Seleksi Fitur dengan *Information Gain* untuk Meningkatkan Deteksi Serangan DDoS Menggunakan *Random Forest*

An Information Gain Feature Selection to Improve DDoS Detection using Random Forest

Kurniabudi[1], Abdul Harris[2], Abdul Rahim[3]
[1]Sistem Komputer, STIKOM Dinamika Bangsa
[2,3]Teknik Informatika, STIKOM Dinamika Bangsa
E-mail: [1]kbudiz@yahoo.com, [2]abdulharris@stikom-db.ac.id, [3]abdulrahim@stikom-db.ac.id

Abstrak

Tantangan deteksi serangan saat ini adalah jumlah trafik yang besar dan beragam serta hadir jenis serangan baru. Disisi lain, pesatnya pertumbuhan teknologi layanan komunikasi, menghasilkan trafik dengan informasi yang beragam. Sehingga diperlukan teknik baru untuk meningkatkan performa deteksi. DDoS merupakan salah satu serangan yang sering terjadi didunia maya. Serangan DDoS susah dibedakan dari trafik normal, selain itu, dalam deteksi serangan tidak semua fitur dianalisa. Terlalu banyak fitur yang tidak relevan akan menghasilkan kategori kelas yang tidak berhubungan dan membebani waktu komputasi. Oleh karena itu diperlukan seleksi fitur, selain itu seleksi fitur dapat meningkatkan akurasi. Penelitian ini bertujuan meningkatkan performa *Random Forest* dalam mendeteksi serangan DDoS dengan seleksi fitur menggunakan teknik *Information Gain.* Berdasarkan hasil eksperimen diperoleh bahwa teknik yang diusulkan mampu meningkatkan akurasi deteksi DDoS hingga 99.99% dengan tingkat alarm palsu 0.001.

Kata kunci: Seleksi fitur, DDoS, *Information Gain*, *Random Forest*, CICIDS-2017

Abstract

The challenge of attack detection now is the large and varied amount of traffic and new types of attacks. In other hand, the rapid growingly of communication services technology, has generated traffic with diverse information. So, the new techniques are needed to improve detection performance. DDoS is one of the most attacks in cyber. Its difficult to differentiate DDoS from norma traffic, in other hand, in attack detection not all feature used for analyzed. Too many irrelevant feature will contribute to unrelated class category and computation overhead. Therefore, feature selection is needed, because can improve accuracy. Feature selection is the one of the method has been used to improve classification algorithm accuracy. This study aims to improve the performance of Random Forest in detecting DDoS attacks by feature selection using Information Gain techniques. Based on the experimental results it was found that the proposed technique is able to increase the accuracy of DDoS detection up to 99.99% with a false alarm rate of 0.001.

Keywords: feature selection, DDoS, Information Gain, Random Forest, CICIDS-2017

1. PENDAHULUAN

Meningkatnya ketergantungan masyarakat terhadap penggunaan sistem komputer pada berbagai aspek seperti keuangan, industri, kesehatan dan lain-lain menjadikan keamanan dunia maya menjadi penting, salah satunya yaitu deteksi intrusi atau serangan[1]. Tantangan deteksi serangan saat ini adalah jumlah trafik yang besar dan beragam serta hadir jenis serangan baru.

Salah satu bentuk serangan yang popular adalah DDoS. Seperti dirilis dalam situs cisco.com, *DDoS* (*Distributed Denial of Service*) merupakan salah satu jenis serangan yang sering terjadi pada dunia maya di tahun 2018. Begitupula pada situs techrepublic.com disebutkan *DDoS* merupakan salah satu dari 6 (enam) jenis serangan di dunia maya yang paling popular di tahun 2018. Pada serangan *DDoS*, penyerang menciptakan jumlah permintaan yang sangat besar ke komputer korban dengan tujuan menolak permintaan normal atau menurunkan kualitas layanan[2]. Serangan *DDoS* berdampak pada kerugian yang sangat besar, seperti berkurangnya pendapatan, kegagalan produksi, buruknya reputasi, pencurian dan lain-lain, hal inilah yang mendorong perlunya teknik pendeteksian dan pencegahan yang baik.[3].

Berbagai penelitian deteksi serangan DDoS telah berhasil mengembangkan berbagai teknik dan metode. Penelitian [4] mengusulkan deteksi serangan DDoS menggunakan teknik entropy, dengan membandingkan entropi alamat IP sumber dan entropi alamat IP tujuan dan berhasil mendeteksi DDoS secara efisien. Penelitian [5] yang mengusulkan model *time series ARIMA* dan *chaotic system,* model mampu mengklasifikasikan serangan hingga 99,5%. Pada penelitian [6] ANN (*Artificial Neural Network)* diaplikasikan untuk mendeteksi DDoS dengan karakterisitik fitur khusus, solusi yang ditawarkan memiliki akurasi hingga 98%. Sedangkan [7] merancang sistem deteksi DDoS pada lingkungan *cloud computing* dengan algorithma C.4.5. hasil eksperimen memperlihatkan hasil yang akurat dibandingan algorithma pembelajaran mesin yang lain.

