e-ISSN. 2685-4066

Jurnal BITe
Vol.1 No.2 (Desember) 2019, Hal 143-152

Klasifikasi *Data Log Intrusion Detection Sistem* (Ids) Dengan *Decision Tree* C4.5

[1]Thifal Baraas, [2]Akbar Juliansyah, [3]Ahmad Ashril Rizal

[1]Universitas Bumigora, thifal@universitasbumigora.ac.id
[2]Universitas Bumigora, akbar.juliansyah@stmikbumigora.ac.id
[3]Universitas Bumigora, ashril.rizal@stmikbumigora.ac.id

Abstrak

Browsing atau kegiatan menjelajahi internet menjadi salah satu aktivitas yang sering dilakukan pada zaman kini. Baik anak-anak hingga orang dewasa menjadi pengguna internet. Akan tetapi para pengguna internet tidak mengetahui jika internet juga bisa menjadi ancaman terutama adanya serangan-serangan yang menyerang sistem keamanan jaringan. Untuk mendeteksi adanya aktivitas yang mencurigakan yang melalui jaringan dibutuhkan bantuan dari IDS (*Intrusion Detection Sistem*). Ketika terjadi banyak serangan yang masuk, IDS tidak bisa menanganinya secara akurat, hal ini mengakibatkan aktivitas normal di dalam jaringan bisa dianggap sebagai serangan dari *hacker* atau sebaliknya. *Data mining* adalah prses yang digunakan untuk menemukan hubungan dari data-data untuk mendapatkan sebuah kesimpulan dari data tersebut. Algoritma C4.5 merupakan salah satu algoritma yang digunakan untuk membuat pohon keputusan. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami. Dengan mengklasifikasi data log IDS dengan algoritma C4.5 dapat mengurangi terjadinya kesalahan IDS dalam menentukan aktivitas yang termasuk serangan atau bukan. Hasil penelitian menunjukkan data log IDS dapat diklasifikasikan dengan algoritma C4.5 dengan tingkat akurasi model adalah 96.371% yang membuktikan bahwa model ini dapat digunakan dalam menentukan aktivitas yang termasuk serangan atau bukan.

Kata Kunci: Klasifikasi, Algoritma C4.5, Data Log, *Intrusion Detection Sistem*

Abstract

Browsing or surfing the internet is one of the activities that are often done today. Both children and adults become internet users. However, internet users do not know the internet can also be a threat, especially the attacks that attack the network security system. To detect suspicious activity through the network, assistance from IDS (Intrusion Detection System) is needed. When there are many incoming attacks, IDS cannot handle it accurately, this results in normal activities on the network can be considered as an attack from hackers or vice versa. Data mining is a process used to find relationships from data to get a conclusion from that data. C4.5 algorithm is one algorithm used to make a decision tree. The decision tree method converts very large facts into decision trees that represent rules. Rules can be easily understood with natural language. By classifying the IDS log data with the C4.5 algorithm it can reduce the occurrence of IDS errors in determining which activities are included or not. The results showed the IDS log data can be classified with the C4.5 algorithm with a 96.371% accuracy rate of the model which proves that this model can be used in determining activities that are included as attacks or not.

***Keyword*:** *Clasification, Algorithm C4.5, Data Log, Intrusion Detection Sistem*

I. PENDAHULUAN

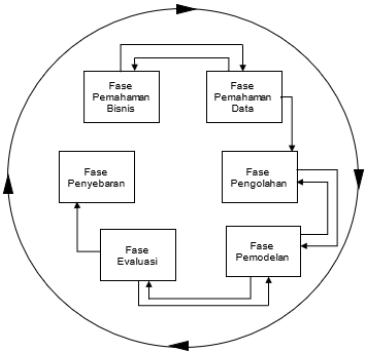
Browsing atau kegiatan menjelajahi internet menjadi salah satu aktivitas yang sering dilakukan pada zaman kini. Baik anak-anak hingga orang dewasa menjadi pengguna internet. Data yang dirilis oleh *We Are Social* per Agustus 2017 jumlah pengguna internet global menyentuh angka 3.800.000.000 dengan penetrasi 51% dari total populasi di dunia[1] Beberapa aktivitas yang sering dilakukan ketika *browsing* adalah membuka sosial media, berkomunikasi, *streaming*, mencari data maupun berbelanja. Akan tetapi, para pengguna internet tidak mengetahui jika internet juga bisa menjadi ancaman terutama adanya serangan-serangan yang menyerang sistem keamanan jaringan. Keamanan sistem dan jaringan dengan pemasangan perangkat *firewall* tidaklah cukup. Peningkatan serangan menyebabkan data yang harus dianalisis menjadi sangat besar, sistem keamanan jaringan internet yang telah ada memiliki keterbatasan dalam kemampuan beradaptasi sejumlah besar data dan jenis serangan baru[2].

