Analisis dan Penanganan Insiden Siber SQL Injection Menggunakan Kerangka NIST SP 800-61R2 dan Algoritma Klusterisasi K-Means

Choerun Asnawi[1], Dedy Hariyadi [2*], Ulfi Saidata Aesyi[3], Puji Winar Cahyo[4]
[1,2,3,4]Universitas Jenderal Achmad Yani Yogyakarta
*email: dedy@unjaya.ac.id

DOI: https://doi.org/10.31603/komtika.v7i2.10527

Received: 12-11-2023, Revised: 25-11-2023, Accepted: 26-11-2023

ABSTRACT

Based on the OWASP Top Ten document in 2021, attacks or vulnerabilities in an application in the form of injection still rank in the top 3. SQL Injection attacks are still classified as injection vulnerabilities so they need special attention from Information & Communication Technology Managers. Badan Siber dan Sandi Negara (BSSN) has published a document related to preventing SQL Injection attacks. However, the document has not included a cyber attack analysis process that uses the K-Means clustering approach. So in this research, a collaborative method of handling cyber attacks in the form of SQL Injection is proposed using the NIST SP 800-61R2 framework as a fundamental for handling cyber attacks and K-Means clustering. Before analyzing cyber attacks, it is better to use a framework or standardization that applies globally. Based on the research conducted, the K-Means clustering algorithm can help cybersecurity analysts in the process of analyzing cyber attacks that occur. The result of this research is that the optimal value is obtained that cyber attacks in the form of SQL Injection, namely 3 clusters. The hope of the research can facilitate cybersecurity analysts in analyzing cyber attacks that are poured into reports to parties in need.
Keywords: *Cyber Attacks, SQL Injection, NIST, K-Means, Cyber Security*

ABSTRAK

Berdasarkan dokumen *OWASP Top Ten* pada tahun 2021 serangan atau kerentanan pada sebuah aplikasi berupa injeksi masih menempati urutan 3 besar. Serangan *SQL Injection* masih tergolong dari kerentan injeksi sehingga perlu perhatian khusus para pengelola Teknologi Komunikasi dan Informasi. Badan Siber dan Sandi Negara (BSSN) telah menerbitkan dokumen terkait pencegahan serangan *SQL Injection*. Namun, pada dokumen tersebut belum disertakan proses analisis serangan siber yang menggunakan pendekatan klusterisasi K-Means. Maka pada penelitian ini diusulkan metode kolaboratif penanganan serangan siber berupa *SQL Injection* menggunakan kerangka NIST SP 800-61R2 sebagai fundamental penanganan serangan siber dan klusterisasi K-Means. Sebelum melakukan analisis serangan siber sebaiknya tetap menggunakan kerangka atau standarisasi yang berlaku secara global. Berdasarkan penelitian yang dilakukan bahwa algoritma klusterisasi K-Means dapat membantu analis keamanan siber dalam melakukan proses analisis serangan siber yang terjadi. Hasil dari penelitian ini bahwa didapatkan nilai optimal bahwa serangan siber berupa *SQL Injection*, yaitu 3 *kluster*. Harapan dari penelitian dapat mempermudah analis keamanan siber dalam menganalisis serangan siber yang dituangkan dalam laporan ke pihak yang membutuhkan.
Keywords: Serangan Siber, *SQL Injection*, NIST, K-Means, Keamanan Siber

PENDAHULUAN

Open Web Application Security Project (OWASP) merupakan komunitas nirlaba yang terbuka dan mendedikasikan dalam melakukan ulasan tingkat celah keamanan secara berkala. Pada dokumen OWASP 2017 menempatkan celah keamanan *Injection* termasuk celah keamanan *SQL Injection* sebagai posisi pertama yang berpeluang dimanfaatkan oleh peretas [1]. Sedangkan pada dokumen OWASP tahun 2021 *SQL Injection* yang merupakan bagian dari kerentanan *Injection* turun pada posisi ketiga[2]. Walaupun celah kerentanan keamanan

Injection mengalami penurunan tetapi potensi ini masih cukup tinggi pada aplikasi berbasis web. Oleh sebab itu Badan Siber dan Sandi Negara (BSSN) melakukan langkah preventif untuk menghadapi serangan dengan potensi kerentanan celah keamanan *SQL Injection* dengan menerbitkan panduan pencegahan serangan *SQL Injection* [3]. Namun, pada dokumentasi tersebut belum dijelaskan tentang deteksi serangan *SQL Injection* baik secara tradisional maupun menggunakan pendekatan kecerdasan buatan. Beberapa penelitian terdahulu telah melakukan usulan dalam mendeteksi serangan *SQL Injection* menggunakan pendekatan beberapa algoritma.

