Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202294107
Vol. 9, No. 2, April 2022, hlm. 245-254 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENGUJIAN *RULE-BASED* PADA *DATASET LOG SERVER* MENGGUNAKAN *SUPPORT VECTOR MACHINE BERBASIS LINEAR DISCRIMINAT ANALYSIS* UNTUK DETEKSI *MALICIOUS ACTIVITY*

Kurnia Adi Cahyanto[*1], Muhammad Anis Al Hilmi[2], Muhamad Mustamiin[3]

[1,2,3]Politeknik Negeri Indramayu, Kabupaten Indramayu
Email: [1]kurnia@polindra.ac.id, [2]alhilmi@polindra.ac.id, [3]mustamiin@polindra.ac.id
*Penulis Korespondensi

(Naskah masuk: 18 September 2020, diterima untuk diterbitkan: 16 Februari 2022)

Abstrak

Web server bertugas menjalankan aplikasi *web* untuk melayani *request* dari klien. Setiap interaksi yang dilakukan klien terhadap aplikasi *web*, tercatat pada catatan *log server*. Dari *log* tersebut, terdapat data detail tentang alamat IP, perangkat dan sumber klien, *request* pengguna, respon *server*, dan keterangan lainnya. Dari informasi pada *log*, dapat dipakai untuk keperluan pengamanan sistem, salah satunya dengan cara melakukan analisis menggunakan *data mining* terhadap aktifitas klien yang tercatat pada *log server*. Selain itu, jika terdapat *file* yang diunggah pengguna, dapat juga dikaitkan dalam analisis *log server* dalam mengenali pola aktifitas dan *malicious file*. Dataset log yang telah didapat, diolah dengan menggunakan pelabelan *rule-based* yang nantinya diuji dengan pemodelan *Support Vector Machine* berbasis *Linear Discriminant Analysis*. Proses mengklasifikasikan data *log server* dapat dilakukan untuk mengenali aktifitas yang termasuk serangan, usaha paksa untuk masuk sistem terhadap server atau bukan. Dari pemodelan yang dilakukan, didapat hasil bahwa algoritma SVM berbasis LDA memiliki tingkat akurasi *training* 90,2% dan *testing* 89,9% dalam melakukan pengujian *rule-based* untuk pelabelan aktifitas pada *web server*.

Kata kunci: *Log server, rule-based, SVM, LDA, malicious.*

Rule-Based Testing on Server Log Dataset Using Support Vector Machine-Radial Basis Function to Detect Malicious Activity

Abstract

The web server is in charge of running web applications to serve requests from clients. Every interaction the client makes to the web application is logged in server logs. From these logs, there are detailed data about IP addresses, client devices and sources, user requests, server responses, and other information. From the information in the logs, it can be used for system security purposes, one of which is by performing analysis using data mining of client activities recorded on the server log. In addition, if there is a file uploaded by a user, it can also be linked in server log analysis in recognizing activity patterns and malicious files. The log dataset that has been obtained is processed using rule-based labeling which will later be tested with a Linear Discriminant Analysis-based Support Vector Machine modeling. The process of classifying server log data can be done to identify activities that include attacks, forced attempts to enter the system against the server or not. From the modeling, the results show that the LDA-based SVM algorithm has a training accuracy rate of 90,2% and testing 89,9% in performing rule-based testing for activity labeling on the web server.

Keywords: *Log server, rule-based, SVM, LDA, malicious.*

1. PENDAHULUAN

Kondisi lalu lintas data di dunia maya semakin terbuka, banyak pengguna yang masih belum siap menggunakan jaringan internet dengan baik dan aman. Keamanan *cyber* menjadi perhatian penting di tengah meluasnya penggunaan internet dan pemanfaatan *website*, apalagi sejak 2017 Kominfo menggalakan program 100 *smart city* di kota dan kabupaten seluruh Indonesia (HILMI, 2019). Banyak data dan informasi penting yang bersifat privat milik pengguna tidak diamankan dengan baik. Keamanan tersebut bukan hanya menjadi tanggung jawab pemilik data semata, adakalanya yang membuat data tersebut menjadi tidak aman adalah pengelola *website* yang tidak menerapkan keamanan yang memadai demi menjaga privasi pengguna web-nya.

Banyak jalan yang dapat dieksploitasi oleh para pengguna internet yang tidak bertanggung jawab. Pengguna internet yang memanfaatkan celah keamanan demi mencuri data atau mengambil keuntungan dari hasil *hack* sistem web atau sering disebut peretas memanfaatkan ketidaksiapan dari pengelola web maupun pengguna web itu sendiri. Salah satu celah yang sering dimanfaatkan oleh peretas adalah sarana upload file di *website*. Secara umum, setiap aktivitas penggunaan *website* akan terekam pada *log server*. Hal ini dapat dimanfaatkan untuk mengenali adanya aktivitas mencurigakan atau berpotensi mengganggu.

Salah satu jenis *server* yang sering digunakan sebagai *web server* adalah Ubuntu *server*, di mana di dalamnya dapat dipasang berbagai layanan yang dapat digunakan dalam pengelolaan *website*. Pada *web server* yang ada dapat dilakukan penggalian data terkait segala akses yang dilakukan oleh pengguna web, yang mungkin bisa jadi ada pengguna web yang melukan kegiatan yang tidak wajar demi tujuan yang tidak baik. Penelitian bertujuan untuk membangun sistem deteksi intrusi (IDS) dengan kemampuan membuat sebuah model secara otomatis dan dapat mendeteksi intrusi dengan menggunakan beberapa *data mining* (DM) untuk mengklasifikasikan audit data lalu lintas akses. Data audit diambil dari *log server* dan *file backdoor*.

