JURNAL SISFOTENIKA
Vol. 16 No. 2, Juli 2026
p-ISSN: 2087-7897 ; e-ISSN : 2460-5344
DOI : 10.30700/sisfotenika.v16i2.699

# Pengembangan JemagoScan sebagai Sistem Pemindai Malware dan Backdoor pada Website

# *Development of JemagoScan as a Malware and Backdoor Detection System for Websites*

**Febryan Hari Purwanto*1, Firmansyah Alfarisi2**
[1,2]Sekolah Tinggi Kesehatan Al-Fatah Bengkulu; Jl. Indragiri gang 3 Serangkai, (0376) 27508
e-mail: *[1]fharipurwanto@gmailcom, [2]firmansyahalfarisi24@gmail.com

**Abstrak**

*Perkembangan layanan berbasis web pada institusi pendidikan meningkatkan kebutuhan terhadap keamanan sistem informasi, khususnya pada website yang dihosting pada lingkungan shared hosting. Salah satu ancaman yang sering ditemukan adalah penyisipan malware, backdoor, web shell, dan konten judi online yang dapat mengganggu integritas serta ketersediaan website. Penelitian sebelumnya telah menghasilkan JemagoWEB sebagai sistem perlindungan file website yang mampu melakukan monitoring, pemulihan, dan pencadangan file secara otomatis. Namun, efektivitas sistem tersebut bergantung pada kondisi awal hosting yang harus terbebas dari file berbahaya. Oleh karena itu, penelitian ini mengembangkan JemagoScan sebagai sistem audit keamanan website yang berfungsi melakukan pemindaian dan identifikasi ancaman sebelum implementasi JemagoWEB dilakukan. JemagoScan dikembangkan menggunakan bahasa pemrograman PHP dan dirancang agar dapat berjalan pada lingkungan shared hosting tanpa memerlukan akses root server. Sistem mampu mendeteksi berbagai indikasi ancaman, seperti malware, backdoor, web shell, file mencurigakan, konfigurasi izin akses yang tidak aman, redirect berbahaya, serta indikasi konten judi online menggunakan metode deteksi berbasis heuristik. Hasil pengujian menunjukkan bahwa JemagoScan mampu membantu proses identifikasi ancaman keamanan pada website sehingga kondisi hosting dapat dipastikan lebih aman sebelum dilakukan perlindungan berkelanjutan menggunakan JemagoWEB. Integrasi kedua sistem tersebut menghasilkan mekanisme keamanan berlapis yang mendukung peningkatan integritas dan keamanan website pada lingkungan shared hosting.*

***Kata kunci****—3-5 keamanan website, deteksi malware dan backdoor, analisis heuristik, integritas website, JemagoScan*

***Abstract***

*The rapid growth of web-based services in educational institutions has increased the need for robust information security, particularly for websites hosted in shared hosting environments. Common threats include malware, backdoors, web shells, and online gambling content that compromise website integrity and availability. Previous research introduced JemagoWEB, a website file protection system that automatically monitors, restores, and backs up website files. However, its effectiveness depends on a hosting environment free from malicious files. Therefore, this study develops JemagoScan, a website security auditing system that scans and identifies*

*security threats before the deployment of JemagoWEB. Developed in PHP for shared hosting environments without requiring root server access, JemagoScan uses a heuristic-based detection approach to identify malware, backdoors, web shells, suspicious files, insecure file permissions, malicious redirects, and indicators of online gambling content. Experimental results show that JemagoScan effectively identifies website security threats, ensuring a safer hosting environment before continuous protection is applied through JemagoWEB. The integration of both systems provides a layered security mechanism that improves website integrity and security in shared hosting environments.*

***Keywords—****3-5 website security, malware and backdoor detection, heuristic analysis, website integrity, JemagoScan*

## 1. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong berbagai institusi pendidikan untuk memanfaatkan sistem berbasis web dalam mendukung layanan akademik dan administrasi [1]. Website jurnal ilmiah dan sistem Penerimaan Mahasiswa Baru (PMB) merupakan dua contoh layanan digital yang saat ini menjadi bagian penting dalam operasional perguruan tinggi [2]. Pemanfaatan sistem informasi akademik berbasis web terbukti memberikan keuntungan yang signifikan, meliputi kemudahan akses, peningkatan transparansi, otomasi pencatatan data, serta peningkatan mutu pelayanan kampus kepada mahasiswa dan masyarakat [3].

Di sisi lain, meningkatnya ketergantungan pada layanan website berbanding lurus dengan ancaman keamanan siber yang makin bervariasi [4]. Berbagai bentuk eksploitasi, seperti malware, web shell, backdoor, defacement, dan penyisipan konten ilegal, semakin sering menargetkan situs web yang dikelola oleh institusi pendidikan [5]. Salah satu tren serangan yang marak dalam beberapa tahun terakhir adalah pemanfaatan kerentanan pada subdomain akademik untuk menyisipkan redirect injection ke halaman situs judi online (slot) [6]. Penyerang umumnya mengeksploitasi celah kerentanan aplikasi web, kelemahan konfigurasi, maupun kerentanan pada fitur unggah (upload) file. Akibatnya, integritas sistem rusak, reputasi institusi menurun tajam, dan domain kampus berisiko dilaporkan (flagged) sebagai situs berbahaya oleh mesin pencari [7].

