Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 6, No. 1, Januari 2022, hlm. 172-180 http://j-ptiik.ub.ac.id

Klasifikasi Serangan *Distributed Denial-of-Service* (DDoS) menggunakan Metode *Data Mining Naïve Bayes*

Muamar Zidane

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ida1996@student.ub.ac.id

Abstrak

Distributed Deniel of Service (DDoS) merupakan salah satu serangan yang paling populer saat ini. DDoS adalah yang bertujuan untuk menyebabkan *crash* pada sistem server dengan cara membanjiri paket ataupun permintaan pada jaringan. Karakteristik serangan *Distributed Denial of Service* (DDoS) sulit di bedakan dari arus lalu lintas jaringan normal, sehingga untuk mengidentifikasi serangan ini diperlukan sistem yang dapat mengklasifikasi serangan DDoS. Pada penelitian ini telah dibangun sistem klasifikasi serangan DDoS dengan menggunakan metode naïve bayes. Dataset yang digunakan yaitu dataset dari CICIDS2018 yang memilki 84 fitur yang dapat membantu kinerja naïve bayes dalam mengklasifikasi serangan DDoS. Sebagai sampel pengujian, data uji di dapatkan dari hasil uji coba serangan menggunakan program Slowloris kemudian arus lalu lintas tersebut di*capture* secara *real time* menggunakan TCPdump. Hasil *capture* tersebut dilakukan ektraksi fitur dan dikonversi menjadi ekstensi .csv menggunakan *tools* CICFlowMeter. Kemudian data tersebut akan di *preprocessing* guna menghilangkan data yang kosong dan dilakukan seleksi fitur yang paling relevan menggunakan *mutual information* untuk mempermudah kinerja metode naïve bayes dalam melakukan klasifikasi. Tingkat akurasi dari hasil klasifikasi dihitung menggunakan *confusion matrix*. Berdasarkan hasil pengujian, peneliti menemukan bahwa metode yang telah diusulkan dapat mengklasifikasi serangan DDoS dengan tingkat akurasi hingga 95%.

Kata kunci: *DDoS, naïve bayes, confusion matrix, mutual information*

Abstract

Distributed Deniel of Service (DDoS) is one of the most popular attacks today. DDoS is what aims to crash the server system by flooding packets or requests on the network. Characteristics of Distributed Denial of Service (DDoS) attacks are difficult to distinguish from normal network traffic flows, so to identify these attacks, a system that can classify DDoS attacks is needed. In this study, a DDoS attack classification system has been built using the nave Bayes method. The dataset used is a dataset from CICIDS2018 which has 84 features that can help nave Bayes performance in classifying DDoS attacks. As a test sample, the test data is obtained from the results of the attack test using the Slowloris program and then the traffic flow is captured in real time using TCPdump. The capture results are extracted and converted into .csv extensions using the CICFlowMeter tool. Then the data will be preprocessed to eliminate empty data and select the most relevant features to facilitate the performance of the nave Bayes method in classifying. The level of accuracy of the classification results is calculated using a confusion matrix. Based on the test results, the researchers found that the proposed method can classify DDoS attacks with an accuracy rate of up to 95%.

Keywords: *DDoS, naïve bayes, confusion matrix, mutual information*

1. PENDAHULUAN

Serangan *Distributed Denial-of-Service* (DDoS) adalah serangan yang menyebabkan *crash* pada server dan sistem di jaringan dengan membanjiri paket atau permintaan di jaringan. Karena makin berkembangnya sistem jaringan, jumlah pengguna didalamnya pun semakin banyak. Oleh karena itu, sangat sulit untuk mengidentifikasi siapa pengguna legal dan siapa peretas (*hacker*). Dan juga seiring berkembangnya teknologi, teknik untuk membuat serangan Ddos juga semakin

meningkat. Mengidentifikasi serangan DDos menjadi masalah yang lebih kompleks karena ada berbagai jenis strategi serangan DDoS. Beberapa jenis serangan DDoS yaitu *ICMP flood, SYN flood, IP packet flood*, dan lain-lain (Priya, S.S., Yuvaraj, D., & Sivaram, M., 2020). Dalam penelitian ini arus lalu lintas jaringan di gunakan untuk sampel data uji yang akan diklasifikasi. Naïve Bayes adalah salah satu metode yang dapat digunakan untuk melakukan klasifikasi data.