Menurut (MBUGUA, 2016), menganalisis *log server* untuk mendeteksi aktivitas yang mencurigakan dianggap sebagai bentuk pertahanan utama (*main defence*), sehingga perlu dilakukan klasifikasi untuk melihat pola-pola serangan tersebut. Namun, dengan ukuran *log server* yang sangat besar membuat analisis ini menjadi sulit. Selain itu, sistem deteksi intrusi tradisional bergantung pada metode berdasarkan teknik pencocokan pola yang tidak dapat dilakukan pengembang tidak dapat mempertahankan berdasarkan tingkat tinggi di mana teknik serangan baru dan belum pernah terlihat sebelumnya diluncurkan setiap hari. Tujuan dari proyek ini adalah untuk mengembangkan sistem deteksi intrusi berbasis log cerdas yang dapat mendeteksi intrusi yang dikenal dan tidak dikenal secara otomatis. Pelatihannya menggunakan algoritma *unsupervised learning*, yaitu *K-Means* untuk melihat sebaran datanya dan menggunakan algoritma SVM *One-Class* untuk mengklasifikasikan atau membuat modelnya, dengan catatan pengembangan sistem dibatasi waktu dan terbatas pada *log* yang dihasilkan mesin karena kurangnya file *access.log* yang sebenarnya. Namun, pengembangan sistem berjalan lancar dan terbukti hingga 85% akurat dalam mendeteksi pola log yang tidak wajar dalam log pengujian.

Sedangkan menurut (V.VIDAPRIYA, 2016), penambangan web adalah integrasi dari banyak informasi yang dikumpulkan oleh teknik dan metodologi penambangan data yang digunakan untuk mengekstrak informasi dari data web mengikuti alur dari data mining. Tiga kategori yang saling berkorelasi dalam penambangan web, yaitu konten web, struktur web dan penggunaan web. Fase yang dilakukan adalah praproses data, penemuan pola dan analisis pola dengan menggunakan klasifikasi seperti *Naïve Bayes* (NB), *Classfication and Regression Tree* (CART), *k-Nearest Neighbor* (k-NN) yang memiliki akurasi maksimum dan *error* yang minimum. Tujuan utama dari makalah ini adalah untuk mengidentifikasi minat pola akses pengguna dari *web log* yang mendefinisikan situs web tertentu. Hasilnya, k-NN memiliki akurasi tertinggi diantara ketiga algoritma tersebut.

Namun, dengan ukuran *log server* yang sangat besar membuat analisis ini menjadi sulit. Selain itu, sistem deteksi intrusi tradisional bergantung pada metode berdasarkan teknik pencocokan pola yang tidak dapat dilakukan pengembang tidak dapat dipertahankan berdasarkan tingkat tinggi di mana teknik serangan baru dan belum pernah terlihat sebelumnya diluncurkan setiap hari. Tujuan dari proyek ini adalah untuk mengembangkan sistem deteksi intrusi berbasis log cerdas yang dapat mendeteksi intrusi yang dikenal dan tidak dikenal secara otomatis.

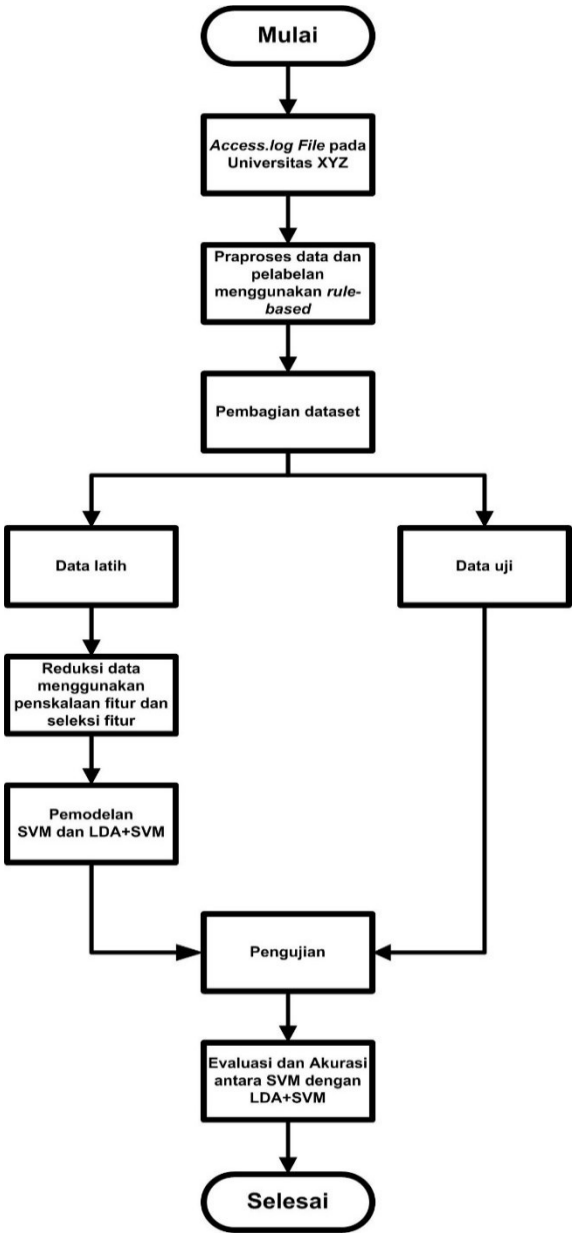
Berdasarkan latar belakang di atas, kami mengusulkan perlu dilakukan analisis pelabelan menggunakan metode berbasis *rule-based* untuk mendeteksi aktivitas *malicious* ini dan yang nantinya diuji akurasinya dengan metode *data mining*. Algoritma yang digunakan adalah *Support Vector Machine* (SVM) dengan *kernel trick Radial Basis Function* (RBF) dan *Linear Discriminant Analysis* (LDA) untuk seleksi fiturnya.