Permasalahan tersebut menjadi semakin kompleks pada lingkungan shared hosting. Model hosting ini paling banyak digunakan oleh institusi pendidikan maupun organisasi skala menengah untuk menyimpan semua data informasi pada website dan pengelolaannya yang praktis [8]. Namun, pengguna shared hosting umumnya tidak memiliki hak akses penuh (root access) ke sistem operasi server, sehingga pergerakan mereka sangat terbatas dalam mengonfigurasi perangkat keamanan tingkat lanjut di sisi backend. Keterbatasan ini sering kali menyebabkan keterlambatan deteksi; di mana file berbahaya dapat bersarang dan aktif di dalam direktori pengguna (user directory) dalam jangka waktu yang lama sebelum akhirnya disadari oleh administrator [9].

Untuk memitigasi peretasan tersebut, penelitian sebelumnya telah mengembangkan JemagoWEB, yakni sistem proteksi file website berbasis PHP dan cron job yang menyerupai mekanisme File Integrity Monitoring (FIM) sebuah metode yang terbukti efektif menekan serangan web defacement [10]. JemagoWEB mampu mendeteksi perubahan file, menghapus anomali, memulihkan file dari cadangan (backup), serta mencatat log aktivitas secara otomatis. Akan tetapi, efektivitas sistem FIM maupun mekanisme backup sangat bergantung pada tingkat integritas awal (baseline) dari direktori yang dilindungi. Apabila pada saat implementasi perlindungan awal sudah terdapat skrip webshell atau backdoor yang tersembunyi, sistem justru berisiko mengidentifikasi file berbahaya tersebut sebagai berkas sah dan menyimpannya ke dalam data backup [11].

Berdasarkan celah tersebut, diperlukan suatu mekanisme audit keamanan (pemeriksaan kerentanan awal) yang mampu memindai hosting secara komprehensif sebelum mekanisme pelindungan pada JemagoWEB diaktifkan. Audit pra-perlindungan ini mutlak diperlukan untuk memastikan bahwa direktori website benar-benar bersih dari ancaman, sehingga sistem pelindung memiliki referensi baseline yang murni. Selain itu, sistem pemindai ini harus dirancang agar dapat beroperasi secara independen di lingkungan shared hosting tanpa memerlukan perizinan root [12].

Sebagai solusi atas kebutuhan tersebut, penelitian ini mengembangkan JemagoScan, yaitu sistem pemindaian keamanan website berbasis PHP yang dirancang secara khusus untuk mendeteksi berbagai indikasi ancaman pada lingkungan shared hosting [13]. Mengingat file backdoor sering kali sulit diidentifikasi dan ditanamkan melalui teknik yang kompleks, diperlukan pendekatan pemindaian yang lebih spesifik. JemagoScan menggunakan pendekatan heuristik untuk mengidentifikasi pola kode berbahaya, terutama skrip PHP yang sengaja disamarkan (diobfuskasi) oleh peretas agar mampu menghindari deteksi Web Application Firewall (WAF) standar [14]. Pemindaian difokuskan pada pelacakan fungsi-fungsi yang lazim digunakan sebagai signature eksploitasi webshell, seperti penyalahgunaan eval(), base64_decode(), dan shell_exec() [14]. Selain itu, sistem ini dibekali kemampuan untuk mendeteksi file PHP tersembunyi, mengaudit konfigurasi perizinan (permission) yang rentan, melacak pengalihan (redirect) ilegal pada file .htaccess, serta menandai indikasi konten judi online berdasarkan ekstraksi kata kunci. Berbeda dengan perangkat lunak keamanan tingkat server yang menuntut hak akses administratif, JemagoScan dapat dijalankan secara mandiri di dalam direktori pengguna, menjadikannya solusi yang praktis bagi pengelola website institusi pendidikan.

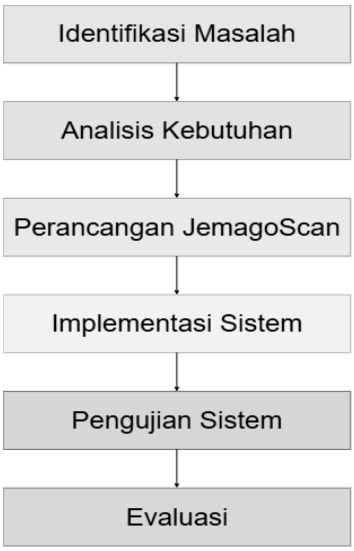
Penelitian ini bertujuan untuk merancang dan menguji efektivitas JemagoScan sebagai tahap audit keamanan (deteksi awal) sebelum perlindungan JemagoWEB (mitigasi lanjutan) diterapkan [11]. Penyatuan kedua sistem ini membentuk ekosistem pertahanan keamanan berlapis (layered security atau defense-in-depth) yang sangat direkomendasikan untuk menekan area serangan (attack surface) pada infrastruktur data perguruan tinggi. Integrasi ini diharapkan dapat menjadi solusi praktis, andal, dan mudah diimplementasikan guna memutus rantai eksploitasi siber dan penyisipan konten judi online pada institusi pendidikan pengguna shared hosting.

