan penelusuran terhadap teori-teori yang relevan dari berbagai sumber seperti buku, artikel, jurnal, dan penelitian terkait yang berkaitan dengan permasalahan yang sedang diteliti. Selain itu, dilakukan pula tinjauan literatur terhadap penelitian sejenis tersebut. Studi literatur ini bertujuan untuk memahami konsep serta metode yang dapat digunakan untuk mengidentifikasi pola serangan DDoS *SYN Flood* menggunakan metode DNN.

3. Persiapan Data
   Dataset yang digunakan pada penelitian ini yaitu CICIoT2023 yang didalamnya memiliki serangan dengan tujuh kategori salah satunya yaitu DDoS yang terdiri dari 168 file dan 46 fitur. Pada tahap ini dilakukan pemilihan file dan jenis serangan yang digunakan serta data normal atau benign. Selain itu dilakukan *feature extraction* menggunakan *Principal Component Analysis* (PCA) pada dataset yang telah dipilih untuk peningkatan kinerja model, efisiensi, dan mengurangi kompleksitas data.
4. Analisis dan Evaluasi Data
   Pada tahap ini, dilakukan analisis data melibatkan pengenalan pola-pola khas dari serangan *SYN Flood*, seperti peningkatan jumlah pesan SYN tanpa diikuti oleh pesan ACK, atau adanya anomali dalam pola komunikasi normal. Kemudian evaluasi dilakukan dengan menggunakan *Confusion Matrix* untuk memperoleh tingkat *accuracy*, *precision*, *recall*, *f1-Score*, guna mengukur sejauh mana kinerja metode DNN dapat mengidentifikasi serangan dengan tepat dan efisien.
5. Prediksi Hasil
   Pada tahap ini, dilakukan pengujian sistem IDS dengan teknik *anomaly-based* dalam deteksi serangan DDoS *SYN Flood* menggunakan metode DNN untuk mendapatkan prediksi hasil berupa *accuracy*, *precision*, *recall*, *f1-Score.* Proses uji coba melibatkan pemberian input kedalam model DNN, dimana model akan mempelajari pola dan karakteristik normal dari lalu lintas jaringan. Teknik *anomaly-based* memungkinkan sistem IDS untuk mengidentifikasi *anomaly* atau perubahan signifikan dari pola normal. Prediksi hasil pengujian metode DNN dalam mendeteksi serangan dapat menunjukkan potensi yang besar dalam meningkatkan keamanan jaringan.
6. Evaluasi
   Pada tahap ini, dilakukan evaluasi mengenai hasil eksperimen mengenai kemampuan model dalam mendeteksi serangan DDoS *SYN Flood* pada jaringan *Internet of Things*.

2.2 Rancangan Eksperimen

Rancangan eksperimen untuk mendeteksi serangan DDoS *SYN Flood* merupakan langkah sistematis yang diperlukan untuk mengetahui tahapan apa saja yang dilakukan dalam pengujian. Adapun rancangan eksperimen yang akan dilakukan pada penelitian ini dapat dilihat pada gambar 2.



Gambar 2. Langkah Eksperimen

Berdasarkan alur eksperimen pada gambar 2, dapat dijelaskan melalui poin-poin berikut:

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

1. Dataset
   Penelitian ini menggunakan dataset CICIoT2023, yang mencakup serangan dengan tujuh kategori yaitu *DDoS*, *DoS*, *Recon*, *Web-based*, *Brute Force*, *Spoofing*, dan *Mirai*. Dataset ini terdiri dari 168 file dengan 46 fitur dan 34 label yang kemudian dipilih data serangan DDoS *SYN Flood* dan benign pada 14 file untuk pengujian, adapun dataset yang digunakan dapat dilihat pada tabel 3.1 dan fitur yang terdapat pada dataset CICIoT2023 dapat dilihat pada tabel 1.

