JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

Klasifikasi Tingkat Kerentanan Website Berdasarkan Hasil Vulnerability Assessment Menggunakan Algoritma K-Means *Clustering*

Dandi Saputra[1], Harry Dhika[2], Siti Fuadah[3]

[1,2,3]Fakultas Teknik dan Ilmu Komputer, Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia
Email*: [1]dandisaputra0122@gmail.com, [2*]dhikatr@yahoo.com, [3]fuadah85@gmail.com
(*: coressponding author)

Abstrak−Tujuan dari penelitian ini adalah untuk merancang dan membangun sistem pendukung keputusan (SPK) yang mampu mengklasifikasikan tingkat kerentanan website berdasarkan hasil *vulnerability assessment* menggunakan algoritma K-Means *Clustering*. Sistem ini bertujuan untuk mengelompokkan kerentanan website menjadi beberapa kategori risiko yang dapat digunakan untuk memprioritaskan tindakan mitigasi. Algoritma K-Means *Clustering* digunakan untuk mengelompokkan data hasil *assessment* yang mencakup jenis kerentanan, tingkat keparahan (*severity*), dan skoring CVSS. Sistem ini dibangun menggunakan bahasa pemrograman Java (*NetBeans*) dan basis data MySQL, serta menghasilkan keluaran berupa kategori tingkat kerentanan yang memudahkan tim keamanan dalam mengambil keputusan. Hasil penelitian menunjukkan bahwa sistem ini mampu memberikan klasifikasi tingkat kerentanan secara efisien dan objektif, yang dapat membantu dalam pengelolaan risiko dan pemilihan prioritas perbaikan kerentanan website secara lebih tepat sasaran.

Kata Kunci: Klasifikasi Tingkat Kerentanan, *Vulnerability Assessment*, K-Means *Clustering*.

Abstract−*The purpose of this research is to design and build a decision support system (DSS) capable of classifying website vulnerability levels based on vulnerability assessment results using the K-Means Clustering algorithm. This system aims to group website vulnerabilities into several risk categories that can be used to prioritize mitigation actions. The K-Means Clustering algorithm is used to group assessment data that includes vulnerability type, severity level, and CVSS scoring (if available). This system is built using the Java programming language (NetBeans) and a MySQL database, and produces output in the form of vulnerability level categories that facilitate security teams in making decisions. The results of the study show that this system is capable of providing efficient and objective vulnerability level classification, which can assist in risk management and selecting priorities for website vulnerability repairs more precisely*.

Keywords: *Vulnerability Level Classification, Vulnerability Assessment, K-Means Clustering.*

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong meningkatnya penggunaan aplikasi berbasis web pada berbagai sektor, termasuk pendidikan dan industri digital (Chen & Babar, 2024). Namun, peningkatan penggunaan website juga berdampak pada meningkatnya risiko serangan siber dan eksploitasi kerentanan (Harzevili *et al.*, 2023). Salah satu dampak nyata adalah meningkatnya penggunaan aplikasi berbasis web dalam aktivitas sehari-hari, baik dalam bidang pendidikan, perdagangan, layanan publik, hingga keuangan. Website tidak hanya berfungsi sebagai media informasi, tetapi juga sebagai sistem transaksi yang menyimpan dan mengelola data penting dan sensitif. Perkembangan pesat *Machine learning* (ML) telah menunjukkan kinerja yang unggul di berbagai bidang, seperti penglihatan komputer, pengenalan video, dan pengenalan suara. Saat ini, ML semakin banyak dimanfaatkan dalam sistem perangkat lunak untuk mengotomatisasi tugas-tugas inti (Chen & Babar, 2024). Namun, peningkatan penggunaan website juga dibarengi dengan meningkatnya ancaman terhadap keamanan siber, khususnya pada sisi kerentanan (*vulnerability*) website itu sendiri. Banyak kasus kebocoran data dan eksploitasi sistem yang terjadi akibat adanya celah keamanan yang tidak terdeteksi atau tidak tertangani dengan baik. Oleh karena itu, proses *vulnerability assessment* deteksi kerentanan perangkat lunak sangat penting dalam keamanan perangkat lunak karena mengidentifikasi potensi bug dalam sistem perangkat lunak, memungkinkan tindakan perbaikan dan mitigasi segera diterapkan sebelum kerentanan tersebut dapat dieksploitasi

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

(Harzevili *et al.*, 2023) menjadi langkah penting dalam mengidentifikasi dan menilai celah keamanan yang ada pada sebuah sistem berbasis web.

