DETEKSI SERANGAN *REMOTE CODE EXECUTION* DAN *CROSS SITE SCRIPTING* MENGGUNAKAN *MACHINE LEARNING*

Hartono[1], Khusnul Khotimah[2], Adi Wibowo[3]

[123]Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Kotabumi
e-mail: hartono@umko.ac.id[1], khusnul.khotimah@umko.ac.id[2], adi.wibowo@umko.ac.id[3]

ABSTRACT

With the increasing connectivity and rapid growth of information technology, Indonesia is confronted with formidable challenges in the domain of cybersecurity. The incidence of cyber attacks has reached an alarming threshold, approaching nearly one billion occurrences throughout the year 2022. This not only serves as a disconcerting indicator of the intensification of cyber threats but also underscores the imperative for efficacious solutions to detect and mitigate these attacks. Such measures are deemed necessary for the comprehensive safeguarding of data security and privacy. This research endeavors to develop a machine learning-based system for the detection of cyberattacks. The swift evolution of technology has rendered cyberattacks increasingly intricate to identify, with methods and vectors becoming more sophisticated and diverse. Consequently, this study employs machine learning methods for detection, with a specific focus on two types of cyberattacks: Remote Code Execution and Cross-Site Scripting. To attain a precise detection model, three algorithms were scrutinized in this research: a) Support Vector Machine; b) Gradient Boosting; and c) Logistic Regression. According to the conducted research, the Support Vector Machine algorithm achieved the highest accuracy rates, specifically 0.9876 for Remote Code Execution and 0.9961 for Cross-Site Scripting. Meanwhile, Logistic Regression yielded accuracy rates of 0.9537 (Remote Code Execution) and 0.9939 (Cross-Site Scripting), while Gradient Boosting demonstrated accuracy rates of 0.9475 (Remote Code Execution) and 0.9939 (Cross-Site Scripting). After conducting penetration tests using the Arachni and ZAP applications, it can be concluded that the Remote Code Execution detection model successfully detected 100% of the attacks. Conversely, the Cross Site Scripting detection model managed to identify up to 70% of the tested attacks.

Keywords*: cyber attack, support vector machine, logistic regression, gradient boosting, cross site scripting, remote code execution*

ABSTRAK

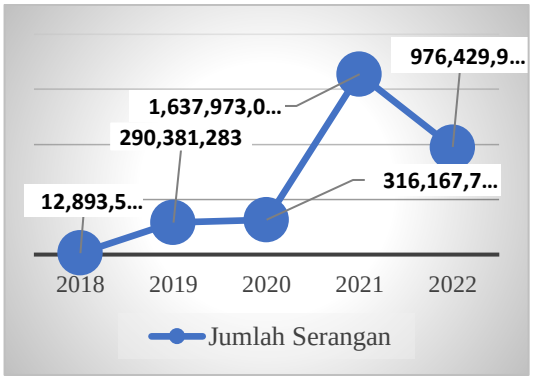
Seiring perkembangan teknologi informasi yang semakin terkoneksi dan bertumbuh secara pesat, Indonesia dihadapkan pada tantangan serius dalam bidang keamanan siber. Tingkat serangan siber bahkan hampir mencapai satu miliar serangan sepanjang tahun 2022. Hal ini menjadi salah satu indikator eskalasi yang mengkhawatirkan dari ancaman siber. Selain itu, serangan siber tersebut menunjukan perlunya solusi efektif untuk mendeteksi dan mengatasi serangan ini guna melindungi keamanan data dan privasi secara menyeluruh. Penelitian ini bertujuan untuk mengembangkan mesin deteksi serangan siber berbasis machine learning. Perkembangan dan kemajuan teknologi yang pesat membuat serangan siber semakin sulit dideteksi. Metode dan vektor serangan siber yang digunakan semakin kompleks dan beragam. Oleh karena itu, metode deteksi menggunakan metode machine learning. Penelitian

ini berfokus pada dua jenis metode serangan siber yaitu: Remote Code Execution dan Cross Site Scripting. Untuk mendapatkan model deteksi yang akurat, penelitian ini mengujikan tiga algoritma yaitu a) Support Vector Machine; b) Gradient Boosting; dan c) Logistic Regression. Berdasarkan penelitian yang telah dilakukan, algoritma Support Vector Machine mampu mencapai tingkat akurasi tertinggi yaitu 0,9876 untuk Remote Code Execution dan 0,9961 untuk Cross Site Scripting. Sementara itu, Logistic Regression mendapatkan tingkat akurasi 0,9537 (Remote Code Execution) dan 0,9939 (Cross Site Scripting) dan Gradient Boosting mendapatkan tingkat akurasi 0,947 (Remote Code Execution) dan 0,9939 (Cross Site Scripting). Hal ini menunjukan bahwa support vector machine menjadi algoritma yang mampu menghasilkan tingkat akurasi tertinggi pada deteksi kedua serangan tersebut. Setelah dilakukan uji coba penyerangan menggunakan aplikasi Arachni dan ZAP, model deteksi Remote Code Execution berhasil mendeteksi 100% serangan, sementara model deteksi Cross Site Scripting berhasil mendeteksi sampai dengan 70% serangan.

Kata kunci: serangan siber, support vector machine, logistic regression, gradient boosting, cross site scripting, remote code execution

I. PENDAHULUAN

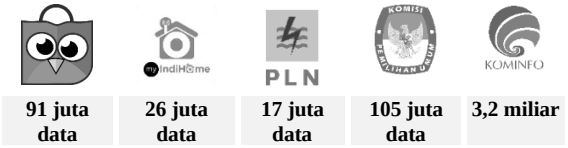
Dalam lima tahun terakhir, serangan siber menjadi subjek atau tema penelitian yang menarik. Terjadi peningkatan yang cukup signifikan pada jumlah serangan siber, bukan hanya pada tingkat lokal tetapi juga pada tingkat global atau internasional. Di Indonesia, peningkatan serangan siber yang cukup signifikan tersebut telah tercatat sejak lima tahun terakhir. Berdasarkan Laporan Tahunan dari Badan Siber dan Sandi Negara (BSSN) Indonesia sejak tahun 2018 sampai dengan 2022, terjadi peningkatan serangan yang signifikan dari tahun ke tahun, terutama pada tahun 2021. Dinamika peningkatan tersebut dapat dilihat pada gambar 1.