Naive Bayes adalah metode atau algoritma klasifikasi sederhana yang dapat berkontribusi pada keputusan akhir, dengan setiap atribut memiliki properti independen (Riadi, I., Umar, R., & Aini, 2019). Naive Bayes merupakan salah satu metode data mining untuk mengklasifikasikan data. Pengoperasian metode Naïve Bayes menggunakan parameter yang ada. Konsep dasar dari naive bayes adalah teorema bayes. *Bayes Optimal Classifier*, sebuah teorema yang digunakan untuk menghitung peluang dalam statistik, menghitung probabilitas satu kelas dari setiap kelompok atribut yang ada untuk menentukan kelas mana yang terbaik.

Banyak penelitian deteksi DDoS telah menghasilkan berbagai metodologi dan metode yang telah berhasil. Penelitian Budi K. yaitu Seleksi Fitur dengan *Information Gain* untuk Meningkatkan Deteksi Serangan DDoS Menggunakan *Random Forest* mampu mengklasifikasi serangan dengan akurasi deteksi DDoS hingga 99.99% dengan tingkat alarm palsu 0.001 (Budi, K., 2020). Pada penelitian Harsono yaitu Klasifikasi Paket Jaringan Berbasis Analisis Statistik dan *Neural Network* mengklasifikasi serangan DDoS dengan tingkat akurasi mencapai 92,99% (Harsono, 2018). Sedangkan Penelitian Aziz M yaitu Implemetasi Jaringan Saraf Tiruan Untuk Mendeteksi Serangan DDoS Pada Forensik Jaringan. Hasil eksperimen menunjukkan hasil akurasi mencapai 95,23 (Aziz, M., 2019).

2. LANDASAN KEPUSTAKAAN

2.1 Ditributed Denial-of-Service (DDoS)

Serangan Jaringan Terdistribusi atau sering disebut serangan Distributed Denial of Service (DDoS) bekerja dengan cara membanjiri beberapa permintaan ke sumber daya server yang diserang dengan tujuan memenuhi kapasitas server sehingga server tidak mampu menangani banyaknya permintaan dan membuat server tidak bekerja dengan benar. (Kaspersky, 2021).

2.2 Slowloris

Slowloris adalah jenis serangan DDoS pada lapisan aplikasi yang menggunakan permintaan HTTP parsial untuk membuka koneksi antara satu komputer dan server *Web* yang ditargetkan, kemudian menjaga koneksi tersebut tetap terbuka selama mungkin, sehingga membebani dan memperlambat kinerja target (Netscout, 2020).

2.3 TCPDump

TCPdump adalah sebuah aplikasi yang digunakan untuk melakukan capture maupun inspeksi pada lalu lintas jaringan yang masuk dari dan ke sistem Anda. Aplikasi ini sering digunakan administrator jaringan baik untuk mengatasi masalah server maupun pengujian keamanan. Hasil capture lalu lintas jaringan dari TCPdump dapat disimpan dalam file berekstensi .pcap (Linuxid, 2020) .

2.4 CICFlowMeter

CICFlowMeter adalah sebuah tools untuk menganalisa arus lalu lintas jaringan. Tools Ini dapat digunakan untuk menghasilkan aliran dua arah, di mana paket pertama menentukan arah maju (sumber ke tujuan) dan mundur (tujuan ke sumber), karenanya lebih dari 80 fitur lalu lintas jaringan statistik seperti Durasi, Jumlah paket, Jumlah byte, Panjang paket, dan lain-lain dapat dihitung secara terpisah dalam arah maju dan mundur. Tools ini menghasilkan output file berekstensi .csv (UNB, 2017).

2.5 Flask

Flask merupakan program API Python yang mempermudah developer dalam membangun aplikasi web. Flask mempuyai kumpulan modul dan perpustakaan yang membantu pengembang untuk menulis aplikasi tanpa menulis kode tingkat rendah seperti protokol, manajemen thread, dan lain-lain (Geeksforgeeks, 2020).

