e-ISSN 2828-0490
JURNAL MEDIA COMPUTER SCIENCE

Development And Evaluation Of A Malware Detection System On Open Journal Systems Using Script Analysis For Cloaking And Backdoor Identification

Indaka Barody [1]; Abdul Jalil [2]; Andri Putra Kesmawan [3]; Karona Cahya Susena [4] Muhammad Yunus [5]

[1] *Pubmedia Digital Sains*
[2] *Program Studi Agroteknologi, Universitas Muhammadiyah Jember*
[3] *Pubmedia Digital Sains*
[4] *Universitas Dehasen Bengkulu*
[5] *Politeknik Negeri Jember*
Email: [1] indakabarody16@gmail.com ;[2] Abduljalil@unmuhjember.ac.id ;[3] andriputrakesmawan@gmail.com
[4] karona.cs@unived.ac.id ;[5] m.yunus@polije.ac.id

How to Cite :
Barody, I., Jalil, A., Kesmawan, A. P., Susena, K. C., & Yunus, M. (2026). Development And Evaluation Of A Malware Detection System On Open Journal Systems Using Script Analysis For Cloaking And Backdoor Identification. Jurnal Media Computer Science, 5(4).

ARTICLE HISTORY
Received [17 Mei 2026]
Revised [30 Oktober 2026]
Accepted [31 Oktober 2026]

KEYWORDS
Malware, Open Journal Systems, Analisis Script, Cloaking, Keamanan Web.

This is an open access article under the CC–BY-SA license

ABSTRAK

Meningkatnya ancaman keamanan berbasis web, khususnya malware yang disisipkan dalam bentuk script, menjadi tantangan serius pada platform Open Journal Systems (OJS). Malware modern cenderung bersifat tersembunyi dan sulit dideteksi menggunakan metode konvensional berbasis signature statis. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi malware berbasis analisis script pada sisi front-end OJS. Metode yang digunakan adalah pendekatan eksperimental dengan rekayasa sistem yang meliputi proses crawling halaman web, ekstraksi HTML dan JavaScript, serta analisis pola script untuk mendeteksi indikasi malware seperti obfuscation, hidden script, dan cloaking. Hasil penelitian menunjukkan bahwa sistem mampu melakukan 393 kali pemindaian terhadap 241 website OJS dengan total 60.396 halaman yang dianalisis, di mana 11,94% halaman terindikasi mengandung malware. Sistem juga terbukti mampu mendeteksi script tersembunyi dan teknik cloaking melalui perbandingan hasil crawling dengan parameter berbeda. Kesimpulan penelitian ini menunjukkan bahwa pendekatan deteksi malware berbasis analisis script pada sisi client-side efektif dalam mengidentifikasi ancaman tersembunyi tanpa memerlukan akses server. Penelitian ini memberikan kontribusi dalam pengembangan metode deteksi malware yang lebih adaptif serta solusi praktis dalam meningkatkan keamanan platform OJS.

ABSTRACT

The growing threat of web-based security attacks, particularly malware embedded in the form of scripts, has become a serious challenge on the Open Journal Systems (OJS) platform. Modern malware tends to be covert and difficult to detect using conventional static signature-based methods. This study aims to develop and evaluate a malware detection system based on script analysis on the front-end side of OJS. The method employed is an experimental approach with system engineering that encompasses web page crawling, HTML and JavaScript extraction, and script pattern analysis to detect malware indicators such as obfuscation, hidden scripts, and cloaking. The

results show that the system was capable of conducting 393 scans across 241 OJS websites with a total of 60,396 pages analyzed, of which 11.94% of pages were indicated to contain malware. The system also proved capable of detecting hidden scripts and cloaking techniques through a comparison of crawling results obtained under different parameters. The conclusions of this study indicate that a client-side script analysis-based malware detection approach is effective in identifying hidden threats without requiring server access. This research contributes to the development of more adaptive malware detection methods as well as practical solutions for enhancing the security of the OJS platform..

PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi signifikan dalam pengelolaan publikasi ilmiah dari sistem konvensional menuju platform digital berbasis web. Salah satu platform yang широко digunakan adalah Open Journal Systems (OJS), yang berperan penting dalam mendukung pengelolaan jurnal ilmiah secara terintegrasi, mulai dari proses submission, review, hingga publikasi. Di Indonesia, OJS menjadi pilihan utama bagi institusi pendidikan dan penelitian karena sifatnya yang open source, fleksibel, serta mendukung prinsip akses terbuka (open access). Namun, di balik kemudahan tersebut, meningkatnya adopsi OJS juga memperluas permukaan serangan (attack surface), sehingga menjadikannya target potensial bagi berbagai ancaman keamanan siber, khususnya serangan berbasis web seperti malware, injeksi script, dan penyisipan konten ilegal pada halaman jurnal.

Ancaman malware pada aplikasi berbasis web merupakan salah satu isu kritis dalam keamanan sistem informasi karena dampaknya yang luas terhadap kerahasiaan, integritas, dan ketersediaan data. Malware dapat menyebabkan kerusakan sistem, pencurian informasi sensitif, serta penyisipan konten berbahaya tanpa sepengetahuan pengelola sistem. Penelitian oleh Ulvi et al. (2026) menunjukkan bahwa sistem berbasis web memiliki tingkat kerentanan yang tinggi terhadap serangan malware yang dapat mengganggu operasional layanan. Lebih lanjut, perkembangan malware saat ini menunjukkan pergeseran pola serangan dari yang sebelumnya berfokus pada sisi server menjadi eksploitasi pada sisi klien (client-side), seperti injeksi JavaScript berbahaya yang dieksekusi langsung pada browser pengguna, sehingga semakin sulit untuk dideteksi menggunakan mekanisme keamanan tradisional.

