**Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer** e-ISSN: 2548-964X
Vol. 5, No. 4, April 2021, hlm. 1329-1333 http://j-ptiik.ub.ac.id

# Deteksi Serangan DDoS Pada Jaringan Berbasis SDN Dengan Klasifikasi Random Forest

**Muhammad Khairullah Harto[1], Achmad Basuki[2]**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]mkharto@student.ub.ac.id, [2]abazh@ub.ac.id

**Abstrak**

Seiring dengan berkembangnya teknologi jaringan, semakin berkembang pula ancaman yang terkait dengan jaringan, salah satunya adalah DDoS (Distributed Denial of Service). Infrastruktur jaringan yang semakin maju dan terjangkau membuat DDoS semakin banyak terjadi. Dari tahun ke tahun, ada peningkatan jumlah kasus serangan DDoS yang signifikan. Oleh karena itu, kajian terhadap berbagai algoritme untuk melakukan klasifikasi data dalam jaringan yang merupakan DDoS dalam masih perlu dilakukan. Jaringan SDN dipilih karena memiliki implementasi yang sederhana dan tidak memerlukan banyak sumber daya karena informasi tentang topologi jaringan serta controller yang akan dibangun menggunakan Ryu. Metode yang digunakan adalah algoritme Random Forest sebagai metode untuk mengklasifikasikan serangan DDoS. Setelah dilakukan penelitian, peneliti menemukan bahwa Random Forest bekerja dengan baik dalam melakukan deteksi terhadap serangan DDoS. Akurasi yang cukup tinggi sekitar 90% dengan waktu deteksi rata-rata 0,3 detik.

**Kata kunci**: *deteksi, DDoS, SDN, Random Forest*

**Abstract**

*Along with the development of network technology, threats related to the network are also growing, one of which is DDoS (Distributed Denial of Service). The increasingly advanced and affordable network infrastructure makes DDoS happen more and more. From year to year, there is a significant increase in the number of DDoS attack cases. Therefore, it is still necessary to study various algorithms for classifying data in networks that constitute deep DDoS. The SDN network was chosen because it has a simple implementation and does not require a lot of resources because of the information about the network topology and the controller that will be built using Ryu. The method used is the Random Forest algorithm as a method for classifying DDoS attacks. After conducting the research, the researchers found that Random Forest performed well in detecting DDoS attacks. The accuracy is quite high around 90% with an average detection time of 0.3 seconds.*

**Keywords**: *detection, DDoS, SDN, Random Forest*

## 1. PENDAHULUAN

DDoS (*Distributed Denial of Service*) merupakan masalah lama dalam jaringan yang terus menjamur dan berkembang. Dilansir dari halaman media digital IT "*infosecurity*" , jumlah kasus DDoS pada kuartal pertama tahun 2019 meningkat sebanyak 84% dari kuartal keempat tahun 2018 (Hill, 2019). DDoS pun terus berevolusi dari waktu ke waktu. Evolusi DDos sendiri terjadi pula karena perkembangan infrastruktur jaringan yang membuat *botnet*/ *zombies* semakin subur. Daya hancur yang dihasilkan pun juga semakin kuat dengan MTU (*Maximum Transmission Unit*) yang semakin besar (Osterweil, et al., 2019). DDoS sendiri merupakan serangan terhadap keamanan jaringan yang bertujuan untuk melumpuhkan jaringan target dengan cara mengirimkan banyak paket secara bertubi-tubi dengan ukuran yang besar (Zargar, et al., 2013).