Serangan *SQL injection* dapat dilihat dengan melalui *log* aktivitas web yang terdapat pada *server*. *Log* ini berisikan data-data perintah seperti *GET* atau *POST* dengan *timestamp* [4]. *Data log* ini perlu diproses sebelum menjadi *dataset* yang akan digunakan pada sistem deteksi serangan *SQL injection*. Pemrosesan data yang dilakukan adalah dengan melakukan analisis oleh pengelola Teknologi Komunikasi dan Informasi sehingga data dapat dibedakan, *data log* yang terindikasi adanya serangan maupun akses normal. Data yang telah dipisahkan kemudian ditandai atau dilakukan pelabelan sehingga dapat digunakan sebagai data latih oleh sistem. Serangan *SQL injection* kerap kali dapat dilihat pada *log* berupa adanya manipulasi menggunakan perintah SQL, penggunaan *comment character*, *operator*, dan penggunaan *semicolon*.

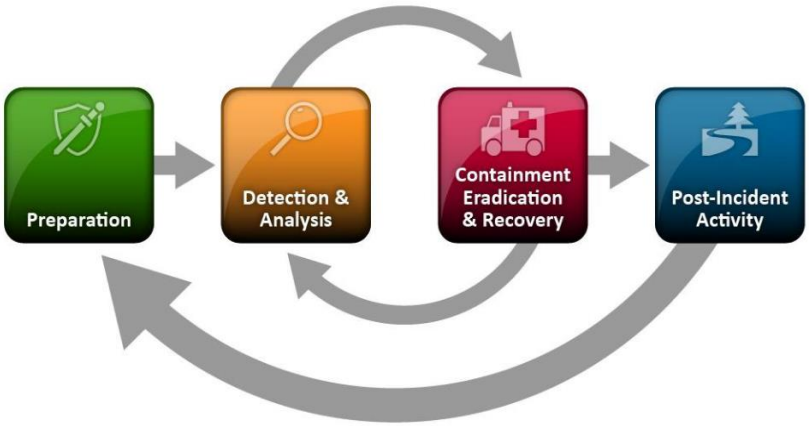
Siti Hajar Nadhirah Harip, dkk dari Universiti Tun Hussein Onn Malaysia pada tahun 2022 telah melakukan penelitian tentang serangan *SQL Injection* menggunakan pendekatan *rule-based*. Dalam penelitian tersebut, serangan *SQL Injection* yang diteliti dibagi menjadi empat kategori, yaitu, *Tautology*, *Inference*, *Basic Query*, dan *Blind Attack*. Berdasarkan hasil pengujian akurasi masing-masing kelompok memiliki nilai, *Tautology* 99%, *Inference* 100%, *Basic Query* 99%, dan *Blind Attack* 100% [5]. Sedangkan peneliti dari Spanyol melakukan analisis serangan SQL Injection bersumber dari data NetFlow, sebuah protokol yang dikembangkan oleh Cisco Systems tentang lalu lintas jaringan dalam rangka untuk memudahkan dalam pengelolaan dan pemantauan sistem pada jaringan komputer oleh pranata sistem jaringan komputer. Berdasarkan data dari NetFlow serangan *SQL Injection* dapat teranalisis dengan akurasi 97% dengan *false alarm* kurang dari 0.07% yang berbasis pemodelan *Logistic Regression* [6].

Berdasarkan penelitian sebelumnya bahwa deteksi dan analisis serangan *SQL Injection* sangat diperlukan karena saat ini serangan ini tergolong membahayakan dengan dampak kebocoran data dan informasi sensitif. Bahkan analisis serangan telah memanfaatkan algoritma untuk mempermudah dan mempercepat analisis. Namun, pada penelitian tersebut tidak dijelaskan secara eksplisit kerangka pengambilan sumber data. Maka pada penelitian ini diusulkan penanganan insiden siber dengan studi kasus serangan *SQL Injection* menggunakan kerangka NIST SP 800-61R2 sebagai kerangka teknis penanganan insiden siber. Sedangkan pada tahapan deteksi dan analisis sesuai NIST SP 800-61R2 menggunakan pendekatan klusterisasi K-Means.