Salah satu tantangan dalam proses *vulnerability assessment* adalah bagaimana mengelompokkan hasil temuan kerentanan menjadi tingkat risiko yang terklasifikasi dengan baik. Pengelompokan ini akan memudahkan tim keamanan siber dalam menentukan prioritas penanganan, sekaligus membantu pengambilan keputusan dalam proses mitigasi. Oleh karena itu, metode analisis data seperti *Clustering* sangat relevan untuk digunakan dalam konteks ini (Bagui *et al.*, 2025) Metode K-Means *Clustering* merupakan salah satu algoritma *unsupervised learning* yang sering digunakan untuk pengelompokan data berdasarkan kesamaan karakteristik (Shahid *et al.*, 2022) Dengan menerapkan metode ini pada hasil *vulnerability assessment*, diharapkan dapat diperoleh klasifikasi tingkat kerentanan yang lebih sistematis dan objektif, sehingga pengelolaan keamanan web dapat dilakukan secara efisien.Penelitian ini dilakukan untuk mengembangkan sebuah pendekatan klasifikasi terhadap tingkat kerentanan website menggunakan hasil dari *vulnerability assessment* yang dianalisis melalui algoritma K-Means. Pendekatan ini diharapkan mampu memberikan gambaran yang lebih jelas terhadap sebaran tingkat risiko dari celah keamanan yang ditemukan.

2. METODE

Penelitian ini menggunakan eksperimental untuk menganalisis pola kerentanan keamanan aplikasi web menggunakan metode *K-Means Clustering*. Pendekatan eksperimental ini umum digunakan dalam riset *machine learning* karena melibatkan manipulasi data dan evaluasi hasil model.(Sarker *et al.*, 2020) Algoritma K-Means bekerja melalui beberapa langkah utama, yaitu:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - c_{jk})^2}$$

Keterangan:

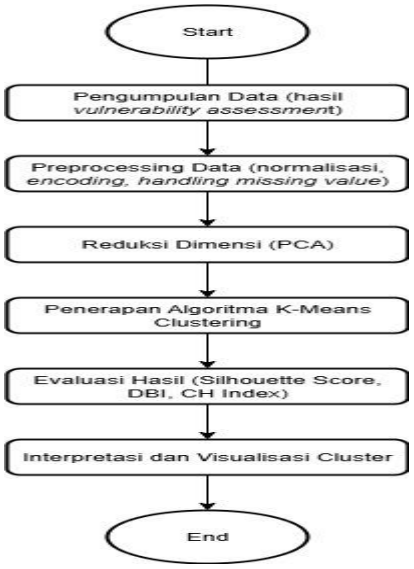
x_i = data ke-i
cj = *centroid Cluster* ke-j
n = jumlah fitur

Tahapan algoritm *K-Means Clustering:*

a. Menentukan jumlah klaster (k) yang ingin dibentuk.
b. Memilih secara acak titik pusat awal (*centroid*) untuk setiap klaster.
c. Menghitung jarak setiap data terhadap masing-masing *centroid* menggunakan rumus *Euclidean Distance*.
d. Mengelompokkan data ke klaster yang memiliki jarak terdekat dengan *centroid* tersebut.
e. Menghitung ulang posisi *centroid* berdasarkan rata-rata data dalam klaster baru.
f. Proses langkah 3–5 diulang hingga posisi *centroid* tidak lagi berubah secara signifikan (konvergen).

K-Means merupakan salah satu algoritma *unsupervised learning* yang paling banyak digunakan karena kesederhanaan dan efisiensinya dalam pengelompokan data. (Arnab Saha, 2021) Setelah pengelompokan K-means dilakukan menggunakan data yang tidak dilabeli, label dari *Dataset* yang dilabeli digunakan untuk menentukan titik-titik kluster mana yang merupakan serangan, dan ini digunakan untuk memprediksi kelas kluster tersebut. Pendekatan ini juga digunakan dalam penelitian vulnerability *Clustering* berbasis semantik. (Stehr & Kim, 2023) Pengelompokan K-means juga belum pernah digunakan pada data berbasis koneksi dalam penelitian sebelumnya (Bagui *et al.*, 2025). Metodologi penelitian yang digunakan mengintegrasikan teknik data mining dengan analisis keamanan siber untuk memberikan pemahaman yang lebih mendalam tentang karakteristik kerentanan aplikasi web (Heiding *et al.*, 2023) Penelitian ini bersifat deskriptif-analitis yang bertujuan untuk mengeksplorasi dan mengelompokkan jenis-jenis kerentanan berdasarkan karakteristik risiko, tingkat keyakinan, dan parameter teknis lainnya.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126



Gambar 1. Flowchart Tahapan Metode Penelitian Menggunakan Algoritma K-Means *Clustering*

Tahapan metode penelitian ini dimulai dari proses pengumpulan data hasil *vulnerability assessment* yang diperoleh melalui pemindaian terhadap website target. Selanjutnya dilakukan *preprocessing* data untuk memastikan data bersih dan siap dianalisis melalui proses normalisasi dan penghapusan duplikasi. Selanjutnya dilakukan reduksi dimensi menggunakan metode *Principal Component Analysis* (PCA) untuk menyederhanakan variabel tanpa kehilangan informasi penting (Landauer *et al.*, 2020).