Gambar 1. Tren serangan siber selama lima tahun terakhir (2018 – 2022)

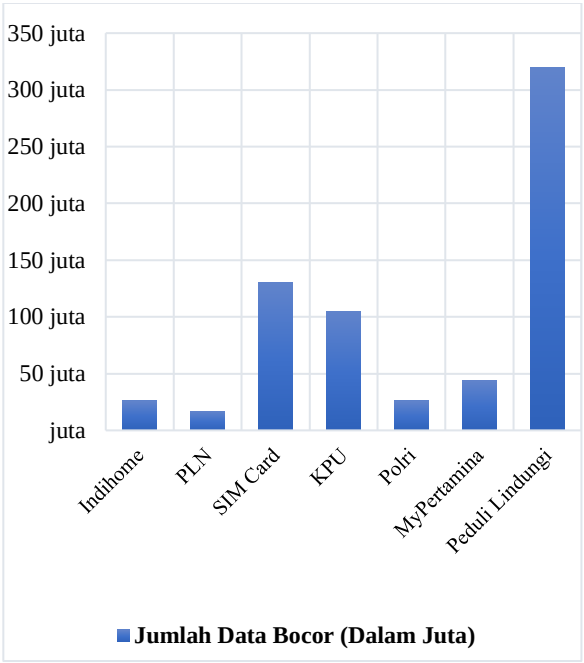
Pada tahun 2019, terjadi peningkatan serangan siber sekitar 25 kali lipat dari tahun sebelumnya [1]. Dengan demikian, rentang tahun 2018 ke 2019 tersebut merupakan gejala awal terjadinya peningkatan intensitas dan eskalasi serangan siber. Tidak berhenti pada tahun tersebut, tren serangan siber juga berlanjut pada tahun 2021, selama masa pandemi, jumlah anomali mencapai angka yang sangat tinggi, yaitu sekitar 1,65 miliar [2]. Pada tahun 2022, jumlah serangan masih mencapai angka yang cukup tinggi, yaitu sebanyak 976.429.996 serangan [3]. Gambar

2 menunjukan kasus *data breach* yang cukup fenomenal sepanjang tahun 2020—2022.



Gambar 2. Lima kasus *data breach* fenomenal di indonesia tahun 2020—2022

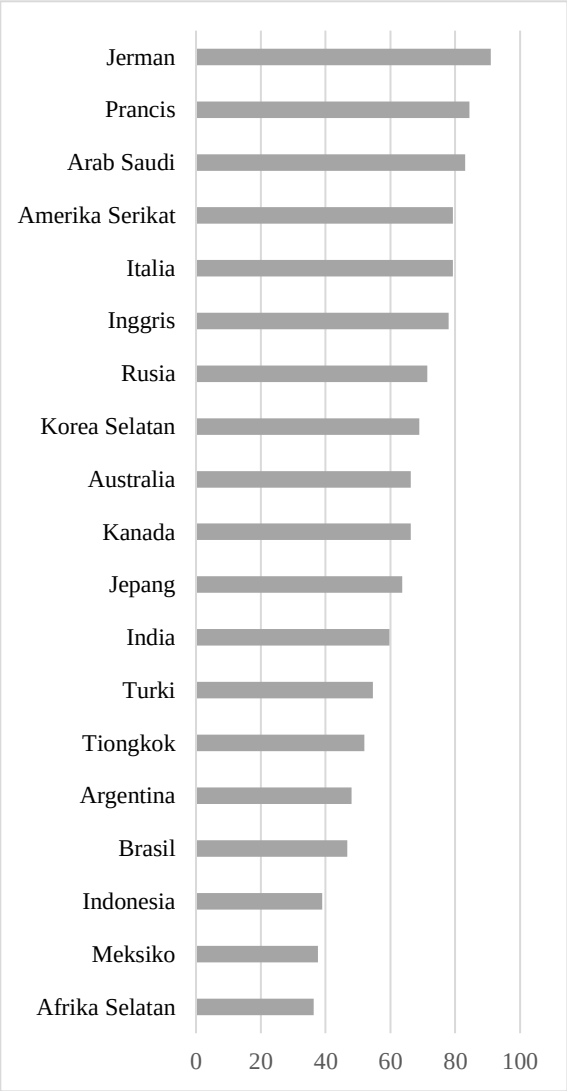
Meskipun total statistik serangan telah menunjukkan bahwa jumlah serangan telah menurun pada tahun 2022 dibandingkan dengan tahun-tahun sebelumnya, namun jumlah kebocoran data sangat tinggi pada tahun tersebut. Ironisnya, kebocoran data ini melibatkan data pribadi jutaan penduduk Indonesia. Data-data ini dijual secara bebas di forum jual beli yang sebagian besar beroperasi di *dark web*. Saat ini, data pribadi jutaan penduduk Indonesia yang seharusnya dijaga kerahasiaannya telah tersebar kepada siapa saja yang mampu membelinya. Padahal, data pribadi penduduk Indonesia adalah aset negara yang harus dijaga keamanannya. Dengan kata lain, serangan siber ini secara tidak langsung telah berdampak dan mengancam stabilitas serta keamanan nasional. Berikut adalah data kebocoran data yang terjadi di Indonesia khusus pada tahun 2022 saja (gambar 3).