METODE

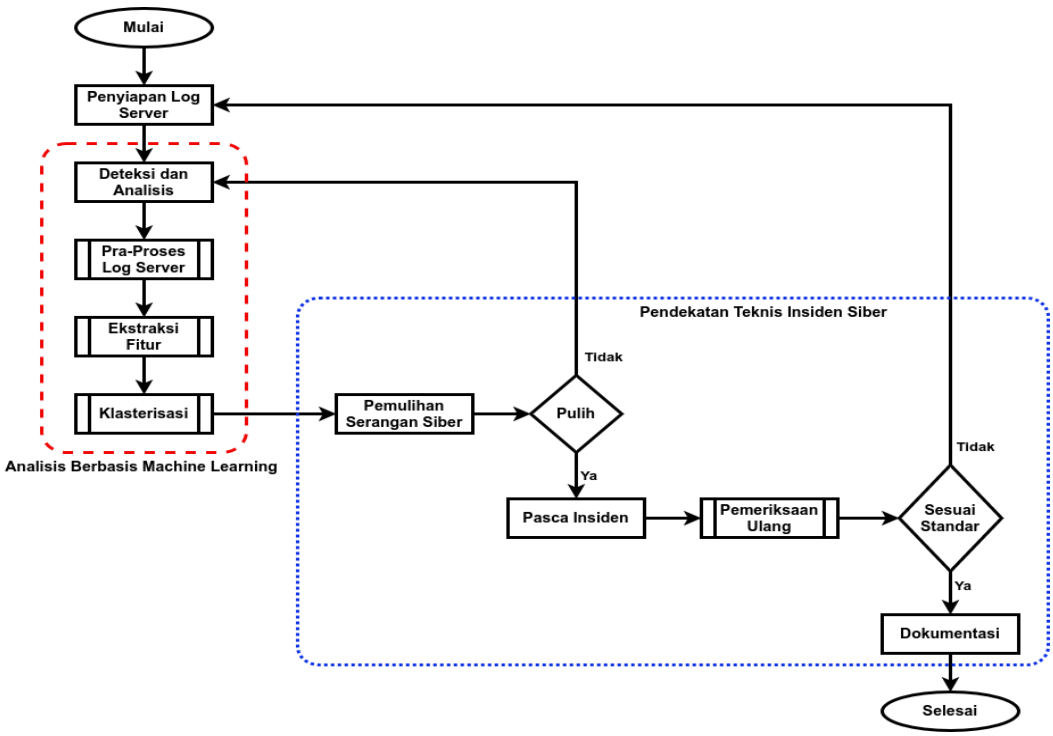
Kementerian Perdagangan Amerika Serikat melalui *National Institute of Standards and Technology* mengeluarkan standarisasi dalam melakukan penanganan insiden siber yang tertuang pada NIST SP 800-61R2. Pada standarisasi tersebut tahapan dalam penanganan siber

terbagi menjadi 4 tahapan, yaitu *preparation, detection & analysis, containment eradication & recovery,* dan *post-incident activity*. 4 tahapan ini dikenal sebagai *Incident Response Life Cycle*, disajikan seperti pada Gambar 1 [7].



Gambar 1. Siklus Penanganan Insiden Siber

Tahap awal, yaitu *preparation* dilakukan dengan mempersiapkan berbagai kebutuhan teknis dan non teknis. Adapun contohnya adalah mengurus perizinan terhadap organisasi yang terdampak serangan siber untuk pengambilan sampel atau barang bukti digital. Selanjutnya pada tahapan *detection & analysis* dapat diselaraskan dengan berbagai keilmuan. Untuk tahapan *detection* penyelarasannya adalah pengambilan sumber data dari *service logs*. Dalam hal ini *service logs* yang digunakan adalah catatan akses dari web server [8]. Sedangkan tahapan *incident analysis* menggunakan algoritma klusterisasi K-Means [9]. Maka alur pada penelitian yang mengintegrasikan NIST SP 800-61R2 dengan algoritma K-Means seperti tampak pada Gambar 2.



Gambar 2. Alur Penelitian

Saat terjadi insiden siber maka langkah awal yang dilakukan sebagai bagian dari persiapan adalah mengamankan barang bukti berupa *log server*. Tahapan analisis dan deteksi dilakukan pra-proses pada *log server* yang dilanjutkan dengan ekstraksi fitur. Proses analisis serangan *SQL Injection* dengan melakukan klusterisasi menggunakan algoritma K-Means. Pada tahapan ini menggunakan *Colab*, produk dari Google untuk melakukan analisis berbasis *Platform as a Service* [10]. Setelah dilakukan klusterisasi, informasi akan divalidasi dengan tim penanganan insider siber untuk proses pemulihan. Jika pemulihan berhasil maka proses klusterisasi menjadi *Body of Knowledge* dalam penanganan insiden siber [11].