2.6 *Mutual Information*

Mutual Information (MI) memiliki konsep untung menghitung seberapa banyak informasi yang terkandung dalam term, dan kontribusinya untuk membuat keputusan klasifikasi yang tepat pada suatu kelas. Untuk memperoleh nilai MI dibutuhkan beberapa nilai pendukung yaitu

frekuensi term x pada kelas A, frekuensi

term lain pada kelas A, frekuensi term x di kelas selain A, frekuensi term selain x di kelas selain A, dan jumlah semua term yang ada.

2.7 Naïve Bayes

Pengklasifikasi Naive Bayes adalah pengklasifikasi stokastik sederhana berdasarkan penerapan teorema Bayes, dan setiap fitur diasumsikan kelas bersyarat independen (Murty, M. N., & Devi, 2011).

Pengklasifikasi naive Bayes mengasumsikan bahwa efek dari nilai satu variabel dalam suatu kelas tidak tergantung pada nilai variabel lain. Asumsi ini disebut kelas independensi bersyarat. Ini dirancang untuk membuat komputasi lebih mudah, dan dalam arti itu naif.

Teorema Bayes merupakan teorema yang mengacu konsep probabilitas bersyarat (Tan et al, 2016). Teorema Bayes dapat dinotasikan pada persamaan 2.1 berikut:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (2.1)$$

Penjelasan :

- (B) Merupakan data dengan class/label yang belum di ketahui
- Merupakan Hipotesis data suata kelas spesific
- P(A|B) Merupakan Probabilitas hipotesis berdasarkan kondisi atau probability posteriory
- P(B|A) Merupakan Probablitas berdasarkan Kondisi hipotesis
- P(A) Merupakan probabilitas A

2.8 *Confusion Matrix*

Untuk menguji hasil klasifikasi pada sistem yang dibangun, diperlukan suatu metode perhitungan untuk evaluasi kinerja. Tahap yang di perlukan untuk evaluasi kinerja yaitu dengan menghitung nilai presisi, *recall*, dan nilai-nilai ukuran f1. Akurasi adalah cara sistem mengklasifikasikan benar dengan data benar dan salah, dan ukuran-f1 adalah untuk menilai kinerja keseluruhan sistem dengan menghitung nilai akurasi dan recall. Perhitungannya dapat dilihat pada matriks konfusi (Han, 2011) :

Tabel 2. 1 Confusion Matrix

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

TP (*True Positive)* merupakan prediksi positif dan nilai sebenarnya positif, TN (*True Negative*) merupakan prediksi negatif dan nilai sebenarnya negative, FP (*False Positive*) merupakan prediksi positif dan nilai sebenarnya negatif dan FN (*False Negative*) merupakan prediksi negatif dan nilai sebenarnya positif.

Rumus dari akurasi dapat dilihat pada persamaan 2.2, 2.3, 2.4, 2.5 berikut :

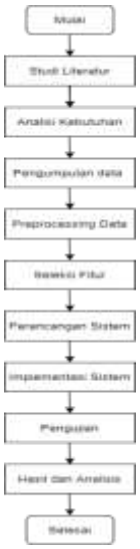
$$Precision = \frac{TP}{TP+FP} \quad (2.2)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+TN}\ x\ 100\% \quad (2.3)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.4)$$

3. METODOLOGI PENELITIAN

Pada bab ini, penulis memaparkan langkah-langkah atau kerangka penelitian ini.



Gambar 3. 1 Kerangka Penelitian

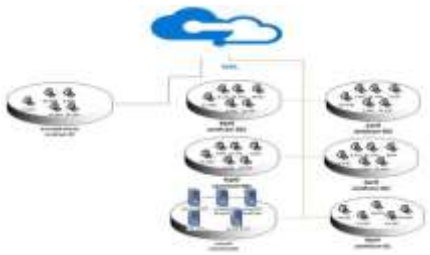
3.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data-data yang diperlukan dalam proses pembuatan sistem. Metode yang dapat dilakukan yaitu melalui studi literatur, yaitu

dengan cara mengumpulkan data-data dan mempelajari refrensi dari berbagai sumber yang berkaitan dengan penelitian yang di lakukan. Data-data yang di perlukan adalah sebagai berikut:

1. Dataset

Dataset yang digunakan dalam penelitian ini yaitu dataset dari CICIDS2018 yang di dapatkan dari kaggle. Dataset CICIDS2018 memiliki 84 Fitur yang di ekstraksi menggunakan tools CICFlowmeter dari 7 skenario serangan yaitu Brute-force, Heartbleed, Botnet, DoS, DDoS, serangan Web, dan infiltrasi jaringan dari dalam. Infrastruktur penyerang mencakup 50 mesin dan organisasi korban memiliki 5 departemen dan mencakup 420 mesin dan 30 server.