$$z = x \, . w$$

Keterangan:

Z = data hasil proyeksi
X = data awal
W = matriks *eigenvector*

Data diambil dari situs: https://demo.owasp-juice.shop/#/. Data yang telah siap kemudian melalui tahap reduksi dimensi menggunakan PCA (*Principal Component Analysis*) untuk menyederhanakan variabel tanpa mengurangi informasi penting. Tahap berikutnya adalah penerapan algoritma *K-Means Clustering* untuk mengelompokkan tingkat kerentanan berdasarkan karakteristik parameter risiko. Setelah proses pengelompokan selesai, dilakukan evaluasi hasil *Clustering* menggunakan metrik seperti *Silhouette Score* dan *Davies-Bouldin Index* guna menilai kualitas pemisahan antar *Cluster* (Bennouk *et al.*, 2024;Chorell & Ekberg, 2024) (Prabowo & Dhika, 2021). Tahap terakhir adalah interpretasi hasil dan visualisasi, yang menampilkan kategori tingkat risiko (tinggi, sedang, rendah) untuk mendukung pengambilan keputusan mitigasi keamanan.

3. ANALISA DAN PEMBAHASAN

3.1 Pembahasan

Proses *Clustering* dilakukan menggunakan algoritma K-Means dengan jumlah klaster (k) = 3, yang ditentukan berdasarkan hasil uji nilai *Elbow Method* dan *Silhouette Score* (Bennouk *et al.*, 2024;Bagui *et al.*, 2025). Pendekatan serupa juga digunakan dalam mendeteksi ancaman keamanan pada sistem Internet of Things (IoT) menggunakan kombinasi algoritma *machine learning*, yang menunjukkan efektivitas pengelompokan berbasis karakteristik risiko (Alfahaid *et al.*, 2025). Data hasil *vulnerability assessment* terlebih dahulu melalui tahap preprocessing seperti normalisasi nilai risiko (*Risk*) dan tingkat kepercayaan (*Confidence*) untuk meningkatkan kualitas analisis (Landauer *et al.*, 2020). Setiap data kerentanan direpresentasikan dalam bentuk numerik untuk memudahkan proses perhitungan jarak menggunakan rumus *Euclidean Distance* (Arnab Saha, 2021). Selanjutnya, sistem mengelompokkan data yang memiliki karakteristik risiko serupa ke dalam klaster yang sama.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

Hasil akhir pengelompokan menunjukkan pembagian kerentanan ke dalam tiga kategori utama yang merepresentasikan tingkat risiko: tinggi, sedang, dan rendah (Sarker *et al.*, 2020).

Berdasarkan hasil *Clustering*:

1. *Cluster 1 – High-Risk Active Attacks*

Cluster ini berisi kerentanan dengan tingkat risiko tinggi (*High*) yang dapat dieksploitasi secara langsung untuk mengakses atau memanipulasi data sensitif. Jenis serangan pada *Cluster* ini mencakup *SQL Injection* dan *Cross-Site Scripting* (XSS) yang termasuk dalam kategori tertinggi pada OWASP Top 10 (OWASP, 2025) memiliki dampak serius pada integritas, kerahasiaan, dan ketersediaan sistem. Kerentanan pada *Cluster* ini memerlukan prioritas penanganan tertinggi karena dapat dieksploitasi dengan tingkat keberhasilan tinggi dan dampak yang signifikan. Oleh karena itu, kerentanan dalam *Cluster* ini memerlukan prioritas penanganan tertinggi dengan implementasi *input sanitization* dan *parameterized query* (Satya *et al.*, 2024).

Contoh Kerentanan:
a. *SQL Injection*: Penyerang dapat menyisipkan perintah SQL berbahaya melalui parameter input sehingga memungkinkan akses atau manipulasi basis data.
b. *Cross-Site Scripting (XSS – Reflected)*: penyerang menyisipkan skrip berbahaya yang dieksekusi di browser korban, memungkinkan pencurian cookie atau data sensitif.
c. *Unencrypted Login Request:* data kredensial dikirimkan tanpa enkripsi, memungkinkan *man-in-the-middle attack*.

Implikasi Keamanan:
a. Pengambilalihan akun pengguna atau admin.
b. Kebocoran data pribadi atau rahasia perusahaan.
c. Penyusupan ke basis data yang dapat merusak integritas informasi.
d. Kerugian reputasi dan potensi pelanggaran hukum terkait perlindungan data.