Dalam konteks OJS, kompleksitas ancaman keamanan semakin meningkat seiring dengan karakteristik sistem yang berbasis template dinamis dan modular. Struktur halaman OJS yang fleksibel memungkinkan penyerang untuk menyisipkan script berbahaya secara tersembunyi, baik melalui manipulasi file maupun injeksi pada elemen tertentu, tanpa memengaruhi tampilan utama secara signifikan. Studi oleh Mulyanto et al. (2022) mengungkapkan bahwa serangan siber, termasuk malware, dapat menargetkan aplikasi berbasis web dan berpotensi merusak integritas sistem serta menurunkan kepercayaan pengguna terhadap layanan tersebut. Selain itu, hasil kegiatan pendampingan keamanan oleh Nur et al. (2025) menunjukkan bahwa banyak pengelola jurnal berbasis OJS belum memiliki mekanisme deteksi malware yang memadai, sehingga serangan yang bersifat tersembunyi sering kali tidak teridentifikasi dalam waktu yang lama.

Sebagian besar pendekatan deteksi malware yang dikembangkan saat ini masih bergantung pada analisis sisi server atau menggunakan metode berbasis signature statis. Pendekatan tersebut memiliki keterbatasan dalam mendeteksi teknik serangan modern yang bersifat dinamis dan adaptif, seperti cloaking dan injeksi script tersembunyi. Cloaking merupakan teknik di mana penyerang menampilkan konten yang berbeda antara pengguna biasa dan sistem deteksi (crawler), sehingga keberadaan malware dapat disembunyikan dari proses pemindaian konvensional. Selain itu, penyisipan konten ilegal, seperti script judi atau redirect tersembunyi, sering kali disamarkan dalam struktur HTML atau JavaScript yang tidak terlihat secara langsung oleh pengguna, sehingga semakin menyulitkan proses identifikasi jika hanya mengandalkan metode deteksi tradisional.

Beberapa penelitian di bidang keamanan sistem informasi menunjukkan bahwa metode deteksi konvensional belum sepenuhnya mampu mengimbangi perkembangan ancaman malware yang semakin kompleks dan beragam. Jelita dan Siregar (2025) menekankan bahwa sistem deteksi malware perlu dikembangkan secara adaptif untuk mampu mengenali pola serangan baru yang terus berkembang. Hal ini diperkuat oleh Dirayati dan Marliana (2024) yang menyatakan bahwa peningkatan kompleksitas malware, terutama yang berbasis script dan obfuscation, memerlukan pendekatan analisis yang lebih dinamis dan kontekstual. Dengan demikian, diperlukan inovasi dalam metode deteksi yang tidak hanya bergantung pada pola statis, tetapi juga mampu menganalisis perilaku dan struktur script secara lebih mendalam.

Berdasarkan permasalahan tersebut, diperlukan suatu pendekatan deteksi malware yang mampu bekerja secara efektif tanpa bergantung pada akses server, serta dapat mengidentifikasi pola serangan tersembunyi pada halaman web. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem deteksi malware berbasis analisis script pada sisi front-end OJS. Pendekatan yang digunakan melibatkan proses crawling halaman web, ekstraksi elemen HTML dan JavaScript, serta analisis pola anomali untuk mengidentifikasi indikasi malware. Sistem ini dirancang untuk mendeteksi berbagai bentuk serangan, termasuk cloaking, backdoor, dan injeksi script tersembunyi yang sering kali tidak terdeteksi oleh metode konvensional.

Dengan pendekatan yang diusulkan, penelitian ini diharapkan dapat memberikan kontribusi baik secara teoritis maupun praktis dalam bidang keamanan sistem informasi, khususnya pada platform OJS. Secara praktis, sistem yang dikembangkan dapat digunakan oleh pengelola jurnal sebagai alat bantu untuk melakukan deteksi mandiri terhadap potensi serangan malware secara lebih cepat dan akurat tanpa memerlukan keahlian teknis yang mendalam. Selain itu, penelitian ini juga diharapkan dapat menjadi referensi dalam pengembangan metode deteksi malware berbasis client-side yang lebih adaptif terhadap karakteristik serangan modern, sehingga dapat meningkatkan tingkat keamanan dan keandalan platform OJS secara keseluruhan..

LANDASAN TEORI

Open Journal Systems (OJS) merupakan platform manajemen jurnal berbasis web yang dikembangkan untuk mendukung proses publikasi ilmiah secara digital, mulai dari pengiriman naskah, proses review, hingga publikasi akhir. OJS banyak digunakan oleh institusi pendidikan karena sifatnya yang open source dan fleksibel dalam pengelolaan konten ilmiah. Namun demikian, karakteristik OJS yang berbasis web dan memanfaatkan berbagai komponen seperti HTML, CSS, dan JavaScript menjadikannya rentan terhadap berbagai ancaman keamanan siber. Menurut Nur, dkk (2025), sistem OJS memiliki risiko terhadap serangan malware dan injeksi script yang dapat memanfaatkan kelemahan pada struktur halaman web. Selain itu, penelitian oleh Yunanri, et al. (2021) menunjukkan bahwa OJS juga memiliki kerentanan yang dapat dieksploitasi melalui berbagai teknik serangan berbasis web.

Malware merupakan perangkat lunak berbahaya yang dirancang untuk merusak, mengganggu, atau memperoleh akses tidak sah terhadap suatu sistem. Dalam konteks aplikasi berbasis web, malware dapat disisipkan dalam bentuk script, file, maupun manipulasi konten halaman. Serangan malware pada web umumnya memanfaatkan celah keamanan seperti input yang tidak tervalidasi, konfigurasi server yang lemah, atau kerentanan pada aplikasi. Ulvi, et al. (2026) menyatakan bahwa malware pada sistem berbasis web dapat menyebabkan gangguan serius terhadap integritas dan ketersediaan sistem informasi. Selain itu, malware modern cenderung bersifat dinamis dan mampu beroperasi pada sisi klien (client-side), sehingga sulit dideteksi menggunakan metode konvensional.