Gambar 3. 2 Topologi Jaringan UNB

Sumber: www.unb.ca

2. Data Latih

Dataset yang didapatkan dibagi menjadi dua kategori yaitu DDoS dan Benign. Kemudian dikarenakan dataset dari CICIDS2018 mempunyai data yang sangat banyak, diperlukan dataset *split* dimana pada proses dataset *split* dihasilkan 1000 data yang di bagi menjadi 500 data untuk label DDoS dan 500 Data untuk label Benign untuk di gunakan sebagai data pelatihan. Sebelum data hasil *split* ini digunakan diperlukan proses preprocessing data.

Gambar 3.3 Contoh Data Latih

3. Data Uji

Data uji pada penelitian di peroleh dari proses mengcapture lalu lintas yang berjalan pada server menggunakan tools TCPdump dengan output berekstensi .pcap. Kemudian file .pcap tersebut di ekstraksi untuk mendapatkan fitur menggunakan tools CICFlowmeter dan juga dikonversi menjadi file berekstensi .csv agar dapat di proses *webservice* sistem.

Tabel 3.1 Contoh Data Uji

id	src_ip	src_ port	dst_ port	timestamp	flow_ duration
276038	10311213961	5678	5678	44392,70575	0
276039	10311225321	11145	8291	44392,70576	90
276040	10311213928	5678	5678	44392,70581	0
276041	16724813392	55820	18003	44392,70581	51
276042	10311213911	33412	5678	44392,70586	0
276043	1,3420911	59076	12474	44392,70595	54
276044	6117717316	27542	22	44392,70603	11941145
276045	0	68	67	44392,70603	4727348
276046	0	5678	5678	44392,70612	164
276047	1031121391	5678	5678	44392,70615	0
276048	1031121392	67	68	44392,70633	34399545
276049	10311213941	5678	5678	44392,70639	0
276050	10311213961	5678	5678	44392,70645	0
276051	10311213929	58975	1900	44392,70646	3024704
276052	10311213922	54256	1900	44392,70649	3006554
276053	10311213928	5678	5678	44392,7065	0
276054	2,21181E11	17735	22	44392,70655	1755634
276055	10311213911	33412	5678	44392,70655	0
276056	8924816520	59527	49583	44392,7066	184544
276057	10311213917	55493	1900	44392,70679	3009064
276058	10311213926	137	137	44392,70681	1505535
276059	10311213926	61412	5355	44392,70681	411179
276060	10311213926	54386	5355	44392,70681	411106

3.2 *Preprocessing Data*

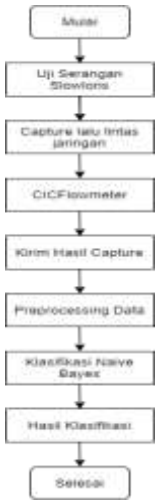
Pada proses ini data latih dan data uji mentah akan dioleh sehingga bisa digunakan dengan sempurna. Pada penelitian proses pengolahan data yang dilakukan yaitu *Data Cleaning* dan *Feature Selection*. *Data cleaning* dilakukan dengan cara menghapus atau memodifikasi data yang salah, duplikat, tidak relevan, dan tidak terformat. Metode yang di

gunakan dalam proses seleksi fitur yaitu *Mutual Information*, yang merupakan salah satu metode seleksi yang menunjukkan seberapa banyak informasi yang ada atau tidaknya sebuah term (istilah) memberikan kontribusi dalam membuat keputusan klasifikasi secara benar atau salah. Mendapatkan fitur yang paling relevan merupakan Tujuan dari pemilihan fitur ini, dimana fitur tersebut akan mempermudah dan mempersingkat metode naïve bayes dalam proses klasifikasi.