Penelitian ini ditujukan untuk mengelompokkan jenis akeses sesuai dengan kategori yang telah ditetapkan menggunakan *rule-based*, pada penelitian ini penulis menggunakan tiga kategori yaitu, aman, dicurigai dan bahaya. Pemilihan metode SVM dikarenakan menurut (A.MUIS & AFFANDES, 2015) menyatakan bahwa berdasarkan penelitian yang telah dia lakukan berpendapat bahwa SVM adalah klasifikasi yang paling tepat untuk melakukan pengklasifikasian teks. Dengan kata lain, SVM merupakan metode yang juga dapat diterapkan untuk mengklasifikasi teks pada variabel *request* dengan tingkat akurasi relatif lebih baik dibanding metode lainnya. Selain itu kami juga ingin mengetahui bagaimana pengaruh LDA terhadap kecepatan komputasi dan hasil akurasi pada SVM.

2. METODE PENELITIAN

2.1 Metode yang Diusulkan

Pada penelitian ini diusulkan sebuah metode SVM berbasis seleksi fitur LDA yang secara khusus diharapkan lebih baik daripada metode SVM yang tanpa seleksi fitur, sedangkan secara umum mampu mendeskripsikan akurasi dari *log server* yang telah dilabeli menggunakan metode *rule-based*.



Gambar 1. Rancangan Metode yang Diusulkan

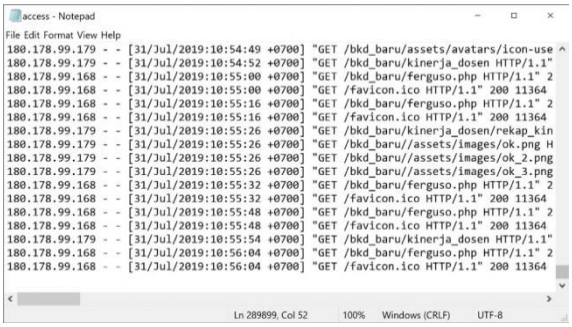
Secara umum, dari algoritma di atas proses dimulai dengan mengumpulkan data yang *access.log* pada sebuah universitas XYZ dari admin *server*-nya setelah juga melakukan serangkaian observasi maupun wawancara. Berikutnya dilakukan pelabelan dengan menggunakan *rule-based* yang akan dijelaskan pada poin berikutnya. Selanjutnya dataset ini akan dibagi menjadi dua, yaitu sebagai data latih dan data uji dengan komposisi 70% untuk data latih dan 30% sebagai data uji. Data yang akan dilatih ini selanjutnya akan direduksi dengan ekstraksi fitur menggunakan LDA yang nantinya hanya akan dilihat atribut yang benar-benar signifikan dengan tujuan untuk mengurangi lamanya proses komputasi. Proses selanjutnya data akn dilatih menggunakan SVM kemudian diuji hasilnya dengan menggunakan data uji untuk kemudian dievaluasi.

2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini diambil dari sebuah kampus XYZ dan wawancara dengan staf sistem administrator. Data yang digunakan adalah data *log server file* yang mencatat informasi setiap kali pengguna meminta sumber daya dari situs web. Data yang didapat berbentuk *file* yang sudah dikompres berisi *file access.log* bulan Januari 2019 sampai dengan Juli 2019 dan contoh file *webshell* yang berpotensi sebagai serangan terhadap web yang dimaksud. Oleh sistem administrator, sampel dari file *webshell* yang berpotensi sebagai serangan ini disimpan sebagai acuan dalam mendeteksi adanya kemungkinan status berbahaya.

2.3 Praproses Data

Data *access.log* di bawah ini berisi dataset sejumlah 289899 baris dan 52 kolom. Didapat setelah melakukan observasi dan wawancara terhadap admin *server* di sebuah universitas XYZ.



Gambar 2. File *access.log*

Kemudian data ini diformat dari *text* menjadi *spreadsheet* supaya kolom antar variabelnya menjadi jelas.

2.4 *Cleaning* Data

Kemudian data yang dihapus adalah data yang diakses dari server lokal dengan IP address dan data yang mengandung file *cron job*. File yang berfungsi sebagai *cron job* adalah yang menggunakan nama "*ferguso.php*". Karena kedua data ini diakses dan dibuat sendiri oleh sistem administratornya, sehingga tidak termasuk ke dalam data yang akan diteliti. Setelah melalui proses penghapusan ini, data yang tersisa berjumlah 37693 baris.

2.5 Pelabelan Kelas Data

Pemberian nama pada atribut dan pelabelan pada kelas data merupakan salah satu hal yang penting dalam praproses, karena data dalam *access.log* terdapat informasi tentang siapa yang mengunjungi, dari mana mereka berasal dan apa yang mereka lakukan dengan server web (MBUGUA, 2016), namun masih berupa konten dan belum memiliki nama, oleh karena itu perlu diberikan nama atribut, yaitu *IP Address, User ID, Timestamp, Request, Status, Size, Referer* dan *User Agent*.

Kemudian data yang tersisa dilabeli dengan menggunakan sistem *rule based labelling* dengan mengamati file-file tertentu yang memiliki jejak ke file *webshell* dan IP address yang mengakses website ini dari luar negeri Indonesia. Dari *rule-based* ini menghasilkan pelabelan sebagai berikut:

1. Dianggap aman, jika lokasi pengaksesan dari Indonesia dan IP address pengakses tidak memiliki jejak ke *webshell*.
2. Dianggap dicurigai, jika lokasi pengaksesan dari luar negeri.
3. Dianggap bahaya, IP address memiliki jejak mengakses file *webshell*.

Akhirnya aturan di atas ini menghasilkan tabel seperti di bawah ini.