Meskipun dari penelitian-penelitian sebelumnya telah dihasilkan berbagai metode dan teknik baru untuk mendeteksi serangan, salah satunya serangan DDoS. Namun, seiring meningkatnya perkembangan sistem keamanan jaringan maya, muncul berbagai bentuk baru serangan yang dimodifikasi dari serangan yang ada. Disisi lain dengan berkembangnya berbagai jenis layanan teknologi komunikasi, menghasilkan trafik jaringan yang sangat beragam dan berukuran besar, hal ini menjadi tantangan dalam deteksi serangan dikarenakan semakin banyak informasi (fitur) yang akan diolah. Pada dasarnya tidak semua fitur yang terdapat pada trafik memiliki pengaruh terhadap algorithma deteksi. Namun, diperlukan pengetahuan untuk menentukan fitur yang tepat dan relevan untuk mendeteksi serangan seperti DDoS. Dikarenakan sangat sulit membedakan DDoS dari trafik normal[8]. Selain itu, terlalu banyak fitur yang tidak relevan akan menghasilkan kategori kelas yang tidak berhubungan[9]. Mengacu pada penelitian [10]&[11]disebutkan bahwa seleksi fitur dapat meningkatkan akurasi algorithma klasifikasi. Oleh karenanya penelitian ini bertujuan meningkatkan performa deteksi serangan DDoS dengan menggunakan teknik seleksi fitur. Pada penelitian ini *Random Forest* digunakan sebagai metoda klasifikasi yang akan ditingkatkan performanya, sedangkan teknik seleksi fitur menggunakan *Information Gain*. Penelitian ini juga menguji teknik seleksi fitur *Information Gain* dalam menghasilkan fitur yang relevan untuk mendeteksi serangan DDoS dan melakukan komparasi fitur hasil seleksi terhadap performa algorithma klasifikasi *Naive Bayes, Bayes Network, OneR, Adaboost, RandomTree* dan *Random Forest* dalam mendeteksi serangan DDoS.

2. METODE PENELITIAN

Pada bagian ini peneliti memaparkan kerangka penelitian, metoda dan dataset yang digunakan pada eksperimen penelitian.

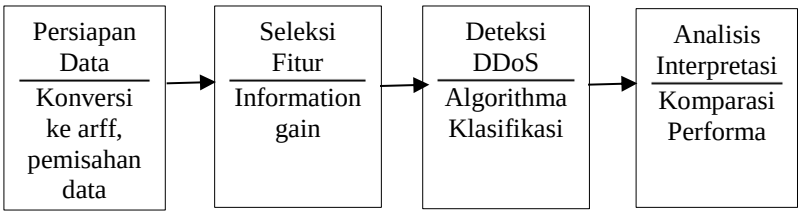
2.1 Kerangka Penelitian

Pada penelitian ini penulis menggunakan beberapa tahapan kegiatan, yang disusun dalam kerangka penelitian pada gambar 1. Berdasarkan gambar 1, dapat dijelaskan tahapan-tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Persiapan data, pada tahapan ini penulis melakukan persiapan data diantaranya, mengubah format data dari csv ke arff. Menghilangkan fitur/atribut yang redundan. Kemudian melakukan pemisahan data sebagai sampel. Dimana pada penelitian ini sebagai sampel digunakan 20% dari dataset.
2. Seleksi fitur, teknik seleksi fitur yang digunakan pada penelitian ini adalah

Information Gain yang merupakan teknik *filtered-based.* Tujuan dari seleksi fitur ini adalah untuk mendapatkan fitur yang paling relevan untuk digunakan dalam deteksi serangan DDoS.