3.2 Perancangan Sistem

Pada tahap ini, penulis menggunakan beberapa tahapan perancangan sistem yang disusun dalam subbab berikut guna mencapai hasil yang sesuai dengan rencana penelitian.

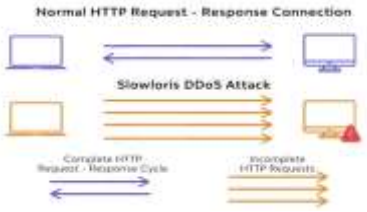
1. Deskripsi Umum Sistem



Gambar 3. 3 Gambaran Umum Sistem

- **Uji Serangan Slowloris**

Uji serangan Slowloris digunakan untuk menciptakan arus lalu lintas yang mengindikasikan serangan DDoS. Slowloris digunakan Karena tidak membutuhkan sumber daya maupun biaya yang banyak namun serangan yang dihasilkan cukup membuat kinerja server melamban. berikut



Gambar 3. 4 Topologi Serangan Slowloris

Sumber: www.wallarm.com

- **Capture Lalu Lintas Jaringan**

Setelah dilakukan skenario serangan Slowloris terhadap server , kemudian dilakukan *capture* lalu lintas jaringan secara *real time* menggunakan tools TCPDump, hasil analisa lalu lintas jaringan tersebut menghasilkan output file berekstensi .pcap.

- **CICFlowmeter**

Setelah capture lalu lintas jaringan sudah mencapai 1000 data maka akan di break selama 5 detik untuk dilakukan tahap CICFlowmeter. Pada tahap ini file *output* dari hasil analisa lalu lintas jaringan akan di konversi ke dalam ekstensi .csv beserta 80 fitur yang diekstraksi menggunakan CICFlowmeter.

- **Kirim Hasil Capture**

Data .csv tersebut kemudian akan di kirimkan ke webservice sistem menggunakan API Flask untuk kemudian dilakukan proses preprocessing data

- **Preprocessing Data**

Pada tahap ini, data yang sudah dikirimkan akan di lakukan proses preprocessing data agar data tersebut dapat digunakan secara sempurna dan dapat menunjang akurasi dari hasil **klasifikasi.**

- **Klasifikasi Naïve Bayes**

Pada tahapan ini, data yang sudah melalui proses preprocessing data akan digunakan sebagai data uji pada proses klasifikasi menggunakan metode naïve bayes dengan data training yang diambil dari dataset CICIDS2018.

- **Hasil Klasifikasi**

Hasil Klasifikasi data uji tersebut akan di tampilkan pada halaman klasifikasi *web service*.

4. HASIL DAN PEMBAHASAN

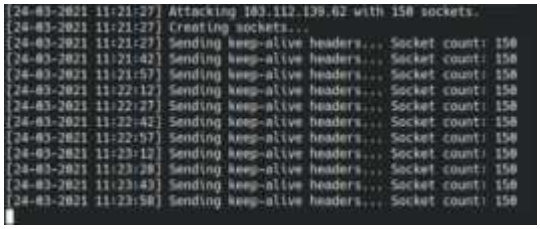
4.1 IMPLEMENTASI

Pada Bab ini menjelaskan tentang penerapan kebutuhan sistem berdasarkan analisis dan perancangan yang telah dibuat. Dalam hal ini implementasi meliputi implementasi Perangkat lunak.

1.1 Implementasi Sistem Server

1. Slowloris

Slowloris melakukan uji coba serangan terhadap server yang digunakan untuk menciptakan arus lalu lintas yang mengindikasikan serangan DDoS.



Gambar 4. 1 Slowloris

2. Sniff

```
From flask import Flask, request, jsonify
import requests
app = Flask(__name__)
@app.route('/api/analisa',
methods=['GET', 'POST']) def
add_message():
 content = request.json URL =
'http://muamarzidane.my.id/api/api'
 response = requests.post(url=URL,
json=content)
 r = response.json()

 sourceIp =
r['results']['results']['data'][0][0]
 hasil = r['results']['results']['hasil']
\
 print("Hasil analisa:")
 if hasil=='ddos':
 cmdCheck = os.popen('iptables -L INPUT -
v -n | grep "'+sourceIp+'"').read()
 if cmdCheck == "":
 myCmdBlock = os.popen('sudo iptables -I
INPUT -s '+sourceIp+' -j DROP').read()
 os.popen('iptables-save') print(hasil+'
- '+sourceIp)
 else:
 print('Benign - '+sourceIp)

 # except:
 # pass

 return jsonify({'status':True})

if __name__ == '__main__':

 app.run(host= '0.0.0.0',debug=True)
```