Dalam penanganan DDoS umumnya diperlukan dua langkah dasar, identifikasi

dan filtering. Identifikasi dilakukan secara manual atau dengan bantuan IDS (*Intrusion Detection System*) maupun IPS (*Intrusion prevention System*). Tujuannya adalah untuk menentukan atau mengidentifikasi kapan anomali terjadi dan asal anomali tersebut. Identifikasi yang dilakukan oleh IDS maupun IPS sangat membantu dalam penerapan tindakan untuk mengurangi dampak dari serangan. Selain itu, karena DDoS memiliki banyak varian, IDS maupun IPS mampu mengenali pola yang tidak dapat dikenali dengan mudah oleh mata manusia. Kombinasi dari IDS ataupun IPS dengan *firewall* merupakan arsitektur *network security* yang umum digunakan pada keamanan jaringan. Penggunaan firewall sangat tergantung dari konfigurasi administrator, sehingga administrator memegang penuh kendali. Tapi jika dikonfigurasi dengan filter yang terlalu ketat, maka aktivitas normal bisa terhambat. Jika dikonfigurasikan terlalu longgar, maka anomali pun juga akan sering terjadi. Oleh karena itu *firewall* biasanya disandingkan dengan IDS ataupun IPS dalam praktiknya. IDS ataupun IPS akan memberikan guideline untuk melakukan konfigurasi pada *firewall*. Namun perlu diketahui juga, bahwa IDS ataupun IPS tetap rentan terhadap kesalahan. Pembaharuan berkala perlu dilakukan untuk mengurangi kesalahan dalam proses identifikasi (Hussain, 2018).

Berdasarkan paparan di atas, penelitian ini mengusulkan mekanisme deteksi terhadap DDoS. Mekanisme yang akan diajukan, memanfaatkan lingkungan jaringan SDN dengan menggunakan *Random Forest*. *Random Forest* dipilih karena dalam melakukan klasifikasi DDoS atribut yang digunakan oleh penyerang memiliki berbagai macam variasi, hal ini bertujuan untuk menghindari proses deteksi. Variasi tersebut menciptakan area abu-abu, dimana serangan yang terjadi sulit dibedakan antara serangan atau bukan. *Random Forest* menggunakan sistem *election*, dimana setiap nilai data *return decision tree* yang digenerate akan menjadi bahan voting. Jumlah *decision tree* yang digenerate akan dibatasi berdasarkan kinerja baik berdasarkan metrik maupun waktu, untuk memaksimalkan kecepatan pengambilan keputusan. Kinerja berdasarkan metrik meliputi akurasi, presisi, recall dan f1-score, yang mengetahui apakah hasil deteksi sudah sesuai sasaran. Sedangkan kinerja berdasarkan waktu adalah waktu yang dibutuhkan sistem untuk mendeteksi apakah lalu lintas yang sedang berjalan adalah serangan atau bukan. Hasil dari kinerja metrik dan waktu pengambilan keputusan akan menentukan jumlah optimal dari *decision tree* yang dibutuhkan dalam *Random Forest*, dimana kinerja secara metrik bernilai mendekati sempurna dengan waktu pengambilan keputusan yang tidak terlalu lama.

## 2. Landasan Kepustakaan

### 2.1 Distributed Denial of Service (DDoS)

DDoS adalah sekumpulan serangan Denial of Service yang memiliki lebih dari satu alamat traceback. Biasanya serangan DDoS menggunakan entitas botnet/zombies. Cara kerja nya adalah dengan mengirimkan paket data secara terus menerus dengan sasaran satu alamat yang sama. Paket yang digunakan biasanya adalah fragment TCP (TCP syn flood), ICMP (ping to death) dan UDP (UDP flood attack). Tujuan dari serangan ini adalah membuat sistem yang diserang kehabisan sumber daya sehingga kewalahan tidak mampu melayani request reguler (S.T Zargar, et al., 2013)

### 2.2 Intrusion Detection System (IDS)

Secara analogi intrusion detection system (IDS) adalah 'alarm pencuri' (atau lebih tepatnya alarm intrusi) dalam keamanan komputer, intrusion prevention system (IPS) adalah satpamnya dan firewall adalah gerbangnya. Data yang masuk akan dilakukan seleksi awal oleh firewall, dengan aturan yang sudah didefinisikan. Ketika sistem firewall telah tertembus oleh serangan, IDS akan memberikan pemberitahuan ke user atau sistem pencegahan untuk menghalau serangan tersebut. Tugas IDS hanya melakukan

pemberitahuan informasi ketika system dijebol saja (V Jaganesh, et al., 2015).