3. *Training*, dataset hasil seleksi fitur, di *training* dengan menggunakan beberapa metode klasifikasi. Tujuannya adalah untuk melihat perbanding performa metode klasifikasi dalam mendeteksi serangan DDoS. Metode yang digunakan dalam *training* adalah : *Naïve Bayes*, *Bayes Network, OneR, AdaBoost*, *Random Tree* dan *Random Forest.* Untuk proses *training* peneliti menggunakan 3 (tiga) mode pengujian yaitu : *use training set, 5-fold cross validation* dan *10-fold cross validation.*
4. Analisis dan Interpretasi, berdasarkan hasil eksperimen kemudian data dianalisis dan diinterprestasi. Dalam penelitian ini, performa metode klasifikasi diukur dengan tingkat *Acuracy, TPR, FPR, Precision, Recall, Time to Building Model* dan *Process Time*



Gambar 1 Tahapan Penelitian

2.2 Information Gain

Teknik seleksi fitur secara luas telah digunakan hampir dibanyak bidang seperti kategorisasi teks, analisi *genomic*, deteksi intrusi, *bioinformatic* dan lain-lain. Salah satu teknik seleksi fitur adalah *information gain. Information Gain* menggunakan pendekatan penyaringan (*filtered-based)*[7]. *Information Gain (IG)* menghitung entropi setiap fitur. Makin tinggi entropi, makin banyak informasi yang dimiliki fitur tersebut. Fitur ini akan dipilih untuk digunakan pada algorithma klasifikasi, untuk membedakan data yang tidak dikenal menjadi kelas serangan[8]. Berdasarkan [3] untuk menghitung *information gain* dilakukan dengan langkah:

- Panggil dataset
- Hitung entropy dataset dengan persamaan 1

$$I(parent) = -\sum_{i=1}^{k} P(h_i).log_2 P(h_i) \quad (1)$$

Dimana $P(h_i)$ merupakan probabilitas dari h_i

- Pisahkan set menjadi subset menggunakan atribut/ fitur yang memiliki entropi minimum.
- Buat simpul *decision tree* dari atribut tersebut
- Hitung entropi dari simpul *child, I(child₁), I(child₂)* menggunakan persamaan (1).
- Hitung entropi atribut/fitur dengan persamaan (2).

$$I(attribute_i) = \sum_{v \in Values(A)} \frac{|child_v|}{|attribute_i|} I(child_v) \quad (2)$$

- Hitung *information gain* menggunakan persamaan (3)

$$G(attribute_i) = I(parent) - I(attribute_i) \quad (3)$$

- Proses berulang untuk semua atribut

2.3 Random Forest

Random Forest termasuk algorithma klasifikasi berbasis *bagging essemble* yang popular. *Random Forest* terdiri atas banyak pohon keputusan (*decission tree).* Luaran dari *random forest* merupakan hasil *voting* dari masing-masing pohon (*tree)*[14]. Seandainya setiap *classifier* dalam *ensemble* merupakan *decision tree classifier*, maka kumpulan *classifier* adalah sebuah "*forest"*[15]. Dalam [16], *random forest* didefenisikan dengan sebuah *forest* terdiri atas *K tree* $\{T_1,...,T_K\}$, sebuah vektor random θ_k dibentuk untuk *tree* k_{th}, $k=1,...,K$. Dimana θ_k merupakan vektor random untuk pemodelan *tree.* Vektor ini didefenisikan sebagai konstruksi *tree.* Misalnya, dalam pemilihan acak, vektor-vektor ini terdiri dari bilangan bulat acak yang dipilih secara acak dari {1,...,*N*} dimana *N* adalah jumlah pemisahan. *Random Forest* memiliki kelebihan yaitu menghasilkan *error* yang rendah, menghasilkan klasifikasi yang baik, dan dapat mengatasi data training yang sangat besar serta efektif untuk mengestimasi *missing data*[14].

Beberapa penelitian tentang deteksi serangan atau intrusi yang menggunakan metoda *Random Forest*, [17] yang membangun model hybrid untuk sistem deteksi intrusi dengan menggabungkan *Rough Set Theory (RST)* dan *Random Forest.* Model yang diusul menunjukkan kinerja yang lebih baik dibandingkan penelitian-penelitian sebelumnya. Sedangkan , dalam penelitiannya menggabungkan *Gain Ration (GR)* sebagai teknik seleksi fitur dan *Random Forest* sebagai *classifier.* Dengan menggunakan NSL-KDD dataset hasil pengujian dengan 30 atribut menghasilkan *Detection Rate (DR)* sebesar 99.73% dan dengan 25 atribut menghasilkan DR sebesar 99.745%[18]. Dalam beberapa penelitian deteksi serangan algorithma *Random Forest* menunjukkan kinerja yang baik, namun masih perlu ditingkatkan. Selain itu, algorithma ini belum diuji untuk data yang berdimensi besar, dengan jumlah fitur yang lebih banyak.

2.4 Pengukuran Performa

Untuk pengukuran performa pada penelitian ini penulis menggunakan *confusion matrix* seperti disajikan pada tabel 1.