Barang bukti berupa *log server* yang diamankan dalam bentuk teks, maka analisis yang cocok adalah analisis menggunakan metode *text mining*. Pada metode *text mining* akan disematkan nilai pembobotan pada teks yang telah ditentukan, yaitu *term* atau dokumen yang terdapat pada [12]. Tingkat kemunculan *term* atau dokumen pada *log server* dianalisis menggunakan TF-IDF, yaitu salah satu metode yang digunakan untuk pembobotan data. Model pembobotan TF-IDF merupakan metode yang mengintegrasikan model *term frequency* (TF) dan *inverse document frequency* (IDF). Adapun formula TF-IDF ditunjukkan seperti pada formula (1) [13].

$$TF - IDF_{t,d} = 1 + \log(TF_{t,d})\, X\, \log(\frac{N}{DF_t}) \qquad (1)$$

Keterangan : TF digunakan untuk menentukan bobot dari setiap kata (*term, t*), sedangkan IDF digunakan untuk pengurangan dominasi kata (*term, t*) yang sering muncul pada suatu dokumen (*document, d*) [14].

K-Means merupakan algoritma untuk mengkonsolidasikan data yang bertujuan untuk menggabungkan titik data berdasarkan kedekatannya dengan pusat kluster [15]. Dalam penentuan jumlah kluster yang optimal dalam algoritma K-Means pada penelitian ini menggunakan metode evaluatif metode *Elbow* dan *Silhouette Score*. Metode *Elbow* digunakan untuk menilai pengelompokan dengan mengukur *Sum of Square Error* (SSE). SSE adalah akumulasi kuadrat jarak antara setiap titik data dan pusat kluster. Metode *Elbow* mencari titik optimal saat terjadi penurunan SSE yang secara signifikan sehingga dapat menunjukkan jumlah kluster yang optimal, seperti pada formula (2).

$$SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left\| x_{ij} - \mu_i \right\|^2 \qquad (2)$$

Keterangan : *k* merupakan jumlah kluster yang dipertimbangkan, *ni* merupakan jumlah data dalam kluster ke-*i, xij* merupakan data ke-*j* dalam kluster ke-*i*, dan *μi* merupakan rata-rata dari data dalam kluster tersebut.

Silhouette Score digunakan sebagai metrik untuk menilai kualitas kluster dengan mempertimbangkan kohesi dalam kluster atau pemisahan antar kluster. *Silhouette Score* memberikan nilai numerik mulai dari -1 hingga 1 untuk setiap titik data dengan nilai yang lebih besar menunjukkan peningkatan kluster, seperti pada formula (3).

$$S = maxS(n,p) \tag{3}$$

Keterangan: *S(n,p)* merupakan hasil yang dihasilkan oleh kriteria *Silhouette* yang diterapkan pada algoritma hierarkis *p* yang membentuk *n* kluster [16].

HASIL DAN PEMBAHASAN

Pada penelitian ini sumber data yang digunakan berupa simulasi serangan *SQL Injection* pada sebuah portal yang telah disiapkan. Serangan *SQL Injection* pada prinsipnya melalui *HyperText Transfer Protocol* (HTTP), sehingga akses dapat tercatat pada web server [17]. Dalam hal ini tercatat pada access.log dari web server Apache2. Pada serangan *SQL Injection* terdapat beberapa teknik maka tipe serangan *SQL Injection* dapat dikategorikan 9 serangan, yaitu: *tautologies, union queries, error-based, boolean-based, time-based, out of banda exploitation technique, piggy – backed queries attack, store procedure injection,* dan *encoding attack* [18]. Berdasarkan pengumpulan data pada simulasi serangan *SQL Injection* yang tercatat pada *access.log* sebanyak 1002 baris akses.