Tabel 1. Dataset

No	Nama Dataset	Serangan	*Benign*
1	*SYNFlood01.csv*	19235	5200
2	*SYNFlood02.csv*	24163	6387
3	*SYNFlood03.csv*	20099	5297
4	*SYNFlood04.csv*	19824	5386
5	*SYNFlood05.csv*	20822	5787
6	*SYNFlood06.csv*	20481	5537
7	*SYNFlood07.csv*	19634	5327
8	*SYNFlood25.csv*	39153	10252
9	*SYNFlood33.csv*	38646	10505
10	*SYNFlood36.csv*	38211	10406
11	*SYNFlood39.csv*	37818	10227
12	*SYNFlood41.csv*	38748	10341
13	*SYNFlood52.csv*	38488	10457
14	*SYNFlood57.csv*	37248	10060
Jumlah		412570	111169
Total		523739	

2. *Data Preprocessing*
   Pada tahap ini, dilakukan data cleaning agar dapat diolah dengan lebih efektif oleh metode yang akan digunakan, kemudian dilakukan one-hot encoding dimana setiap element vektor dipresentasikan sebagai nilai biner, dan dilakukan *feature extraction* menggunakan *Principal Component Analysis* (PCA). Adapun tahapan preprocessing sebagai berikut:
   a. *Data Cleaning*
      Pada tahap ini dilakukan data *cleaning*, yang meliputi mencari dan menghilangkan data duplikat, menghilangkan data yang tidak lengkap, dan menghilangkan data yang tidak berharga (kosong).
   b. *One-Hot Encoding*
      Pada tahap ini, dilakukan *one-hot encoding* dimana setiap element vektor dipresentasikan sebagai nilai biner, dimana elemen memiliki nilai 1 (serangan) yang menunjukkan kategori atau label yang sesuai, sementara elemen lainnya bernilai 0 (*benign*).
   c. *Feature Ekstraction*
      Pada tahap ini, dilakukan fitur ekstraksi dan pemilihan fitur atau atribut tertentu dari dataset. Tujuannya adalah untuk mengurangi dimensi data dan meningkatkan efisiensi serta kinerja dari metode yang digunakan. Proses *feature extraction* menggunakan *Principal Component Analysis* (PCA).
3. *Split Data*
   Pada langkah ini, dataset displit menjadi dua bagian, yaitu dataset *training* dan dataset *testing* dengan proporsi 80% untuk *training* dan 20% untuk *testing*. Dataset training digunakan untuk melatih model algoritma, sementara dataset *testing* digunakan untuk menguji kinerja model yang telah dilatih. Setelah itu, model DNN dilatih untuk mendeteksi serangan menggunakan dataset *training*, sehingga mendapatkan model DNN yang akan digunakan dalam proses pengujian.
4. Model DNN
   Setelah data terbagi menjadi dua *training* dan data *testing*, langkah berikutnya melakukan pengujian pada model DNN yang telah dilatih. Proses ini bertujuann untuk mengevaluasi sejauh mana model mampu menggeneralisasi informasi dari data latih ke data yang belum pernah dilihat sebelumnya. Pengujian model mencakup penggunaan data uji untuk mengukur kinerja model dalam menghasilkan akurasi yang baik dalam mendeteksi serangan.
5. Pengujian Sistem Deteksi DNN

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

Pada tahap ini, dilakukan pengujian menggukanan DNN dimana model akan mempelajari pola dari lalulintas jaringan yang normal. Pengujian dilakukan dengan menggunakan data *training* dan hasilnya dibandingkan dengan label data serangan. Proses pengujian dilakukan dengan tiga tingkat *epoch*, yaitu 10, 50, dan 100.

6. Hasil Deteksi
   Setelah melalui tahap pengujian deteksi serangan DDoS *SYN Flood* menggunakan metode DNN, langkah berikutnya evaluasi hasil deteksi menggunakan *Confusion Matrix* untuk melihat nilai *accuracy*, *precision*, *recall*, *dan f1-score* dari model yang digunakan. Proses ini memberikan pemahaman mengenai efektivitas model DNN dalam mengidentifikasi serangan DDoS *SYN Flood*.

3. HASIL DAN PEMBAHASAN

Pada bagian ini, desajikan hasil dan analisis dari eksperimen yang telah dilakukan, termasuk hasil ekstraksi fitur serta evaluasi kinerja model DNN dalam melakukan deteksi serangan.

3.1 Hasil Ekstraksi Fitur

Pada penelitian ini, *Principal Component Analysis* (PCA) digunakan sebagai teknik ekstraksi fitur dengan tujuan mengurangi dimensi dataset dan meningkatkan efisiensi serta kinerja dari model yang digunakan dalam mendeteksi serangan DDoS *SYN Flood*. Adapun hasil ekstraksi fitur dapat dilihat pada tabel 2.