Tabel 1. Dataset *log server* yang Telah Dilabeli dengan metode *Rule-Based*

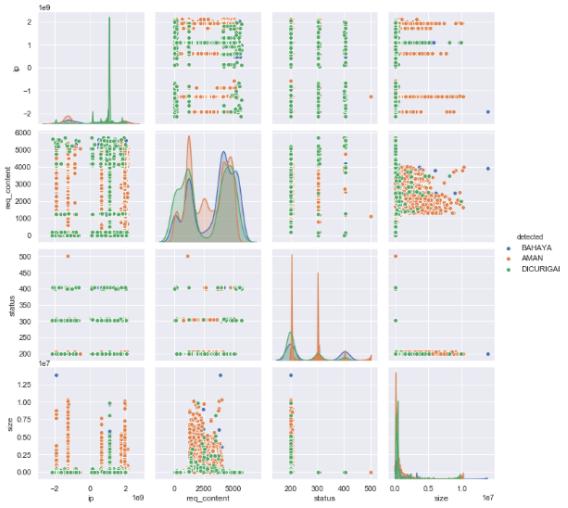
IP	114.125.221.132
Datetime	01-Jul 2019 10.55.38
GMT	+0700]
Request	POST /bkd_baru/awubahon.php HTTP/1.1
Status	200
Size	12421
Referrer	http://xyz.ac.id/bkd_baru/awubahon.php
Browser	Mozilla/5.0 (Linux; Android 5.1.1; SM-J111F Build/LMY47V)
Country	AppleWebKit/537.36 (KHTML, like
Detected	Gecko) Chrome/55.0.2883.91 Mobile Safari/537.36 Indonesia BAHAYA

IP	114.125.207.5
Datetime	01-Jul 2019 10.55.38
GMT	+0700]
Request	GET / HTTP/1.1
Status	302
Size	-
Referrer	-
Browser	Mozilla/5.0 (Linux; Android 5.1.1; SM-J111F Build/LMY47V)
Country	AppleWebKit/537.36 (KHTML, like
Detected	Gecko) Chrome/55.0.2883.91 Mobile Safari/537.36 Indonesia AMAN

Ada tiga buah kategori kelas atau pelabelan, yaitu: aman (warna hijau), dicurigai (warna kuning) dan bahaya (warna merah). Sebagai contoh, dengan menggunakan filter pada Ms. Excel, maka file *backdoor awubahon.php* akan tampil hasilnya, lalu dilabeli sebagai kelas bahaya, lalu saya filter lagi IP *address* yang terdeteksi pernah mengakses file *webshell* tersebut, dan dilabeli sebagai bahaya lagi.

2.6 Visualisasi Data

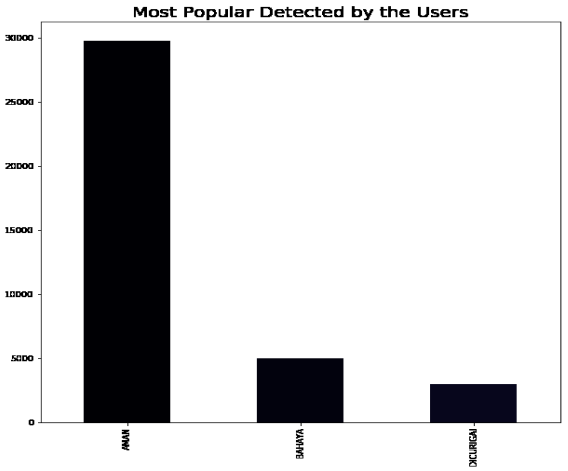
Data mentah selanjutnya akan dianalisis dengan tujuan memilih metode untuk model yang paling tepat, dengan harapan dari data tersebut kita sudah dapat melihat gambaran umum dari data yang akan diolah. Oleh karena itu diperlukan visualisasi data awal dalam rangka untuk melihat sebaran data dan karakter datanya, sehingga diharapkan dengan menganalisis data yang ada di dalamnya dapat digunakan untuk mengambil model yang tepat dalam mendeteksi intrusi.



Gambar 3. Sebaran Data Awal

Dari gambar di atas dapat dilihat bahwa terdapat titik yang berwarna biru yang terdeteksi sebagai "bahaya" memiliki kecenderungan posisi yang lebih jauh dari titik-titik yang lain.

Variabel awal yang akan digunakan adalah *IP adress user, HTTP request, status, size* dan *detected*. Tiap-tiap atribut tersebut kemudian divisualisasikan menggunakan grafik diagram batang, sehingga lebih mudah untuk dipahami.



Gambar 4. *Rule Based* untuk Pelabelan Kelas

2.7 Transformasi Data

Transformasi dilakukan dalam rangka untuk mengubah data menjadi bentuk yang sesuai untuk penambangan sekaligus langkah sebelum melakukan pemodelan (MAIMON, 2010). Sedangkah poin-poin transformasi yang dilakukan adalah sebagai berikut

1. Pada fitur yang berisi *HTTP request* dapat dipisah menjadi *req_method* (metode request),

req_content (isi request), *req_version* (versi request). Kemudian melakukan penanganan data yang hilang atau kosong dengan merubah nilainya menjadi nol pada atribut *status*, *size*, *req_method*, *req_content*, *req_version* dan menghapus baris yang terdapat data *NaN*.

2. Untuk IP address dikonversi menjadi bernilai *integer* supaya lebih mudah untuk diolah dengan model yang memang memerlukan jenis bukan *string*.
3. Untuk atribut *HTTP request* di-encode supaya bernilai *integer*.
4. Kelas data dikonversi menjadi numerik, yaitu: AMAN = 0, DICURIGAI = 1, BAHAYA = 2.
5. Hasil konversi IP address diabsolutkan nilanya untuk menghindari nilai yang bersifat negatif.