Untuk mendeteksi adanya aktivitas yang mencurigakan yang melalui jaringan dibutuhkan bantuan dari IDS (*Intrusion Detection System*). IDS memberikan pertahanan pertama yang sangat penting dalam menghadapi penyusupan. Jika penyusup berusaha masuk ke server jaringan, mungkin dapat ditemukan bukti dalam sistem log, meskipun *hacker* pintar akan menghapus file log. *Host* IDS mengamati aktivitas yang tidak layak pada setiap sistem. Jika penyusup berusaha mengganggu *server* yang sama menggunakan serangan fregmentasi, mungkin dapat diketahui apa yang terjadi dengan melihat log [3]. Menurut Khaerani (2010), pada IDS terdapat masalah yaitu, ketika terjadi banyak serangan yang masuk dan IDS tidak bisa menanganinya secara akurat, maka hal ini mengakibatkan aktifitas normal di dalam jaringan dianggap sebagai serangan dari *hacker* (*False Positive*) sedangkan ketika terjadi serangan yang sebenarnya IDS tidak mengirimkan *alert* (*False Negative*)[4].

Data mining adalah proses yang digunakan untuk menemukan hubungan dari data-data untuk mendapatkan sebuah kesimpulan dari data tersebut. Teknik yang digunakan dalam *data mining* adalahpengenalan pola dan menggunakan matematika dalam menyelesaikannya. Salah satu algoritma yang sering digunakan adalah algoritma C4.5. Algoritma C4.5 merupakan salah satu algoritma yang digunakan untuk klasifikasi, membuat pohon keputusan dan mempresentasikan aturan-aturan dari pohon keputusan tersebut. Pemilihan algoritma C4.5 dikarenakan algoritma ini merupakan algoritma yang biasa digunakan untuk klasifikasi [5]. Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah [6]. Berdasarkan pada pemaparan masalah tersebut, akan dilakukan penelitian menganalisa algoritma C4.5 untuk mengklasifikasi *data log* IDS dengan tujuan untuk mengetahui cara klasifikasi serangan yang ada dalam data log.

II. METODOLOGI

Metodologi penelitian yang digunakan dalam penelitian ini adalah CRISP-DM (*Cross-Industry Standard Process for Data Mining*). CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian [4]. Proses *data mining* berdasarkan CRISP-DM terdiri dari 6 fase yaitu *Business Understanding* (Pemahaman Bisnis), *Data Understanding* (Pemahaman Data), *Data Preparation* (Pengolahan Data), *Modeling* (Pemodelan), *Evaluation* (Evaluasi), dan *Deployment* (Penyebaran).



Gambar 1. Proses Data Mining CRISP-DM

Dari 6 fase tersebut yang dilakukan pada penelitian ini hanya 5 fase saja yaitu *Business Understanding, Data Understanding, Data Preparation, Modeling,* dan *Evaluation*.

1. ***Business Understanding***

   Pada fase pemahaman bisnis akan menentukan tujuan penelitian, manfaat penelitian dan batasan penelitian. Tujuan penelitian ini adalah mengetahui cara klasifikasi dan tingkat akurasi data log IDS dengan metode C4.5. Manfaat penelitian ini adalah mengetahui cara klasifikasi data log IDS dengan metode C4.5, mengetahui tingkat akurasi model yang dibuat, mengetahui jenis-jenis serangan pada data log IDS. Batasan masalah dalam penelitian ini yaitu sebagai berikut:
   a. Pada penelitian ini hanya menganalisa *data log* IDS
   b. Metode data mining yang digunakan adalah algoritma C4.5
   c. Metode reduksi variabel yang digunakan adalah *Principal Component Analysis* (PCA)
   d. Aplikasi yang digunakan adalah Rapidminer
   e. Dataset yang digunakan adalah Dataset NSL-KDD
   f. Pengukuran evaluasi dari perbandingan algoritma *data mining* menggunakan *confusion matrix*.
2. ***Data Understanding***

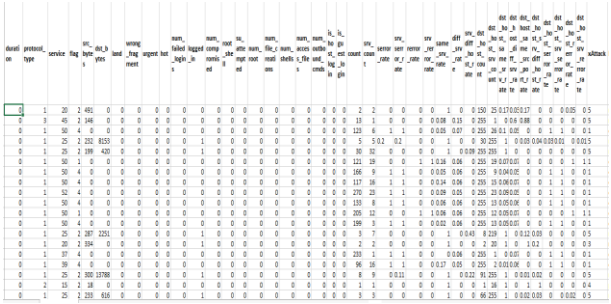
   Dataset yang digunakan adalah NSL-KDD yang terdiri dari 41 atribut yang tersedia dan atribut ke-42 berisi tentang kelas normal dan empat kelas serangan [7]. Dataset yang digunakan berasal dari http://github.com/FransHBotes/NSLKDD-Dataset Dataset terdiri dari dua data yaitu data *training* dan data *testing*. Data *training* digunakan untuk diklasifikasi dengan algoritma C4.5 sedangkan data *testing* digunakan untuk memeriksa akurasi dari percobaan algoritma pada data *training*. Adapun atribut beserta deskripsinya dapat dilihar dari tabel 1.