Gambar 3. Kasus data breach fenomenal di Indonesia pada tahun 2022

Seperti yang terlihat pada gambar 3, terdapat perubahan variasi signifikan dalam jumlah kebocoran data pada berbagai kasus. Kasus *data breach* PeduliLindungi menjadi yang paling besar dengan 3,2 miliar data bocor, sementara kasus PLN mengalami jumlah kebocoran data terkecil yaitu 17 juta data. Ini menunjukkan pergeseran tren dari tahun-tahun sebelumnya, di mana serangan lebih banyak ditujukan kepada bisnis *e-commerce*, sementara pada tahun 2022, target serangan lebih banyak ditujukan pada lembaga milik pemerintah atau BUMN. Berita tentang data bocor ini tersebar luas di berbagai portal berita daring, dan data tersebut dijual di forum-forum *dark web*. Tingkat kebocoran data yang tinggi ini mengindikasikan bahwa keamanan siber di Indonesia masih belum cukup memadai dan harus segera diperkuat. Oleh karena itu,

penelitian dan analisis serangan siber menjadi sangat mendesak. Selain itu, berdasarkan KataData, indeks keamanan siber Indonesia berada posisi ke-3 terendah pada lingkup negara G-20 (Gambar 4).



Gambar 4. Indeks keamanan siber indonesia di antara negara G20

Mendeteksi serangan siber bukanlah hal yang mudah, karena penyerang semakin canggih dalam membangun serangan yang kompleks dan sulit terdeteksi. Metode-metode konvensional yang berbasis pada aturan atau *rule-based* tidak lagi efektif dalam mendeteksi serangan. Bahkan dalam beberapa kasus, penyerang berhasil mengelabui aplikasi deteksi yang berbasis *rule* seperti OWASP *Mod Security* CRS atau *Comodo Web Application Firewall*, meskipun aplikasi tersebut menggunakan *ekspresi reguler* (Regex) [4]. Aplikasi berbasis *rule* dan *Regex* hanya memblokir serangan secara statis dan tidak dapat mendeteksi serangan yang berada di luar cakupan aturan dan ekspresi Regex. Dalam era perkembangan teknologi yang pesat ini, metode deteksi serangan haruslah adaptif, cerdas, efisien, dan akurat. Oleh karena itu, penelitian ini memilih metode *machine learning* sebagai pendekatan untuk mengatasi kompleksitas serangan siber yang terjadi saat ini dan memfokuskan pada dua teknik serangan yaitu *Remote Code Execution* (RCE) dan *Cross Site Scripting* (XSS). Sampai saat ini, kedua teknik serangan siber ini telah beberapa kali masuk sebagai bagian dari *OWASP Top 10 Vulnerabilities*, yang berarti celah keamanan ini masih sering ditemukan pada banyak kasus serangan siber [5].

Terdapat beberapa penelitian sebelumnya yang terkait dengan penelitian ini. Penelitian-penelitian tersebut memiliki kelebihan dan kelemahan tersendiri. Berikut ini adalah lima penelitian sebelumnya yang juga mengujikan model deteksi serangan RCE dan XSS.

Penelitian oleh Nivetha et al [6] mengusulkan pendekatan multi algoritma *machine learning* dan fitur seleksi dengan *correlation filter*. Meskipun memiliki keunggulan penggunaan beragam algoritma, penelitian ini memiliki keterbatasan pada ukuran dataset kecil, ketergantungan pada aplikasi tertentu, dan fokus yang terbatas pada jenis serangan. Saha [7] mengembangkan kerangka kerja deteksi serangan yang cepat dan efisien dengan tujuan meminimalkan kerusakan dan biaya pemulihan. Namun, penelitian ini memiliki keterbatasan dalam teknik pengambilan fitur dan jenis serangan yang dapat dideteksi. Prasetio et al [8] membuat pendekatan fitur hybrid yang berbasis signature dan *behavior-based* menggunakan dataset besar. Walaupun memiliki potensi baik dalam meningkatkan akurasi, pendekatan ini memerlukan waktu dan daya komputasi besar. Aibamov [9] mengembangkan metode deteksi dengan akurasi baik dan fleksibilitas tinggi. Meskipun demikian, penelitian ini terbatas dalam mendeteksi serangan baru yang belum teridentifikasi sebelumnya. Bhardwaj [10] menggunakan beberapa algoritma dan *fitur hybrid* dengan dataset besar namun memerlukan sumber daya komputasi besar dan pengujian kompleks. Berdasarkan pada penelitian-penelitian tersebut, *novelty* atau kebaruan penelitian ini sebagai berikut:

1) penggunaan *multi features* dan *comprehensive feature enginering*;
2) penggunaan dua aplikasi simulasi serangan (Arachni dan ZAP) untuk memastikan dan mengujikandaya tahan dan kehandalan deteksi;
3) pemilihan, pengujian, pengukuran, dan perbandingan tiga algoritma statistik atau *machine learning* ;
4) pemanfaatan NLP untuk semakin meningkatkan akurasi deteksi; dan
5) pengujian terhadap dua jenis vektor serangan yaitu RCE dan XSS.

II. METODE PENELITIAN

2.1 Dataset

Untuk mendapatkan tingkat akurasi deteksi yang tinggi, berkinerja optimal, dan waktu komputasi yang relatif cepat, peneliti memilih dua dataset untuk masing-masing teknik atau vektor serangan siber. Dataset yang dipergunakan pada proses penelitian ini diungkapkan pada tabel 1.