Tabel 2. Hasil *Feature Extraction* PCA

[0.3500068 0.13372558 0.09370266 0.08680644 0.06285212 0.04676881 0.03987176 0.03663694 0.03249668 0.02939277]
[0.327727 0.10831433 0.09862819 0.08683474 0.06466736 0.04673577 0.04420577 0.04102544 0.03723662 0.03559623]
[0.27030408 0.104237 0.09597805 0.08707327 0.07728829 0.06161074 0.04651702 0.04139247 0.03570862 0.03414366]
[0.22783322 0.12901269 0.11895179 0.08698499 0.05953695 0.05358374 0.0520233 0.04222871 0.03702514 0.03613981]
[0.27407722 0.10679336 0.09076606 0.0831784 0.06112167 0.04804827 0.04352739 0.04169831 0.03955106 0.03553716]

Berdasarkan tabel 2 hasil ekstraksi fitur dengan menggunakan PCA yang mengkonversi dari 46 fitur menjadi 10 fitur, hasil ekstraksi fitur tersebut kemudian digunakan dalam proses pengolahan data *training* IDS dengan menggunakan metode DNN.

3.2 Proses Deteksi Serangan Menggunakan DNN

Pada tahap ini, dilakukan eksperimen dengan menggunakan *epoch* 10, 50, dan 100 pada model DNN serta proses pelatihan model. Adapun model yang digunakan dengan *epoch* 10, 50, dan 100 dapat dilihat pada tabel 3.

Tabel 3. Hasil Membangun Model DNN

Model: "DNN-model"
model.add(Dense(128, input_dim=X_train.shape[1], activation='relu')) *model.add(Dense(64, activation='relu'))* *model.add(Dense(32, activation='relu'))* *model.add(Dense(1, activation='sigmoid'))*
Total params: 16385 (64.00 KB) *Trainable params: 16385 (64.00 KB)* *Non-trainable params: 0 (0.00 Byte)*

Berdasarkan tabel 3 model DNN yang dibangun terdiri dari empat *layer*, dimana *layer* pertama merupakan *input layer* yang memiliki 128 *neuron* dengan fungsi aktifitas ReLU (*Rectified Linear Unit*) digunakan untuk memperkenalkan *non-linearitas* pada model, lapisan ini menerima *input* dengan dimensi yang sesuai dengan jumlah fitur pada data *training*. Lapisan kedua merupakan *hidden layer* pertama dengan 64 *neuron* serta fungsi

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

aktivitas ReLU digunakan kembali untuk menambah *non-linearitas* pada model, lapisan ini bertugas untuk mengekstraksi fitur dari *input*. Lapisan ketiga merupakan *hidden layer* kedua dengan 32 *neuron* dan menggunakan fungsi aktivitas ReLU, semakin mendalamnya lapisan tersembunyi dapat membantu model dalam mengekstraksi hierarki fitur yang lebih kompleks. Terakhir adalah lapisan keempat yang merupakan *output layer* yang memiliki 1 *neuron* karena memiliki tugas klasifikasi *biner*, fungsi aktivitas yang digunakan adalah *sigmoid* untuk menghasilkan *output* dalam rentang 0 dan 1 yang dapat diinterpretasikan sebagai probabilitas kelas positif. Total dari keseluruhan parameter DNN berjumlah 16385 parameter.

3.3 Hasil Deteksi

Pada tahap ini, menyajikan hasil deteksi serangan menggunakan metode DNN yang telah dilakukan pengujian dengan menggunakan tiga parameter *epoch*, hasil deteksi serangan *accuracy*, *precision*, *recall*, *dan f1-score.* Berikut pada tabel 4 merupakan hasil deteksi menggunakan *epoch* 10.

Tabel 4. Hasil Performa DNN Dengan *Epoch* 10

No	Dataset	*Accuracy*	*Precision*	*Reccall*	*F1-Score*
1	*SYNFlood01*	98,02%	97,92%	99,61%	98,76%
2	*SYNFlood02*	98,79%	98,77%	99,71%	99,24%
3	*SYNFlood03*	98,66%	98,49%	99,85%	99,17%
4	*SYNFlood04*	99,19%	99,03%	99,95%	99,49%
5	*SYNFlood05*	97,95%	98,27%	99,14%	98,71%
6	*SYNFlood06*	98,35%	98,23%	99,71%	98,96%
7	*SYNFlood07*	98,40%	98,37%	99,64%	99,00%
8	*SYNFlood25*	98,75%	98,64%	99,80%	99,21%
9	*SYNFlood33*	98,12%	97,87%	99,79%	98,82%
10	*SYNFlood36*	99,21%	99,07%	99,93%	99,50%
11	*SYNFlood39*	98,77%	98,80%	99,66%	99,23%
12	*SYNFlood41*	98,05%	97,72%	99,88%	98,79%
13	*SYNFlood52*	98,23%	98,03%	99,77%	98,89%
14	*SYNFlood57*	98,26%	98,17%	99,64%	98,90%