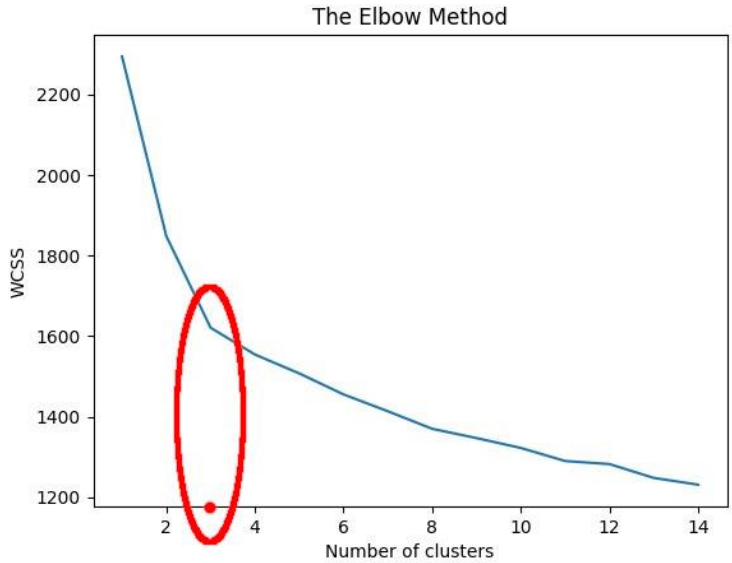
Pengelola server pada umumnya kesulitan dalam membaca atau menganalisis *access.log* baris per baris. Maka dari itu diperlukan alat bantu untuk menganalisis atau memantau aktivitas *web server* sehingga dapat melakukan deteksi jika terjadi akses yang bersifat anomali seperti serangan *SQL Injection* [19]. Untuk mendapatkan berkas *access.log* mengikuti kaidah forensik digital yang menjaga keutuhan barang bukti digital, yaitu melakukan akuisisi secara logikal pada *log server* [20]. Adapun tahapan analisis pada penelitian ini melakukan identifikasi serangan *SQL Injection* menggunakan K-Means dengan tujuan melakukan pengelompokan informasi pada *access.log* sehingga mempermudah mencari informasi terkait serangan *SQL Injection* [21].

Secara umum format *log* dari *web server apache* mencatat informasi sumber pengakses yang terdiri dari *IP Address, username* (jika dikonfigurasi), waktu, *request type* (GET/POST), kode status respon HTTP, ukuran, *user-agent*, dan sumber perujuk [22]. Implementasi TF-IDF pada dokumen berupa *log web server*, yaitu melakukan *pre-processing*, melakukan perhitungan frekuensi dari *term*, melakukan perhitungan frekuensi dari inversi dokumen, dan pembobotan. Setiap data yang diperoleh dari *log server* dilakukan pembobotan kata (TF), kemudian dihitung jumlah kemunculan setiap kata (IDF). Pada penelitian ini data setiap *log server* dilakukan pembobotan kata dan dihitung jumlah kemunculan setiap kata. Contoh hasil dari tahapan TF-IDF ditunjukan pada Tabel 1.

Tabel 1. Tahapan TF-IDF

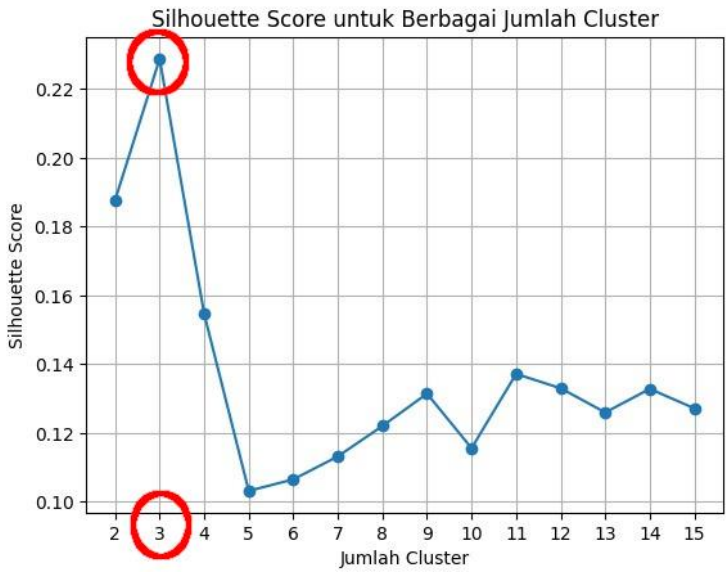
Sumber Data	**Term Frequency (TF)**	**Inverse Document Frequency**
GET /.env HTTP/1.1" 404 199 "-	[0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]	[0. 0. 0. 0. 0.50723605 0. 0. 0. 0. 0.50723605 0. 0. 0. 0. 0. 0. 0.63950056 0. 0.20012736 0. 0. 0. 0.19081735 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
GET /home.php HTTP/1.1" 200 18 "http://103.150.90.214:80/	[1 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1 2 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]	[0.17909919 0.17909919 0.47190241 0. 0. 0.41480944 0.17909919 0. 0. 0. 0.47190241 0.17909919 0. 0. 0. 0. 0. 0. 0.14767866 0. 0. 0.27670026 0.28161717 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.27670026 0. 0. 0. 0. 0. 0.]
GET /home.php HTTP/1.1" 200 1908 "-	[0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]	[0. 0. 0. 0.61461422 0. 0.54025531 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.19233935 0. 0. 0.36037942 0.18339164 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.36037942 0. 0. 0. 0. 0. 0.]
GET /images/footer-bg.png HTTP/1.1" 304 0 "http://103.150.90.214/css/style.css	[1 1 0 0 0 0 1 0 1 0 0 1 0 0 1 2 0 1 1 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0]	[0.18259304 0.18259304 0. 0. 0. 0. 0.18259304 0. 0.21240081 0. 0. 0.18259304 0. 0. 0.32339644 0.47276467 0. 0.48110824 0.15055956 0. 0. 0. 0.28711094 0.21240081 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.23638233 0. 0. 0. 0.23638233 0.]