### 2.3 Algoritme Random Forest

Random Forest pada dasarnya adalah sekumpulan decision tree yang melakukan seleksi untuk satu kasus, lalu untuk menentukan hasilnya setiap decision tree dalam random forest akan melakukan voting berdasarkan nilai return dari masing-masing tree (Tin Kam Ho, 1995). Dengan adanya voting, nilai return akan jauh lebih akurat karena dapat menyentuh data yang ambigu.

Berdasarkan data training yang sudah direkam terlebih dahulu, dibentuk beberapa pohon keputusan. Setiap pohon akan melakukan klasifikasi untuk data baru. Hasil klasifikasi kemudian dihimpun dan dihitung berdasarkan hasilnya. Hasil terbanyak akan menjadi kelas dari data tersebut (A. Liaw, et al.,2002).

### 2.5 SDN

Software Defined Network (SDN) adalan paradigma baru dalam membangun suatu jaringan komputer. SDN memberikan harapan baru untuk merubah keterbatasan dari infrastruktur jaringan yang ada. Dalam susunan infrastrukturnya, SDN memiliki beberapa perbedaan dengan jaringan onvensional. Pertama, SDN mengurai integrasi vertikal dengan memisahkan pusat kontrol (control plane) dan perangkat forwarding (Data Plane). Kedua, pemisahan ini switch pada jaringan hanya berperan sebagai forwarding device saja dan pengontrol logika direpresentasikan sebagai logically centralized controller(Diego Kreutz, et al., 2014).

Selain dari sisi arsitekturnya, dari sisi menajemennya juga berbeda. Jika router menggunakan router OS sebagai pusat manajemen, kini user bisa memprogram sendiri bagaimana jaringan tersebut beroperasi. Controller yang programmable dapat diubah dan dimodifikasi sendiri oleh pengembang melalui middle box. Tidak terbatas pada rules dan flow table saja, controller juga dapat diprogram untuk melakukan monitoring jaringan dimana ia beroperasi (Izzat Alsmadi, et al.,2015).

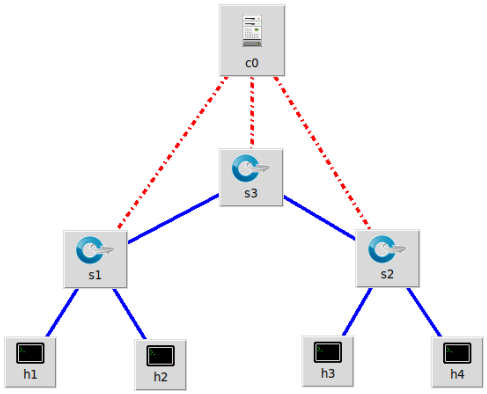
## 3. METODOLOGI

Cara kerja dari sistem yang akan dibangun bisa dilihat pada gambar 1. Dataset dimasukkan yang kemudian akan dibuat menjadi model Random Forest. Kemudian sistem akan merekan lalu lintas yang terjadi dan akan diuji berdasarkan permodelan yang sudah dibuat.


Gambar 1 Diagram Alir Sistem

Dalam perancangan sistem deteksi dengan metode menggunakan *Software Defined Network* dan *Open Flow* sebagai *controller*. Penelitian ini akan mengimplementasikan arsitektur SDN dengan satu buah server yang telah terkonfigurasi dan bersedia untuk melayani client, satu *controller* SDN dan tiga SDN switch yang nantinya

akan mengarahkan *request* dari client menuju ke server. Untuk detail dan bentuk topologi, dapat dilihat pada gambar 2.