Pada potongan kode program di atas berfungsi untuk menangkap lalu lintas jaringan yang masuk ke dalam server. Trafik jaringan akan ditangkap secara realtime menggunakan library sniff dan menyimpan hasil tangkapannya ke file berekstensi .pcap. Setiap 5 detik sekali, file .pcap tersebut akan di ubah formatnya menjadi .csv kemudian di kirim ke api route server.py

3. Server

Pada potongan program diatas berfungsi untuk server. Tahapan pertama mengimport modul flask, *request*, dan jsonify. Selanjutnya mendeklarasikan *routing* API dan analisa. Selanjutnya membuat function *add message*. Pada content = request.json mengambil argument dari data sniff.py yang akan dikirimkan pada *function add message*. Selanjutnya data tersebut di kirim ke *webservice* untuk dilakukan klasifikasi. Kemudian sistem akan mengambil data hasil url analsia untuk menetukan ddos atau bukan.

4. *Preprocessing*

Preprocessing data yang dilakukan yaitu *data cleaning* dan seleksi fitur. Dari hasil seleksi fitur menggunakan *mutual information* didapatkan 8 fitur paling relevan berdasarkan score paling tinggi

Tabel 4.1 Fitur

Fitur	Deskripsi
Dst Port	Port penerimaan paket
Fwd Seg Size Min	Ukuran terkecil segmen yang diamati dari arah maju (dari *source* ke *destination*)
Init Bwd Win Byts	Jumlah total byte yang dikirim di jendela awal ke *backward direction* (dari *destination ke source*)
Flow IAT Min	Waktu paling minimum yang dibutuhkan antar kedatangan paket
Src Port	*Port* yang di gunakan pengiriman paket dari *source*
Sub Flow Bwd Byts	Jumlah rata-rata *byte* dalam sub aliran *backward direction* (dari *destination* ke *source*)
Bwd Header Len	Total *byte* yang digunakan untuk *header* pada *backward direction* (dari *destination* ke *source*)
Fwd Header Len	Total *byte* yang digunakan untuk header pada *forward direction* (dari *destination* ke *source*)

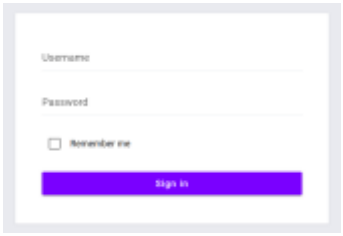
5. Klasifikasi

```
public function doNaiveBayes($data)
 {
$data = json_encode($data);
$data = json_decode($data, true);
$X_train_temp =
$this>M_datatraining>getRandDataTraining(
1000);
$X_train = [];
$y_train = [];
 foreach ($X_train_temp as $trkey =>
$trvalue) {
 $arraySamples = [];
 foreach ($trvalue as $skey => $svalue) {
 if ($skey != 'id' AND $skey != 'kag_id'
AND $skey != 'flow_id' AND $skey !=
'label' AND $skey != 'predict') {
 // if ($skey == 'src_port' OR $skey ==
'dst_port' OR $skey == 'tot_fwd_pkts' OR
$skey == 'flow_duration' ) {
 if ($skey == 'dst_port'
 OR $skey == 'fwd_seg_size_min'
 OR $skey == 'init_bwd_win_byts'
 OR $skey == 'flow_iat_min'
 OR $skey == 'fwd_header_len'
 OR $skey == 'src_port'
 OR $skey == 'subflow_bwd_byts'
 OR $skey == 'bwd_header_len') {
 $arraySamples[] = $svalue;
 }
 }
 }
 $X_train[] = $arraySamples;
 $y_train[] = $trvalue['label'];
 }
 $X_test = [];
 foreach ($data['columns'] as $dtkey =>
$dtvalue) {
 // if ($dtvalue == 'src_port' OR
$dtvalue == 'dst_port' OR $dtvalue ==
'tot_fwd_pkts' OR $dtvalue ==
'flow_duration' ) {
 if ($dtvalue == 'dst_port'
 OR $dtvalue == 'fwd_seg_size_min'
 OR $dtvalue == 'init_bwd_win_byts'
 OR $dtvalue == 'flow_iat_min'
 OR $dtvalue == 'fwd_header_len'
 OR $dtvalue == 'src_port'
 OR $dtvalue == 'subflow_bwd_byts'
 OR $dtvalue == 'bwd_header_len') {
 $X_test[] = $data['data'][0][$dtkey];
 }
 }
 $classifier = new NaiveBayes();

 $classifier->train($X_train, $y_train);

 $predict = $classifier-
>predict($X_test);

 return $predict;

 }

}
```