Akumulasi kuadrat jarak antara setiap titik data dan pusat kluster berupa nilai SSE untuk menentukan data jumlah kluster yang optimal berdasarkan nilai TF-IDF dari titik data. Dengan menghitung SSE menggunakan nilai TF-IDF, metode *Elbow* terbukti mengidentifikasi jumlah kluster yang paling optimal pada serangan *SQL Injection* pada log server sebanyak 3 kluster. Pada Gambar 3 menunjukkan nilai optimal yaitu 3 kluster ditunjukan dengan titik siku atau "*elbow*".



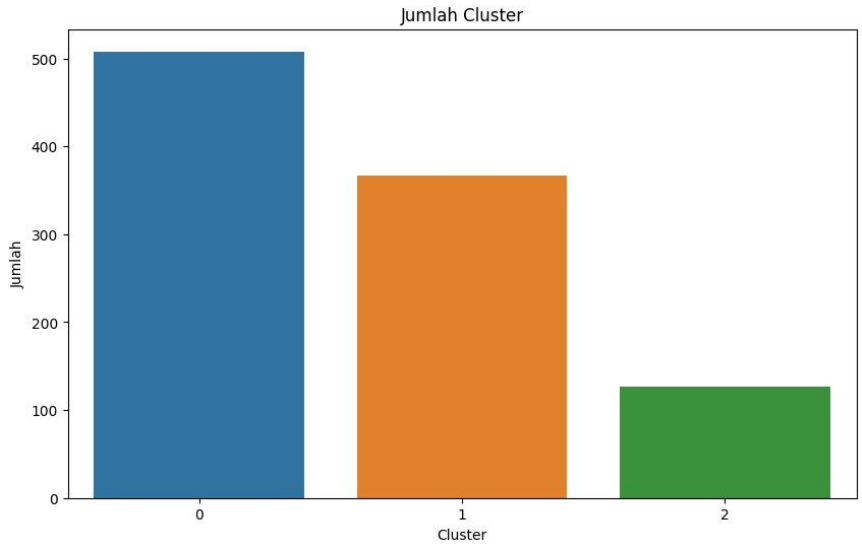
Gambar 3. Kluster Optimal

Jumlah kluster yang optimal pada *log server* dilakukan pengujian menggunakan *Silhouette Score*. Nilai yang dihasilkan menunjukkan kluster yang optimal, yaitu 3 kluster. Nilai tersebut sebagai rekomendasi dan memvalidasi jumlah kluster yang ditentukan dalam menjalankan algoritma K-Mean pada kumpulan data pada *log server*. Hasil terbaik dari proses klusterisasi yang menghasilkan nilai optimal 3 kluster ditunjukan pada Gambar 4.



Gambar 4. *Silhouette Score* Optimal

Berdasarkan perhitungan tersebut yang menghasilkan nilai optimal klusterisasi 3 kluster maka visualisasi serangan *SQL Injection* pada penelitian ini dapat ditunjukan pada Gambar 5.



Gambar 5. Diagram Batang Hasil Klusterisasi

Sedangkan pada Tabel 2 merupakan kutipan serangan *SQL Injection* pada kluster 0 yang menunjukan serangan terbanyak pada pukul 02.40 sampai dengan 02.43 pada hari yang sama. Berdasarkan waktu serangan bahwa IP 182.253.163.96 mendominasi dalam melakukan serangan *SQL Injection* yang ditunjukan pada bagian *Request* dengan berbagai payload.

