SISTEM REKOMENDASI *WEB AUTOMATION VULNERABILITY SCANNER* BERBASIS *LARGE LANGUAGE MODEL (LLM)* DAN *CONTENT-BASED FILTERING* PADA *SOFTWARE HOUSE* NEXTY LABS

Ramdhani Hadi Winarno, Afu Ichsan Pradana, Bondan Wahyu Pamekas
Program Studi Teknik Informatika, Universitas Duta Bangsa Surakarta
Jl. Bhayangkara No.55 Kota Surakarta, Jawa Tengah, Indonesia
ramdhanihw420@gmail.com

ABSTRAK

Keamanan aplikasi web menjadi aspek penting dalam pengembangan sistem, seiring meningkatnya ancaman serangan siber. Permasalahan yang dihadapi adalah proses analisis hasil *vulnerability scanning* yang masih dilakukan secara manual sehingga kurang efisien dan sulit dipahami karena data bersifat teknis. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi berbasis *Large Language Model* (LLM) dan *Content-Based Filtering* untuk meningkatkan efisiensi analisis serta menghasilkan rekomendasi mitigasi yang relevan. Metode yang digunakan meliputi *vulnerability scanning* menggunakan Nuclei dan WhatWeb terhadap 12 domain, *preprocessing* data, implementasi LLM melalui Ollama, ekstraksi fitur menggunakan TF-IDF, serta perhitungan kemiripan *menggunakan Cosine Similarity*. Hasil penelitian menunjukkan bahwa sistem mampu menghasilkan rekomendasi mitigasi dengan nilai similarity tertinggi mencapai 0.5431 dan rata-rata *similarity* sebesar 0.50. Selain itu, penggunaan LLM mampu menyederhanakan data teknis menjadi informasi terstruktur sehingga meningkatkan efisiensi proses analisis dibandingkan metode manual. Dengan demikian, pendekatan yang diusulkan terbukti efektif dalam mendukung pengambilan keputusan pada proses *vulnerability scanning*.

Kata kunci : *vulnerability scanning*, LLM, Ollama, *content-based filtering*, TF-IDF, *cosine similarity*

1. PENDAHULUAN

Keamanan aplikasi web menjadi salah satu aspek penting dalam pengembangan sistem informasi modern, seiring dengan meningkatnya penggunaan layanan berbasis internet yang diikuti oleh meningkatnya ancaman serangan siber. Kerentanan pada aplikasi web dapat muncul akibat kesalahan konfigurasi, kelemahan validasi input, maupun bug pada sistem yang tidak terdeteksi selama proses pengembangan [1]. Kondisi ini menuntut adanya proses pengujian keamanan yang efektif dan efisien untuk mengidentifikasi serta meminimalkan potensi risiko yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab.

Software House NEXTY LABS sebagai perusahaan yang bergerak di bidang pengembangan website menghadapi tantangan dalam menjaga keamanan aplikasi yang dihasilkan. Setiap sistem yang dikembangkan perlu melalui proses *vulnerability scanning* untuk memastikan tidak terdapat celah keamanan. Penggunaan *tools* seperti Nuclei dan WhatWeb telah membantu dalam proses deteksi kerentanan secara otomatis [2].

Meskipun demikian, hasil yang dihasilkan oleh *tools* tersebut masih berupa data mentah dalam format teknis yang sulit dipahami, sehingga tetap memerlukan proses interpretasi secara manual. Selain itu, perkembangan *Large Language Model* (LLM) menunjukkan bahwa teknologi ini mampu memahami dan mengolah teks kompleks serta menghasilkan informasi yang lebih terstruktur [3]. Dalam bidang *cybersecurity*, LLM juga dimanfaatkan untuk membantu interpretasi data teknis menjadi lebih mudah dipahami [4].

Penggunaan LLM dalam penelitian ini diimplementasikan melalui *framework* Ollama yang memungkinkan model dijalankan secara lokal tanpa ketergantungan pada layanan berbasis *cloud*. Pendekatan ini memberikan keunggulan dalam aspek keamanan data serta efisiensi dalam pemrosesan informasi [5].

Berdasarkan hasil observasi, proses analisis hasil *vulnerability scanning* di NEXTY LABS masih dilakukan secara manual, sehingga memerlukan waktu yang lama dan kurang efisien. Selain itu, data hasil scanning yang bersifat teknis menyebabkan kesulitan dalam proses interpretasi serta pengambilan keputusan terkait mitigasi yang tepat.

Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi berbasis *Large Language Model* (LLM) dan *Content-Based Filtering* guna meningkatkan efisiensi proses analisis *vulnerability scanning* serta menghasilkan rekomendasi mitigasi yang sesuai dengan karakteristik kerentanan yang ditemukan.

Permasalahan tersebut diselesaikan dengan mengintegrasikan *Large Language Model* (LLM) yang dijalankan menggunakan Ollama untuk melakukan proses *summarization* dan klasifikasi tingkat keparahan (*severity*) terhadap hasil *vulnerability scanning*.