Hasil dari transformasi tersebut dapat dilihat pada gambar di bawah ini.

ip	datetime	eq_metho	eq_conten	eq_versio	status	size	detected
1,9E+09	2019-07-01 10.54.15	GET	/bkd_baru	HTTP/1.1	200	12133	BAHAYA
1,9E+09	2019-07-01 10.54.23	GET	/bkd_baru	HTTP/1.1	200	15491	BAHAYA
1,9E+09	2019-07-01 10.54.42	POST	/bkd_baru	HTTP/1.1	200	16305	BAHAYA
1,9E+09	2019-07-01 10.55.08	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,9E+09	2019-07-01 10.55.28	GET	/bkd_baru	HTTP/1.1	200	1735	BAHAYA
1,9E+09	2019-07-01 10.55.38	POST	/bkd_baru	HTTP/1.1	200	12421	BAHAYA
1,9E+09	2019-07-01 10.55.57	GET	/	HTTP/1.1	302	0	AMAN
1,9E+09	2019-07-01 10.55.57	GET	/bkd_baru	HTTP/1.1	301	360	AMAN
1,9E+09	2019-07-01 10.55.57	GET	/bkd_baru	HTTP/1.1	200	2416	AMAN
1,9E+09	2019-07-17 21.11.54	GET	/bkd_baru	HTTP/1.1	200	31568	AMAN
1,9E+09	2019-07-17 21.11.55	GET	/bkd_baru	HTTP/1.1	200	317017	AMAN
1,9E+09	2019-07-17 21.11.55	GET	/bkd_baru	HTTP/1.1	404	1130	DICURIGAI
1,9E+09	2019-07-17 21.12.03	GET	/bkd_baru	HTTP/1.1	200	25074	AMAN
1,9E+09	2019-07-17 21.12.03	GET	/bkd_baru	HTTP/1.1	200	283099	AMAN
1,9E+09	2019-07-17 21.12.03	GET	/bkd_baru	HTTP/1.1	404	1130	DICURIGAI
1,1E+09	2019-07-17 21.12.03	GET	/bkd_baru	HTTP/1.1	200	38520	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	200	10965	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,1E+09	2019-07-17 21.12.04	GET	/bkd_baru	HTTP/1.1	404	1130	BAHAYA
1,9E+09	2019-07-17 21.12.15	GET	/bkd_baru	HTTP/1.1	200	28106	AMAN
1,9E+09	2019-07-17 21.12.15	GET	/bkd_baru	HTTP/1.1	404	1130	DICURIGAI

Gambar 5. Data setelah melalui Praproses

Setelah melalui transformasi ini, variabel akhir yang akan digunakan adalah *IP adress user, datetime, req_method*, *req_content*, *req_version, status, size* dan *detected*.

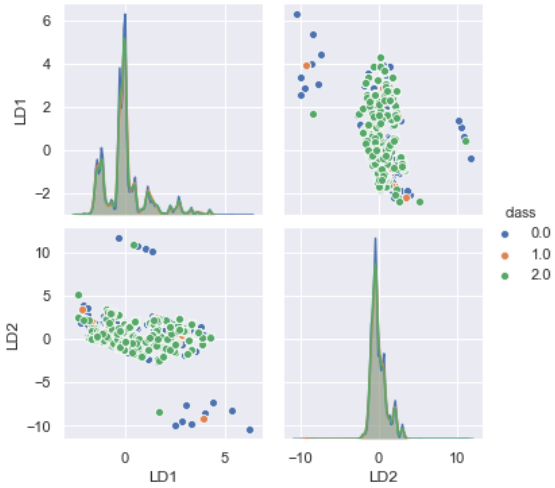
2.8 Pembagian Data

Setelah data melalui praproses, kemudian dataset dibagi menjadi data *training* dan data *testing* dengan rasio data training 70% dan data testing 30%, sehingga formatnya menjadi 26065 buah dataset sebagai data *training* dan 11171 buah dataset sebagai data *testing*.

2.9 Seleksi Fitur menggunakan LDA

Seleksi fitur menggunakan LDA supaya fitur data terpisah menjadi dua dimensi saja sehingga membantu mempercepat pemodelan karena hanya menggunakan fitur yang memiliki korelasi terbaik saja. *Linear Discriminant Analysis* (LDA) merupakan salah satu metode *supervised learning* dalam hal reduksi dimensi data untuk masalah klasifikasi (ALPAYDIN, 2010). Setelah direduksi, sebaran data untuk kelasnya menjadi lebih sedikit dan jelas.

Melihat visualisasi pada gambar 6, terlihat bahwa beberapa data berwarna biru yang mewakili BAHAYA sebagai sebuah *outlier* atau semacam anomali dari kumpulan data tersebut. Selain itu, data yang berwarna hijau (DICURIGAI) dan kuning (AMAN) lebih terlihat mengelompok menandakan mereka memiliki banyak kemiripan dari setiap fiturnya.



Gambar 6. Sebaran Data Hasil Reduksi oleh LDA

Reduksi yang dilakukan oleh LDA mmenjadikan visualisasi lebih simpel dan mudah dipahami namun tidak mengurangi esensi dari keberadaan data tersebut.

2.10 Validasi dan Evaluasi Sistem

Perlu dilakukan tahapan untuk memvalidasi dan mengevaluasi model yang telah diusulkan suaya dapat diketahui kesesuaian dengan yang diharapkan. Validasi model ini dengan menggunakan *cross validation* (CV).