Tabel 2. Kutipan Hasil Klusterisasi

IP	Timestamp	Request
182.253.163.96	23/Apr/2023:02:40:07 +0000	GET /mod.php?kategori=lounge&id=-9310%20UNION%20ALL%20SELECT%2013%2CCONCAT%280x716b6b7a71%2CJSON_ARRAYAGG%28CONCAT_WS%280x767a70646b6a%2Cgrantee%2Cprivilege_type%29%29%2C0x716a787a71%29%2C13%2C13%2C13%2C13%2C13%20FROM%20INFORMATION_SCHEMA.USER_PRIVILEGES--%20- HTTP/1.1" 200 2092 "-
182.253.163.96	23/Apr/2021:02:41:11 +0000	GET /mod.php?kategori=lounge&id=-1179%20UNION%20ALL%20SELECT%2032%2C32%2C32%2C32%2C32%2C32%2C32%2C32%2C32--%20- HTTP/1.1" 200 2092 "-
182.253.163.96	23/Apr/2021:02:41:25 +0000	GET /mod.php?kategori=lounge&id=2%20AND%20%28SELECT%20%28NULL%20SETEQ%20NULL%29%29%20IS%20NULL HTTP/1.1" 200 2092 "-
...	...	...
182.253.163.96	23/Apr/2021:02:43:29 +0000	GET /mod.php?kategori=lounge&id=-5676%20UNION%20ALL%20SELECT%2032%2C32%2CCONCAT%280x716a766271%2C0x537a686e714869424269625 6796a59596d46776442596e45666f6d5272666b486e4549494d796c62%2C0x716a767071%29%2C32%2C32%2C32%2C32--%20- HTTP/1.1" 200 2410 "-

KESIMPULAN

Pada penelitian ini bertujuan menentukan klusterisasi serangan *SQL Injection* menggunakan kerangka NIST SP 800-61R2 dan klusterisasi K-Means dalam upaya melakukan analisis serangan siber yang komprehensif dan mempermudah analis keamanan siber. Untuk melakukan analisis serangan menggunakan kerangka NIST SP 800-61R2 sebagai fundamental penanganan serangan siber yang diimprovisasi mengunakan pendekatan klusterisasi K-Means. Berdasarkan analisis serangan *SQL Injection* menggunakan K-Means ditemukan 3 kluster serangan yang terbanyak. Maka pada laporan analisis serangan *SQL Injection* akan didukung dari diagram yang dihasilkan pada proses klusterisasi ini. Harapannya analis keamanan siber memiliki wawasan dan upaya yang taktis dalam melakukan penanganan serangans siber secara umum. Proses analisis serangan siber tidak cukup hanya dilakukan klusterisasi saja tetapi perlu tahapan lebih lanjut seperti dilakukan klasifikasi untuk mempermudah dalam deskripsi laporan analisis penanganan serangan siber ke pihak manajerial sehingga dapat dilakukan tindakan mitigasi dan kolaborasi dengan pihak terkait.

UCAPAN TERIMA KASIH

Terima kasih peneliti ucapkan kepada Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat (DRTPM), Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, yang telah memberikan dukungan kepada tim peneliti melalui hibah penelitian pada skema Penelitian Dosen Pemula (PDP). Penelitian ini diajukan sesuai dengan Surat Kontrak 0423.20/LL5-INT/AL.04/2023 dengan pelaksanaan mono tahun pada tahun 2023. Harapan tim peneliti adalah dengan adanya penelitian ini, dapat memberikan manfaat kepada Analis pada *Computer Security Incident Response Team* (CISRT) dan peneliti keamanan siber dalam upaya meminimalisir upaya serangan siber di Indonesia.

DAFTAR PUSTAKA

[1] OWASP, “The Ten Most Critical Web Application Security Risks,” The Open Web Application Security Project (OWASP), 2017.

[2] B. Glas, A. van der Stock, T. Gigler, dan N. Smithline, “OWASP Top 10 2021,” OWASP Project, 2021.

[3] Badan Siber dan Sandi Negara, “Mengenal Sql Injection dan Cara Mencegahnya,” Badan Siber dan Sandi Negara, Jakarta, 2019.

[4] R. Andriani, E. S. Pramukantoro, dan M. Data, “Pengembangan Sistem Visualisasi Access Log untuk Mengetahui Informasi Aktivitas Pengunjung pada Sebuah Website,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 6, hlm. 2104–2112, 2018.

[5] S. H. N. Harip, I. R. A. Hamid, N. Murli, dan N. Hassan, “Classification of SQL injection attack using K-Means clustering algorithm,” dipresentasikan pada International Conference on Applied Science And Technology, Kuala Lumpur, Malaysia, 2022, hlm. 040004. doi: 10.1063/5.0104348.

[6] I. S. Crespo-Martínez, A. Campazas-Vega, Á. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Álvarez-Aparicio, dan C. Fernández-Llamas, "SQL injection attack detection in network flow data," *Computers & Security*, vol. 127, hlm. 103093, Apr 2023, doi: 10.1016/j.cose.2023.103093.