2. *Cluster 0 – Configuration Weaknesses*

Cluster ini mencakup kelemahan konfigurasi yang tidak langsung menyebabkan serangan, tetapi membuka peluang bagi penyerang untuk mengeksploitasi sistem melalui teknik lanjutan atau serangan berantai Kerentanan ini biasanya berasal dari pengaturan sistem yang tidak aman atau default yang tidak diubah.

Contoh kerentanan:
a. *Information Disclosure*: File konfigurasi atau data sensitif dapat diakses secara publik, seperti *.git/config*.
b. *CSRF (Cross-Site Request Forgery):* Penyerang memanfaatkan sesi login korban untuk menjalankan perintah tanpa izin.
c. *Directory Browsing Enabled*: Penyerang dapat melihat daftar file dan direktori yang seharusnya tersembunyi.

Potensi Kombinasi Serangan:
a. *CSRF + XSS*: Digabungkan untuk memaksa korban mengeksekusi skrip berbahaya saat mengakses halaman tertentu.
b. *Information Disclosure + SQL Injection*: Data konfigurasi bocor digunakan untuk mempermudah eksekusi serangan *SQL Injection*.
c. *Directory Browsing + File Upload*: Memungkinkan penyerang mengunggah skrip berbahaya dan mengeksekusinya langsung.

3. *Cluster 2 – Low-Risk Implementation Issues*

Cluster ini berisi kerentanan dengan risiko rendah seperti *Insecure Cookie* dan *Missing Security Headers*. Meskipun tidak langsung dimanfaatkan untuk mendapatkan akses ilegal, isu ini tetap berpengaruh terhadap keamanan keseluruhan sistem (Al Moaiad *et al.*, 2022). Studi Bennouk *et al.* (2024) juga menegaskan pentingnya *hardening* lapisan aplikasi melalui penerapan *Content-*

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

Security-Policy (CSP), *X-Frame-Options*, dan *Strict-Transport-Security* (HSTS) untuk mengurangi risiko eksposur data.

Contoh kerentanan:

a. *Insecure Cookie*: *Cookie* tidak diberi atribut *Secure* atau *HttpOnly*, meningkatkan risiko pencurian cookie.
b. *Missing Security Headers*: Tidak adanya header keamanan seperti *Content-Security-Policy* atau *X-Frame-Options* yang melindungi dari serangan umum.

Rekomendasi Perbaikan:

a. Mengaktifkan atribut keamanan pada *cookie* (*Secure*, *HttpOnly*, *SameSite*).
b. Menambahkan header keamanan standar seperti:
    1. *Content-Security-Policy* (CSP) untuk mencegah XSS.
    2. *X-Frame-Options* untuk mencegah *clickjacking*.
    3. *Strict-Transport-Security* (HSTS) untuk memaksa penggunaan HTTPS.
c. Melakukan audit keamanan rutin untuk memastikan perubahan konfigurasi tetap aman.

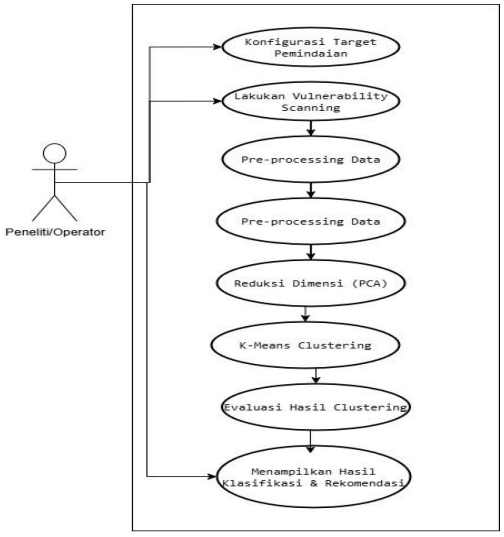
3.2 Implementasi Jangka Panjang dan Rekomendasi Strategis

Implementasi metode *Clustering* untuk klasifikasi kerentanan tidak hanya relevan untuk penanganan insiden saat ini, tetapi juga memberikan manfaat strategis dalam pengelolaan keamanan jangka panjang.

1. Penguatan Kebijakan Keamanan
   Data klasifikasi dapat menjadi landasan pembaruan kebijakan keamanan internal perusahaan atau organisasi, termasuk *patch management* dan audit berkala.
2. Optimalisasi Sumber Daya
   Penentuan prioritas berbasis data membantu memfokuskan sumber daya teknis dan finansial pada area yang paling membutuhkan perbaikan.
3. Perluasan Model Analisis
   Ke depan, model *Clustering* ini dapat dikombinasikan dengan metode *predictive analytics* untuk memprediksi tren kerentanan dan mencegah potensi serangan sebelum terjadi.