JavaScript merupakan bahasa pemrograman yang digunakan untuk meningkatkan interaktivitas pada halaman web. Namun, sifatnya yang dieksekusi pada sisi klien menjadikannya sebagai salah satu media utama dalam penyebaran malware berbasis web. Teknik injeksi script seperti Cross-Site Scripting (XSS) memungkinkan penyerang menyisipkan kode berbahaya ke dalam

halaman web yang kemudian dieksekusi oleh browser pengguna. Kuswara dan Fachri (2025) menjelaskan bahwa injeksi JavaScript dapat digunakan untuk menyebarkan malware, mencuri data pengguna, atau melakukan redirect ke situs berbahaya. Oleh karena itu, analisis terhadap script menjadi aspek penting dalam mendeteksi potensi ancaman keamanan pada aplikasi web.

Cloaking merupakan teknik yang digunakan oleh penyerang untuk menyembunyikan konten berbahaya dengan menampilkan konten yang berbeda berdasarkan jenis pengguna atau sistem yang mengakses halaman tersebut. Teknik ini sering digunakan untuk menghindari deteksi oleh sistem keamanan atau crawler. Dalam praktiknya, cloaking dapat dilakukan dengan membedakan user-agent, alamat IP, atau parameter tertentu. Akibatnya, halaman yang tampak normal bagi pengguna biasa dapat mengandung script berbahaya yang hanya ditampilkan dalam kondisi tertentu. Pendekatan ini menjadikan cloaking sebagai salah satu teknik yang sulit dideteksi dengan metode berbasis signature statis.Analisis keamanan berbasis web umumnya melibatkan proses crawling dan parsing untuk mengekstraksi serta menganalisis struktur halaman web. Crawling digunakan untuk mengambil konten halaman secara otomatis, sedangkan parsing digunakan untuk mengidentifikasi elemen-elemen penting seperti HTML dan JavaScript. Menurut Wiharja, et al. (2024), teknik ini efektif dalam menganalisis struktur halaman web dan mendeteksi anomali yang berpotensi sebagai indikasi malware. Dengan melakukan ekstraksi dan analisis terhadap script yang terdapat pada halaman web, sistem dapat mengidentifikasi pola mencurigakan seperti obfuscation, hidden script, dan komunikasi dengan domain berbahaya.

Pendekatan deteksi malware berbasis analisis script merupakan metode yang berfokus pada identifikasi pola perilaku dan struktur kode yang mencurigakan. Berbeda dengan metode berbasis signature, pendekatan ini lebih adaptif terhadap variasi malware yang terus berkembang. Analisis script dapat mencakup deteksi penggunaan fungsi berbahaya (seperti eval()), encoding yang kompleks, serta pola obfuscation yang bertujuan menyembunyikan kode asli. Pendekatan ini dinilai lebih efektif dalam mendeteksi malware modern yang bersifat dinamis dan tersembunyi dalam struktur halaman web.

Berdasarkan kajian teoritik yang telah diuraikan, dapat disimpulkan bahwa platform OJS memiliki kerentanan terhadap serangan malware berbasis web, khususnya yang memanfaatkan injeksi script dan teknik cloaking. Metode deteksi konvensional yang berbasis server atau signature statis memiliki keterbatasan dalam mendeteksi malware yang bersifat dinamis dan tersembunyi. Oleh karena itu, diperlukan pendekatan alternatif yang mampu menganalisis konten pada sisi klien melalui proses crawling, parsing, dan analisis script. Penelitian ini mengembangkan sistem deteksi malware berbasis analisis script pada halaman front-end OJS untuk mengidentifikasi pola mencurigakan, hidden script, serta teknik cloaking. Dengan pendekatan ini, diharapkan sistem mampu meningkatkan akurasi deteksi malware serta memberikan solusi praktis bagi pengelola jurnal dalam menjaga keamanan sistem.

METODE PENELITIAN

Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan pendekatan eksperimental yang dikombinasikan dengan rekayasa sistem (system development research), dengan tujuan untuk mengembangkan sekaligus mengevaluasi kinerja sistem deteksi malware pada platform Open Journal Systems (OJS). Pendekatan ini dipilih karena penelitian tidak hanya berfokus pada analisis fenomena keamanan, tetapi juga menghasilkan suatu sistem yang dapat diimplementasikan dan diuji secara langsung dalam mendeteksi keberadaan malware pada halaman web. Dengan demikian, penelitian ini mencakup proses perancangan, implementasi, hingga evaluasi sistem secara terintegrasi.

Pendekatan yang digunakan mengacu pada konsep analisis keamanan berbasis web (web-based security analysis) yang melibatkan proses crawling, parsing, dan analisis script untuk mengidentifikasi indikasi malware pada halaman web. Teknik ini dinilai efektif karena mampu

mengekstraksi serta menganalisis struktur HTML dan JavaScript yang menjadi komponen utama dalam penyisipan malware modern. Metode serupa telah digunakan dalam penelitian sebelumnya untuk mendeteksi ancaman berbasis script pada aplikasi web (Wiharja et al., 2024), sehingga relevan untuk diadaptasi dalam konteks sistem OJS.