Berdasarkan data pada tabel 4 dapat disimpulkan bahwa dataset ke-10 memiliki nilai tertinggi untuk *accuracy* sebesar 99,21%, nilai tertinggi untuk *precision* juga terdapat pada dataset ke-10 yakni sebesar 99,07%, nilai tertinggi untuk *recall* terdapat pada dataset ke-4 yakni sebesar 99,95%, dan nilai tertinggi untuk *F1-Score* terdapat pada dataset ke-10 yakni sebesar 99,50%. Data dengan *accuracy* tertinggi, ditandai dengan warna kuning. Selanjutnya, pada tabel 5 merupakan hasil deteksi menggunakan *epoch* 50.

Tabel 5. Hasil Performa DNN Dengan *Epoch* 50

No	Dataset	*Accuracy*	*Precision*	*Reccall*	*F1-Score*
1	*SYNFlood01*	98,85%	98,77%	99,79%	99,28%
2	*SYNFlood02*	99,10%	99,05%	99,81%	99,43%
3	*SYNFlood03*	99,06%	98,95%	99,88%	99,41%
4	*SYNFlood04*	99,39%	99,40%	99,82%	99,61%
5	*SYNFlood05*	98,91%	99,12%	99,50%	99,31%
6	*SYNFlood06*	98,65%	98,58%	99,73%	99,15%
7	*SYNFlood07*	99,76%	99,72%	99,92%	99,82%
8	*SYNFlood25*	99,35%	99,26%	99,91%	99,59%
9	*SYNFlood33*	99,30%	99,30%	99,82%	99,56%
10	*SYNFlood36*	99,56%	99,53%	99,91%	99,72%
11	*SYNFlood39*	99,24%	99,18%	99,87%	99,52%
12	*SYNFlood41*	99,30%	99,34%	99,78%	99,56%
13	*SYNFlood52*	99,01%	98,91%	99,85%	99,38%
14	*SYNFlood57*	99,33%	99,32%	99,84%	99,58%

Berdasarkan data pada tabel 5 dapat disimpulkan bahwa dataset ke-7 memiliki nilai tertinggi untuk *accuracy* sebesar 99,76%, nilai tertinggi untuk *precision* juga terdapat pada dataset ke-7 yakni sebesar 99,72%,

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

nilai tertinggi untuk *recall* juga terdapat pada dataset ke-7 yakni sebesar 99,92%, dan nilai tertinggi untuk *F1-Score* juga terdapat pada dataset ke-7 yakni sebesar 99,82%. Data dengan *accuracy* tertinggi, ditandai dengan warna kuning. Selanjutnya, pada tabel 6 merupakan hasil deteksi menggunakan *epoch* 100.

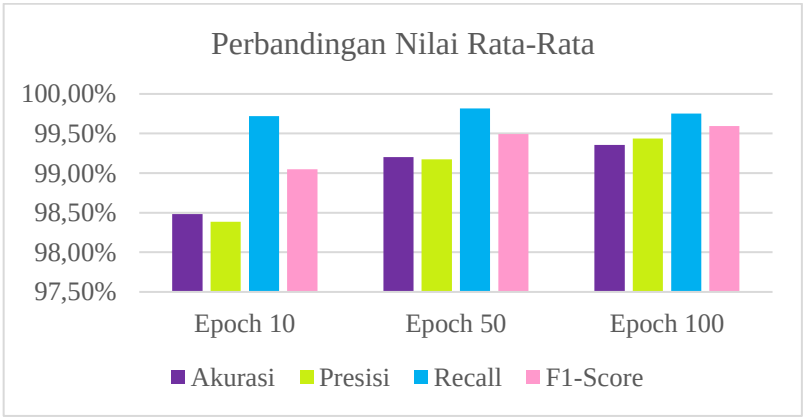
Tabel 6. Hasil Performa DNN Dengan *Epoch* 100