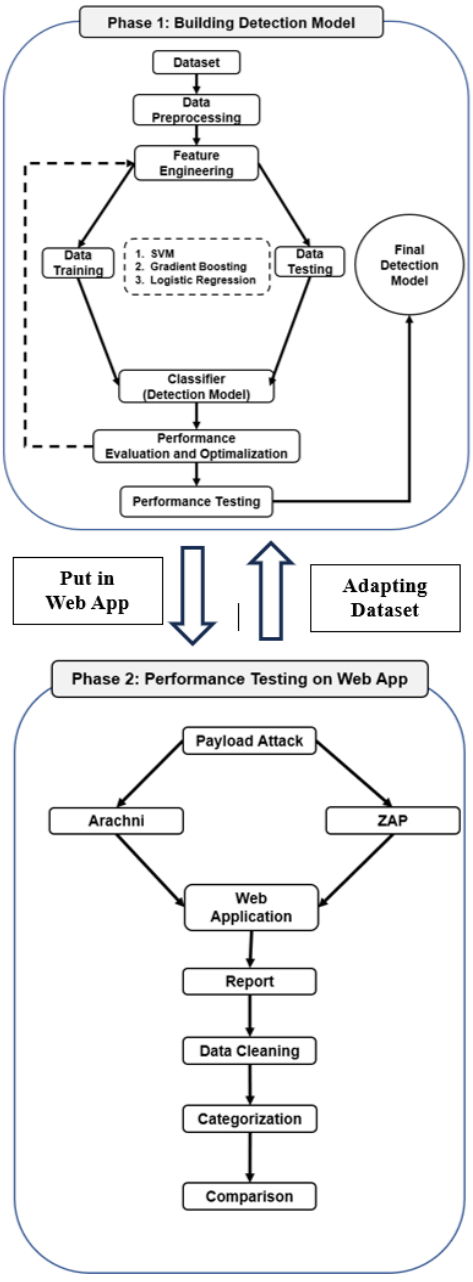
Tabel 1. Dataset dan Teknik Serangan

No	Teknik Serangan	Sumber Data
1	Remote Code Execution (RCE) [11]	https://github.com/payloadbox/command-injection-payload-list https://www.kaggle.com/datasets/antonyj453/urldataset
2	Cross Site Scripting (XSS) [12]	https://github.com/fmereani/Cross-Site-Scripting-XSS https://github.com/payloadbox/xss-payload-list

2.2 Proposed Method

Penelitian ini menggunakan *machine learning* untuk mengenali dan mendeteksi

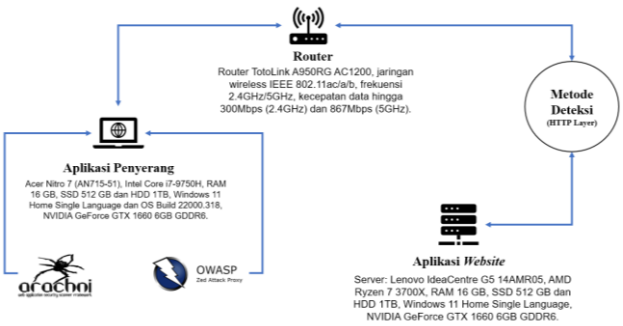
pola-pola pada serangan siber. Secara umum, penelitian ini berjalan dalam dua tahap, yaitu a) tahap pembangunan model deteksi dan b) tahap evaluasi performa model pada aplikasi berbasis *website* (Gambar 5).



Gambar 5. Dua fase penelitian untuk pembangunan dan pengujian deteksi

Untuk mengevaluasi kemampuan model deteksi dalam mengidentifikasi serangan siber secara akurat, peneliti telah melakukan pengujian pada aplikasi berbasis *website*. Pengujian ini dilakukan menggunakan dua aplikasi simulasi serangan rekursif, yakni Arachni dan OWASP ZAP. Setelah model deteksi terintegrasi dalam aplikasi *website*, kedua aplikasi tersebut melakukan simulasi serangan ke aplikasi *website* tersebut.

Tingkat keberhasilan minimal yang diharapkan dalam deteksi serangan adalah sekitar 89%. Jika tingkat ini belum tercapai, vektor atau payload serangan dari kedua aplikasi dimasukkan ke dalam dataset pada tahap pertama. Proses ini bertujuan untuk menguji sejauh mana metode deteksi tersebut dapat diandalkan. Metode deteksi ini bertugas untuk menyaring permintaan yang dikirim pengguna (gambar 6).



Gambar 6. Skenario pengujian metode deteksi menggunanakan aplikasi penyerang

Penelitian ini mengujikan tiga algoritma yaitu 1) *Support Vector Machine* (SVM); 2) *Gradient Boosting* (G-Boost); dan 3) *Logistic Regression* (LogReg). Ketiga algoritma tersebut dilihat 1) tingkat akurasi; 2) komputasi yang digunakan; dan 3) kecepatan kinerja. SVM merupakan suatu algoritma yang digunakan untuk menetapkan batas keputusan (*decision boundary*) dalam sebuah klasifikasi [13]. Algoritma ini memanfaatkan

model linear untuk menentukan batas keputusan. Aktivitas ini direpresentasikan dalam bentuk persamaan rumus 1 berikut:

$$\boldsymbol{y(x) = w^T \emptyset(x) + b}$$

Rumus 1. *Decision Boundary* Menggunanakan Model Linear

Keterangan:

y(x) = Prediksi model.
w^T = Vektor bobot.
∅(x) = Transformasi fitur input.
B = Bias model.