Tahapan Penelitian

Tahapan penelitian ini disusun secara sistematis untuk memastikan proses pengembangan sistem berjalan terstruktur dan dapat dievaluasi dengan baik. Secara umum, tahapan penelitian meliputi pengumpulan data, perancangan sistem, implementasi, serta pengujian sistem. Pada tahap pengumpulan data, data yang digunakan dalam penelitian ini berupa URL website berbasis OJS yang dijadikan sebagai objek pengujian. Selain itu, dikumpulkan pula halaman web yang mengandung indikasi malware sebagai data uji kasus, serta halaman web normal sebagai data pembanding. Data diperoleh melalui proses crawling terhadap website OJS serta pengujian langsung terhadap halaman publik yang dapat diakses secara terbuka. Tahap ini bertujuan untuk menyediakan dataset yang representatif dalam menguji kemampuan sistem dalam mendeteksi malware.

Selanjutnya, pada tahap perancangan sistem, dilakukan perancangan arsitektur sistem deteksi malware berbasis web. Sistem dirancang untuk menerima input berupa URL OJS yang akan dianalisis, kemudian memproses data melalui beberapa tahapan utama, yaitu crawling halaman, ekstraksi HTML dan JavaScript, serta analisis script. Hasil dari proses tersebut berupa informasi status keamanan halaman, apakah tergolong aman atau terindikasi malware, beserta jenis indikasi yang terdeteksi seperti cloaking, backdoor, atau script tersembunyi. Tahap berikutnya adalah implementasi sistem, di mana rancangan yang telah dibuat direalisasikan ke dalam bentuk aplikasi berbasis web. Sistem menggunakan komponen utama berupa web crawler untuk mengambil konten halaman, HTML parser untuk mengekstraksi elemen script, serta modul analisis yang bertugas mendeteksi pola-pola mencurigakan. Proses analisis difokuskan pada identifikasi obfuscation, suspicious pattern, dan hidden content yang menjadi karakteristik umum malware berbasis web.

Tahap terakhir adalah pengujian sistem, yang dilakukan untuk mengevaluasi kinerja sistem dalam mendeteksi malware. Pengujian dilakukan dengan menguji sistem pada halaman OJS normal serta halaman yang telah disisipi script malware. Hasil deteksi kemudian dibandingkan untuk menilai tingkat akurasi sistem. Parameter evaluasi yang digunakan meliputi akurasi deteksi, kemampuan mendeteksi teknik cloaking, serta kemampuan mengidentifikasi script tersembunyi. Tahap ini menjadi penting untuk membuktikan efektivitas metode yang diusulkan.

Metode Deteksi Malware

Metode deteksi malware yang diusulkan dalam penelitian ini merupakan inti dari kontribusi penelitian, yang dirancang untuk mengidentifikasi malware berbasis script pada halaman front-end OJS tanpa memerlukan akses ke sisi server. Metode ini terdiri dari beberapa tahapan utama yang saling terintegrasi.

Tahap pertama adalah crawling halaman web, di mana sistem mengambil konten halaman OJS berdasarkan URL yang diberikan oleh pengguna. Proses ini menghasilkan data berupa struktur HTML lengkap dari halaman yang akan dianalisis. Crawling dilakukan untuk memastikan seluruh elemen halaman, termasuk resource yang terkait, dapat diakses dan dianalisis secara menyeluruh.

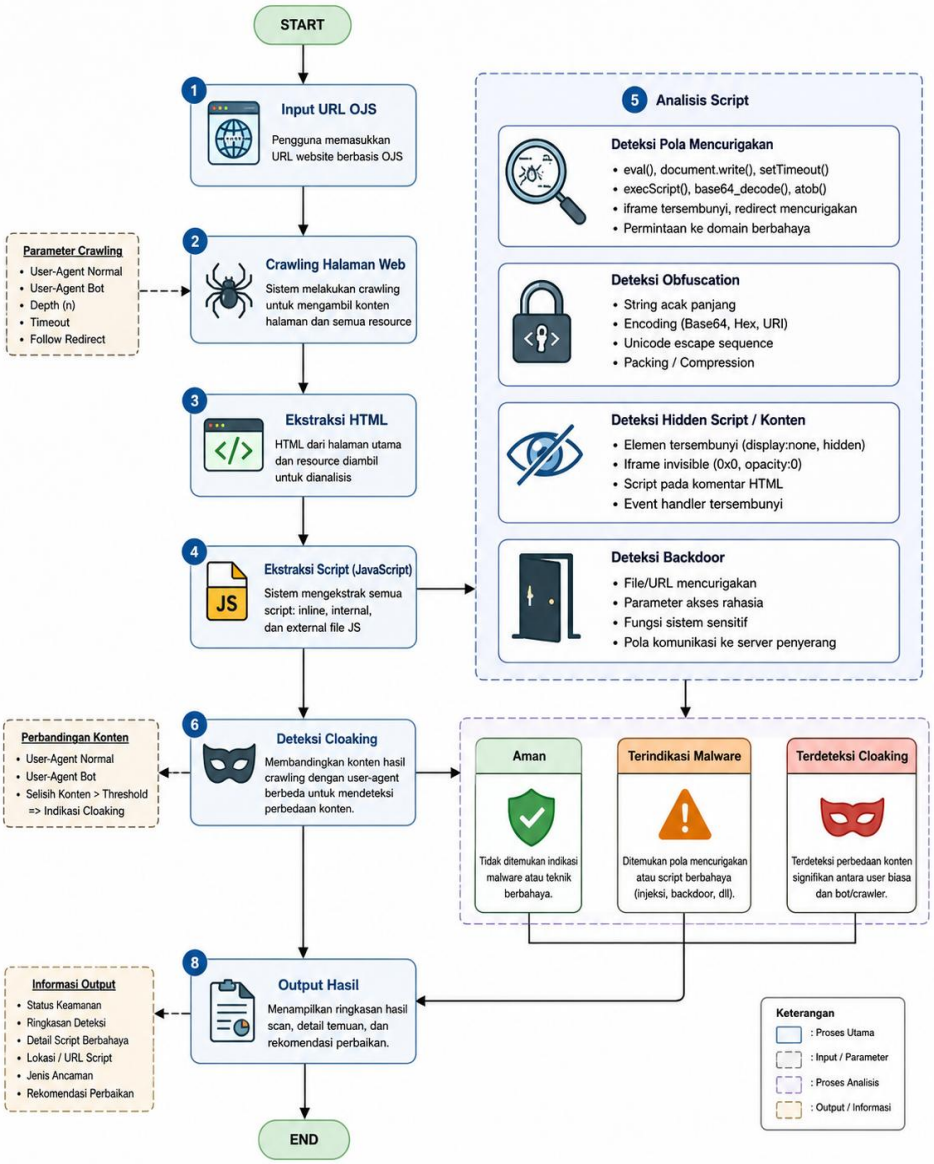
Tahap kedua adalah ekstraksi script, yaitu proses pengambilan elemen JavaScript dari dokumen HTML yang telah diperoleh. Script yang diekstraksi meliputi JavaScript internal, JavaScript eksternal yang terhubung melalui file terpisah, serta inline script yang ditanam langsung dalam elemen HTML. Tahap ini bertujuan untuk mengisolasi bagian-bagian kode yang berpotensi mengandung malware.