No	Dataset	*Accuracy*	*Precision*	*Reccall*	*F1-Score*
1	*SYNFlood01*	99,22%	99,23%	99,79%	99,51%
2	*SYNFlood02*	99,28%	99,32%	99,77%	99,55%
3	*SYNFlood03*	99,27%	99,53%	99,56%	99,54%
4	*SYNFlood04*	99,52%	99,65%	99,75%	99,70%
5	*SYNFlood05*	98,84%	99,00%	99,52%	99,26%
6	*SYNFlood06*	99,06%	99,18%	99,64%	99,41%
7	*SYNFlood07*	99,68%	99,70%	99,90%	99,80%
8	*SYNFlood25*	99,46%	99,45%	99,86%	99,66%
9	*SYNFlood33*	99,49%	99,64%	99,72%	99,68%
10	*SYNFlood36*	99,59%	99,66%	99,82%	99,74%
11	*SYNFlood39*	99,40%	99,41%	99,83%	99,62%
12	*SYNFlood41*	99,44%	99,62%	99,68%	99,65%
13	*SYNFlood52*	99,36%	99,32%	99,87%	99,59%
14	*SYNFlood57*	99,38%	99,38%	99,83%	99,60%

Berdasarkan data pada tabel 6 dapat disimpulkan bahwa dataset ke-7 memiliki nilai tertinggi untuk *accuracy* sebesar 99,68%, nilai tertinggi untuk *precision* juga terdapat pada dataset ke-7 yakni sebesar 99,70%, nilai tertinggi untuk *recall* juga terdapat pada dataset ke-7 yakni sebesar 99,90%, dan nilai tertinggi untuk *F1-Score* juga terdapat pada dataset ke-7 yakni sebesar 99,80%. Data dengan *accuracy* tertinggi, ditandai dengan warna kuning.

3.4 Perbandingan Performa DNN

Berdasarkan rangkuman nilai keseluruhan proses, maka dapat dilihat pada tabel perbandingan hasil rata-rata dari keseluruhan data selama proses yang mencakup seluruh *epoch*. Adapun hasil rata-rata dari keseluruhan nilai data dapat dilihat pada gambar 3.



Gambar 3. Grafik Perbandingan Nilai Rata-Rata Seluruh *Epoch*

Berdasarkan gambar 3 dapat disimpulkan perbandingan kinerja pada parameter *epoch* 10, 50, dan 100 menunjukkan bahwa *epoch* 100 memberikan hasil performa tertinggi, dapat dilihat dari nilai rata-rata *accuracy* sebesar 99,36%, nilai *precision* sebesar 99,44%, nilai *recall* sebesar 99,75% dan nilai *f1-score* sebesar 99,59%.

3.5 Perbandingan Penelitian Terdahulu

Perbandingan penelitian yang dilakukan dan penelitian terdahulu dapat memberikan wawasan dalam pengembangan ilmiah. Oleh karena itu melalui perbandingan antara penelitian yang dilakukan dan penelitian terdahulu dahulu, penulis dapat memahami sejauh mana kemajuan yang telah dicapai. Adapun perbandingan penelitian yang dilakukan dan penelitian terdahulu dapat dilihat pada tabel 7.

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at http://ejournal.unama.ac.id/index.php/jakakom
Volume 4, Nomor 1, April 2024,
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI https://doi.org/10.33998/jakakom.v4i1

Tabel 7. Perbandingan Dengan Penelitian Terdahulu

No	Penulis	Serangan	Metode	*Accuracy*
1	Xiaoyong Yuan et al. (2017) [35]	DDoS	RNN LSTM dan *Random Forest*	97,60%
2	Abbdullah Emir Cil et al. (2021) [8]	DDoS	DNN	94,57%
3	Penulis	DDoS	DNN	99,36%

Berdasarkan tabel 4.15 dapat disimpulkan bahwa setelah dilakukan perbandingan hasil antara penelitian yang dilakukan dan penelitian terdahulu, menghasilkan peningkatan *accuracy* dalam mendeteksi serangan DDoS menggunakan metode DNN.