Parameter bobot diwakili oleh huruf w (*weight*), sedangkan ∅(x) adalah fungsi basis dan b adalah bias. Model linear yang paling sederhana dalam menentukan batas keputusan adalah y(x) = wtx + w0, dimana x adalah vektor, w adalah vektor bobot, dan w0 adalah bias. Selain SVM, peneliti menggunakan G-Boost berbasis *decision tree* [14]. Algoritma ini memproses dataset secara sekuensial dengan menambahkan prediktor sebelumnya ke *ensemble data* sehingga kesalahan prediksi sebelumnya dapat diperbaiki. Algoritma ketiga adalah Regresi Logistik. Algoritma ini memodelkan variabel target diskrit sebagai fungsi beberapa variabel fitur. Algoritma ini menggunakan variabel y diskrit. Setiap observasi dianalisis untuk menentukan probabilitas bahwa y = 1, diprediksi sebagai fungsi logistik dari kombinasi linear dari nilai-nilai fitur yang ada [15] (Rumus 2).

$$\mathbf{Y} = \frac{\exp(B_0 + B_1 X)}{(1 + \exp(B_0 + B_1 X))}$$

Rumus 2. Klasifikasi Menggunanakan *Logistic Regresion Classifier*

Keterangan:

Y = Variabel target atau hasil.
X = Variabel prediktor/fitur input.
B_0, B_1 = Parameter model.
exp() = Fungsi eksponensial.
1+exp() = Pembentuk probabilitas

2.3 Data Preprocessing and Analysis

Metode deteksi serangan pada penelitian ini dibangun dengan bantuan metode *Natural Language Processing* (NLP) *based machine learning*. Baik serangan RCE maupun XSS adalah serangan berbasis *string* atau teks, yang berarti serangan ini dapat dianalisis dalam lingkup atau konteks corpus [16]. Oleh karena itu, pada tahapan *data* preprocesing, peneliti melakukan konversi dataset menjadi data berbasis *corpus*.

Dengan pemanfaatan NLP, pola-pola serangan dapat lebih mendalam dan efektif ketika dianalisis dan dieksplorasi. Penelitian ini juga menggunakan dua dataset yang komprehensif dan representatif dengan vektor serangan siber nyata, sehingga sangat representatif dengan serangan sebenarnya. Setelah model deteksi dibangun, model tersebut dievaluasi dan dioptimalisasi seperti yang tertuang pada tabel 2. Selain itu, model deteksi tersebut juga diujikan kinerjanya untuk mendeteksi serangan RCE dan XSS yang dilakukan dengan bantuan aplikasi Arachni dan ZAP [17].

Tabel 2. Evaluasi dan Optimalisasi

Evaluasi Metode	Optimalisasi Metode
1) Penggunaan *confusion matrix* untuk mengevaluasi performa model	1) Memilih parameter yang memberikan hasil terbaik (*parameter tuning)*.
2) Mengukur proporsi prediksi positif benar terhadap total prediksi positif.	2) *Feature engineering* yang komprehensif
3) F1-score adalah rata-rata harmonik antara *precision* dan *recall*, yang memberikan perbandingan yang seimbang.	3) *Ensemble learning* atau menggabungkan model menjadi satu model yang lebih tinggi akurasinya.

III. HASIL DAN PEMBAHASAN

a. *Karakteristik Dataset*

Penelitian ini menggunakan 4 dataset dengan rincian 2 dataset untuk RCE dan 2 dataset XSS. Sebelum dianalisis, dilatih, dan diujikan, dataset kedua teknik serangan tersebut dikompilasi. Berikut ini adalah karakteristik dari dataset RCE dan XSS setelah dilakukan kompilasi.

Tabel 3. Sumber Dataset Berdasarkan Teknik Serangan

No	Teknik Serangan	Payload	Non Payload	*Total Item*	*Total Eligible*	*Total Removed*
1	RCE [11]	1034	1037	2071	1619	452
2	XSS [12]	21164	28068	49830	49232	598

Berdasarkan tabel 3, jumlah dataset yang digunakan pada penelitian berada pada kolom *total eligible*, yaitu 1619 data RCE dan 49232 data XSS. Penentuan *eligible* atau tidaknya ditentukan oleh (1) ada tidaknya kolom/baris yang kosong; (2) duplikasi data; dan (3) penyaringan minimal karakter.

b. *Konfigurasi Parameter Deteksi*

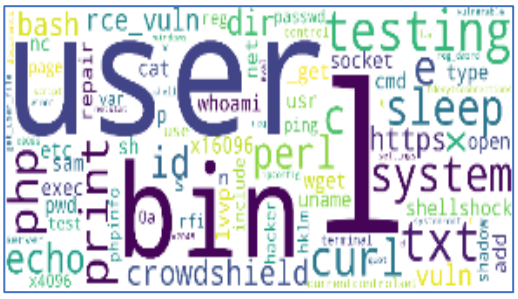
Pembangunan dan pengujian model deteksi pada penelitian ini menggunakan Python. Proses pembangunan model deteksi dipengaruhi oleh lima parameter yaitu margin, ukuran pengujian, konversi *lowercase*, penyaringan alfanumerik, dan penghilangan tanda baca. Ukuran margin akan berpengaruh pada ukuran atau pembatasan vektor dengan persamaan (sum(*n of words*) / len(*n of words*)). Konversi ke *lowercase* dilakukan karena vektor serangan tidak *case-sensitive*. Selain itu, vektor serangan juga didominasi oleh karakter non *alphanumeric* dan tanda baca sehingga tidak dilakukan penyaringan pada kedua parameter tersebut. Ketika penyaringan diaktifkan, tingkat akurasi menjadi menurun dan kinerja model deteksi menurun. Dalam hal ini, konfigurasi akhir dan teroptimal yang digunakan pada penelitian ini adalah sebagai berikut.