Tahap ketiga adalah analisis script, yang dilakukan untuk mengidentifikasi pola-pola mencurigakan yang sering digunakan dalam malware berbasis web. Analisis ini mencakup beberapa aspek, antara lain deteksi penggunaan fungsi berisiko seperti eval() dan document.write(),

identifikasi teknik encoding seperti base64 atau hexadecimal, serta deteksi obfuscation yang ditandai dengan penggunaan string acak panjang atau encoding berlapis. Selain itu, sistem juga mendeteksi keberadaan hidden content, seperti script yang tidak ditampilkan pada antarmuka pengguna atau penggunaan iframe tersembunyi yang sering dimanfaatkan untuk aktivitas berbahaya.

Tahap keempat adalah deteksi cloaking, yang dilakukan dengan membandingkan hasil crawling halaman menggunakan parameter yang berbeda, khususnya user-agent. Sistem akan mengakses halaman menggunakan user-agent normal dan user-agent crawler, kemudian membandingkan perbedaan konten yang dihasilkan. Jika ditemukan perbedaan signifikan antara kedua hasil tersebut, maka halaman diindikasikan menggunakan teknik cloaking untuk menyembunyikan malware.

Tahap terakhir adalah klasifikasi hasil deteksi, di mana sistem mengelompokkan hasil analisis ke dalam beberapa kategori, yaitu halaman aman atau halaman yang terindikasi malware. Indikasi malware selanjutnya diklasifikasikan lebih lanjut berdasarkan jenisnya, seperti cloaking, backdoor, atau script tersembunyi. Hasil klasifikasi ini kemudian ditampilkan kepada pengguna sebagai informasi akhir dari proses deteksi.



Gambar 1. Flowchart Sistem

e-ISSN 2828-0490
JURNAL MEDIA COMPUTER SCIENCE

Berdasar gambar 1 Alur sistem dimulai dari proses input URL website berbasis OJS yang akan dianalisis. Sistem kemudian melakukan crawling untuk mengambil konten halaman web dalam bentuk HTML. Selanjutnya, dilakukan ekstraksi elemen script, khususnya JavaScript, yang kemudian dianalisis untuk mendeteksi pola mencurigakan, obfuscation, dan script tersembunyi. Sistem juga melakukan deteksi cloaking dengan membandingkan hasil akses menggunakan parameter berbeda. Berdasarkan hasil analisis, sistem mengklasifikasikan apakah halaman tersebut aman atau terindikasi mengandung malware, dan hasilnya ditampilkan kepada pengguna.

HASIL DAN PEMBAHASAN

Hasil

Berdasarkan hasil pengujian sistem deteksi malware berbasis analisis script pada platform Open Journal Systems (OJS), diperoleh data pemindaian terhadap website OJS selama periode 30 April 2026 hingga 16 Mei 2026. Pengujian dilakukan menggunakan sistem yang telah dikembangkan dengan tahapan crawling, ekstraksi HTML dan JavaScript, serta analisis script untuk mendeteksi indikasi malware.

Secara keseluruhan, sistem telah melakukan 393 kali proses pemindaian terhadap 241 website OJS, dengan total 60.396 halaman yang berhasil dianalisis. Dari jumlah tersebut, 47.255 halaman (78,26%) teridentifikasi dalam kondisi bersih, 7.216 halaman (11,94%) terindikasi mengandung malware, dan 5.925 halaman (9,81%) mengalami error saat proses crawling.

Persentase temuan ini menunjukkan bahwa lebih dari satu dari sepuluh halaman OJS yang diuji memiliki indikasi malware. Angka ini mengindikasikan bahwa tingkat kerentanan keamanan pada platform OJS masih relatif tinggi dan membutuhkan mekanisme deteksi yang lebih adaptif.

Tabel 1. Rekapitulasi Hasil Pemindaian Sistem

No	Indikator Pengujian	Hasil	Keterangan
1	Jumlah Website OJS	241	Website yang diuji
2	Total Proses Scan	393	Jumlah pengujian
3	Total Halaman Dipindai	60.396	Hasil crawling
4	Halaman Bersih	47.255	Tidak terdeteksi malware
5	Halaman Terinfeksi	7.216	Mengandung indikasi malware
6	Halaman Error	5.925	Gagal diakses/crawling

Berdasarkan Tabel 1, terlihat bahwa sistem mampu mengidentifikasi kondisi halaman secara komprehensif, baik halaman normal maupun halaman yang mengandung indikasi malware. Temuan ini menunjukkan bahwa pendekatan berbasis analisis script dapat diterapkan secara efektif dalam lingkungan nyata (real-world scenario), khususnya pada platform OJS.