Selanjutnya, metode *Content-Based Filtering* digunakan dengan teknik TF-IDF dan *Cosine Similarity* untuk mengukur tingkat kemiripan antara hasil analisis dan database mitigasi. Pendekatan ini memungkinkan sistem menghasilkan rekomendasi mitigasi berdasarkan tingkat kesamaan karakteristik data yang dianalisis.

Dengan demikian, sistem yang diusulkan diharapkan mampu mengotomatisasi proses analisis serta meningkatkan efisiensi dan akurasi dalam pengambilan keputusan terkait mitigasi kerentanan pada aplikasi web.

2. TINJAUAN PUSTAKA

Beberapa penelitian sebelumnya telah membahas penerapan sistem rekomendasi dan pemanfaatan kecerdasan buatan dalam berbagai bidang, termasuk keamanan siber. *Large Language Model* (LLM) diketahui mampu memahami konteks teks serta melakukan tugas seperti klasifikasi dan *summarization* secara efektif. Selain itu, metode *Content-Based Filtering* (CBF) banyak digunakan dalam sistem rekomendasi berbasis teks karena mampu menghasilkan rekomendasi berdasarkan kesamaan karakteristik menggunakan TF-IDF dan *Cosine Similarity*. Beberapa penelitian lain juga menunjukkan efektivitas pendekatan *Content-Based Filtering* dalam berbagai domain aplikasi.

Di sisi lain, *vulnerability scanner* seperti Nuclei dan WhatWeb memungkinkan deteksi kerentanan secara otomatis, namun hasilnya masih berupa data teknis yang sulit dipahami oleh pengguna *non-security*. Oleh karena itu, penelitian ini mengintegrasikan LLM dan *Content-Based Filtering* untuk menghasilkan pendekatan yang mampu menginterpretasikan hasil *scanning* serta memberikan rekomendasi mitigasi yang relevan.

2.1. Penelitian Terdahulu

Penelitian terkait pemanfaatan Large Language Model (LLM) menunjukkan bahwa model ini memiliki kemampuan dalam memahami teks kompleks serta melakukan analisis berbasis konteks secara efektif [6]. Penelitian lain juga menunjukkan bahwa penggunaan LLM dalam bidang *software engineering* mampu meningkatkan kualitas analisis berbasis teks secara signifikan [7]. Selain itu, kajian terbaru terkait perkembangan LLM menunjukkan bahwa model ini memiliki potensi besar dalam berbagai aplikasi berbasis pemrosesan bahasa alami, termasuk dalam bidang keamanan siber [8].

Di sisi lain, metode *Content-Based Filtering* (CBF) telah banyak digunakan dalam sistem rekomendasi berbasis teks. Metode ini memanfaatkan teknik TF-IDF dan *Cosine Similarity* untuk mengukur tingkat kemiripan antar data dan menghasilkan rekomendasi yang relevan [9]. Metode ini tidak bergantung pada data interaksi pengguna sebelumnya sehingga dapat digunakan secara fleksibel. Beberapa penelitian lain juga menunjukkan bahwa pendekatan *Content-Based Filtering* efektif dalam menghasilkan rekomendasi pada berbagai domain aplikasi [10].

Berdasarkan penelitian-penelitian tersebut, dapat disimpulkan bahwa integrasi antara LLM dan *Content-Based Filtering* memiliki potensi untuk meningkatkan kualitas analisis serta menghasilkan rekomendasi yang lebih relevan dalam konteks *vulnerability scanning*.

2.2. *Large Language Model*

Large Language Model (LLM) merupakan model berbasis arsitektur *transformer* yang dilatih menggunakan data teks dalam skala besar. Model ini memiliki kemampuan dalam memahami konteks bahasa, melakukan klasifikasi, serta menghasilkan teks secara otomatis [11].

Kemampuan LLM dalam melakukan *summarization* memungkinkan sistem untuk menyederhanakan informasi kompleks menjadi ringkasan yang lebih terstruktur. Selain itu, LLM juga dapat digunakan untuk melakukan klasifikasi tingkat keparahan (*severity*) serta membantu dalam menghasilkan rekomendasi berdasarkan hasil analisis yang dilakukan.

Dalam penelitian ini, LLM diimplementasikan menggunakan *framework* Ollama yang memungkinkan model dijalankan secara lokal tanpa ketergantungan pada layanan berbasis *cloud*. Pendekatan ini memberikan keunggulan dalam aspek keamanan data karena seluruh proses dilakukan secara lokal tanpa mengirimkan data ke pihak eksternal, serta meningkatkan efisiensi dalam penggunaan sumber daya komputasi.

Penggunaan LLM yang dijalankan melalui Ollama dalam penelitian ini bertujuan untuk melakukan proses *summarization*, klasifikasi tingkat *severity*, serta transformasi data hasil *vulnerability scanning* menjadi representasi teks yang lebih terstruktur. Dengan kemampuan tersebut, LLM mampu meningkatkan kualitas interpretasi data serta mendukung proses analisis dalam menghasilkan rekomendasi mitigasi yang relevan.