Tabel 1 *Binary Confusion Matrix*

		Prediksi	
		Normal	**Serangan**
Aktual	**Normal**	TP	FP
	Serangan	FN	TN

Dalam konteks IDS tabel 1 dapat dijelaskan sebagai berikut :

- FP (*False Positive*) : didefenisikan sebagai jumlah aktual normal yang terdeteksi sebagai serangan
- FN (*False Negative*) : didefenisikans sebagai kesalahan prediksi, dimana aktual serangan dideteksi sebagai normal
- TP (*True Positive*) : didefenisikan sebagai ketepatan prediksi, aktual normal terdeteksi sebagai normal
- TN (*True Negative*) : didefenisikan sebagai aktual serangan terdeteksi sebagai serangan.

Selanjutnya berdasarkan *confusion matrix*, peforma algorithma klasifikasi diukur berdasarkan:

a. *Accuracy:* didefinisikan sebagai tingkat kedekatan antara nilai pengkategorian dengan nilai aktual

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (4)$$

b. *True Positive Rate* : yang didefenisikan sebagai aktual positif dikategorikan sebagai kelas positif.

$$True\ Positive\ Rate = \frac{TP}{FN+TP} \quad (5)$$

c. *False Positive Rate*: yang didefenisikan sebagai aktual negatif dikategorikan sebagai kelas positif

$$False\ Positive\ Rate = \frac{\text{FP}}{\text{FP+TN}} \quad (6)$$

d. *Recall :* yang didefinisikan sebagai aktual positif dikategorikan dengan benar sebagai kelas positif.

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \quad (7)$$

e. *Precission :* didefenisikan sebagai ukuran estimasi probabilitas prediksi positif yang benar

$$Precission = \frac{\text{TP}}{\text{TP+FP}} \quad (8)$$

Sedangkan untuk pengukuran *Time Building Model* dihitung, saat system membangun model training oleh aplikasi weka. Untuk *Proses Time* dihitung saat proses *training* dimulai hingga selesai.

2.5 Dataset

Penelitian ini menggunakan dataset CICIDS-2017 dari ISCX. Pada dataset CICIDS-2017 terdapat trafik Normal dan Serangan. Pemilihan dataset ini, dikarenakan dataset ini mewakili beberapa hal diantaranya : kompleksitas, heterogenitas, interaksi lengkap dan lain-lain[19]. Untuk trafik serangan terdiri beberapa jenis serangan secara umum yaitu : *Bot, Brute-Force, DoS/DDoS, Infiltration, PortScan* dan *Web-Attack.* Dalam penelitian ini penulis hanya menggunakan 20% data dari dataset. Pada penelitian ini dataset yang digunakan telah dilakukan seleksi, yaitu hanya berisi trafik Normal dan serangan Dos/DDoS, detail data disajikan pada tabel 2. Menurut [20], pada dataset CICIDS-2017, untuk trafik serangan DDoS terdiri atas beberapa jenis yaitu : *DDoS, DoS GoldenEye, DoS Hulk, DoS Slow-httptest, DoS Slowloris* dan *Hearthbleed*. Setelah melalui proses persiapan data, terdapat 78 fitur pada dataset yang akan digunakan pada eksperimen.

Tabel 2 Profil Data Penelitian

Jenis Data	**Jumlah**
Normal	454306
Dos/DDoS	76445
Total	**530751**

2.6 Lingkungan Eksperimen

Pada penelitian ini, untuk mendukung eksperimen seleksi fitur dan deteksi DDoS dengan algorithma klasifika digunakan Notebook dengan *Processor Core i7-7500U 2,7 GHz(4 CPU)* dan *8 GB RAM.* Sistem operasi yang digunakan Windows 10 Pro 64-bit. Alat bantu analis menggunakan weka 3.9. Weka merupakan perangkat lunak *datamining* dan *machine learning. Open-source* yang pertama kali diimplementasikan di *University of Waikato, New Zealand* pada tahun 1997 [21]. Weka memiliki sejumlah besar skema pembelajaran untuk klasifikasi, klasterisasi dan prediksi regresi numerik[22].

3. HASIL DAN PEMBAHASAN

Pada bagian ini peneliti memaparkan hasil eksperimen seleksi fitur dan hasil *training* beberapa algorithma klasifikasi untuk mendeteksi serangan DDoS. Pada bagian ini juga disampaikan analisis terhadap komparasi performa klasifikasi.