Dalam *k-fold cross-validation*, data awal secara acak dipartisi menjadi k subset atau "*folds*" yang saling eksklusif, D1, D2,…, Dk, masing-masing berukuran kira-kira sama. Pelatihan dan pengujian dilakukan k-kali (HAN, 2011). Dalam iterasi i, partisi Dk dicadangkan sebagai set pengujian, dan partisi yang tersisa digunakan secara kolektif untuk melatih model.

Disini kami menggunakan *k-fold* CV = 10, karena CV ini adalah salah satu *k-fold* CV yang direkomendasikan untuk pemilihan model terbaik karena cenderung memberikan estimasi akurasi yang kurang bias dibandingkan dengan CV biasa, leave-one-out CV dan bootstrap (SIHOMBING & HENDARSIN, 2020). Setelah didapatkan nilai akurasi yang diharapkan, maka dilakukan evaluasi dengan membandingkan tingkat akurasi antara SVM dengan LDA+SVM.

3. STUDI LITERATUR

Digunakan sumber pustaka yang relevan untuk mengumpulkan informasi yang diperlukan dalam penelitian ini (CAHYANTO, MULYANI, & MUHAMAD, 2019), yaitu penelitian terkait dan sumber pustaka yang berupa buku, jurnal, prosiding seminar nasional, skripsi maupun tesis.

3.1 *Rule-Based*

Sistem berbasis aturan (*rule-based*) adalah jenis khusus dari sistem pakar yang terdiri dari sekumpulan aturan. Dalam praktiknya, sistem berbasis aturan dapat dibangun dengan menggunakan pengetahuan ahli atau belajar dari data nyata (LIU & GEGOV, 2016). Karena ukuran data yang luas dan terus meningkat dengan istilah *big data*, maka pendekatan yang terakhir menjadi sangat populer untuk membangun sistem berbasis aturan.

3.2 Log Server

File *log server* web adalah file teks biasa sederhana yang mencatat informasi setiap kali pengguna meminta sumber daya dari situs web (SALAMA, 2011). File ini dibuka saat layanan web server dimulai dan tetap terbuka saat server merespons permintaan pengguna.

File *log server* web ini memberi administrator web banyak jenis informasi yang berguna seperti:
- Halaman mana dari situs web Anda yang diminta
- Kesalahan apa yang ditemui orang
- Apa status yang dikembalikan oleh server atas permintaan pengguna
- Berapa banyak paket data yang dikirim dari server ke pengguna

Umumnya ada empat jenis log server, yaitu:
1. *Access log file*
2. *Error log file*
3. *Agent log file*
4. *Referrer log file*

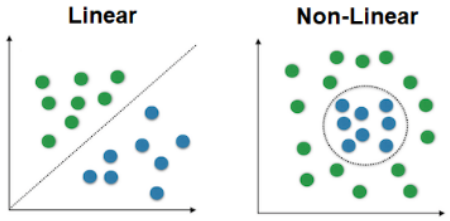
Pada penelitian ini, kami menggunakan file *access log file*.

3.3 SVM-RBF

Teknik SVM diperkenalkan oleh Vapnik (APOSTOLIDIS-AFENTOULIS & LIOUFI, 2015) dan berkembang pesat dalam beberapa tahun terakhir. Beberapa penelitian melaporkan bahwa SVM, secara umum, mampu memberikan akurasi klasifikasi yang lebih tinggi daripada algoritma klasifikasi lain yang ada.

Sebagai salah satu algoritma klasifikasi yang sering digunakan, SVM bekerja dengan cara mencari sebuah *hyperplane* atau garis pembatas pemisah antar kelas yang memiliki margin atau jarak antar *support vector* (SV) dengan data paling terdekat pada setiap kelas yang paling besar (SOMANTRI & APRILIANI, 2018).

Prinsipnya SVM bekerja secara linear, dan dikembangkan untuk dapat diterapkan pada masalah non-linear. Dengan menggunakan metode *kernel trick* yang mencari *hyperplane* dengan cara mentransformasi dataset ke ruang vektor yang berdimensi lebih tinggi (*feature space*), kemudian proses klasifikasi dilakukan pada *feature space tersebut*. Penentuan fungsi kernel yang digunakan akan sangat berpengaruh terhadap hasil prediksi. Misalkan {x1,.....xn} 1 adalah dataset dan yi ∈ {+1, −1} adalah kelas dari data xi .. Pada gambar 1 dapat dilihat berbagai alternatif bidang pemisah yang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar. Data yang berada tepat pada bidang pemisah disebut sebagai *support vector* (SV).



Gambar 7. Data Linear dan Non-Linear

Gambar di atas menunjukkan perbedaan antara SVM yang menghasilkan garis pemisah untuk yang data linera dengan data yang non-linear.

Berdasarkan data yang digunakan terdapat data berupa teks maka proses SVM yang digunakan akan proses non-linear. Salah satu dari *kernel trick* pada SVM non-linear ini adalah *Radial Basis Function* (RBF).

Secara linear, SVM menggunakan formulasi sebagai berikut:

$$f(x)\frac{+1, w*x_i+b\geq+1}{-1, w*x_i+b\geq-1} \quad (1)$$

Sedangkan untuk *kernel trick* untuk non-linearnya menggunakan RBF dengan paramater C dan gamma.

$$K(X_1, X_2) = eksponen(-\gamma\|X_1, X_2\|^2) \quad (2)$$

3.4 LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) adalah adalah cara lain untuk menemukan transformasi linier data yang mengurangi jumlah dimensi yang diperlukan untuk merepresentasikannya (WITTEN, 2017). Ini sering digunakan untuk pengurangan dimensi sebelum klasifikasi, tetapi juga dapat digunakan sebagai teknik klasifikasi itu sendiri. Tidak seperti analisis komponen utama dan independen, analisis ini menggunakan data berlabel, sehingga dimungkinkan untuk dikombinasikan dengan algoritma *supervised learning* lainnya.

$$y_c = x^T\Sigma^{-1}\mu_c - \frac{1}{2\mu_c^T\Sigma^{-1}\mu_c} + \log(\frac{n_c}{n}) \quad (3)$$

Dimana n_c adalah jumlah kelas dari c dan n jumlah total dari sampel data.