2.3. Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang digunakan untuk memberikan saran atau rekomendasi kepada pengguna berdasarkan analisis terhadap data tertentu. Sistem ini bertujuan untuk membantu pengguna dalam mengambil keputusan dengan menyediakan informasi yang relevan sesuai dengan kebutuhan [12].

Terdapat beberapa pendekatan dalam sistem rekomendasi, antara lain *Collaborative Filtering*, *Content-Based Filtering*, dan *hybrid approach*. *Collaborative Filtering* bekerja berdasarkan data interaksi pengguna, sedangkan *Content-Based Filtering* berfokus pada kesamaan karakteristik antar item.

Dalam penelitian ini, pendekatan yang digunakan adalah *content-based filtering* karena tidak bergantung pada data pengguna sebelumnya. Metode ini memanfaatkan atribut atau fitur dari data untuk menghasilkan rekomendasi yang relevan berdasarkan tingkat kemiripan antar item.

Dengan demikian, sistem rekomendasi dalam penelitian ini berperan dalam menghasilkan solusi mitigasi yang sesuai dengan jenis kerentanan yang ditemukan melalui proses analisis berbasis teks.

2.4. *Vulnerability Scanner* dan *Web Automation*

Vulnerability scanner merupakan alat yang digunakan untuk mendeteksi kelemahan atau potensi kerentanan dalam sistem atau aplikasi web. Tools seperti Nuclei digunakan untuk mendeteksi kerentanan berbasis template CVE secara otomatis [13].

WhatWeb digunakan untuk mengidentifikasi teknologi yang digunakan pada suatu website serta memberikan informasi terkait struktur aplikasi web [14]. Selain itu, penelitian terkait pengujian aplikasi web menunjukkan bahwa pendekatan otomatis dapat meningkatkan efektivitas proses pengujian serta mengurangi risiko kesalahan manual [15].

Dengan adanya otomatisasi melalui *vulnerability scanner*, proses deteksi kerentanan menjadi lebih cepat dan sistematis, namun masih membutuhkan proses interpretasi untuk memahami hasil yang diperoleh.

2.5. *Content-Based Filtering* (TF-IDF dan *Cosine Similarity*)

Content-Based Filtering merupakan metode dalam sistem rekomendasi yang bekerja dengan menganalisis kesamaan konten antar data. Metode ini menghasilkan rekomendasi berdasarkan karakteristik atau atribut yang dimiliki oleh setiap item, tanpa bergantung pada data interaksi pengguna sebelumnya . Pendekatan ini banyak digunakan dalam sistem rekomendasi berbasis teks karena mampu memberikan hasil yang relevan berdasarkan kesamaan fitur yang terkandung dalam data.

Dalam penelitian ini, pendekatan *Content-Based Filtering* digunakan untuk menghasilkan rekomendasi mitigasi berdasarkan hasil analisis *vulnerability scanning*. Representasi data dilakukan menggunakan teknik TF-IDF (*Term Frequency-Inverse Document Frequency*), yang berfungsi untuk menentukan bobot kata berdasarkan tingkat kepentingannya dalam suatu dokumen. Kata yang sering muncul dalam suatu dokumen namun jarang muncul pada dokumen lain akan memiliki nilai bobot yang lebih tinggi, sehingga dianggap lebih representatif dalam menggambarkan isi dokumen tersebut [16].

TF-IDF dihitung berdasarkan dua komponen utama, yaitu *Term Frequency* (TF) yang mengukur frekuensi kemunculan suatu kata dalam dokumen, dan *Inverse Document Frequency* (IDF) yang mengukur tingkat kelangkaan kata dalam keseluruhan dokumen. Kombinasi kedua komponen ini menghasilkan bobot kata yang mampu merepresentasikan kepentingan suatu kata dalam konteks dokumen secara lebih akurat.

Selanjutnya, *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antar data dalam bentuk vektor. Metode ini menghitung nilai kesamaan berdasarkan sudut antara dua vektor, di mana nilai yang mendekati 1 menunjukkan tingkat kemiripan yang tinggi, sedangkan nilai mendekati 0 menunjukkan kemiripan yang rendah [17]. Dengan menggunakan metode ini, sistem dapat menentukan sejauh mana kesamaan antara hasil analisis *vulnerability scanning* dengan data mitigasi yang tersedia.

Penggunaan kombinasi TF-IDF dan Cosine Similarity dalam penelitian ini memungkinkan sistem untuk mengidentifikasi kesamaan antara hasil analisis dan database mitigasi secara efektif. Dengan demikian, sistem dapat menghasilkan rekomendasi mitigasi yang lebih relevan, spesifik, dan sesuai dengan karakteristik kerentanan yang ditemukan. Pendekatan ini juga mendukung proses otomatisasi dalam sistem rekomendasi berbasis teks, sehingga mampu meningkatkan efisiensi dan akurasi dalam proses pengambilan keputusan.