Pada potongan program diatas menjelaskan proses klasifikasi data uji baru dengan metode naïve bayes menggunakan data training yang sudah di peroleh sebelumnya. Pada proses klasifikasi hanya ini menggunakan 8 fitur yang paling relevan. Kemudian hasil dari klasifikasi akan di simpan ke database server untuk ditampilkan pada halaman klasifikasi.

1.2 Implementasi Tampilan

1. Halaman *Login*



Gambar 4. 2 Tampilan Halaman *Login*

2. Halaman *User*

Gambar 4.3 Tampilan Halaman *User*

3. Halaman Kategori

Gambar 4. 4 Tampilan Halaman Kategori

4. Halaman *Data Training*

Gambar 4. 5 Halaman *Data Training*

5. Halaman Klasifikasi

Gambar 4. 6 Klasifikasi

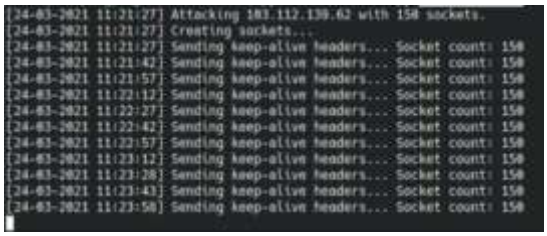
4.2 PENGUJIAN

Skenario pengujian yang dijalankan ditentukan berdasarkan desain yang telah ditetapkan. Dalam skenario ini diharapkan sistem akan berjalan dan menghasilkan hasil seperti yang dirancang.

1. Skenario Uji Coba Serangan

- **Slowloris**

Uji coba serangan dengan Slowloris dengan mengirim *packet* serangan ke server yang digunakan.



Gambar 5. 1 Slowloris

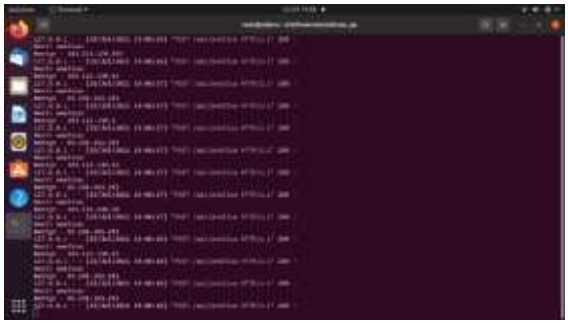
- **Sniff**

Sniff melakukan capture lalu lintas jaringan secara realtime pada eth0 dan hasil dari capture tersebut di simpan dalam ekstensi .pcap. Setelah sudah mencapai 1000 capture maka program sniff.py break selama 5 detik untuk mengubah ekstensi .pcap ke .csv yang kemudian akan dikirim ke server.py melalui Api route.

Gambar 5. 2 Sniff

- **Server**

Server mengambil data yang dikirimkan oleh sniff.py untuk kemudian dikirimkan lagi ke *webservice* untuk dilakukakan klasifikasi berdasarakan hasil *capture* lalu lintas jaringan oleh sniff.py



Gambar 5. 3 Server

2. Skenario Uji Coba Halaman

Pada skenario uji coba Halaman ini seluruh fungsi pada *webservice* berjalan dengan baik sesuai apa yang sudah di rancang.