Tabel 1. Deskripsi Atribut NSL-KDD

No Atribut	Nama Atribut	Deskripsi	Tipe
1	Duration	Panjang durasi waktu	*Continuous*
2	protocol_type	Penggunaan protokol dalam koneksi (TCP, UDP, ICMP)	Simbolik
3	Services	Layanan tujuan jaringan yang digunakan	Simbolik
4	Flag	Status koneksi (normal atau *error*)	Simbolik
5	src_bytes	Jumlah *byte* data yang ditransfer dari sumber ke tujuan dalam satu koneksi	*Continuous*
6	dst_bytes	Jumlah *byte* data yang ditransfer dari tujuan untuk sumber di satu koneksi	*Continuous*
7	land	Jika sumber dan IP *adress* tujuan dan nomor port sama, variabel ini bernilai 1 jika tidak 0	Simbolik
8	wrong_fragment	Total nomor pada fragmen yang salah pada koneksi	*Continuous*
9	urgent	Jumlah paket yang *urgent* pada koneksi.	*Continuous*
10	hot	Banyaknya indikator '*hot*' dalam konten seperti: memasuki sistem direktori, membuat progra, dan melaksanakan program	*Continuous*
11	num_failed_login	Menghitung usaha login yang gagal	*Continuous*
12	logged_in	Status login: jika 1	*Discrete*

		berhasil login dan 0 sebaliknya.	
13	num_compromised	Nomor dari kondisi yang bisa dikompromosikan	*Continuous*
14	root_shell	Jika 1, *shell root* bisa diakses dan 0 sebaliknya	Simbolik
15	su_attempted	Jika 1, percobaan perintah "*su root*" dapat digunakan dan 0 sebaliknya	Simbolik
16	num_root	Banyaknya mengakses "*root*" atau jumlah kegiatan yang dilakukan sebagai *root* di dalam koneksi	*Continuous*
17	num_file_creations	Jumlah operasi pembuatan file dalam koneksi	*Continuous*
18	num_shells	Banyaknya jumlah *shell prompt*	*Continuous*
19	num_access_files	Nomor dalam operasi akses kontrol file	*Continuous*
20	num_outbound_cmds	Banyaknya jumlah dari perintah outbound dalam sesi ftp	*Continuous*
21	num_hot_login	Jika 1, login termasuk daftar hot jika 0 maka tidak	Simbolik
22	is_guest_login	Jika 1, login adalah login '*guest*' jika 0, maka tidak	Simbolik
23	Count	Jumlah koneksi ke *host* tujuan yang sama dalam 2 detik terakhir	*Continuous*
24	srv_count	Jumlah koneksi pada *service* sama untuk koneksi yang sama dalam 2 detik terakhir	*Continuous*
25	serror_rate	Persentase koneksi yang terdapat pada flag (4) seperti s0, s1, s2 atau s3 yang dikumpulkan pada count (23)	*Continuous*
26	srv_serror_rate	Persentase koneksi yang dihubungkan diantara koneksi	*Continuous*
		flag (4) s0, s1, s2 atau s3 yang dikumpulkan pada srv_count (24)	
27	rerror_rate	Persentase koneksi yang telah diaktifkan pada koneksi flag (24) REJ, yang dikumpulkan pada count (23)	*Continuous*
28	srv_rerror_rate	Persentase koneksi yang sudah mengaktifkan flag (4) REJ, dikumpulkan di antara koneksi srv_count (24)	*Continuous*
29	same_srv_rate	Persentase koneksi ke *service* (3) yang sama, yang dikumpulkan pada koneksi count (23)	*Continuous*
30	diff_srv_rate	Persentase koneksi ke *service* yang berbeda, dikumpulkan dalam koneksi count (23)	*Continuous*
31	srv_diff_host_rate	Persentase koneksi ke mesin tujuan yang berbeda dikumpulkan diantara koneksi srv_count (24)	*Continuous*
32	dst_host_count	Nomor yang memiliki koneksi yang sama dengan tujuan *host* IP *address*	*Continuous*
33	dst_host_srv_count	Nomor dari koneksi yang memiliki *port* yang sama	*Continuous*
34	dst_host_same_srv_rate	Persentase koneksi ke layanan yang sama, dikumpulkan di antara koneksi dst_host_count (32)	*Continuous*
35	dst_host_diff_srv_rate	Persentase koneksi ke *service* yang berbeda, dikumpukan diantara koneksi dst_host_count (32)	*Continuous*
36	dst_host_same_src_port_rat	Persentase koneksi ke sumber *port* yang sama, dikumpulkan	*Continuous*