3. METODE PENELITIAN

Bagian ini menjelaskan metode yang digunakan dalam mengimplementasikan pendekatan analisis *vulnerability scanning* berbasis *Large Language Model* (LLM) dan *Content-Based Filtering*. Metode penelitian mencakup jenis dan sumber data, teknik pengumpulan data, serta metode yang digunakan dalam penelitian ini.

3.1. Jenis dan Sumber Data

Data primer adalah data yang diperoleh secara langsung dari lapangan, baik melalui observasi maupun wawancara dengan narasumber. Dalam penelitian ini, data primer diperoleh dari hasil observasi proses *vulnerability scanning* yang dilakukan di *Software House* NEXTY LABS, serta wawancara dengan tim *developer* dan *cybersecurity* untuk memahami alur kerja dan kebutuhan analisis.

Data sekunder diperoleh melalui studi pustaka, yaitu pengumpulan referensi dari jurnal ilmiah, dokumentasi *tools* seperti Nuclei dan WhatWeb, serta literatur terkait *Large Language Model* (LLM) dan *Content-Based Filtering*. Data ini digunakan sebagai dasar dalam perancangan pendekatan dan pemilihan metode yang digunakan.

Adapun metode pengumpulan data dalam penelitian ini adalah sebagai berikut:

a. Observasi
   Observasi dilakukan dengan mengamati secara langsung proses *vulnerability scanning* yang dilakukan oleh tim di NEXTY LABS. Proses yang diamati meliputi penggunaan *tools scanning*, pembacaan hasil output, serta proses analisis kerentanan yang masih dilakukan secara manual.
b. Wawancara
   Wawancara dilakukan dengan developer dan tim *cybersecurity* untuk memperoleh informasi mengenai kebutuhan sistem, kendala yang dihadapi dalam proses *scanning*, serta harapan terhadap sistem yang akan dikembangkan.
c. Studi Pustaka
   Studi pustaka dilakukan dengan mengkaji literatur dari jurnal, buku, dan dokumentasi resmi yang berkaitan dengan metode yang digunakan, seperti LLM, *Content-Based Filtering*, TF-IDF, dan *Cosine Similarity*.

3.2. Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimental dengan fokus pada implementasi *Large Language Model* (LLM) dan metode *Content-Based Filtering* dalam proses analisis *vulnerability*. Tahapan penelitian dilakukan secara sistematis sebagai berikut:

a. Pengumpulan Data
   Pada tahap ini dilakukan pengumpulan data berupa hasil *vulnerability* scanning dari beberapa domain website menggunakan *tools* Nuclei dan WhatWeb. Data yang diperoleh berupa informasi kerentanan dalam format JSON yang masih bersifat teknis.
b. *Preprocessing* Data
   Data hasil *scanning* yang masih berupa format mentah kemudian diproses untuk mempersiapkan input ke dalam model. Tahap ini meliputi pembersihan data, ekstraksi informasi penting, serta transformasi data menjadi bentuk teks yang dapat dipahami oleh model LLM.
c. Implementasi *Large Language Model* (LLM)
   Data yang telah diproses selanjutnya dianalisis menggunakan Large Language Model (LLM) yang dijalankan secara lokal melalui framework Ollama. Penggunaan Ollama memungkinkan model dijalankan tanpa ketergantungan pada layanan berbasis cloud sehingga proses analisis menjadi lebih aman dan efisien. LLM yang dijalankan melalui Ollama digunakan untuk melakukan proses summarization, klasifikasi tingkat severity, serta transformasi data hasil vulnerability scanning menjadi representasi teks yang lebih terstruktur. Input yang diberikan ke sistem berupa teks hasil preprocessing, sedangkan output yang dihasilkan berupa ringkasan kerentanan serta klasifikasi tingkat severity yang digunakan pada tahap selanjutnya.
d. Ekstraksi Fitur Menggunakan TF-IDF
   Hasil analisis dari LLM kemudian diubah menjadi vektor fitur menggunakan metode TF-IDF untuk menentukan bobot kata berdasarkan tingkat kepentingannya dalam dokumen.
e. Perhitungan Kemiripan Menggunakan *Cosine Similarity*
   Selanjutnya dilakukan perhitungan kemiripan antara hasil analisis dengan database mitigasi menggunakan metode *Cosine Similarity* untuk menemukan rekomendasi yang paling relevan.
f. Evaluasi Hasil
   Tahap akhir dilakukan evaluasi terhadap hasil rekomendasi yang dihasilkan dengan membandingkan relevansi rekomendasi terhadap jenis kerentanan yang ditemukan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

Pengumpulan data dilakukan melalui proses *vulnerability scanning* terhadap 12 domain website menggunakan *tools* Nuclei dan WhatWeb. Proses scanning dilakukan secara menyeluruh untuk mengidentifikasi teknologi yang digunakan serta mendeteksi potensi kerentanan berbasis *template* CVE.

Seluruh proses *scanning* berjalan dengan baik tanpa adanya kegagalan, sehingga data yang diperoleh dapat digunakan pada tahap selanjutnya. Data hasil scanning berupa informasi kerentanan dalam format JSON yang masih bersifat teknis dan belum terstruktur.