3.1 Hasil Seleksi Fitur

Seperti dijelaskan diawal, dalam eksperimen ini penulis menggunakan teknik *Information Gain* sebagai teknik seleksi fitur. Melalui teknik ini diharapkan akan didapatkan fitur yang relevan untuk digunakan dalam mendeteksi serangan DDoS. Eksperimen seleksi fitur dilakukan menggunakan alat bantu Weka. Proses seleksi fitur menggunakan weka sebagai berikut:

```
Started weka.attributeSelection.InfoGainAttributeEval
Command: weka.attributeSelection.InfoGainAttributeEval -s "weka.attributeSelection.Ranker -T -
1.7976931348623157E308 -N -1"
Filter command: weka.filters.supervised.attribute.AttributeSelection -E
"weka.attributeSelection.InfoGainAttributeEval " -S "weka.attributeSelection.Ranker -T -
1.7976931348623157E308 -N -1"
Meta-classifier command: weka.classifiers.meta.AttributeSelectedClassifier -E
"weka.attributeSelection.InfoGainAttributeEval " -S "weka.attributeSelection.Ranker -T -
1.7976931348623157E308 -N -1" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Finished weka.attributeSelection.InfoGainAttributeEval weka.attributeSelection.Ranker
```

Selanjutnya dari proses seleksi fitur menggunakan *Information Gain* di dapatkan hasil seperti disajikan pada tabel 3. *Information Gain* menghasilkan perangkingan fitur berdasarkan nilai bobot. Dalam penelitian ini penulis menetapkan fitur dengan bobot > 0.3 yang akan digunakan sebagai fitur terpilih serperti disajikan pada tabel 4. Berdasarkan hasil eksperimen, dari 78 fitur, dengan menggunakan teknik Information Gain dan penetapan bobot minimal maka didapatkan 19 fitur terpilih. Selanjutnya fitur-fitur ini digunakan untuk mendeteksi serangan DDoS.

Tabel 3 Rangking Fitur berdasarkan *Information Gain*

Rank.#	Bobot	ID	Nama Fitur	Rank.#	Bobot	ID	Nama Fitur
1	0.4034879	65	Subflow Bwd Bytes	40	0.1898138	74	Idle Mean
2	0.4034879	13	Total Length of Bwd Packets	41	0.1880105	77	Idle Min
3	0.3742486	54	Avg Bwd Segment Size	42	0.1738925	62	Subflow Fwd Packets
4	0.3742486	20	Bwd Packet Length Mean	43	0.1738925	10	Total Fwd Packets
5	0.3717176	18	Bwd Packet Length Max	44	0.1607988	31	Bwd IAT Min
6	0.3708806	66	Init_Win_bytes_forward	45	0.1498899	6	Bwd IAT Std
7	0.3708107	8	Destination Port	46	0.145804	2	Flow Bytes/s
8	0.3517825	52	Average Packet Size	47	0.1428483	19	Bwd Packet Length Min
9	0.3499329	39	Max Packet Length	48	0.1341145	3	Flow Packets/s
10	0.3381345	41	Packet Length Std	49	0.1275302	37	Fwd Packets/s
11	0.3356896	63	Subflow Fwd Bytes	50	0.1266325	15	Fwd Packet Length Min
12	0.3356896	12	Total Length of Fwd Packets	51	0.1262031	38	Min Packet Length
13	0.3314553	14	Fwd Packet Length Max	52	0.1116554	27	Fwd IAT Min
14	0.3289957	1	Bwd Packet Length Std	53	0.1099881	7	Bwd Packets/s
15	0.3221081	67	Init_Win_bytes_backward	54	0.0868865	23	Flow IAT Min
16	0.3220426	26	Fwd IAT Max	55	0.084843	68	act_data_pkt_fwd
17	0.3142141	40	Packet Length Mean	56	0.0508056	51	Down/Up Ratio
18	0.3021613	24	Fwd IAT Total	57	0.0404799	47	ACK Flag Count
19	0.3004814	22	Flow IAT Max	58	0.0311414	43	FIN Flag Count
20	0.2958327	42	Packet Length Variance	59	0.0299443	75	Idle Std
21	0.2855234	36	Bwd Header Length	60	0.0205767	48	URG Flag Count

22	0.2736278	9	Flow Duration	61	0.018671	71	Active Std
23	0.2735282	25	Fwd IAT Mean	62	0.0075712	69	min_seg_size_forward
24	0.2486764	17	Fwd Packet Length Std	63	0.0063766	32	Fwd PSH Flags
25	0.243348	64	Subflow Bwd Packets	64	0.0063766	44	SYN Flag Count
26	0.243348	11	Total Backward Packets	65	0.0003598	46	PSH Flag Count
27	0.2361221	55	Fwd Header Length	66	0.0000567	50	ECE Flag Count
28	0.2346869	5	Fwd IAT Std	67	0.0000567	45	RST Flag Count
29	0.2327158	30	Bwd IAT Max	68	0.0000224	34	Fwd URG Flags
30	0.225487	70	Active Mean	69	0.0000224	49	CWE Flag Count
31	0.2249796	21	Flow IAT Mean	70	0	35	Bwd URG Flags
32	0.223046	73	Active Min	71	0	56	Fwd Avg Bytes/Bulk
33	0.2206013	72	Active Max	72	0	59	Bwd Avg Bytes/Bulk
34	0.2149794	28	Bwd IAT Total	73	0	33	Bwd PSH Flags
35	0.2149494	16	Fwd Packet Length Mean	74	0	61	Bwd Avg Bulk Rate
36	0.2149494	53	Avg Fwd Segment Size	75	0	60	Bwd Avg Packets/Bulk
37	0.2014082	29	Bwd IAT Mean	76	0	57	Fwd Avg Packets/Bulk
38	0.1984862	4	Flow IAT Std	77	0	58	Fwd Avg Bulk Rate
39	0.197735	76	Idle Max				