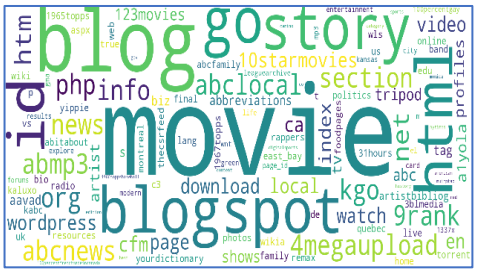
Tabel 4. Konfigurasi Parameter Teroptimal Model Deteksi RCE dan XSS

	Parameter	Nilai Parameter	
		RCE	XSS
1	Margin	16	6
2	Test Size	0,2	0,2
3	Lowercase Conversion	True	True
4	Alphanumeric Filter	False	False
5	Punctuation Remover	False	False

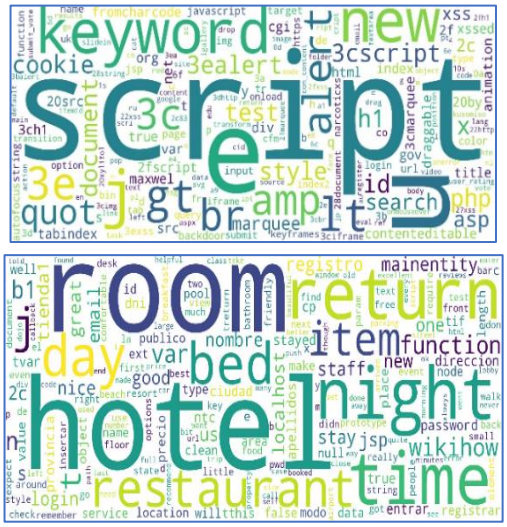
c. *Feature Set Vector*

Seperti yang telah diungkapkan pada poin B sebelumnya, tinggi dan rendahnya margin akan menentukan jumlah himpunan fitur. Pada dasarnya, deteksi serangan siber RCE dan XSS adalah tindakan untuk mengenali pola teks yang berupa serangan. Dengan demikian, jumlah vektor atau fitur akan menentukan tingkat akurasi model deteksi. Penelitian ini menggunakan modul Python NLTK kit untuk mempermudah identifikasi dan mencari jumlah vektor yang paling optimal. Gambar 7 dan 8 adalah vektor yang digunakan pada RCE dan XSS.





Gambar 7. *Wordcloud* Serangan RCE



Gambar 8. *Wordcloud* Serangan XSS

d. *Tingkat Akurasi*

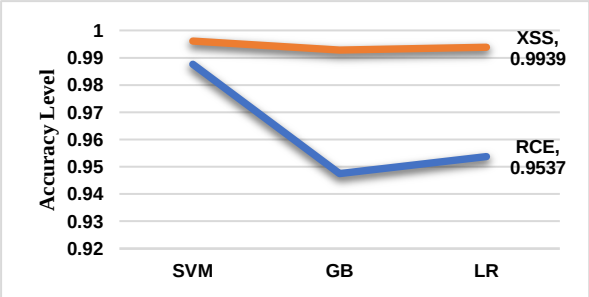
Setelah melakukan *multiple switching* pada berbagai parameter seperti margin, ukuran tes, konversi *lowecase*, serta penyaringan alphanumerik dan tanda baca, penelitian ini pada akhirnya mendapatkan konfigurasi parameter yang ideal (tabel 4) dan model deteksi mendapatkan tingkat akurasi deteksi yang tinggi. Pada penelitian ini, model deteksi serangan RCE dan XSS diujikan menggunakan 3 algoritma yaitu: a) Support Vector Machine (SVM); b) Gradient Boosting (GB); and c) Logistic Regression (LR). Berikut ini adalah tingkat akurasi pada masing-masing serangan dan algoritma. Nilai akurasi diambil berdasarkan akurasi tertinggi

masing-masing algoritma.

Tabel 5. Tingkat Akurasi Berdasarkan Serangan dan Algoritma

	Teknik Serangan	Tingkat Akurasi		
		SVM	GB	LR
1	Remote Code Execution (RCE)	**0,9876**	0,9475	0,9537
2	Cross Site Scripting (XSS)	**0,9961**	0,9928	0,9939

Berdasarkan tabel 5 tersebut, SVM terbukti menjadi algoritma yang mampu menghasilkan tingkat akurasi tertinggi pada serangan RCE dan XSS. Meskipun menggunakan dataset dengan karakteristik yang berbeda, tingkat akurasi yang diberikan pada penelitian ini cukup konsisten. Gambar 9 adalah grafik visualisasi tingkat akurasi berdasarkan teknik serangan dan algoritma.



Gambar 9. Tingkat Akurasi Masing-Masing Algoritma dan Teknik Serangan

Berikut ini adalah *confussion matrix* untuk serangan jenis RCE dan XSS pada masing-masing algoritma:

Tabel 6. *Confussion Matrix* Serangan RCE

	Non-payload	*Payload*
Non-payload	<208>	1
Payload	3	<112>

	Label	*Precision*	*Recall*	*F-Measure*
0	non-payload	0,98578	0,99521	0,99047
1	payload	0,99115	0,97391	0,98245

Tabel 7. *Confussion Matrix* Serangan XSS

	Non-payload	Payload
Non-payload	<5601>	18
Payload	20	<4208>

	Label	*Precision*	*Recall*	*F-Measure*
0	non-payload	0,99644	0,99679	0,99661
1	payload	0,99574	0,99527	0,99550

e. *Optimalisasi dan Pengujian Kinerja*

Penelitian ini melakukan optimalisasi model deteksi berdasarkan tiga hal (1) pemilihan algoritma yang memiliki tingkat akurasi terbaik; (2) pencarian parameter; dan (3) penggunaan aplikasi Arachni dan ZAP untuk menguji kinerja model deteksi. Selain mengukur tingkat akurasi, peneliti juga melakukan benchmarking pada waktu eksekusi masing-masing teknik serangan dan algoritma. Berikut ini adalah optimalisasi dan pengujian kinerja yang dilakukan.