Tabel 1. Data target pengujian

No	Domain	Jenis Scan
1	Domain A	*Full Scan*
2	Domain B	*Full Scan*
3	Domain C	*Full Scan*
4	Domain D	*Full Scan*
5	Domain E	*Full Scan*
6	Domain F	*Full Scan*
7	Domain G	*Full Scan*
8	Domain H	*Full Scan*
9	Domain I	*Full Scan*
10	Domain J	*Full Scan*
11	Domain K	*Full Scan*
12	Domain L	*Full Scan*

Berdasarkan hasil perhitungan tersebut, setiap domain memiliki nilai *similarity* tertinggi yang menunjukkan tingkat kesesuaian antara hasil analisis dan solusi mitigasi yang tersedia. Nilai *similarity* yang lebih tinggi menunjukkan bahwa mitigasi yang dipilih memiliki relevansi yang lebih besar terhadap jenis kerentanan yang ditemukan.

Dengan demikian, pendekatan ini memungkinkan sistem untuk secara otomatis memilih solusi mitigasi yang paling sesuai untuk masing-masing domain berdasarkan tingkat kemiripan tertinggi.

4.2. Hasil *Preprocessing* Data

Data hasil *vulnerability scanning* yang masih dalam format mentah kemudian diproses melalui tahap *preprocessing*. Tahap ini meliputi pembersihan data, penghapusan informasi yang tidak relevan, serta ekstraksi informasi penting seperti jenis kerentanan dan deskripsi teknis.

Data yang semula berbentuk struktur JSON kemudian diubah menjadi representasi teks agar dapat digunakan dalam proses analisis. Hasil *preprocessing* menunjukkan bahwa data yang sebelumnya kompleks berhasil disederhanakan menjadi format yang lebih terstruktur dan mudah dipahami.

4.3. Hasil Implementasi *Large Language Model* (LLM)

Pada tahap ini, data hasil preprocessing dianalisis menggunakan *Large Language Model* (LLM) yang dijalankan melalui *framework* Ollama. Penggunaan Ollama memungkinkan proses analisis dilakukan secara lokal sehingga lebih aman dan efisien.

LLM yang dijalankan melalui Ollama menghasilkan interpretasi dalam bentuk teks melalui

proses *summarization* serta klasifikasi tingkat *severity*. Input berupa teks hasil *preprocessing* diproses untuk menghasilkan ringkasan kerentanan yang lebih terstruktur dan mudah dipahami.

Sebagai contoh, pada domain A, sistem mampu mengidentifikasi kerentanan terkait konfigurasi SSL/TLS yang lemah serta menghasilkan ringkasan yang lebih jelas dibandingkan data mentah. Hasil ini menunjukkan bahwa penggunaan LLM melalui Ollama mampu menyederhanakan informasi teknis menjadi representasi yang lebih informatif.

Dengan demikian, penggunaan LLM yang dijalankan melalui Ollama memberikan kontribusi dalam meningkatkan efisiensi proses analisis serta mendukung pengambilan keputusan dalam menentukan solusi mitigasi yang sesuai.

4.4. Hasil Ekstraksi Fitur Menggunakan TF-IDF

Hasil interpretasi dari *Large Language Model* (LLM) yang dijalankan melalui *framework* Ollama kemudian diubah menjadi representasi vektor menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). Proses ini bertujuan untuk menentukan bobot kata berdasarkan tingkat kepentingannya dalam dokumen hasil analisis *vulnerability scanning*.

Pada penelitian ini, setiap domain diperlakukan sebagai satu dokumen dalam korpus. Dengan demikian, proses perhitungan TF-IDF dilakukan terhadap seluruh domain (Domain A hingga Domain L) secara keseluruhan. Nilai TF (*Term Frequency*) dihitung berdasarkan frekuensi kemunculan suatu kata dalam satu domain, sedangkan nilai IDF (*Inverse Document Frequency*) dihitung berdasarkan distribusi kata tersebut pada seluruh domain yang digunakan dalam penelitian.

Pendekatan ini memungkinkan sistem untuk mengidentifikasi kata-kata yang memiliki tingkat kepentingan tinggi dalam suatu domain, sekaligus mempertimbangkan seberapa umum atau jarang kata tersebut muncul pada domain lainnya. Kata yang sering muncul pada satu domain tetapi jarang muncul pada domain lain akan memiliki nilai TF-IDF yang lebih tinggi, sehingga dianggap sebagai fitur utama dalam merepresentasikan karakteristik kerentanan.

Untuk memberikan gambaran hasil yang representatif, pembobotan TF-IDF disajikan dengan menampilkan tiga kata dengan nilai TF-IDF tertinggi pada masing-masing domain. Hal ini dilakukan agar hasil tetap ringkas namun tetap mencerminkan karakteristik utama dari setiap domain yang dianalisis.