3.3 Permodelan Perangkat Lunak

1. *Use Case Diagram*



Gambar 2. *Use Case* Diagram

Use Case Diagram menggambarkan hubungan interaksi antara pengguna (Peneliti/Operator) dengan sistem klasifikasi tingkat kerentanan website yang dibangun. Diagram ini menunjukkan aktivitas utama yang dilakukan dalam proses analisis mulai dari konfigurasi target pemindaian

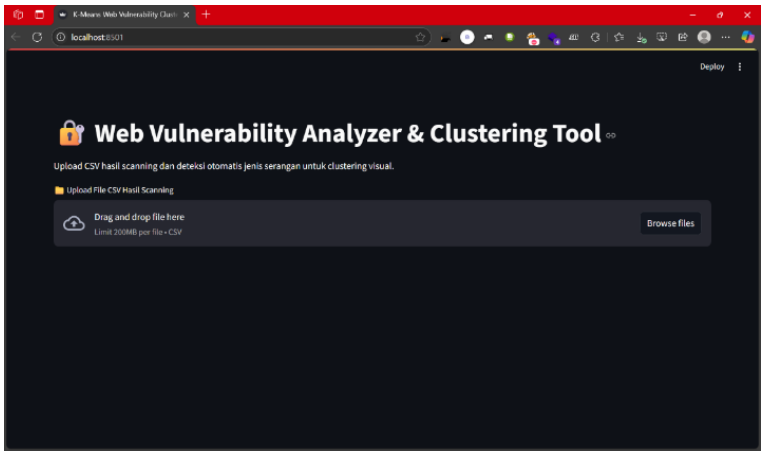
JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

hingga penampilan hasil klasifikasi. Setiap tahapan menggambarkan alur kerja sistem berdasarkan metode K-Means *Clustering* yang digunakan dalam penelitian ini.

Berdasarkan Gambar 2, aktor utama yaitu Peneliti/Operator berperan untuk menginisiasi seluruh proses mulai dari konfigurasi target pemindaian, pelaksanaan *vulnerability scanning*, hingga pengolahan data hasil pemindaian. Setelah itu, sistem melakukan pre-processing data, reduksi dimensi menggunakan PCA (*Principal Component Analysis*), dan penerapan algoritma K-Means *Clustering* untuk mengelompokkan tingkat risiko. Langkah terakhir adalah evaluasi hasil *Clustering* serta penampilan hasil klasifikasi dan rekomendasi mitigasi kepada pengguna. Diagram ini membantu memvisualisasikan keseluruhan fungsi sistem secara sistematis dan mudah dipahami.

3.4 Tahapan Penggunaan Aplikasi dan Penjelasannya

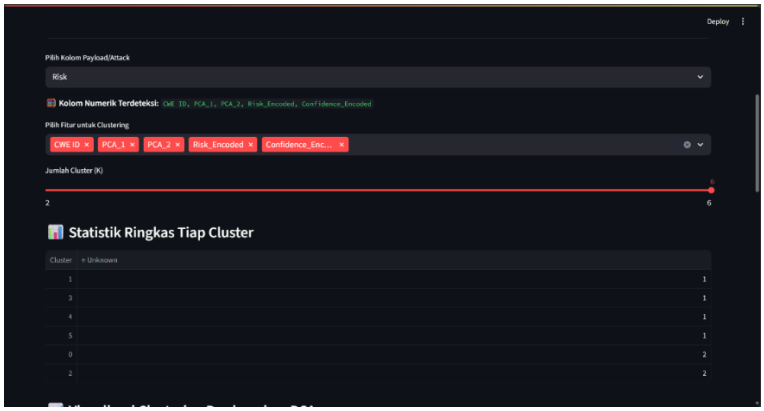
1. Upload File CSV Hasil Scanning



Gambar 3. Upload file CSV

Gambar 3 menunjukkan tampilan awal aplikasi *Web Vulnerability Analyzer* & *Clustering Tool* yang digunakan dalam penelitian ini. Pada tahap ini, pengguna atau peneliti dapat mengunggah file hasil *vulnerability assessment* dalam format .CSV melalui tombol Browse files atau dengan cara *drag and drop* ke area unggahan. File CSV tersebut berisi data hasil pemindaian dari tools seperti OWASP ZAP atau Burp Suite, yang mencakup informasi kerentanan seperti tingkat risiko (*Risk*), tingkat kepercayaan (*Confidence*), dan parameter serangan (*Attack Vector*). Tahap ini merupakan langkah pertama sebelum sistem melakukan proses *preprocessing* data dan pengelompokan menggunakan algoritma K-Means *Clustering*.