Tabel 8. Rekam Jejak Optimalisasi dan Waktu Eksekusi Model Deteksi RCE

	Algoritma	Akurasi	Waktu Eksekusi	Margin	Test	Vektor
1	SVM	0,987654	0:00: 00.777743	16	0,2	189
2	SVM	0,966049	0:00: 00.747749	15	0,2	177
3	LR	0,953704	0:00: 00.670415	15	0,2	177
4	GB	0,947531	0:00: 01.334170	15	0,2	177

	Algoritma	Akurasi	Waktu Eksekusi	Margin	Test	Vektor
5	LR	0,947531	0:00: 00.799561	17	0,2	200
6	GB	0,941358	0:00: 01.661327	17	0,2	200

Waktu eksekusi pada tabel 8 dan 9 bukanlah waktu eksekusi per-*request* namun waktu eksekusi keseluruhan data pengujian (*data testing*). Selain serangan RCE, rekam jejak optimalisasi dan waktu eksekusi model deteksi XSS dapat dilihat pada tabel 9.

Tabel 9. Rekam Jejak Optimalisasi dan Waktu Eksekusi Model Deteksi XSS

	Algoritma	Akurasi	Waktu Eksekusi	Margin	Vektor
1	SVM	0,996	0:05:16.024	6	606
2	SVM	0,9952	0:04:28.789	5	505
3	LR	0,9939	0:01:08.352	6	606
4	LR	0,9936	0:00:58.411	5	505
5	GB	0,9928	0:03:50.434	5	505
6	GB	0,9893	0:06:08.905	7	707

Setelah model deteksi mendapatkan tingkat akurasi tertinggi, model tersebut disimpan dalam bentuk *pickle* dan disematkan pada aplikasi berbasis *website* Python Django. model deteksi disematkan pada lapisan *http middleware*, sehingga dapat memonitor *request* yang dikirimkan Arachni dan ZAP. Untuk efisiensi waktu eksekusi dan analisis, *request* serangan Arachni dan ZAP dikompilasi dan disimpan dalam csv. Cara ini juga dapat mempermudah dalam membaca waktu eksekusi. Hasil pengujian model dapat dilihat pada tabel 10 berikut ini.

Tabel 10. Pengujian Model Deteksi dalam Suasana Penyerangan

	Model Deteksi	Payload Serangan	Terdeteksi	Tidak Terdeteksi	Persentase
1	RCE	323	323	0	100%
2	XSS	10917	7734	3183	70%

Model deteksi RCE berhasil mendeteksi 100% serangan, sementara model deteksi XSS berhasil mendeteksi 70% serangan. Angka ini cukup signifikan memberikan dampak pada keamanan sistem, karena pada saat simulasi serangan dilakukan, fitur *firewall* dinonaktifkan. Hasil pengujian ini menunjukan bahwa kinerja model deteksi menunjukan hasil yang cukup baik.

f. *Kelemahan Penelitian*

Secara umum, terdapat dua kelemahan pada penelitian ini. Pertama, model deteksi serangan XSS pada saat dilakukan simulasi serangan hanya mencapai 70% seperti yang tertuang pada tabel 10. Kedua, model deteksi serangan memerlukan spesifikasi server yang mampu melayani *request* dalam jumlah yang besar. Setiap *request* yang dikirimkan akan diperiksa satu persatu sehingga memerlukan kinerja server yang prima optimal.

g. *Perbandingan Penelitian Sebelumnya*

Seperti yang telah dijelaskan pada hasil dan pembahasan tingkat akurasi serangan RCE mencapai 0,9876 dan XSS mencapai 0,9961. Perbandingan antara penelitian ini dengan penelitian sebelumnya dapat dibagi

menjadi tiga, yaitu berdasarkan (1) dataset; (2) tingkat akurasi; dan (3) kinerja, pengujian dan peforma. Berdasarkan dataset, penelitian Nivetha *et al* [6] menggunakan dataset kecil (*9 instances* dan 35 *features*), sementara itu penelitian ini menggunakan NLP Vector atau fitur sebanyak 606 dan 505 sehingga lebih representatif dengan data serangan. Berdasar pada tingkat akurasi, penelitian ini mampu mencapai 0,9876 (RCE) dan 0,9961 (XSS). Tingkat akurasi ini lebih tinggi dari beberapa penelitian seperti Nivetha *et al* [6] sebesar 0,91, Bhardwaj [10] sebesar 0,994, dan Abaimov [9] sebesar 0.95. Sementara itu, berdasar pada kinerja, pengujian, dan peforma model, kinerja dan peforma model diujikan oleh aplikasi simulasi serangan dan mampu mendeteksi 100% XSS dan 70% RCE. Hal ini tidak dilakukan pada penelitian Prasetio *et al* [8], tepatnya pada kasus serangan XSS.

h. *Implikasi*

Hasil penelitian ini menawarkan potensi untuk meningkatkan ketahanan sistem terhadap serangan berbasis *website* yang sering dimanfaatkan penyerang. Penerapan model *machine learning* dalam deteksi XSS dan RCE dapat meningkatkan respon sistem terhadap ancaman dengan mendeteksi pola dan perilaku serangan yang belum teridentifikasi secara statis. Hal ini tentunya memungkinkan sistem untuk secara dinamis mengidentifikasi dan merespons serangan yang berkembang seiring waktu.

Selain itu. hasil penelitian ini dapat membimbing pengembangan algoritma deteksi yang lebih canggih, memungkinkan sistem untuk mengidentifikasi serangan dengan tingkat akurasi yang lebih tinggi dan mengurangi *false positive*. Dengan demikian, implikasi praktis termasuk penerapan model deteksi ini dalam lingkungan produksi untuk memastikan perlindungan terus-menerus pada serangan XSS dan RCE.

Penelitian ini diharapkan dapat memberi landasan untuk pengembangan lebih lanjut dalam bidang keamanan siber dengan fokus pada deteksi ancaman berbasis *machine learning*. Inisiatif penelitian selanjutnya dapat memperluas cakupan untuk mengatasi serangan lainnya dan meningkatkan daya tangkal sistem secara menyeluruh.