Tabel 2. Contoh Hasil Deteksi pada Website OJS

No	URL Website	Total Halaman	Clean	Infected	Keterangan
1	journal.pubmedia.id	200	160	40	Terdapat script mencurigakan
2	jrtppi.id	50	21	29	Terindikasi malware tinggi
3	journal.sttsimpson.ac.id	200	194	6	Relatif aman
4	(beragam domain)	Variatif	Variatif	Variatif	Hasil bervariasi

Variasi tingkat infeksi pada masing-masing website menunjukkan bahwa keamanan OJS sangat bergantung pada konfigurasi sistem, pembaruan software, serta pengelolaan keamanan oleh administrator. Website dengan tingkat infeksi tinggi cenderung memiliki kelemahan keamanan yang memungkinkan terjadinya injeksi script berbahaya.

Pembahasan

Hasil penelitian menunjukkan bahwa sistem deteksi malware berbasis analisis script yang dikembangkan mampu mengidentifikasi keberadaan malware pada halaman Open Journal Systems (OJS) secara efektif. Hal ini ditunjukkan dengan keberhasilan sistem dalam mendeteksi 7.216 halaman yang terindikasi mengandung malware, di mana sebagian besar tidak terlihat secara langsung pada tampilan antarmuka pengguna. Temuan ini menegaskan bahwa pendekatan berbasis client-side analysis memiliki kemampuan yang signifikan dalam mengungkap ancaman tersembunyi yang tidak dapat dideteksi melalui inspeksi visual biasa. Hal ini sejalan dengan penelitian Kapravelos, et al. (2014) yang menekankan bahwa analisis pada sisi klien sangat penting untuk mengidentifikasi perilaku berbahaya yang tidak terdeteksi pada sisi server.

Fenomena ini juga menguatkan karakteristik malware modern yang cenderung bersifat stealthy dan memanfaatkan teknik penyembunyian berbasis script. Dalam konteks ini, malware sering disisipkan dalam bentuk JavaScript yang dieksekusi pada sisi klien. Kuswara dan Fachri (2025) menyatakan bahwa injeksi JavaScript melalui teknik seperti Cross-Site Scripting (XSS) menjadi salah satu metode utama dalam serangan web karena memungkinkan eksekusi kode berbahaya tanpa interaksi langsung dari pengguna. Temuan ini diperkuat oleh penelitian Cova, et al. (2010) yang menunjukkan bahwa sebagian besar serangan web modern, seperti drive-by download, memanfaatkan JavaScript sebagai media utama untuk menyisipkan dan mengeksekusi payload berbahaya. Selain itu, Sohan, et al. (2020) dalam kajian sistematisnya juga menegaskan bahwa JavaScript menjadi vektor dominan dalam penyebaran malware web karena fleksibilitas dan kemampuannya untuk menyamarkan kode berbahaya dalam struktur halaman.

Lebih lanjut, kemampuan sistem dalam mendeteksi script tersembunyi dan obfuscation menunjukkan keunggulan signifikan dibandingkan metode berbasis signature statis. Jelita dan Siregar (2025) mengungkapkan bahwa metode signature memiliki keterbatasan dalam mendeteksi malware yang telah dimodifikasi atau disamarkan melalui teknik encoding. Hal ini diperkuat oleh penelitian Likarish, et al. (2009) yang menyatakan bahwa teknik obfuscation pada JavaScript digunakan secara luas untuk menghindari deteksi berbasis pola tetap. Selain itu, Kolbitsch, et al. (2012) menunjukkan bahwa malware modern secara aktif menggunakan teknik penyamaran untuk menghindari sistem deteksi konvensional. Dalam penelitian ini, pendekatan analisis struktur dan perilaku script memungkinkan sistem untuk mengidentifikasi karakteristik mencurigakan seperti penggunaan fungsi eval(), encoding berlapis, serta pola string acak, sehingga lebih adaptif terhadap variasi malware yang terus berkembang. Pendekatan ini juga sejalan dengan metode yang diusulkan oleh Curtsinger, et al. (2013) melalui sistem Zozzle yang menekankan pentingnya analisis struktur JavaScript dalam mendeteksi malware secara efisien.

Salah satu kontribusi penting dari penelitian ini adalah kemampuan sistem dalam mengidentifikasi teknik cloaking, yaitu metode penyembunyian konten berbahaya dengan menampilkan hasil yang berbeda berdasarkan jenis akses (misalnya user biasa dan crawler). Teknik ini secara luas digunakan dalam praktik serangan web untuk menghindari deteksi oleh sistem keamanan otomatis. Nur, dkk (2025) menyatakan bahwa serangan pada OJS sering kali dilakukan secara tersembunyi dan sulit terdeteksi oleh mekanisme standar.

Hal ini juga didukung oleh Zhang, et al. (2021) yang menunjukkan bahwa teknik cloaking banyak digunakan dalam serangan phishing dan malware berbasis web untuk mengecoh sistem deteksi berbasis crawling. Selain itu, penelitian Canali, et al. (2011) menunjukkan bahwa pendekatan berbasis crawling perlu dikombinasikan dengan analisis lanjutan untuk dapat mendeteksi halaman berbahaya secara efektif.