2. Memilih Kolom Payload/Attack

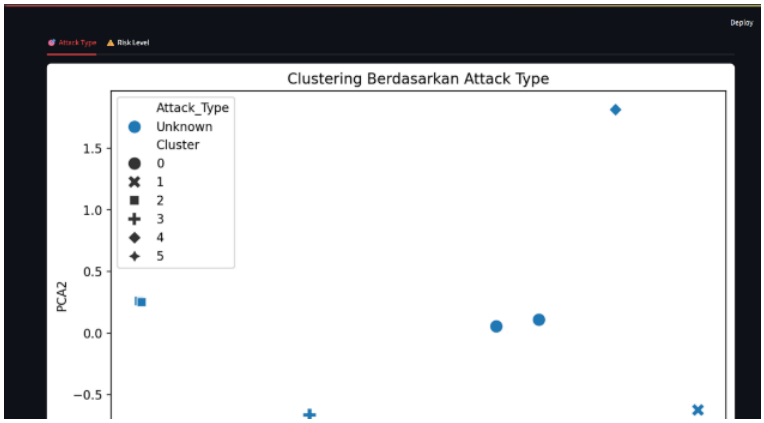


Gambar 4. Memilih kolom *Payload/Attack*

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

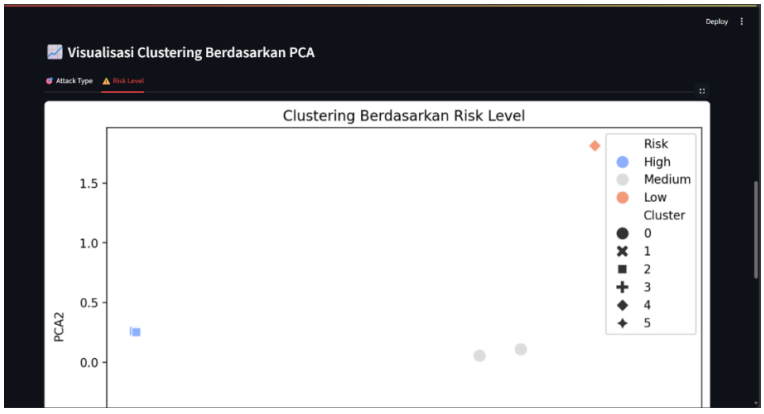
Gambar 4 menampilkan tahap kedua dalam penggunaan aplikasi, yaitu proses pemilihan kolom yang berisi data *payload* atau *attack* vector dari file CSV yang telah diunggah sebelumnya. Tahap ini berfungsi untuk menentukan sumber data yang akan dianalisis oleh sistem guna mendeteksi jenis serangan secara otomatis. Aplikasi akan menjalankan fungsi detect_*attack*_type() yang bertugas mengenali pola serangan seperti XSS, SQL *Injection*, LFI, dan RCE berdasarkan isi *payload* yang terdeteksi. Melalui tahap ini, sistem dapat mengklasifikasikan setiap entri kerentanan sesuai pola serangan yang ada, sebelum masuk ke proses pengkodean risiko (*Risk Encoding*) dan pengelompokan menggunakan algoritma K-Means *Clustering*.

3. Reduksi Dimensi & Visualisasi PCA



Gambar 5. Visualisasi *Attack* Type

Gambar 5 menampilkan hasil visualisasi *Clustering* berdasarkan jenis serangan (*Attack* Type) setelah dilakukan reduksi dimensi menggunakan metode Principal Component Analysis (PCA). Proses ini mengubah data berdimensi tinggi menjadi dua komponen utama, yaitu PCA1 dan PCA2, untuk mempermudah interpretasi secara visual. Setiap titik pada grafik merepresentasikan satu entri kerentanan, sedangkan warna dan bentuk titik menunjukkan kategori serangan seperti XSS, SQL Injection, LFI, atau RCE. Dengan visualisasi ini, peneliti dapat mengamati persebaran serangan dalam setiap klaster dan melihat kecenderungan pola serangan yang dominan.