3.5 EVALUASI MODEL

Evaluasi model akan dihitung menggunakan akurasi data sebagai berikut:

$$akurasi = \frac{jumlah\ klasifikasi\ benar}{jumlah\ klasifikasi\ keseluruhan} x100\% \quad (4)$$

4. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan komputer dengan spesifikasi *hardware* prosesor Core i7 dan memori 16 Gb, *software* untuk SO Windows 10 64 Bit, dan untuk *tools* untuk proses DM-nya menggunakan *Anaconda3-2020.02*.

4.1. Hasil Evaluasi dan Akurasi

Dataset yang telah diperoleh dan telah siap digunakan untuk pemodelan dengan variabel bebas (X) yang akan digunakan yaitu *ip, req_method, req_content, req_version, status,* dan *size* dan variabel tak bebasnya (y) adalah *detected*.

Kemudian SVM yang digunakan dikonfigurasikan dengan parameter sebagai berikut:

- *Kernel trick = Radial Basis Function;*
- *Cache size* = 200;
- C = 1;
- γ = auto;
- degree = 3;

Sedangkan LDA yang akan digunakan dikonfigurasi dengan parameter *n_component* = 2 yang bertujuan mereduksi dataset untuk diseleksi fiturnya sehingga menghasilkan dimensi yang lebih rendah.

Untuk kategori kelas ada 3, yaitu, 0 = AMAN, 1 = DICURIGAI, 2 = BAHAYA

1. Pemodelan menggunakan SVM-RBF
   - Waktu untuk melakukan komputasi 324.8567113876343 detik
   - Waktu untuk melakukan validasi 4935.598784685135 detik

Tabel 2. Hasil Pemodelan SVM-RBF

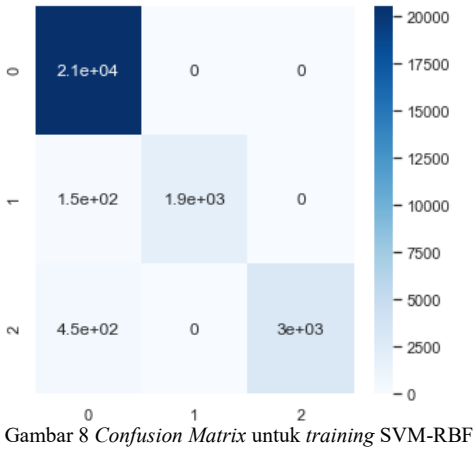
	Accuracy (%)	*Precision* (%)			*Recall* (%)		
		0	1	2	0	1	2
Training	97,72	97	100	100	100	93	87
Testing	79,97	80	100	100	100	3	8
Validasi	80,08						

Sedangkan jika dibuat diagram menggunakan *confusion matrix* (CM) adalah seperti di bawah ini.

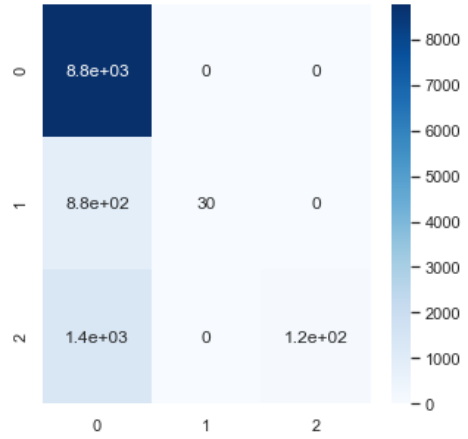
Dari tabel dan gambar di atas akurasi untuk *training* sebesar 97,72% dan akurasi *testing* sebesar 79,97%.

Untuk pemodelan menggunakan SVM-RBF, terlihat bahwa akurasi training dan testing sangat timpang dan juga durasi komputasi memakan waktu yang sangat lama.

Training berikutnya akan ditambahkan LDA sebagai fungsi optimasi dengan tujuan untuk memperbaiki hasil akurasi dan mempercepat proses komputasi.



Gambar 8 *Confusion Matrix* untuk *training* SVM-RBF

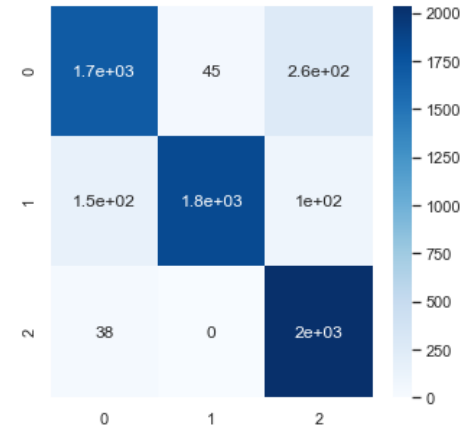


Gambar 9. *Confusion Matrix* untuk *testing* SVM-RBF

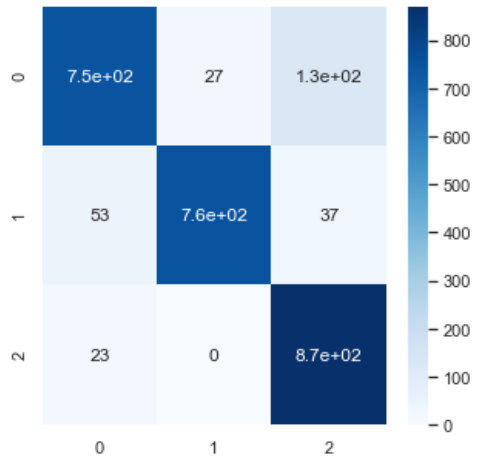
2. Pemodelan menggunakan LDA+SVM-RBF
   - Waktu untuk melakukan komputasi 1.3290517330169678 detik
   - Waktu untuk melakukan validasi 4.3212831020355225 detik