Tabel 4 Fitur-fitur hasil Seleksi

Rank.#	Bobot	ID	Nama Fitur	Rank.#	Bobot	ID	Nama Fitur
1	0,403488	65	Subflow Bwd Bytes	11	0,335690	63	Subflow Fwd Bytes
2	0,403488	13	Total Length of Bwd Packets	12	0,335690	12	Total Length of Fwd Packets
3	0,374249	54	Avg Bwd Segment Size	13	0,331455	14	Fwd Packet Length Max
4	0,374249	20	Bwd Packet Length Mean	14	0,328996	1	Bwd Packet Length Std
5	0,371718	18	Bwd Packet Length Max	15	0,322108	67	Init_Win_bytes_backward
6	0,370881	66	Init_Win_bytes_forward	16	0,322043	26	Fwd IAT Max
7	0,370811	8	Destination Port	17	0,314214	40	Packet Length Mean
8	0,351783	52	Average Packet Size	18	0,302161	24	Fwd IAT Total
9	0,349933	39	Max Packet Length	19	0,300481	22	Flow IAT Max
10	0,338135	41	Packet Length Std				

3.2 Performa Klasifikasi

Untuk menguji performa klasifikasi dilakukan eksperimen deteksi DDoS menggunakan fitur hasil seleksi yang telah dibahas pada bagian sebelumnya. Untuk performa deteksi pada penelitian menggunakan *Accuracy*, *True Positive Rate*, *False Positive Rate, Recall dan Precission* secara konsep menggunakan persamaan (4)-(8). Pada eksperimen ini untuk menguji pengaruh fitur hasil seleksi terhadap algorithma klasifikasi *Random Tree* dan dikomparasi dengan *Naïve bayes*, *Bayes Network*, *OneR*, *Adaboost* dan *Random Tree* penulis menggunakan weka. Detail proses *training* algorithma klasifikasi menggunakan weka disajikan sebagai berikut :

a. Proses *training* naive bayes

Started weka.classifiers.bayes.NaiveBayes
Command: weka.classifiers.bayes.NaiveBayes
Finished weka.classifiers.bayes.NaiveBayes

b. Proses *training* Bayes nework

Started weka.classifiers.bayes.BayesNet
Command: weka.classifiers.bayes.BayesNet -D -Q
weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E
weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A
Finished weka.classifiers.bayes.BayesNet

c. Proses *training* OneR

Started weka.classifiers.rules.OneR
Command: weka.classifiers.rules.OneR -B 6
Finished weka.classifiers.rules.OneR

d. Proses *training* Adaboost

Started weka.classifiers.meta.AdaBoostM1
Command: weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W
weka.classifiers.trees.DecisionStump
Finished weka.classifiers.meta.AdaBoostM1

e. Proses *training* Random Tree

Started weka.classifiers.trees.RandomTree
Command: weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1
Finished weka.classifiers.trees.RandomTree

f. Proses *training* Random Forest

Started weka.classifiers.trees.RandomForest
Command: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Finished weka.classifiers.trees.RandomForest

Dalam eksperimen digunakan 3 (tiga) jenis pengujian yaitu, 1)*use training set, 20 5-Fold Cross Validation* dan 3) *10-Fold Cross Validation.* Berdasarkan hasil ekperimen, dari sisi akurasi secara umum *Random Forest* memiliki tingkat akurasi yang lebih tinggi dibandingkan *Naïve bayes, Bayes Network, OneR, Adaboost* dan *Random Tree*. Dari 3 (tiga) proses pengujian yang digunakan pada eksperimen, hasil yang di sajikan pada tabel 5 memperlihatkan *Random Forest* memiliki akurasi yang lebih tinggi.