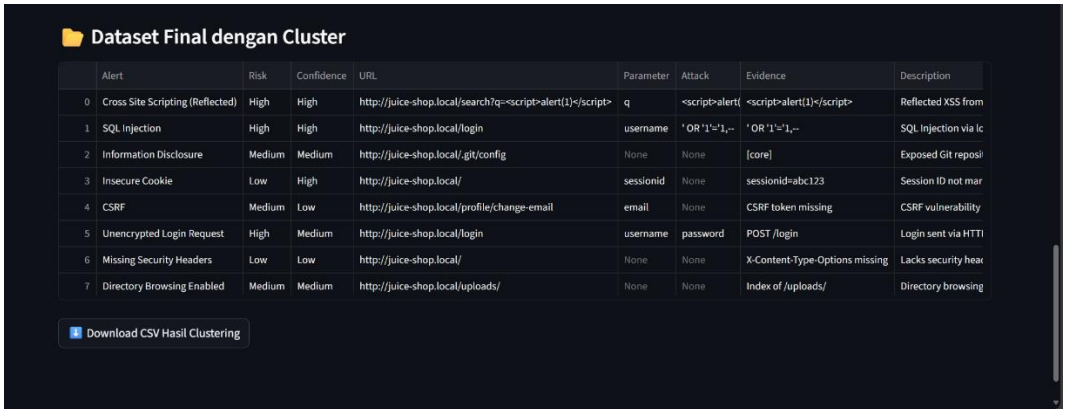
Gambar 6. Visualisasi Risk Level

Gambar 6 menggambarkan hasil *Clustering* yang ditampilkan berdasarkan tingkat risiko (Risk Level) dengan menggunakan hasil reduksi dimensi PCA yang sama. Dalam visualisasi ini, warna titik menandakan tingkat risiko — merah untuk High, biru untuk Medium, dan abu-abu untuk Low — sedangkan bentuk titik menunjukkan posisi klaster yang terbentuk. Visualisasi ini membantu menilai hubungan antara tingkat risiko dengan distribusi data pada tiap klaster. Melalui pendekatan ini, sistem dapat mempermudah proses identifikasi kerentanan yang memiliki potensi

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

ancaman tertinggi serta memprioritaskan tindakan mitigasi berdasarkan hasil pengelompokan tersebut.

4. Melihat Dataset Final & Download



	Alert	Risk	Confidence	URL	Parameter	Attack	Evidence	Description
0	Cross Site Scripting (Reflected)	High	High	http://juice-shop.local/search?q=<script>alert(1)</script>	q	<script>alert(	<script>alert(1)</script>	Reflected XSS from
1	SQL Injection	High	High	http://juice-shop.local/login	username	' OR '1'='1,--	' OR '1'='1,--	SQL Injection via lc
2	Information Disclosure	Medium	Medium	http://juice-shop.local/.git/config	None	None	[core]	Exposed Git reposi
3	Insecure Cookie	Low	High	http://juice-shop.local/	sessionid	None	sessionid=abc123	Session ID not mar
4	CSRF	Medium	Low	http://juice-shop.local/profile/change-email	email	None	CSRF token missing	CSRF vulnerability
5	Unencrypted Login Request	High	Medium	http://juice-shop.local/login	username	password	POST /login	Login sent via HTTI
6	Missing Security Headers	Low	Low	http://juice-shop.local/	None	None	X-Content-Type-Options missing	Lacks security heac
7	Directory Browsing Enabled	Medium	Medium	http://juice-shop.local/uploads/	None	None	Index of /uploads/	Directory browsing

Gambar 7. *Dataset* Final dengan *Cluster*

Gambar 7 menampilkan hasil akhir dari proses *Clustering* yang dilakukan menggunakan algoritma K-Means. *Dataset* ini merupakan hasil integrasi antara data mentah hasil *vulnerability assessment* dengan proses *preprocessing*, *encoding*, serta reduksi dimensi PCA. Kolom yang dihasilkan meliputi *Attack_Type*, *Risk_Encoded*, *Confidence_Encoded*, *Cluster*, PCA1, dan PCA2, yang merepresentasikan hasil klasifikasi tiap kerentanan ke dalam klaster risiko rendah, sedang, dan tinggi. Tampilan ini membantu peneliti untuk memahami distribusi kerentanan berdasarkan tingkat risiko serta jenis serangan yang terdeteksi. Selain itu, fitur *Download* CSV Hasil *Clustering* disediakan agar pengguna dapat menyimpan hasil analisis ke dalam file CSV untuk kebutuhan dokumentasi atau analisis lanjutan.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan mengenai, dapat disimpulkan beberapa hal sebagai berikut: metode K-Means *Clustering* terbukti mampu mengelompokkan kerentanan secara efektif berdasarkan kesamaan karakteristik parameter risiko, tingkat kepercayaan (*confidence*), dan tipe serangan yang ditemukan. Hasil pengelompokan menghasilkan tiga *Cluster* utama yang merepresentasikan tingkatan risiko: tinggi, sedang, dan rendah. Tahap pra-pemrosesan data (data *preprocessing*) dan reduksi dimensi (PCA) berperan penting dalam meningkatkan akurasi visualisasi serta memudahkan interpretasi hasil *Clustering*. Proses normalisasi dan pengkodean kategori membuat data lebih seragam dan siap digunakan dalam proses pembelajaran mesin. Evaluasi model menggunakan *Silhouette Score*, *Davies-Bouldin Index* (DBI), dan *Calinski-Harabasz Index* menunjukkan bahwa hasil *Clustering* yang diperoleh berada pada tingkat kualitas yang baik, dengan *Silhouette Score* yang mengindikasikan pemisahan antar *Cluster* yang jelas. Integrasi hasil analisis dengan proses *vulnerability assessment* memberikan gambaran yang lebih terstruktur terkait prioritas mitigasi. Kerentanan dengan risiko tinggi yang teridentifikasi pada *Cluster* tertentu dapat langsung diarahkan untuk penanganan prioritas oleh tim keamanan. Berdasarkan temuan dalam penelitian ini, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah: Penggunaan *Dataset* yang lebih beragam dan berskala besar, sehingga model K-Means dapat diuji terhadap variasi pola serangan yang lebih kompleks dan heterogen. Integrasi metode *feature selection* atau *dimensionality reduction* yang lebih canggih, seperti t-SNE atau UMAP, untuk mendapatkan visualisasi dan pemisahan *Cluster* yang lebih optimal. Eksperimen dengan algoritma *Clustering* lain, seperti DBSCAN atau *Hierarchical Clustering*, untuk membandingkan performa dan stabilitas hasil pengelompokan terhadap data kerentanan. Pengembangan sistem berbasis web yang terotomasi, sehingga proses *vulnerability assessment*, analisis *Clustering*, dan pembuatan laporan dapat berjalan secara terpadu tanpa perlu menjalankan setiap tahap secara manual.