Gambar 2 Desain topologi

Untuk dataset yang digunakan sebagai data latih, menggunakan dataset dari Universitas Muhammadiyah yang dihimpun oleh Oxicusa Gugi Housman pada tahun 2020. Data tersebut nantinya akan dilatih untuk membuat pemodelan Random Forest yang digunakan untuk proses klasifikasi.

## 4. HASIL PENGUJIAN DAN ANALISIS

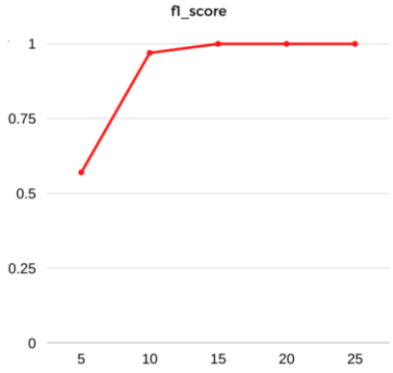
Pengujian dilakukan dengan proses meliputi pengujian kinerja metrik berdarkan jumlah pohon keputusan (n_estimator) yang dibuat dan waktu pengambilan keputusan berdasarkan jumlah pohon keputusan.

### 4.1 Analisis kinerja metrik berdasarkan jumlah pohon keputusan

Tabel 1 Kinerja Random Forest dalam mengidentifikasi DDoS

| n_estimator | Akurasi | Presisi | *Recall* | *F1_score* | *False rate* |
|---|---|---|---|---|---|
| 5 | 67% | 0.53 | 0.67 | 0.57 | 0.33 |
| 10 | 97% | 0.97 | 0.97 | 0.97 | 0.03 |
| 15 | 100% | 1 | 1 | 1 | 0 |
| 20 | 100% | 1 | 1 | 1 | 0 |
| 25 | 100% | 1 | 1 | 1 | 0 |
| Average | 92.8% | 0.9 | 0.928 | 0.908 | 0.072 |

Pada tabel 1 dapat dilihat masing- masing kinerja dalam random forest dengan jumlah pohon keputusan yang berbeda-beda. Ketika jumlah pohon keputusan hanya 5, tingkat false rate nya sangat tinggi. Ketika jumlah pohon keputusan ada 10, kinerja dari sistem meningkat drastis. False rate yang dihasilkan pun sangat rendah. Ketika pohon keputusan berjumlah 15 dan seterusnya, kinerja sudah mendekati sempurna dan false rate yang dihasilkan nyaris tidak ada. Untuk melihat dinamika kinerja dari sistem, bisa dilihat pada grafik pada gambar 2.




Gambar 3. Grafik kinerja berdasarkan f1_score

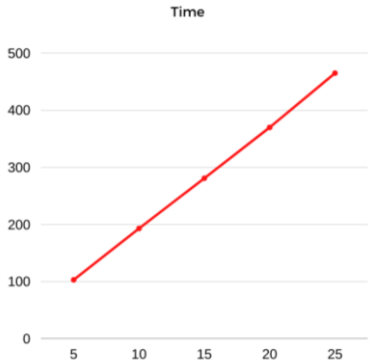
Pada gambar 3, grafik menunjukan perbedaan kinerja pada sistem berdasarkan f1_score-nya. Perbedaan dari 5 pohon ke 10 pohon, ada lompatan kinerja yang lumayan besar, sedangkan dari 10 ke 15, lompatan kecil. Setelah 15 tree, grafik kinerja landai tanpa ada penurunan kinerja. Sehingga dapat disimpulkan puncak dari kinerja tercapai pada jumlah n_estimator adalah 15, penambahan n_estimator lebih dari itu tidak akan memberi penambahan atau pengurangan kinerja

### 4.2 Analisis kinerja berdasarkan waktu

Tabel 2 Kinerja Random Forest berdasarkan waktu

| n_estimator | Time* |
|---|---|
| 5 | 103 |
| 10 | 193 |
| 15 | 281 |
| 20 | 370 |
| 25 | 465 |
| Average | 282.4 |