	e	diantara koneksi dst_host_srv_count (33)	
37	dst_host_srv_diff_host_rate	Persentase koneksi ke mesin tujuan yang berbeda, dikumpulkan diantara koneksi dst_host_srv_count (33)	*Continuous*
38	dst_host_serror_rate	Persentase koneksi yang telah diaktifkan oleh flag (4) s0, s1, s2 atau s3 dikumpulkan diantara koneksi di dst_host_count (32)	*Continuous*
39	dst_host_srv_serror_rate	Persentase koneksi yang telah diaktifkan oleh flag (4) s0, s1, s2 atau s3 dikumpulkan diantara koneksi dst_host_srv_count (33)	*Continuous*
40	dst_host_rerror_rate	Persentase koneksi yang telah diaktidkan flag (4) REJ, dikumpulkan diantara koneksi dst_host_count (32)	*Continuous*
41	dst_host_srv_rate_rerror_rate	Persentase koneksi yang telah diaktidkan diantara koneksi dst_host_srv_count (33)	*Continuous*
42	xAttack	Menyatakan label jenis serangan. 1 = dos, 2 = u2r, 3 = r2l, 4 = probe, 5 = normal	*Simbolik*



Gambar 2. Data NSL-KDD

3. *Data Preparation*

NSL-KDD memiliki 41 atribut dimana jumlah atribut ini sangat banyak jika ingin diolah dengan sebuah algoritma. Oleh karena itu dilakukan reduksi variabel. Proses reduksi variabel dilakukan menggunakan metode *Principal Component Analysis* (PCA). Metode PCA sangat berguna digunakan jika data yang ada memiliki jumlah variabel yang sangat besar dan memiliki korelasi antar variabelnya . Tujuan dari analisa PCA adalah untuk mereduksi variabel yang ada menjadi lebih sedikit tanpa harus kehilangan informasi yang termuat dalam data asli/awal. Pada penelitian ini dilakukan reduksi variabel dengan bantuan aplikasi Matlab. Adapun langkah-langkah yang digunakan untuk mereduksi dataset NSL-KDD dengan PCA adalah:

a. Mempersiapkan dataset NSL-KDD
b. Mengurangi rata-rata dari masing-masing dimensi data (standarisasi)

$$s^2 = \frac{\sum_{i=1}^{n}(X_i-\bar{X})^2}{(n-1)} \quad (1)$$

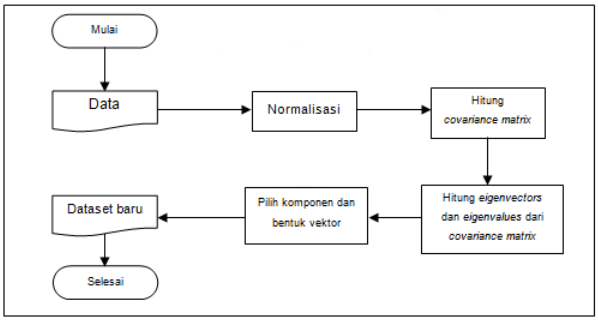
Keterangan:
s^2 = Varian
X_i = Nilai X ke-i
$\bar{X}$ = Rata-rata
n = Ukuran sampel

c. Menghitung *covariance matrix*

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{(n-1)} \quad (2)$$

Keterangan:
$X_i - \bar{X}$ = Nilai X_i dikurangi rata-ratanya
$Y_i - \bar{Y}$ = Nilai Y_i dikurangi rata-ratanya
n = Ukuran sampel

d. Menghitung *eigenvectors* dan *eigenvalues* dari *covariance matrix*
e. Memilih komponen dan membentuk vector fitur



Gambar 3. Alur PCA

4. *Modeling*
Alur saat penentuan akar (*root*) adalah [8] :
   a. Menghitung total nilai informasi dari data trainingnya
   b. Mendaftar atribut A
   c. Tiap-tiap atribut akan dihitung nilai *Entropy* dan *Gain*nya.
   d. Sistem akan membandingkan nilai *Gain* terbesar dari tiap-tiap atribut
   e. Setelah nilai *Gain* terbesar didapat, maka sistem akan memilih atribut dengan *Gain* terbesar sebagai atribut terbaik untuk dijadikan akar
   f. Sistem akan melakukan proses ini sampai semua atribut dalam daftar habis dihitung