[7] P. Cichonski, T. Millar, T. Grance, dan K. Scarfone, "NIST Special Publication 800-61 Revision 2: Computer Security Incident Handling Guide Recommendations," 2012. doi: 10.6028/NIST.SP.800-61r2.

[8] A. D. Djayali, "Analisa Serangan SQL Injection pada Server pengisian Kartu Rencana Studi (KRS) Online," *Jurnal Ilmiah Manajemen Informatika & Komputer*, vol. 1, no. 1, hlm. 16–24, 2020.

[9] M. Zulfadhilah, Y. Prayudi, dan I. Riadi, "Cyber Profiling using Log Analysis and K-Means Clustering A Case Study Higher Education in Indonesia," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 7, hlm. 430–435, 2016.

[10] R. Gelar Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7," *JTEKSIS*, vol. 5, no. 1, hlm. 55–60, Feb 2023, doi: 10.47233/jteksis.v5i1.750.

[11] A. Handa, R. Negi, dan S. K. Shukla, *Implementing Enterprise Cybersecurity with Open-Source Software and Standard Architecture*. River Publishers, 2021.

[12] R. Ramadhan, Y. A. Sari, dan P. P. Adikara, "Perbandingan Pembobotan Term Frequency-Inverse Document Frequency dan Term Frequency-Relevance Frequency terhadap Fitur N-Gram pada Analisis Sentimen," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, hlm. 5075–5079, 2021.

[13] A. Lahitani, U. S. Aesyi, N. Wulandari, dan B. D. Santosa, "Cosine Similarity untuk Mengukur Tingkat Kesadaran pada Topik Software Security Berbasis Teks Komentar di Media Sosial Youtube," *JSI*, vol. 8, no. 2, Des 2022, doi: 10.34128/jsi.v8i2.535.

[14] S. M. Fani, R. Santoso, dan S. Suparti, "Penerapan Text Mining untuk Melakukan Clustering Data Tweet Akun Blibli Pada Media Sosial Twitter Menggunakan K-Means Clustering," *Jurnal Gaussian*, vol. 10, no. 4, hlm. 583–593, Des 2021, doi: 10.14710/j.gauss.10.4.583-593.

[15] S. Handoko, F. Fauziah, dan E. T. E. Handayani, "Implementasi Data Mining Untuk Menentukan Tingkat Penjualan Paket Data Telkomsel Menggunakan Metode K-Means Clustering," *tekno*, vol. 25, no. 1, hlm. 76–88, 2020, doi: 10.35760/tr.2020.v25i1.2677.

[16] R. Ananda dan Achmad Zaki Yamani, "Penentuan Centroid Awal K-means pada proses Clustering Data Evaluasi Pengajaran Dosen," *RESTI*, vol. 4, no. 3, hlm. 544–550, Jun 2020, doi: 10.29207/resti.v4i3.1896.

[17] S. Chowdhury, A. Nandi, M. Ahmad, A. Jain, dan M. Pawar, "A Comprehensive Survey for Detection and Prevention of SQL Injection," *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, hlm. 434–437, 2021, doi: 10.1109/ICACCS51430.2021.9442012.

[18] J. Hu, W. Zhao, dan Y. Cui, "A Survey on SQL Injection Attacks, Detection and Prevention," *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, hlm. 483–488, 2020, doi: 10.1145/3383972.3384028.

[19] A. M. Al Hilmi dan E. Khujaemah, "Network Security Monitoring With Intrusion Detection System," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 2, hlm. 249–253, 2022, doi: 10.20884/1.jutif.2022.3.2.117.

[20] D. Hariyadi, M. Kusuma, A. Sholeh, dan Fazlurrahman, "Digital Forensics Investigation on Xiaomi Smart Router Using SNI ISO/IEC 27037:2014 and NIST SP 800-86 Framework," dalam *Proceedings of the International Conference on Science and Engineering (ICSE-UIN-SUKA 2021)*, Yogyakarta, 2021. doi: 10.2991/aer.k.211222.023.

[21] W. N. I. Al-Obaydy, H. A. Hashim, Y. A. Najm, dan A. A. Jalal, "Document classification using term frequency-inverse document frequency and K-means clustering," *IJEECS*, vol. 27, no. 3, hlm. 1517, Sep 2022, doi: 10.11591/ijeecs.v27.i3.pp1517-1524.

[22] L. Wirz, R. Tanthanathewin, A. Ketphet, dan S. Fugkeaw, "Design and Development of A Cloud-Based IDS using Apache Kafka and Spark Streaming," dalam *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Bangkok, Thailand: IEEE, Jun 2022, hlm. 1–6. doi: 10.1109/JCSSE54890.2022.9836264.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License