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 8, Januari Tahun 2026
ISSN 3025-0919 (media online)
Hal 2118-2126

REFERENCES

Al Moaiad, Y., Matar, N., Hassan Hassan, A., ABaker El-Ebiary, Y., Zawaideh, F. H., Mohamed Abdelrahman Tarshany, Y., & Ts, A. (2022). Cyber Attack detection Using K-means Machine Learning. *International Journal of Special Education*, *37*(3), 6570–6579.

Alfahaid, A., Alalwany, E., Almars, A. M., Alharbi, F., Atlam, E., & Mahgoub, I. (2025). Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey. *Sensors*, *25*(11), 1–48. https://doi.org/10.3390/s25113341

Arnab Saha. (2021). *K-Means Cluster and It'S Use Case in Cyber Security.* Medium.

Bagui, S. S., Carvalho, G. C. S. De, Mishra, A., Mink, D., Bagui, S. C., & Eager, S. (2025). Detecting Cyber Threats in UWF-ZeekDataFall22 Using K-Means Clustering in the Big Data Environment. *Future Internet*, *17*(6). https://doi.org/10.3390/fi17060267

Bennouk, K., Ait Aali, N., El Bouzekri El Idrissi, Y., Sebai, B., Faroukhi, A. Z., & Mahouachi, D. (2024). A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies. *Journal of Cybersecurity and Privacy*, *4*(4), 853–908. https://doi.org/10.3390/jcp4040040

Chen, H., & Babar, M. A. (2024). Security for Machine Learning-based Software Systems: A Survey of Threats, Practices, and Challenges. *ACM Computing Surveys*, *56*(6). https://doi.org/10.1145/3638531

Chorell, I., & Ekberg, C. (2024). *A Comparative Analysis of Open Source Dynamic Application Security Testing Tools*. 59.

Harzevili, N. S., Belle, A. B., Wang, J., Wang, S., Ming, Z., Jiang, & Nagappan, N. (2023). *A Survey on Automated Software Vulnerability Detection Using Machine Learning and Deep Learning*. *37*(4).

Heiding, F., Katsikeas, S., & Lagerström, R. (2023). Research communities in cyber security vulnerability assessments: A comprehensive literature review. *Computer Science Review*, *48*, 100551. https://doi.org/10.1016/j.cosrev.2023.100551

Landauer, M., Skopik, F., Wurzenberger, M., & Rauber, A. (2020). System log clustering approaches for cyber security applications: A survey. *Computers and Security*, *92*, 101739. https://doi.org/10.1016/j.cose.2020.101739

OWASP. (2025). *OWASP Top Ten*. OWASP.

Prabowo, J. A., & Dhika, H. (2021). Safe Routing Model and Balanced Load Model for Wireless Sensor Network. In *Cyberspace: Jurnal Pendidikan Teknologi Informasi* (Vol. 5, Issue 1, p. 44). Universitas Islam Negeri Ar-Raniry. https://doi.org/10.22373/cj.v5i1.8420

Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00318-5

Satya, P., Kiran, S., & Valluri, D. (2024). ^ ĐŝĞŶĤĐĞ ŝđĞĐŀ ^ ĐŝĞŶĤĐĞ ŝđĞĐŀ ScienceDirect Web Application Security through Comprehensive Web Application Security through Comprehensive Vulnerability Assessment Vulnerability Assessment. *Procedia Computer Science*, *230*(2023), 168–182. https://doi.org/10.1016/j.procs.2023.12.072

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences (Switzerland)*, *12*(8). https://doi.org/10.3390/app12084077

Stehr, M.-O., & Kim, M. (2023). *Vulnerability Clustering and other Machine Learning Applications of Semantic Vulnerability Embeddings*.