Berikut langkah dalam penentuan cabang pada pohon keputusan [8]:
   a. Memilih atribut dengan *gain* tertinggi
   b. Nilai yang ada pada atribut tertinggi akan diklasifikasikan berdasarkan target yang ingin dicapai yaitu jenis serangan Normal (N), DOS (D), U2R (U), R2L (R), dam *Probe* (P).
   c. Tiap nilai atribut akan dihitung *entropy* masing-masing hingga semua atribut habis.
   d. Nilai *entropy* yang nol (0) akan dikoreksi untuk penentuan klasifikasi kasus.
   e. Bila nilai *entropy* lebih dari nol (0) maka nilai tersebut akan dijadikan cabang pada node selanjutnya.

Berikut adalah langkah penentuan *node* pada pembentukan pohon keputusan [8]:
   a. Menghitung jumlah kasus pada atribut dengan *gain* tertinggi
   b. Menghitung nilai *entropy* total dari atribut dengan *gain* tertinggi.
   c. Mendaftar atribut A selain atribut dengan *gain* tertinggi.
   d. Menghitung *entropy* dari masing-masing nilai atribut ke-n hingga habis terhitung semua
   e. Menghitung nilai *gain* dari masing-masing atribut
   f. Menentukan nilai *gain* maksimal sebagai penentuan *node* selanjutnya
   g. Setelah *node* selanjutnya terpilih maka proses perhitungan akan berulang lagi mulai dari penentuan cabang dan penentuan *node* hingga semua atribut habis dieksekusi dan mencapai *end of tree*.

Untuk menghitung *gain* digunakan rumus seperti tertera dalam persamaan berikut [9]:

$$Gain\,(S,A) = entropy\,(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|}\, Entropy\,(A) \quad (3)$$

Keterangan :
S = Himpunan Kasus
A = Atribut
n = Jumlah partisi atribut A
$|Si|$ = Jumlah kasus pada partisi ke-i
$|S|$ = jumlah kasus dalam S

Entropy dapat dikatakan sebagai kebutuhan bit untuk menyatakan suatu kelas. Semakin kecil nilai *Entropy* maka akan semakin *Entropy* digunakan dalam mengekstrak suatu kelas. *Entropy* digunakan untuk mengukur ketidakaslian [10]. Perhitungan nilai *entropy* dapat dilihat pada persamaan berikut [9]:

$$Entropy\,(S) = \sum_{i=1}^{n} - p_i \log_2(p_i) \quad (4)$$

Keterangan :
S = Himpunan kasus
n = Jumlah partisi S
p_i = Proporsi Si terhadap S

5. *Evaluation*
Evaluasi pada penelitian ini lebih difokuskan kepada model yang dihasilkan dalam tahap pemodelan yaitu pohon keputusan dari algoritma C4.5. Setelah mendapatkan pohon keputusan, kemudian

dilakukan pengambilan keputusan dari pohon keputusan tersebut. Selanjutnya menentukan akurasi menggunakan *confusion matrix*.

Confusion matrix mengandung informasi mengenai kelas sebenarnya dan kelas prediksi dari suatu proses klasifikasi. Pada dasarnya *confusion matrix* membandingkan hasil klasifikasi yang dihasilkan oleh suatu system dengan hasil klasifikasi sebenarnya [9].

Tabel 2. Confussion Matrix

		Actual Class	
		Negative class (normal)	Positive class (attack)
Predicted class	Negative class (normal)	True Negative (TN)	False Negative (FN)
	Positive class (attack)	False Positive (FP)	True Positive (TP)

Secara khusus, langkah-langkah berikut ini akan digunakan untuk menilai kinerja IDS:

a. *Accuracy* adalah proporsi dari total jumlah prediksi yang benar. Ditentukan dengan menggunakan persamaan:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

b. *Precision* didefinisikan sebagai proporsi kasus negatif yang diklasifikasikan dengan benar, sebagaimana dihitung menggunakan persamaan:

$$P = \frac{TP}{TP+FP} \quad (6)$$

c. *Recall or True Positive Rate or Detection Rate* (DR) adalah proporsi kasus positif yang diidentifikasi dengan benar, seperti yang dihitung menggunakan persamaan :

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

Evaluasi dilakukan secara mendalam dengan tujuan menyesuaikan model yang didapat agar sesuai dengan sasaran yang ingin dicapai dalam fase pertama.