Tabel 2. Hasil Pembobotan TF-IDF Tertinggi Setiap Domain

No	Domain	*Term 1*	*TF-IDF*	*Term 2*	*TF-IDF*	Term 3	*TF-IDF*
1	Domain A	*tls*	0.4123	*cipher*	0.3987	*security*	0.3652
2	Domain B	*header*	0.3891	*security*	0.3542	*missing*	0.3310
3	Domain C	*cipher*	0.4012	*deprecated*	0.3723	*tls*	0.3488
4	Domain D	*tls*	0.4255	*version*	0.3977	*weak*	0.3661
5	Domain E	*csp*	0.3921	*header*	0.3684	*missing*	0.3422
6	Domain F	*cipher*	0.4102	*suite*	0.3815	*weak*	0.3554
7	Domain G	*hsts*	0.3982	*missing*	0.3675	*header*	0.3441
8	Domain H	*tls*	0.4188	*config*	0.3892	*error*	0.3520
9	Domain I	*ssl*	0.4321	*outdated*	0.4010	*protocol*	0.3795
10	Domain J	*header*	0.3875	*frame*	0.3589	*missing*	0.3366
11	Domain K	*encryption*	0.4054	*weak*	0.3722	*security*	0.3490
12	*Domain L*	*header*	0.3910	*security*	0.3663	*missing*	0.3418

Berdasarkan hasil pembobotan TF-IDF pada Tabel 2, setiap domain memiliki kata-kata utama dengan nilai tertinggi yang merepresentasikan karakteristik kerentanan yang ditemukan. Kata seperti *tls*, *cipher*, *security*, dan *header* muncul sebagai fitur dominan pada beberapa domain, yang menunjukkan bahwa sebagian besar kerentanan berkaitan dengan konfigurasi keamanan, protokol komunikasi, serta pengelolaan header pada aplikasi web.

Nilai TF-IDF yang dihasilkan pada penelitian ini dipengaruhi oleh distribusi kata pada seluruh domain yang digunakan sebagai korpus. Oleh karena itu, nilai yang diperoleh dapat berbeda dengan perhitungan sederhana yang hanya menggunakan sedikit dokumen, karena mempertimbangkan variasi dan frekuensi kemunculan kata secara global pada seluruh dataset.

Hasil ekstraksi fitur ini kemudian digunakan sebagai representasi vektor pada tahap perhitungan *Cosine Similarity* untuk menentukan tingkat kemiripan antara hasil analisis dan database mitigasi. Dengan demikian, proses ini menjadi dasar dalam menghasilkan rekomendasi mitigasi yang paling relevan untuk setiap domain.

4.5. Hasil *Vulnerability Scanning*

Setelah proses analisis awal, diperoleh ringkasan hasil *vulnerability scanning* yang menunjukkan jenis kerentanan serta tingkat *severity* pada masing-masing domain.

Tabel 3. Hasil *Vulnerability Scanning*

No	Domain	Jenis Kerentanan	*Severity*
1	Domain A	*Weak* SSL/TLS *Configuration*	*Medium*
2	Domain B	*Missing Security Header*	*Low*
3	Domain C	*Deprecated Cipher*	*Medium*
4	Domain D	*Weak* TLS *Version*	*High*
5	Domain E	*Missing* CSP *Header*	*Low*
6	Domain F	*Weak Cipher Suite*	*Medium*

No	Domain	Jenis Kerentanan	*Severity*
7	Domain G	*Missing* HSTS	*Low*
8	Domain H	TLS *Misconfiguration*	*Medium*
9	Domain I	*Outdated* SSL *Protocol*	*High*
10	Domain J	*Missing X-Frame Header*	*Low*
11	Domain K	*Weak Encryption*	*Medium*
12	Domain L	*Security Header Missing*	*Low*

Tabel tersebut menunjukkan bahwa sebagian besar kerentanan yang ditemukan berkaitan dengan konfigurasi keamanan pada protokol komunikasi serta header keamanan.

4.6. Hasil Perhitungan *Cosine Similarity*

Setelah dilakukan pembobotan menggunakan metode TF-IDF, tahap selanjutnya adalah perhitungan kemiripan menggunakan metode *Cosine Similarity* antara hasil analisis dan database mitigasi.

Sebagai studi kasus, perhitungan dilakukan pada domain A untuk menentukan mitigasi yang paling sesuai.

Tabel 4. Hasil Perhitungan *Cosine Similarity* pada Domain A

No	Domain	*Query*	Mitigasi	*Similarity*
1	Domain A	*weak tls cipher security*	Perbaikan SSL/TLS Lemah	0.5179
2	Domain B	*missing security header*	Penambahan *Security Header*	0.4892
3	Domain C	*deprecated cipher*	*Update Cipher Suite*	0.5011
4	Domain D	*weak tls version*	*Upgrade* TLS Version	0.5345
5	Domain E	*missing csp header*	Implementasi CSP Header	0.4723
6	Domain F	*weak cipher suite*	Perbaikan *Cipher Suite*	0.4987
7	Domain G	*missing hsts*	Implementasi HSTS	0.4655
8	Domain H	*tls misconfiguration*	Perbaikan Konfigurasi TLS	0.5122
9	Domain I	*outdated ssl protocol*	*Upgrade* SSL *Protocol*	0.5431
10	Domain J	*missing x-frame header*	Penambahan *X-Frame Header*	0.4789
11	Domain K	*weak encryption*	Perbaikan Enkripsi	0.5066
12	*Domain L*	*security header missing*	Penambahan *Security Header*	0.4910

Hasil menunjukkan bahwa mitigasi Perbaikan SSL/TLS Lemah memiliki nilai *similarity* tertinggi, sehingga dipilih sebagai rekomendasi utama. Nilai *similarity* yang lebih tinggi menunjukkan tingkat kesesuaian yang lebih besar antara hasil analisis dan solusi mitigasi.