Dalam penelitian ini, penggunaan pendekatan perbandingan hasil crawling dengan parameter berbeda terbukti mampu mengidentifikasi perbedaan konten secara signifikan, yang menjadi indikator kuat adanya cloaking. Tingginya jumlah halaman yang terinfeksi, yaitu sebesar 11,94% dari total halaman yang dianalisis, menunjukkan bahwa tingkat keamanan pada platform OJS masih perlu ditingkatkan.

Temuan ini konsisten dengan penelitian Yunanri, et al. (2021) yang menyatakan bahwa OJS memiliki berbagai kerentanan yang dapat dimanfaatkan oleh penyerang jika tidak dilakukan pembaruan dan pengamanan secara berkala. Selain itu, studi oleh Herman, dkk (2023) menunjukkan bahwa banyak aplikasi berbasis web di Indonesia masih memiliki celah keamanan yang dapat dieksploitasi melalui injeksi script dan manipulasi konten. Lebih lanjut, Nikiforakis, et al. (2012) juga menekankan bahwa penggunaan script eksternal dalam halaman web dapat menjadi sumber risiko keamanan yang signifikan jika tidak dilakukan validasi secara ketat. Hal ini menguatkan bahwa pendekatan analisis script yang digunakan dalam penelitian ini sangat relevan dalam mengidentifikasi potensi ancaman dari berbagai sumber script.

Di sisi lain, hasil penelitian juga menunjukkan adanya 5.925 halaman (9,81%) yang mengalami error saat proses crawling. Kondisi ini menunjukkan adanya keterbatasan teknis dalam implementasi sistem, terutama dalam menghadapi pembatasan akses server (rate limiting), struktur halaman yang kompleks, serta ketergantungan pada konektivitas jaringan. Wiharja, et al. (2024) menyatakan bahwa proses crawling dalam analisis keamanan web sering menghadapi kendala teknis yang dapat memengaruhi kelengkapan data yang diperoleh. Hal ini juga diperkuat oleh Wang (2018) yang menegaskan bahwa analisis malware berbasis web sangat dipengaruhi oleh kondisi lingkungan saat proses pengambilan data, termasuk respons server dan konfigurasi halaman.

Meskipun terdapat keterbatasan tersebut, sistem yang dikembangkan menunjukkan sejumlah keunggulan yang signifikan dibandingkan pendekatan konvensional. Pertama, sistem ini tidak memerlukan akses ke sisi server (client-side detection), sehingga dapat digunakan secara fleksibel oleh pengguna tanpa hak administratif. Kedua, kemampuan dalam mendeteksi script tersembunyi dan obfuscated memberikan nilai tambah dalam menghadapi malware modern yang bersifat dinamis. Ketiga, fitur deteksi cloaking menjadi keunggulan utama yang jarang ditemukan pada sistem deteksi konvensional. Keempat, sistem dirancang dengan pendekatan praktis sehingga dapat digunakan oleh pengelola jurnal tanpa memerlukan keahlian teknis yang mendalam.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa pendekatan deteksi malware berbasis analisis script pada sisi front-end tidak hanya relevan secara teoritis, tetapi juga efektif secara praktis dalam meningkatkan keamanan platform OJS. Pendekatan ini mampu menjawab keterbatasan metode konvensional dan memberikan kontribusi baru dalam pengembangan sistem deteksi malware berbasis web, khususnya dalam konteks platform jurnal ilmiah.

KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini menyimpulkan bahwa sistem deteksi malware berbasis analisis script pada sisi front-end mampu secara efektif mengidentifikasi ancaman keamanan pada platform Open Journal Systems (OJS) tanpa memerlukan akses ke sisi server. Metode yang dikembangkan terbukti mampu mendeteksi malware tersembunyi melalui analisis struktur dan perilaku script, termasuk injeksi JavaScript, obfuscation, serta teknik cloaking yang tidak dapat diidentifikasi oleh pendekatan konvensional berbasis signature statis. Selain itu, kemampuan sistem dalam membandingkan hasil crawling untuk mendeteksi perbedaan konten menunjukkan efektivitas dalam mengungkap teknik penyembunyian malware yang adaptif.

Temuan ini juga menunjukkan bahwa tingkat kerentanan pada platform OJS masih signifikan, sehingga diperlukan pendekatan deteksi yang lebih dinamis. Secara umum, penelitian ini menegaskan bahwa analisis script berbasis client-side merupakan solusi alternatif yang relevan dan efektif dalam meningkatkan keamanan aplikasi web, khususnya pada sistem pengelolaan jurnal ilmiah.

Saran

Pengelola OJS disarankan menerapkan sistem deteksi berbasis analisis script disertai pemantauan berkala. Pengembangan selanjutnya perlu mengintegrasikan metode adaptif seperti machine learning serta meningkatkan performa crawling. Selain itu, pengujian pada berbagai platform dan pengembangan deteksi real-time terintegrasi sisi server menjadi langkah penting untuk meningkatkan keamanan secara komprehensif.

DAFTAR PUSTAKA

Canali, D., Cova, M., Vigna, G., & Kruegel, C. (2011, March). Prophiler: a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th international conference on World wide web (pp. 197-206). https://doi.org/10.1145/1963405.1963486

Cova, M., Kruegel, C., & Vigna, G. (2010, April). Detection and analysis of drive-by-download attacks and malicious JavaScript code. In Proceedings of the 19th international conference on World wide web (pp. 281-290). https://doi.org/10.1145/1772690.1772767

Curtsinger, C., Livshits, B., Zorn, B., & Seifert, C. (2011, August). Zozzle: Low-overhead mostly static javascript malware detection. In Proceedings of the usenix security symposium (pp. 3-3).

Dirayati, F., & Iin Marliana, T. (2024). Studi Keamanan Sistem Operasi Android: Analisis dan Pencegahan Malware Berbasis Mobile. Jurnal Multimedia dan Android (JMA), 5(02).