Tabel 5 Akurasi

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	94.14	99.61	94.89	96.89	99.99	99.99
5-Fold Cross Validation	94.44	99.59	94.89	96.89	99.92	99.95
10-Fold Cross Validation	94.35	99.59	94.89	96.89	99.92	99.95

Pada tabel 6 disajikan komparasi *True Positive Rate* untuk masing-masing algorithma klasifikasi dengan 3 (tiga) proses pengujian. Hasil memperlihatkan *Random Tree dan Random Forest* memiliki nilai TPR yang sama jika dibandingkan *Naïve bayes, Bayes network, OneR* dan *Adaboost.* Meskipun pada proses *training* dengan *use training set* dan *5-fold cross validation* memperoleh nilai yang sama dengan *Random Tree.*

Tabel 6 *True Positive Rate*

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	0.996	0.941	0.949	0.969	1.000	1.000
5-Fold Cross Validation	0.996	0.944	0.949	0.969	0.999	0.999
10-Fold Cross Validation	0.996	0.943	0.949	0.969	0.999	1.000

Sedangkan untuk nilai *False Positive Rate* yang disajikan pada tabel 7, memperlihatkan *Random Forest* memiliki nilai FPR lebih rendah dibandingkan *Naïve bayes, Bayes network, OneR, Adaboost* dan *Random Tree*.

Tabel 7 *False Positive Rate*

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	0.006	0.012	0.303	0.045	0.000	0.000
5-Fold Cross Validation	0.006	0.012	0.303	0.045	0.002	0.001
10-Fold Cross Validation	0.006	0.012	0.303	0.045	0.002	0.001

Untuk nilai *precision* yang disajikan pada tabel 8, memperlihatkan pada pengujian menggunakan *user training set* dan *5-fold cross validation, Random Forest* dan *Random Tree* memiliki nilai precission yang sama dan lebih tinggi dibandingkan *Naïve bayes, Bayes network, OneR* dan *AdaBoost.* Sedangkan untuk pengujian menggunakan *10-fold cross validation, Random Forest* memiliki nilai *precision* yang paling tinggi dibandingkan *Naïve bayes, Bayes network, OneR, AdaBoost* dan *Random Tree.*

Tabel 8 *Precision*

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	0.958	0.996	0.952	0.971	1.000	1.000
5-Fold Cross Validation	0.960	0.996	0.952	0.971	0.999	0.999
10-Fold Cross Validation	0.959	0.996	0.952	0.971	0.999	1.000

Untuk nilai *recall* pada tabel 9, memperlihatkan pada pengujian menggunakan *user training set* dan *5-fold cross validation, Random Forest* dan *Random Tree* memiliki nilai *recall* yang sama dan lebih tinggi dibandingkan *Naïve bayes, Bayes network, OneR* dan *AdaBoost.* Sedangkan untuk pengujian dengan *10-fold cross validation, Random Forest* memiliki nilai *recall* yang tertinggi jika dibandingkan dibandingkan *Naïve bayes, Bayes network, OneR, AdaBoost* dan *Random Tree.*

Tabel 9 *Recall*

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	0.941	0.996	0.949	0.969	1.000	1.000
5-Fold Cross Validation	0.944	0.996	0.949	0.969	0.999	0.999
10-Fold Cross Validation	0.943	0.996	0.949	0.969	0.999	1.000

Selanjutnya untuk waktu yang diperlukan membangun model disajikan pada tabel 10. Data pada tabel 10 memperlihatkan bahwa jika dibandingkan dengan *Naïve bayes, Bayes network, OneR, Adaboost* dan *Random Tree, Random Forest* memiliki waktu yang lebih lama.

Tabel 10 *Time Building Model(s)*

Mode Pengujian	Naïve Bayes	Bayes Net	OneR	AdaBoost	Random Tree	Random Forest
Use training Set	1.84	21.56	2.38	38.79	2.96	295.09
5-Fold Cross Validation	1.68	20.63	3.30	39.15	2.74	291.79
10-Fold Cross Validation	1.61	20.64	3.33	46.20	2.73	290.28

Begitupula halnya dengan waktu proses yang disajikan pada tabel 11. Data pada tabel 11 memperlihatkan *Random Forest* membutuhkan waktu proses yang lebih lama jika dibandingkan dengan *Naïve bayes, Bayes network, OneR, Adaboost* dan *Random Tree.*

Berdasarkan data pada tabel 10 dan 11, dapat disimpulkan, meskipun dari sisi akurasi, *TPR, FPR, precision* dan *recall*, *Random Forest* unggul dibandingkan algorithma lain, namun *Random Forest* memiliki waktu yang lebih lama untuk membangun model dan proses. Hal ini tentunya menjadi tantangan bagi peneliti untuk menghasilkan metode deteksi yang akurat namun memiliki waktu komputasi yang rendah.