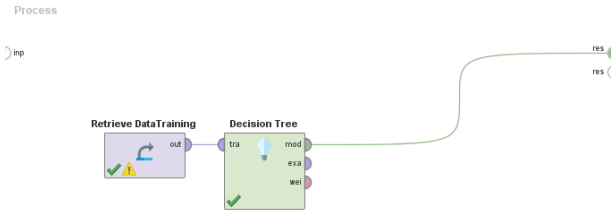
III. HASIL DAN PEMBAHASAN

1. Reduksi Variabel
   Reduksi variabel dengan menggunakan *Principal Component Analysis* menghasilkan 13 variabel dari 41 variabel yang terdapat pada tabel 3.

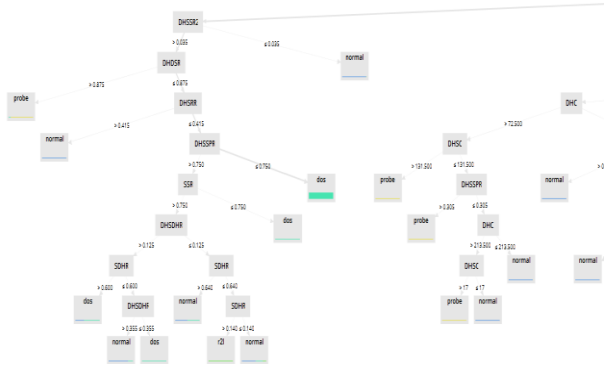
Tabel 3. Variabel Dataset NSL-KDD Setelah Direduksi

Id	**Variabel**	**Inisial**
29	same_srv_rate	SSR
30	diff_srv_rate	DSR
31	srv_diff_host_rate	SDHR
32	dst_host_count	DHC
33	dst_host_srv_count	DHSC
34	dst_host_same_srv_rate	DHSSR
35	dst_host_diff_srv_rate	DHDSR
36	dst_host_same_src_port_rate	DHSSPR
37	dst_host_srv_diff_host_rate	DHSDHR
38	dst_host_serror_rate	DHSR
39	dst_host_srv_serror_rate	DHSSR2
40	dst_host_rerror_rate	DHRR
41	dst_host_srv_rate_rerror_rate	DHSRR

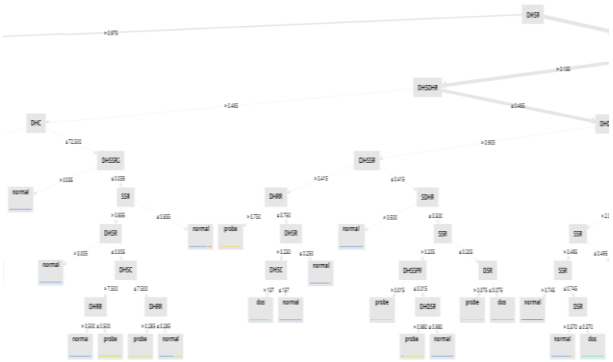
2. Pohon Keputusan
   Pembuatan pohon keputusan menggunakan data *training* yang terdiri dari 125973 data dan menggunakan 13 variabel.



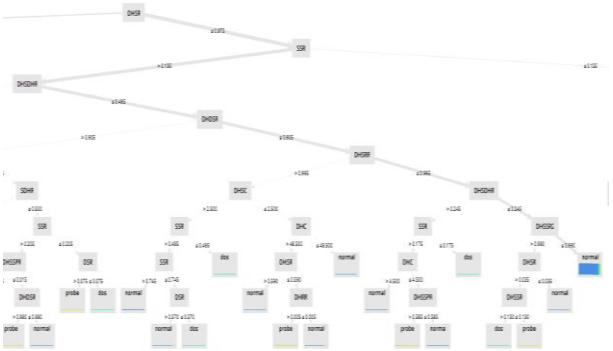
Gambar 4. Proses Membuat Pohon Keputusan dengan Rapidminer



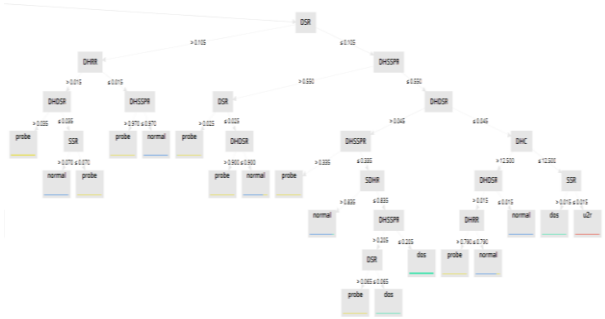
Gambar 5. Pohon Keputusan Bagian 1



Gambar 6. Pohon Keputusan Bagian 2



Gambar 7. Pohon Keputusan Bagian 3



Gambar 8. Pohon Keputusan Bagian 4

3. Evaluasi

Evaluasi dilakukan dengan menggunakan data *testing* yang terdiri dari 22543 data. Dimana hasil klasifikasi pada data *training* akan diuji menggunakan data *testing*. Hasil uji coba dengan menggunakan data *testing* adalah sebagai berikut:

Tabel 4. *Confusion Matrix*

Normal	Probe	DoS	R2L	U2R	Pengukuran
9303	23	85	289	10	**Normal**
18	2388	4	5	6	**Probe**
34	3	7391	27	1	**DoS**
257	4	11	2480	2	**R2L**
11	2	6	20	163	**U2R**

a. Menghitung Nilai Akurasi Serangan Normal

Tabel 5. *Confusion Matrix* Serangan Normal

Predicted Class	**Actual Class**	
	Negative class (normal)	Positive class (attack)
Negative class (normal)	12513	407
Positive class (attack)	320	9303

$$Accuracy = \frac{9303 + 12513}{9303 + 407 + 320 + 12513} = \frac{21816}{22543} = 0.967750521$$

$$Precision = \frac{9303}{9303 + 320} = \frac{9303}{9623} = 0.966746337$$

$$Recall = \frac{9303}{9303 + 407} = \frac{9303}{9710} = 0.958084449$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan normal yang didapatkan adalah 0.967750521 atau sama dengan 96.775%, nilai *precision* adalah 0.966746337 atau sama dengan 96.675% dan nilai *recall* adalah 0.958084449 atau sama dengan 95.808%

b. Menghitung Nilai Akurasi Serangan DoS

Tabel 6. *Confusion Matrix* Serangan DoS

	Actual Class	
Predicted Class	Negative class (normal)	Positive class (attack)
Negative class (normal)	14981	65
Positive class (attack)	106	7391

$$Accuracy = \frac{7391 + 14981}{7391 + 106 + 14981 + 65} = \frac{22378}{22543} = 0.992414497$$

$$Precision = \frac{7391}{7391 + 106} = \frac{7391}{7597} = 0.985861011$$

$$Recall = \frac{7391}{7391 + 65} = \frac{7391}{7456} = 0.991282189$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan DoS yang didapatkan adalah 0.992414497 atau sama dengan 99.241%, nilai *precision* adalah 0.985861011 atau sama dengan 98.586% dan nilai *recall* adalah 0.991282189 atau sama dengan 99.128%

c. Menghitung Nilai Akurasi Serangan R2L

Tabel 7, *Confusion Matrix* Serangan R2L

	Actual Class	
Predicted Class	Negative class (normal)	Positive class (attack)
Negative class (normal)	19448	274
Positive class (attack)	341	2480

$$Accuracy = \frac{2480 + 19448}{2480 + 274 + 341 + 19448} = \frac{21928}{22543} = 0.972718804$$

$$Precision = \frac{2480}{2480 + 341} = \frac{2480}{2821} = 0.879120879$$

$$Recall = \frac{2480}{2480 + 274} = \frac{2480}{2754} = 0.900508351$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan R2L yang didapatkan adalah 0.972718804 atau sama dengan 97.272%, nilai *precision* adalah 0.879120879 atau sama dengan 87.912% dan nilai *recall* adalah 0.900508351 atau sama dengan 90.051%

d. Menghitung Nilai Akurasi Serangan U2R

Tabel 8. *Confusion Matrix* Serangan U2R

	Actual Class	
Predicted Class	Negative class (normal)	Positive class (attack)
Negative class (normal)	22322	39
Positive class (attack)	19	163

$$Accuracy = \frac{163 + 22322}{163 + 39 + 19 + 22322} = \frac{22485}{22543} = 0.997427139$$

$$Precision = \frac{163}{163 + 19} = \frac{163}{182} = 0.895604396$$

$$Recall = \frac{163}{163 + 39} = \frac{163}{202} = 0.806930693$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan U2R yang didapatkan adalah 0.997427139 atau sama dengan 99.743%, nilai *precision* adalah 0.895604396 atau sama dengan 89.56% dan nilai *recall* adalah 0.806930693 atau sama dengan 80.693%

e. Menghitung Nilai Akurasi Serangan Probe

Tabel 9. *Confusion Matrix* Serangan Probe

	Actual Class	
Predicted Class	Negative class (normal)	Positive class (attack)
Negative class (normal)	20090	33
Positive class (attack)	32	2388

$$Accuracy = \frac{2388 + 20090}{2388 + 33 + 32 + 20090} = \frac{22478}{22543} = 0.997116622$$

$$Precision = \frac{2388}{2388 + 32} = \frac{2388}{2420} = 0.98677686$$

$$Recall = \frac{2388}{2388 + 33} = \frac{2388}{2421} = 0.986369269$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan *Probe* yang didapatkan adalah 0.997116622 atau sama dengan 99.712%, nilai *precision* adalah 0.98677686 atau sama dengan 98.678% dan nilai *recall* adalah 0.991282189 atau sama dengan 99.128%

f. Menghitung Nilai Akurasi Keseluruhan
Untuk menghitung nilai keseluruhan, dihitung dengan cara memilih *instances* yang diklasifikasikan dengan benar yang dipilih dari tabel *Confusion Matrix*.