*)dalam milisecond(ms)




Gambar 4. Grafik kinerja berdasarkan waktu pengambilan keputusan

Waktu pengambilan keputusan sangat sensitif dengan jumlah n_estimator. Pada tabel 2 dan gambar 3, dapat dilihat peningkatan waktu pengambilan keputusan terus bertambah seiring dengan penambahan n_estimator. Perlu diketahui, library sklearn yang digunakan untuk melakukan pemodelan, menjalankan n_estimator secara serial. Sehingga semakin banyak pohon yang di-generate maka semakin lama juga waktu pengambilan keputusan. Rata-rata penambahan waktu untuk setiap 5 n_estimator adalah 90 millisecond(ms).

Berdasarkan data-data diatas, peneliti menyimpulkan bahwa jumlah optimum pohon yang di-generate adalah 15. Hal ini karena setelah 15 grafik kinerja landai, tidak ada kenaikan ataupun penurunan. Selain itu waktu pengambilan keputusan yang dibutuhkan juga tidak terlalu panjang, masih dibawah 0.5 detik (500 milidetik). Selain itu jika pohon yang di-generate ditambah lagi, kemungkinan beban kerja yang dihasilkan juga akan bertambah dari sisi waktu dan sumber daya

## 5. Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan dapat ditarik beberapa kesimpulan, yakni:

1. Berdasarkan pengujian yang telah dilakukan, kinerja Random forest dalam melakukan deteksi serangan DDoS dengan cukup baik. Rata-rata akurasi yang dihasilkan adalah 92,8% dengan akurasi maksimum bisa mencapai 100%. Untuk presisi dan recall memiliki rata-rata hasil 0.90 untuk presisi dan 0.93 untuk recall. Untuk rata-rata f1_score nya didapatkan nilai 0.9 dengan false rate 7.2%. Waktu pengambilan keputusannya juga terbilang singkat dengan rata-rata 282.4 ms atau sekitar 0.3 detik.
2. Jumlah optimum pohon yang di-generate adalah 15. Hal ini karena setelah 15 grafik kinerja landai, tidak ada kenaikan ataupun penurunan. Selain itu waktu pengambilan keputusan yang dibutuhkan juga tidak terlalu panjang, 281 ms, 1 ms dibawah rata-rata. Selain itu dengan jumlah 15 pohon yang di-generate, beban kerja mesin juga tidak terlalu berat .

## 6. DAFTAR PUSTAKA


Alsmadi, I., & Xu, D. 2015. *Security of Software Defined Networks: A survey. Computers and Security*, 53, 79

Farnaaz, N., & Jabbar, M. A. 2016. *Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science*, 89, 213–217.

Ho, T. K. 1995. *Random decision forests. Proceedings of the International Conference on Document Analysis and Recognition*, ICDAR, 1, 278–282.

Housman, Oxicusa Gugi; Isnaini, Hafida; Sumadi, Fauzi Dwi Setiawan 2020. *SDN-DDOS (ICMP,TCP,UDP). Mendeley Data.* v1.

Hussain, A. 2018. *Use of Firewall and Ids To Detect and Prevent Network Attacks. International Journal of Technical Research & Science*, 3(IX), 289–292.

Jaiganesh, V., & Karthikeyan, M. M. 2015. *Intrusion Detection Systems : A survey and Analysis of Security Issues*. 4(6), 553–556.

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. 2015. *Software-defined networking: A comprehensive survey. Proceedings of the IEEE*, 103(1), 14–76.

Liaw, A., & Wiener, M. 2002. *Classification and regression by randomForest. R news*, 2(3), 18-22.

Osterweil, E., Stavrou, A., & Zhang, L. 2019. *20 Years of DDoS: a Call to Action*. 1(1), 1–11.

Zargar, S. T., Joshi, J., & Tipper, D. 2013. *A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. IEEE Communications Surveys and Tutorials*, 15(4), 2046–2069.