4. KESIMPULAN

Penerapan metode DNN dalam mendeteksi serangan DDoS *SYN Flood* menunjukkan hasil deteksi yang sangat baik dengan tingkat akurasi, presisi, recall dan *f1-score* yang tinggi, mencerminkan kemampuannya dalam mengatasi serangan kompleks pada jaringan IoT. Pengujian dengan tiga parameter (*epoch*) menunjukkan bahwa *epoch* 100 memberikan performa tertinggi dengan rata-rata *accuracy* sebesar 99,36%, nilai *precision* sebesar 99,44%, nilai *recall* sebesar 99,75% dan nilai *f1-score* sebesar 99,59%. Kelebihan metode DNN adalah kemampuannya dalam mengekstraksi fitur-fitur kompleks dan abstrak dari data menggunakan PCA, serta dapat disesuaikan dengan data besar dan kompleks untuk memproses informasi dari jaringan yang luas. Namun, kekurangannya adalah membutuhkan sejumlah besar data untuk mencapai kinerja yang optimal, dan mungkin tidak efektif jika data pelatihnya terbatas dalam jumlah dan keragaman.

REFERENCES

[1] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial Internet of Things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, Oct. 2023, doi: 10.1016/j.aej.2023.09.023.

[2] W. Najib, S. Sulistyo, and Widyawan, "Tinjauan Ancaman dan Solusi Keamanan pada Teknologi Internet of Things," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 9, no. 4, pp. 375–384, 2020, doi: 10.22146/jnteti.v9i4.539.

[3] K. B. Virupakshar, M. Asundi, K. Channal, P. Shettar, S. Patil, and D. G. Narayan, "Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-based Private Cloud," *Procedia Comput Sci*, vol. 167, pp. 2297–2307, 2020, doi: 10.1016/j.procs.2020.03.282.

[4] M. Aljanabi, R. Hayder, S. Talib, A. H. Ali, M. A. Mohammed, and T. Sutikno, "Distributed denial of service attack defense system-based auto machine learning algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 544–551, Feb. 2023, doi: 10.11591/eei.v12i1.4537.

[5] Lukman and M. Suci, "Analisis Perbandingan Kinerja Snort Dan Suricata Sebagai Intrusion Detection System Dalam Mendeteksi Serangan Syn Flood Pada Web Server Apache," *Jurnal Teknologi Informasi*, vol. 15, no. 2, pp. 6–15, Jul. 2020, doi: 10.35842/jtir.v15i2.343.

[6] Y. Ariyanto, H. A. V. Firdaus, and H. Pramana, "Klasifikasi Jenis serangan DOS dan Probing pada IDS menggunakan metode K-Nearest Neighbor," *Seminar Informatika Aplikatif Polinema (SIAP)*, pp. 472–476, 2020.

[7] V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*, Oct. 2018, doi: 10.1109/ICCCNT.2018.8494096.

[8] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Syst Appl*, vol. 169, May 2021, doi: 10.1016/j.eswa.2020.114520.

[9] N. Rezaee, S. M. Zanjirchi, N. Jalilian, and S. M. H. Bamakan, "Internet of things empowering operations management; A systematic review based on bibliometric and content analysis," *Telematics and Informatics Reports*, vol. 11, Sep. 2023, doi: 10.1016/j.teler.2023.100096.

[10] Ferdiansyah Zulkifli and Handy N, *INTERNET OF THINGS (IOT) MEDIA PEMBELAJARAN PRAKTIKUM ERA 4.0*. CV. Eureka Media Aksara, 2022.

[11] F. Behmann and Kwok wu., *Collaborative Internet Of Things (C-IoT): for future smart connected life and business*. Texas: John Wiley & Sons, 2015.

[12] Mambang, *BUKU AJAR TEKNOLOGI KOMUNIKASI INTERNET (Internet of Things)*. Purwokerto: CV. Pena Persada, 2022.

[13] R. Vivin, N. Riza, A. Erna, D. Astuti, M. Pramudia, and D. Rahmawati, *FUNDAMENTAL INTERNET OF THINGS (IOT) TEORI DAN APLIKASI PENERBIT CV.EUREKA MEDIA AKSARA*. Jawa Tengah: CV. Eureka Media Aksara, 2023.

[14] K. Lone and S. A. Sofi, "A review on offloading in fog-based Internet of Things: Architecture, machine learning approaches, and open issues," *High-Confidence Computing*, vol. 3, no. 2, Jun. 2023, doi: 10.1016/j.hcc.2023.100124.

[15] Sharipuddin *et al.*, "Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction," *Indonesian Journal of Electrical and Engineering and Informatics (IJEEI)*, vol. 9, no. 3, pp. 747–755, 2021, doi: 10.52549/ijeei.v9i3.3134.