IV. SIMPULAN

Berdasarkan pengujian dan analisis yang telah dilakukan, algoritma SVM terbukti memiliki kinerja dan tingkat deteksi yang paling baik dibandingkan algoritma GB dan LR. Algoritma ini mendapatkan tingkat akurasi deteksi RCE sebesar 0,9876 dan XSS sebesar 0,9961. Dengan melakukan optimalisasi, tingkat akurasi deteksi serangan RCE dan XSS menjadi semakin baik. Model deteksi RCE mampu mengenali 100% vektor serangan Arachni dan ZAP, dan model deteksi XSS mampu mengenali 70% vektor serangan. Penelitian yang dilakukan oleh Nivetha [6] mendapatkan akurasi deteksi

serangan XSS sebesar 0,91 setelah mengujikan algoritma XGBoost, Decision tree, KNN, Naive Bayes, and AdaBoost. Penelitian Abaoimov [9] mendapatkan tingkat akurasi serangan RCE sebesar 0,957.

DAFTAR PUSTAKA

[1] A. Yusuf, *Laporan Tahunan 2020 Honeynet Project BSSN - IHP*. Badan Siber dan Sandi Negara, 2022.

[2] S. Parulian, D. A. Pratiwi, dan M. C. Yustina, "Studi Tentang Ancaman dan Solusi Serangan Siber di Indonesia," *Telecommun. Netw. Electron. Comput. Technol. TELNECT*, vol. 1, no. 2, Art. no. 2, Des 2021.

[3] "BSSN: Hampir 1 Miliar Serangan Siber Hantam RI di 2022." Diakses: 9 April 2023. [Daring]. Tersedia pada: https://www.cnnindonesia.com/teknologi/20230119144028-192-902537/bssn-hampir-1-miliar-serangan-siber-hantam-ri-di-2022

[4] K. D. Ayunda, A. Widjajarto, dan A. Budiono, "Implementation and Analysis ModSecurity on Web-Based Application with OWASP Standards," vol. 8, no. 3, hlm. 12, 2021.

[5] S. -, I. Riadi, dan P. Ananda, "Vulnerability Analysis of E-voting Application using Open Web Application Security Project (OWASP) Framework," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, 2019, doi: 10.14569/IJACSA.2019.0101118.

[6] G. Nivetha, "IDENTIFYING THE CROSS SITE SCRIPTING (XSS) ATTACK USING XSSER TOOL AND DETECTION USING SUPERVISED LEARNING ALGORITHM," *Ind. Eng. J.*, no. 1, 2023.

[7] T. Saha, T. Al Rahat, N. Aaraj, Y. Tian, dan N. K. Jha, "ML-FEED: Machine Learning Framework for Efficient Exploit Detection," dalam *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, Des 2022, hlm. 140–149. doi: 10.1109/TPS-ISA56441.2022.00027.

[8] D. A. Prasetio, K. Kusrini, dan M. R. Arief, "Cross-site Scripting Attack Detection Using Machine Learning with Hybrid Features," *J. INFOTEL*, vol. 13, no. 1, hlm. 1–6, Feb 2021, doi: 10.20895/infotel.v13i1.606.

[9] S. Abaimov dan G. Bianchi, "CODDLE: Code-Injection Detection With Deep Learning," *IEEE Access*, vol. 7, hlm. 128617–128627, 2019, doi: 10.1109/ACCESS.2019.2939870.

[10] A. Bhardwaj, S. S. Chandok, A. Bagnawar, S. Mishra, dan D. Uplaonkar, "Detection of Cyber Attacks: XSS, SQLI, Phishing Attacks and Detecting Intrusion Using Machine Learning Algorithms," dalam *2022*

IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), New Delhi, India: IEEE, Sep 2022, hlm. 1–6. doi: 10.1109/GlobConPT57482.2022.9938367.

[11] Swissky, "Payloads All The Things." 9 April 2023. Diakses: 9 April 2023. [Daring]. Tersedia pada: https://github.com/swisskyrepo/PayloadsAllTheThings

[12] F. Mereani dan J. Howe, "Exact and Approximate Rule Extraction from Neural Networks with Boolean Features," dipresentasikan pada 11th International Conference on Neural Computation Theory and Applications, Jul 2022, hlm. 424–433. Diakses: 1 Juli 2022. [Daring]. Tersedia pada: https://www.scitepress.org/Link.aspx?doi=10.5220/0008362904240433

[13] L. Ali, I. Wajahat, N. Amiri Golilarz, F. Keshtkar, dan S. A. C. Bukhari, "LDA–GA–SVM: improved hepatocellular carcinoma prediction through dimensionality reduction and genetically optimized support vector machine," *Neural Comput. Appl.*, vol. 33, no. 7, hlm. 2783–2792, Apr 2021, doi: 10.1007/s00521-020-05157-2.

[14] R. Sarkhani Benemaran, M. Esmaeili-Falak, dan A. Javadi, "Predicting resilient modulus of flexible pavement foundation using extreme gradient boosting based optimised models," *Int. J. Pavement Eng.*, vol. 0, no. 0, hlm. 1–20, Jul 2022, doi: 10.1080/10298436.2022.2095385.

[15] F. Handayani, "Komparasi Support Vector Machine, Logistic Regression Dan Artificial Neural Network Dalam Prediksi Penyakit Jantung," *J. Edukasi Dan Penelit. Inform. JEPIN*, vol. 7, no. 3, hlm. 329, Des 2021, doi: 10.26418/jp.v7i3.48053.

[16] J. K. Cage, "Python Natural Language Processing (NLP) Exercises : From Basics to BERT," hlm. 163.

[17] G. A. Supriatmaja, I. P. M. Y. Pratama, K. Mahendra, I. M. E. Listartha, dan G. A. J. Saskara, "Perbandingan Vulnerability Analysis Pada Website Menggunakan Tools Wapiti, Skipfish, Dan Arachni," *JurTI J. Teknol. Inf.*, vol. 6, no. 2, Art. no. 2, Des 2022, doi: 10.36294/jurti.v6i2.2990.