Dari tabel 4 diketahui *instances* yang diklasifikasikan dengan benar terdapat pada garis diagonal yang dapat dilihat pada tabel 10.

Tabel 10. Serangan yang Diklasifikasi Benar

Serangan	Jumlah yang diklasifikasikan benar
Normal	9303
Probe	2388
DoS	7391
R2L	2480
U2R	163

$$Accuracy = \frac{9303 + 2388 + 7391 + 2480 + 163}{22543} = \frac{21725}{22543} = 0.963713791$$

Dari hasil perhitungan di atas diketahui bahwa tingkat akurasi serangan keseluruhan yang didapatkan adalah 0.963713791 atau sama dengan 96.371%.

IV. SIMPULAN DAN SARAN

Berdasarkan pada hasil pembahasan dan pengujian yang dilakukan, maka didapat kesimpulan sebagai berikut :

1. Data log IDS dapat diklasifikasikan dengan menggunakan algoritma C4.5
2. Algoritma C4.5 menghasilkan 66 rule untuk menentukan serangan dengan tingkat akurasi 96.371%

Adapun saran-saran untuk pengembangan lebih lanjut dari skripsi ini adalah sebagai berikut:

1. Menggunakan algoritma *data mining* selain C4.5 dan algoritma-algoritma yang pernah digunakan dalam penelitian sebelumnya.
2. Menggunakan dataset UNSW-NB15
3. Memilih metode reduksi variable selain *Principal Component Analysis* (PCA)
4. Hasil data mining dapat diimplementasikan dalam sebuah aplikasi pendeteksi serangan atau aplikasi IDS seperti *snort* dan *suricata*.

V. UCAPAN TERIMA KASIH

Ucapan terima kasih kepada :

1. Allah SWT atas rahmat, nikmat, kebahagiaan serta anugerahNya.
2. Yang tercinta Abi, Umi, Nadhif , Galbi dan keluarga yang selalu memberikan kasih sayang, do'a, dorongan, dan semangat.
3. Bapak Akbar Juliansyah, S.Kom., M.MT dan Bapak Ahmad Ashril Rizal, M.Cs, selaku Dosen Pembimbing.
4. Bapak/Ibu Dosen yang telah memberikan ilmu selama dalam perkuliahan.
5. Semua teman-teman yang membantu khususnya Ami, Irfa, Tya, Kak Dara, Niko, teman-teman kelas B angkatan 2015, teman-teman TRIGORA, panitia OPPS 2018, dan anggota NETCOM.

REFERENSI

[1] J. Iqbal, "Jumlah Pengguna Internet Dunia Sentuh 3,8 Miliar," 2017.

[2] D. Mongkareng, N. A. Setiawan, and A. E. Permanasari, "Implementasi Data Mining dengan Seleksi Fitur untuk Klasifikasi Serangan pada Intrusion Detection System (IDS)," *CITEE*, pp. 314–321, 2017.

[3] S. A. Budiman, C. Iswahyudi, and M. Sholeh, "Implementasi Intrusion Detection System (IDS) Menggunakan Jejaring Sosial Sebagai Media Notifikasi," in *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*, 2014, pp. 255–262.

[4] F. Zikrillah, "Perbandingan Algoritma C4.5 dengan Naive Bayes untuk Data Miing pada Data Log Intrusion Detection System (IDS)," STMIK Bumigora Mataram, 2016.

[5] A. Balon-perin, "Ensemble-based methods for intrusion detection," Norwegian University of Science and Technologi, 2012.

[6] Kusrini and E. T. Luthfi, *Algoritma Data Mining*. Yogyakarta: Andi Offset, 2009.

[7] L. Dhanabal and S. P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[8] S. Mashlahah, "Prediksi Kelulusan Mahasiswa Menggunakan Metode Decision Tree dengan Penerapan Algoritma C4.5," Universitas Islam Negeri Maulana Malik Ibrahim Malang, 2013.

[9] E. Prasetyo, *Data Mining Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: Andi Offset, 2012.

[10] D. Aprilla, D. A. Baskoro, L. Ambarwati, and I. W. S. Wicaksana, *Belajar Data Mining Dengan RapidMiner*. Jakarta, 2013.