Tabel 11 *Process Time(s)*

Mode Pengujian	**Naïve Bayes**	**Bayes Net**	**OneR**	**AdaBoost**	**Random Tree**	**Random Forest**
Use training Set	25	27	4	40	6	303
5-Fold Cross Validation	15	112	16	226	26	1454
10-Fold Cross Validation	38	255	29	585	70	5208

4. KESIMPULAN DAN SARAN

Berdasarkan data hasil eksperimen maka dapat disimpulkan bahwa teknik seleksi fitur *Information Gain* mampu meningkatkan performa metoda klasifikasi khususnya *Random Forest* yang memiliki performa yang lebih baik dibandingkan *Naïve Bayes, Bayes Network, OneR, AdaBoost* dan *Random Tree* dengan tangkat akurasi 99.99% pada pengujian seluruh data training dan 99.95% pada pengujian menggunakan *10-fold cross validation.* Namun disisi lain, *Random Forest* memiliki waktu yang lebih lama untuk membangun model dan proses *training* jika dibanding *Naïve Bayes, Bayes Network, OneR, AdaBoost* dan *Random Tree.*

Pada eksperimen yang dilakukan pada penelitian ini, peneliti menggunakan *Information Gain* sebagai teknik seleksi fitur terhadap dataset CICIDS-2017 dalam mendeteksi serangan DDoS. Untuk penelitian selanjutnya dapat digunakan teknik seleksi fitur lain yang mungkin dapat meningkatkan performa deteksi serangan DDoS. Selain penggunaanteknik klasifikasi yang lain perlu dipertimbangkan dalam penelitian berikutnya, khususnya yang memiliki performa yang lebih baik dengan waktu komputasi yang rendah.

DAFTAR PUSTAKA

[1] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and Big Heterogeneous Data : a Survey," 2015.

[2] J. David and C. Thomas, "DDoS attack detection using fast entropy approach on flow-based network traffic," *Procedia Comput. Sci.*, vol. 50, pp. 30–36, 2015.

[3] N. A. Singh, J. Singh, and T. De, "Distributed denial of service attack detection using naive bayes classifier through info gain feature selection," *ACM Int. Conf. Proceeding Ser.*, vol. 25-26-Augu, 2016.

[4] Y. Chen, "DDoS Detection Method Based on Chaos Analysis of Network Traffic Entropy," vol. 18, no. 1, pp. 114–117, 2014.

[5] S. M. T. Nezhad, M. Nazari, and E. A. Gharavol, "A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 700–703, 2016.

[6] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[7] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," *Proc. 2017 Int. Conf. Cloud Comput. Technol. Appl. CloudTech 2017*, vol. 2018-Janua, pp. 1–7, 2018.

[8] Y. Chen, X. Ma, and X. Wu, "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 1052–1054, 2013.

[9] K. J. Singh and T. De, "Efficient Classification of DDoS Attacks Using an Ensemble Feature Selection Algorithm," *J. Intell. Syst.*, 2017.

[10] A. I. Madbouly, A. M. Gody, and T. M. Barakat, "Relevant Feature Selection Model Using Data Mining for Intrusion Detection System," *Int. J. Eng. Trends Technol.*, vol. 9, no. 10, pp. 501–512, 2014.

[11] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," *Ski. 2014 - 8th Int. Conf. Software, Knowledge, Inf. Manag. Appl.*, 2014.

[12] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, "Feature selection using information gain for improved structural-based alert correlation," *PLoS One*, vol. 11, no. 11, pp. 1–18, 2016.

[13] Akashdeep, I. Manzoor, and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Syst. Appl.*, vol. 88, pp. 249–257, 2017.

[14] D. Summeet and D. Xian, *Data Mining and Machine Learning in Cybersecurity*. CRC Press, 2011.

[15] J. Han, *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. 2011.

[16] B. LEO, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[17] J. Jiang, Q. Wang, Z. Shi, B. Lv, and B. Qi, "RST-RF: A hybrid model based on rough set theory and random forest for network intrusion detection," *ACM Int. Conf. Proceeding Ser.*, pp. 77–81, 2018.

[18] R. K. Singh, S. Dalal, V. K. Chauhan, and D. Kumar, "Optimization of FAR in Intrusion Detection System by Using Random Forest Algorithm," *SSRN Electron. J.*, pp. 3–6, 2019.

[19] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018.

[20] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol.*, vol. 7, no. 3.24 Special Issue 24, pp. 479–482, 2018.

[21] T. Garg and S. S. Khurana, "Comparison of classification techniques for intrusion detection dataset using WEKA," *Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2014*, 2014.

[22] B. Cui and S. He, "Anomaly detection model based on hadoop platform and weka interface," *Proc. - 2016 10th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2016*, pp. 84–89, 2016.