4.7. Hasil Rekomendasi Mitigasi

Berdasarkan nilai *similarity* tertinggi, dihasilkan rekomendasi mitigasi yang paling relevan untuk setiap domain.

Tabel 5. Hasil *Rekomendasi Mitigasi*

No	Domain	Rekomendasi Mitigasi
1	Domain A	Perbaikan SSL/TLS Lemah
2	Domain B	Penambahan *Security Header*
3	Domain C	*Update Cipher Suite*
4	Domain D	*Upgrade* TLS Version
5	Domain E	Implementasi CSP Header
6	Domain F	Perbaikan *Cipher Suite*
7	Domain G	Implementasi HSTS
8	Domain H	Perbaikan Konfigurasi TLS
9	Domain I	*Upgrade* SSL *Protocol*
10	Domain J	Penambahan *X-Frame Header*
11	Domain K	Perbaikan Enkripsi
12	Domain L	Penambahan *Security Header*

Hasil menunjukkan bahwa pendekatan yang digunakan mampu menyesuaikan rekomendasi mitigasi berdasarkan karakteristik kerentanan pada masing-masing domain.

4.8. Alur Proses Analisis

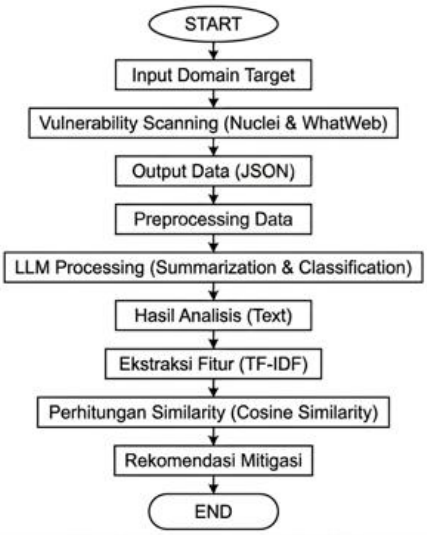
Alur proses analisis pada penelitian ini menggambarkan tahapan implementasi metode dalam menghasilkan rekomendasi mitigasi berdasarkan hasil *vulnerability scanning*. Proses dimulai dari input berupa domain target yang akan dianalisis, kemudian dilakukan *vulnerability scanning* menggunakan *tools* Nuclei dan WhatWeb untuk memperoleh data kerentanan dalam bentuk JSON yang masih bersifat teknis.

Data hasil *scanning* selanjutnya melalui tahap *preprocessing* untuk membersihkan dan mengekstraksi informasi penting agar dapat digunakan sebagai input dalam proses analisis. Setelah itu, data diproses menggunakan *Large Language Model* (LLM) untuk menghasilkan interpretasi dalam bentuk teks yang lebih terstruktur, termasuk ringkasan kerentanan serta klasifikasi tingkat keparahan.

Hasil interpretasi dari LLM kemudian digunakan sebagai representasi teks yang akan diproses menggunakan metode TF-IDF untuk menentukan

bobot kata berdasarkan tingkat kepentingannya. Selanjutnya dilakukan perhitungan kemiripan menggunakan metode *Cosine Similarity* antara hasil analisis dengan database mitigasi yang tersedia.

Berdasarkan nilai kemiripan tertinggi, dihasilkan rekomendasi mitigasi yang paling relevan dengan jenis *vulnerability* yang ditemukan. Alur proses ini menunjukkan integrasi antara LLM dan metode *Content-Based Filtering* dalam mendukung analisis vulnerability secara lebih sistematis dan efisien.



Gambar 1. Alur Proses Analisis

4.9. Evaluasi Hasil

Evaluasi dilakukan untuk menilai efektivitas pendekatan yang digunakan dalam menghasilkan rekomendasi mitigasi berdasarkan hasil *vulnerability scanning*. Proses evaluasi mencakup seluruh tahapan metode yang telah diimplementasikan.

Hasil evaluasi menunjukkan bahwa penggunaan *Large Language Model* (LLM) mampu mengubah data hasil scanning yang bersifat teknis menjadi informasi yang lebih terstruktur dan mudah dipahami. Proses ini memberikan kontribusi penting dalam meningkatkan kualitas data sebelum dilakukan analisis lebih lanjut. Selain itu, metode TF-IDF mampu mengidentifikasi kata kunci yang relevan sehingga menghasilkan representasi vektor yang sesuai dengan karakteristik kerentanan.