Herman, H., Riadi, I., Kurniawan, Y., & Rafiq, I. A. (2023). Analisis Keamanan Website Menggunakan Information System Security Asessment Framework (ISSAF). Jurnal Teknologi Informatika dan Komputer, 9(1), 126-136. https://doi.org/10.37012/jtik.v9i1.1439

Jelita, N. B. A., & Siregar, H. (2025). Systematic Literature Review: Evolusi Ancaman Siber Dan Metode Deteksi Malware Di Sistem Operasi Android (2020–2025). Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI), 4(1), 227-235. https://doi.org/10.62712/juktisi.v4i1.395

Kapravelos, A., Grier, C., Chachra, N., Kruegel, C., Vigna, G., & Paxson, V. (2014). Hulk: Eliciting malicious behavior in browser extensions. In 23rd USENIX Security Symposium (USENIX Security 14) (pp. 641-654).

Kolbitsch, C., Livshits, B., Zorn, B., & Seifert, C. (2012, May). Rozzle: De-cloaking internet malware. In 2012 IEEE Symposium on Security and Privacy (pp. 443-457). IEEE. https://doi.org/10.1109/SP.2012.48

Kuswara, R., & Facrhi, F. (2025). Analisis Keamanan Website Di Smk Wongsorejo Gombong Terhadap Serangan Cross-Site Scripting (Xss) Menggunakan Penetration Testing. JATI (Jurnal Mahasiswa Teknik Informatika), 9(2), 2700-2707. https://doi.org/10.36040/jati.v9i2.13158

Likarish, P., Jung, E., & Jo, I. (2009, October). Obfuscated malicious javascript detection using classification techniques. In 2009 4th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 47-54). IEEE. https://doi.org/10.1109/MALWARE.2009.5403020

Mulyanto, Y., Zaen, M. T. A., Yuliadi, Y., & Sihab, S. (2022). Analisis Keamanan Website SMA Negeri 2 Sumbawa Besar Menggunakan Metode Penetration Testing (Pentest). Journal of Information System Research (JOSH), 4(1), 202-209. https://doi.org/10.47065/josh.v4i1.2335

Nikiforakis, N., Invernizzi, L., Kapravelos, A., Van Acker, S., Joosen, W., Kruegel, C., ... & Vigna, G. (2012, October). You are what you include: large-scale evaluation of remote javascript inclusions. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 736-747). https://doi.org/10.1145/2382196.2382274

Nur, M. N. A., Lolok, A., Samsul, A. S., & Barata, L. O. A. (2025). Strategi Keamanan Jurnal Berbasis OJS melalui Program Pendampingan dan Pencegahan Malware di Universitas Halu Oleo. Jurnal Pengbadian Masyarakat Ilmu Terapan (JPMIT), 7(2), 182-192.

Prabowo, C. R., Rohadi, E. ., & Irmanto, D. . (2024). Study and Analysis of Docker Internal System Security From Docker Daemon Attack and DDOS on The Open Journal System. JPUA: Jurnal Perpustakaan Universitas Airlangga: Media Informasi Dan Komunikasi Kepustakawanan, 14(1), 61–68. https://doi.org/10.20473/jpua.v14i1.2024.61-68.

Sohan, M. F., & Basalamah, A. (2020). A systematic literature review and quality analysis of Javascript malware detection. IEEE Access, 8, 190539-190552. https://doi.org/10.1109/ACCESS.2020.3031690

Ulvi, K., Rasikha, G., Aulia, R., Rika, S., & Hidayani, M. (2026). Kajian Ancaman Malware dan Upaya Pencegahan melalui Penguatan Keamanan Sistem Komputer. JIKUM: Jurnal Ilmu Komputer, 2(1), 108-110. https://doi.org/10.62671/jikum.v2i1.155

Junjie, W. (2018). Detection and Analysis of Web-based Malware and Vulnerability (Doctoral dissertation, Doctoral Thesis, Nanyang Technological University). https://doi.org/10.32657/10220/47659

Wiharja, S., Pradeka, D., & Suteddy, W. (2024). Comparative study of the effect of datasets and machine learning algorithms for pdf malware detection. Digital Zone: Jurnal Teknologi Informasi dan Komunikasi, 15(1), 80-93. https://doi.org/10.31849/digitalzone.v15i1.19744

Yulisusianto, G., Kurniawan, M. A., & Cahyono, Y. D. (2026). Qualitative Analysis of the Security of URLs Extracted from QR Codes Using Artificial Intelligence: A Review of Current Literature: Rekayasa Keamanan Siber. Jurnal Telkommil, 6(2), 183–193.

Yunanri, W., Yuwono, D. T., & Yuliadi, Y. (2021). Deteksi Serangan Vulnerability Pada Open Jurnal System Menggunakan Metode Black-Box. Jurnal Informatika dan Rekayasa Elektronik, 4(1), 68-77. https://doi.org/10.36595/jire.v4i1.365

Zakariya, R. A. I., & Ramli, K. (2024). Desain Penilaian Risiko Privasi pada Aplikasi Seluler Melalui Model Machine Learning Berbasis Ensemble Learning dan Multiple Application Attributes. Jurnal Teknologi Informasi dan Ilmu Komputer, 10(4), 831-842. https://doi.org/10.25126/jtiik.2024107029

Zhang, P., Oest, A., Cho, H., Sun, Z., Johnson, R. C., Wardman, B., ... & Ahn, G. J. (2021, May). Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In 2021 IEEE Symposium on Security and Privacy (SP) (pp. 1109-1124). IEEE. https://doi.org/10.1109/SP40001.2021.00021