4.3 EVALUASI PERFORMASI

Tabel 5. 1 Confusion Matrix

Data Aktual	Data Prediksi	
	DDoS	Benign
DDoS	93	5
Benign	5	97

Pada tabel 5.2 dijelaskan pada percobaan ini menggunakan dataset yang telah di*split* secara *random* dengan data pelatihan 80% dan data pengujian 20% di dapatkan hasil klasifikasi yaitu DDoS DDoS sebanyak 93, Benign DDos sebanyak 5, DDos Benign sebanyak 5 dan Benign Benign sebanyak 97. Kemudian dari hasil tersebut dapat dihitung nilai presisi , *recall* dan akurasi dari metode klasifikasi Naïve Bayes.

Kelas DDoS:

$$Precision = \frac{93}{(93+5)} = 0.948$$

$$Recall = \frac{93}{(93+5)} = 0.948$$

Kelas Benign:

$$Precision = \frac{97}{(97+5)} = 0.95$$

$$Recall = \frac{97}{(97+5)} = 0.95$$

Akurasi Total:

$$Akurasi = \frac{93+97}{(93+5+97+5)} * 100\% = 95\%$$

4.4 PEMBAHASAN

Berdasarkan hasil pengujian sistem yang sudah dibuat berjalan dengan baik dan seluruh fitur yang ada sudah dapat digunakan dengan

baik. Selanjutnya uji coba serangan dengan Slowloris juga dapat diterima oleh sistem dan pada hasil evaluasi performasi menggunakan *confusion matrix* didapatkan tingkat akurasi dari metode klasifikasi *Naïve bayes* dengan akurasi sebesar 95%

5. KESIMPULAN SARAN

Bab ini memuat kesimpulan dan saran terhadap penelitian yang telah dilakukan. Kesimpulan dan saran disajikan secara terpisah dengan penjelasan sebagai berikut:

Kesimpulan

Berdasarkan hasil implementasi dan uji coba yang telah dilakukan, maka dapat disimpulkan hasil penelitian ini adalah sebagai berikut :

1. Penelitian ini telah menghasilkan sistem klasifikasi serangan DDoS dengan dua kategori yaitu DDoS dan Benign .
2. Hasil penelitian menunjukkan nilai akurasi sebesar 95% menggunakan metode naïve bayes sehingga sistem dinilai cukup baik untuk melakukan klasifikasi serangan DDOS.

Saran

1. Sistem ini dapat dikembangkan menjadi aplikasi seluler, yang memungkinkan admin untuk mengklasifikasikan serangan DDoS di perangkat smartphone mereka.

2. Untuk penelitian selanjutnya, diharapkan penelitian ini dapat dimanfaatkan sebagai sistem identifikasi awal dalam pengembangan alat pendektesian serangan DDoS

6. DAFTAR PUSTAKA

Priya, S.S., Yuvaraj., D., dan Sivaram, M., 2016. *A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks*, 20(4). Tersedia melalui:IEEE<https://ieeexplore.ieee.org/document/7381613> [Diakses 20 Februari 2021]

Harsono, H., Khambali, M., & Muhammad, A. W. 2018. Klasifikasi Paket Jaringan Berbasis Analisis Statistik dan *Neural Network*. Jurnal Informatika: Jurnal Pengembangan IT, 3(1), 67-70.

Kurniabudi, K., Harris, A., & Rahim, A. 2020. Seleksi Fitur dengan *Information Gain* untuk Meningkatkan Deteksi Serangan DDoS Menggunakan *Random Forest*. Techno. Com, 19(1), 56-66.

Aziz, M., Umar, R., & Ridho, F. 2019. Implemetasi Jaringan Saraf Tiruan Untuk Mendeteksi Serangan DDoS Pada Forensik Jaringan. Query: *Journal of Information Systems*, 3(1).

Vadehra, R., Chowdhary, N., & Malhotra, J. 2015. *Impact evaluation of distributed denial of service attacks using ns2. International Journal of Security and Its Applications*, 9(8), 303-316.

Riadi, I., Umar, R., & Aini, F. D. 2019. Analisis Perbandingan *Detection Traffic Anomaly* Dengan Metode Naive Bayes Dan *Support Vector Machine* (SVM). ILKOM Jurnal Ilmiah, 11(1), 17-24.

Soofi, A. A., & Awan, A. 2017. *Classification techniques in machine learning: applications and issues. Journal of Basic and Applied Sciences*, 13, 459-465.