Penggunaan *Cosine Similarity* menunjukkan kemampuan dalam menentukan tingkat kemiripan antara hasil analisis dan database mitigasi. Nilai kemiripan yang dihasilkan mampu membedakan tingkat relevansi antar solusi, sehingga rekomendasi yang dipilih merupakan solusi yang paling sesuai dengan kondisi kerentanan yang ditemukan.

Secara keseluruhan, pendekatan yang digunakan mampu meningkatkan efisiensi proses analisis dibandingkan metode manual yang sebelumnya dilakukan. Integrasi antara LLM dan *Content-Based Filtering* menghasilkan proses analisis yang lebih sistematis, terstruktur, dan mendukung pengambilan keputusan secara lebih efektif.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan pendekatan berbasis *Large Language Model* (LLM) dan *Content-Based Filtering* dalam proses analisis *vulnerability scanning*. Pendekatan ini mampu menyederhanakan data teknis menjadi informasi yang lebih mudah dipahami serta menghasilkan rekomendasi mitigasi yang relevan berdasarkan karakteristik kerentanan yang ditemukan.

Metode TF-IDF dan *Cosine Similarity* menunjukkan hasil yang efektif dalam menentukan tingkat kemiripan antara hasil analisis dan data mitigasi. Selain itu, penggunaan LLM memberikan kontribusi dalam meningkatkan kualitas interpretasi data sehingga proses analisis menjadi lebih efisien.

Untuk pengembangan selanjutnya, pendekatan ini dapat ditingkatkan dengan penambahan dataset mitigasi yang lebih luas serta integrasi dengan sumber *vulnerability* secara *real-time* untuk meningkatkan kualitas rekomendasi.

DAFTAR PUSTAKA

[1] X. Du, Y. Li, and H. Zhang, "Vulnerability Severity Prediction Using Multi-task Learning," *Computers & Security*, vol. 136, pp. 103123, 2024.

[2] M. A. Ferrag et al., "Large Language Models for Cybersecurity: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 1-25, 2024.

[3] M. Huda, A. Prasetyo, and R. Kurniawan, "Sistem Rekomendasi Content-Based Filtering Menggunakan TF-IDF dan Cosine Similarity," *Jurnal Teknologi Informasi*, vol. 18, no. 2, pp. 120–130, 2022.

[4] Herimanto, D. Saputra, and R. Wijaya, "Comparative Analysis of Content-Based Filtering and TF-IDF Approaches," *Journal of Information Systems*, vol. 20, no. 1, pp. 45–55, 2024.

[5] W. X. Zhao et al., "A Survey of Large Language Models," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 1-38, 2023.

[6] R. Hoda, N. Salleh, and J. Grundy, "The Rise and Evolution of Agile Software Development," *IEEE Software*, vol. 40, no. 2, pp. 10-17, 2023.

[7] Y. Li, Z. Chen, and J. Xu, "Large Language Models for Software Engineering: A Systematic Review," *IEEE Access*, vol. 11, no. 1, pp. 1-15, 2023.

[8] A. Naseer et al., "Applications of Large Language Models in Cybersecurity: A Survey," *IEEE Access*, vol. 11, no. 1, pp. 1-20, 2023.

[9] A. Rahman, I. Indra, N. Zulkarnaim, M. Mukhram, and A. Rizaldi, "Analisis Implementasi Nuclei Vulnerability dan OWASP-ZAP Scanner untuk Deteksi Kerentanan Keamanan (Secure System) pada Platform Web Based," *Jurnal Komputer Terapan*, vol. 11, no. 1, pp. 10-15, 2025.

[10] M. Ridwansyah, S. Subartini, and Sylviani, "Penerapan Content-Based Filtering dalam Sistem Rekomendasi," *Jurnal Sistem Informasi dan Teknologi*, vol. 6, no. 1, pp. 50-60, 2024.

[11] Y. Sheng et al., "Large Language Models in Software Security," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1-35, 2025.

[12] A. Kar and S. Sen, "Web Server Fingerprinting with HTTP Request Fuzzing," *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT)*, pp. 260-267, 2022.

[13] H. Zhang, Q. Liu, and S. Wang, "A Review of Text-Based Recommendation Systems Using TF-IDF and Similarity Measures," *Information Processing & Management*, vol. 59, no. 6, pp. 102890, 2022.

[14] J. Wei et al., "Emergent Abilities of Large Language Models," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1-24, 2023.

[15] S. K. Singh and R. Kumar, "Automated Web Application Security Testing: A Systematic Literature Review," IEEE Access, vol. 11, no. 1, pp. 1-18, 2023.

[16] S. K. Jain and V. K. Sharma, "A Comparative Study of TF-IDF and Word Embedding Techniques for Text Similarity Analysis," *Expert Systems with Applications*, vol. 213, pp. 118945, 2023.

[17] B. Huang, Z. Liu, and X. Wang, "Improved Cosine Similarity Measures for Text Classification," *Journal of Information Science*, vol. 48, no. 2, pp. 200-210, 2022.