Tabel 3. Hasil Pemodelan LDA+SVM-RBF

	Accuracy (%)	*Precision* (%)			*Recall* (%)		
		0	1	2	0	1	2
Training	90,26	90	98	85	85	88	98
Testing	89,95	91	97	84	83	89	97
Validasi	89,88						



Gambar 10. *Confusion Matrix* untuk *training* LDA+SVM-RBF



Gambar 11. *Confusion Matrix* untuk *testing* LDA+SVM-RBF

Dari tabel dan gambar di atas akurasi untuk *training* sebesar 90,26% dan akurasi *testing* sebesar 89,95%.

Dari hasil evaluasi di atas maka dapat dibuat tabel ringkasan *f1-score* sebagai berikut:

Tabel 4. *F1-Score* untuk *Training* dan *Testing*

Model Algoritma	*Training* (%)				*Testing* (%)			
	Acc	*F1-score*			*Acc*	*F1-score*		
		0	1	2		0	1	2
SVM-RBF	98	99	96	93	78	89	6	15
LDA+SVM-RBF	90	87	92	91	90	87	93	90
Rata-rata	98				84			

5. KESIMPULAN

Metode pengujian *rule-based* menggunakan algoritma SVM-RBF dan LDA+SVM-RBF yang telah diterapkan untuk melabeli setiap aktifitas yang masuk pada *web server* memiliki tingkat akurasi rata-rata untuk *training* sebesar 98% dan untuk *testing* sebesar 84%. Sedangkan pemodelan menggunakan LDA+SVM-RBF mampu meningkatkan akurasi dari SVM-RBF untuk *testing* sebesar 10%. Untuk pemodelan menggunakan LDA+SVM-RBF, terlihat bahwa waktu komputasi jauh lebih cepat daripada SVM-RBF. Selain itu, akurasi *training, validasi,* dan *testing* juga tidak jauh berbeda dan cenderung stabil. Pada penelitian selanjutnya disarankan untuk menggunakan algoritma yang lebih efektif dan efisien dalam menemukan pemodelan yang lebih baik lagi untuk meningkatkan akurasi dalam mendeteksi *malicious activity*.

UCAPAN TERIMAKASIH

Terimakasih kami ucapkan kepada DRPM Direktorat Jenderal Penguatan Riset dan Pengembangan Kemenristek Dikti atas pendanaan untuk penelitian yang diberikan melalui skema pendanaan Penelitian Dosen Pemula (PDP) tahun anggaran 2020.

DAFTAR PUSTAKA

A.MUIS, I., & AFFANDES, M. 2015. Penerapan Metode Support Vector Machine (SVM) Menggunakan Kernel Radial Basis Function (RBF) Pada Klasifikasi Tweet. *Jurnal Sains, Teknologi dan Industri*, 189-197.

ALPAYDIN, E. 2010. *Introduction to Machine Learning Second Edition.* London: The MIT Press.

APOSTOLIDIS-AFENTOULIS, V., & LIOUFI, K.-I. 2015. *Svm Classification With Linear & Rbf Kernels.* Thessaloniki: University of Macedonia.

CAHYANTO, K. A., MULYANI, E., & MUHAMAD, F. P. 2019. Penerapan Dizcretize By Frequency Dalam Meningkatkan Akurasi Algoritma C4.5 Dalam Memprediksi Cuaca Pada Jalur Pantura Tegal-Pekalongan-Semarang. *Jurnal Tenologi Terapan (JTT), 5*, 78-85.

HAN, J. 2011. *Data Mining Concepts and Techniques.* Massachusetts: Morgan Kaufmann Publishers.

HILMI, M. A. 2019. Uji Performa dan Website Responsiveness Institusi dan Smart City se-Jawa Barat. *Sentrinov.* arXiv preprint arXiv:1912.13346.

LIU, H., & GEGOV, A. 2016. Rule Based Systems and Networks: Deterministic and Fuzzy Approaches. *International Conference on Intelligent Systems*, (hal. 316-321).

MAIMON, O. 2010. *Data Mining and Knowledge Discovery Handbook Second Edition.* London: Springer.

MBUGUA, J. G. 2016. *Automated Log Analysis Using Ai: Intelligent Intrusion Detection System.* Bondo: Jaramogi Oginga Odinga University Of Science And Technology.

SALAMA, S. E. 2011. Web Server Logs Preprocessing for Web Intrusion Detection. *Computer and Information Science, 4*, 123-133.

SIHOMBING, P. R., & HENDARSIN, O. P. 2020. Perbandingan Metode Artificial Neural Network (ANN) dan Support Vector Machine (SVM) untuk Klasifikasi Kinerja Perusahaan Daerah Air Minum (PDAM) di Indonesia. *Jurnal Ilmu Komputer VOL. XIII No. 1*, 9-20.

SOMANTRI, O., & APRILIANI, D. 2018. Support Vector Machineberbasis Feature Selection Untuksentiment Analysiskepuasan Pelanggan Terhadap Pelayanan Warung Dan Restoran Kuliner Kota Tegal. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 537-547.

V.VIDAPRIYA. 2016. Identifying Web Users from Weblogs Using Classification Algorithms. *International Journal of Innovative Research in Computer and Communication Engineering, 4*(7), 13722-13728.

WITTEN, I. H. 2017. *Data Mining Practical Machine Learning Tools and Techniques.* Cambridge: Elsevier.

Halaman ini sengaja dikosongkan