## 2. METODE PENELITIAN

### 2.1 *Metode Penelitian*

Penelitian ini menggunakan metode *Research and Development* (R&D) yang bertujuan untuk mengembangkan sistem audit keamanan website bernama JemagoScan sekaligus menguji kemampuannya dalam mendeteksi berbagai ancaman keamanan pada lingkungan *shared hosting*. Metode R&D dipilih karena penelitian tidak hanya berfokus pada identifikasi permasalahan, tetapi juga menghasilkan solusi berupa perangkat lunak yang dirancang, diimplementasikan, dan dievaluasi secara sistematis [15]. Pendekatan ini sesuai dengan karakteristik penelitian pengembangan sistem informasi yang menekankan proses inovasi, validasi, serta evaluasi produk secara bertahap hingga menghasilkan solusi yang dapat diimplementasikan sesuai kebutuhan pengguna [16]. Selain itu, penerapan metode R&D memberikan fleksibilitas dalam melakukan pengembangan perangkat lunak secara iteratif sehingga setiap tahapan dapat dievaluasi dan disempurnakan berdasarkan hasil implementasi maupun pengujian sistem.

Dalam penelitian ini, metode R&D dimodifikasi sesuai dengan karakteristik pengembangan sistem keamanan website. Proses penelitian disederhanakan menjadi enam tahapan utama, yaitu identifikasi masalah, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Penyederhanaan tahapan dilakukan agar proses pengembangan lebih efektif tanpa mengurangi kualitas hasil penelitian, sekaligus memastikan bahwa sistem yang dikembangkan mampu menjawab kebutuhan pengguna dan permasalahan keamanan website pada lingkungan *shared hosting*.




Gambar 1. Tahapan Metode Penelitian

### *2.2 Identifikasi Masalah*

Tahap awal penelitian dilakukan melalui observasi terhadap website yang dioperasikan pada lingkungan *shared hosting*. Hasil observasi menunjukkan bahwa website masih rentan terhadap berbagai ancaman keamanan, seperti *malware*, *web shell*, *backdoor*, modifikasi file tanpa izin, serta penyisipan konten judi online. Ancaman tersebut berpotensi mengganggu integritas file, menurunkan ketersediaan layanan, dan merusak reputasi institusi apabila website berhasil disusupi atau dimanfaatkan sebagai media penyebaran konten ilegal.

Penelitian sebelumnya telah mengembangkan JemagoWEB sebagai sistem *File Integrity Monitoring* (FIM) yang mampu memantau perubahan file, melakukan *auto backup*, *restore*, serta menghapus file yang terindikasi mencurigakan [11]. Meskipun demikian, efektivitas sistem tersebut sangat dipengaruhi oleh kondisi awal direktori yang dijadikan *baseline*. Apabila proses pembentukan *baseline* dilakukan ketika hosting telah terinfeksi *malware* atau *backdoor*, maka file berbahaya berpotensi tersimpan sebagai data referensi sehingga menurunkan efektivitas mekanisme perlindungan. Oleh karena itu, penelitian ini mengembangkan JemagoScan sebagai mekanisme *pre-security assessment* untuk memastikan kondisi hosting telah bersih dari file berbahaya sebelum proses pemantauan integritas file dijalankan oleh JemagoWEB.

### 2.3 *Analisis Kebutuhan*

Tahap analisis kebutuhan dilakukan untuk menentukan fungsi-fungsi yang harus dimiliki oleh JemagoScan berdasarkan karakteristik ancaman yang ditemukan pada tahap identifikasi masalah. Analisis difokuskan pada berbagai pola serangan yang umum ditemukan pada website berbasis PHP, khususnya pada lingkungan *shared hosting* yang memiliki keterbatasan akses terhadap konfigurasi server. Oleh karena itu, sistem harus mampu melakukan pemeriksaan keamanan secara mandiri tanpa memerlukan hak akses root maupun instalasi perangkat lunak tambahan pada sisi server.

Berdasarkan hasil analisis, JemagoScan dirancang sebagai sistem Website Security Assessment yang mampu melakukan pemindaian secara rekursif terhadap seluruh file dan direktori website. Sistem menerapkan pendekatan heuristic analysis, yaitu mendeteksi indikasi ancaman berdasarkan karakteristik kode maupun perilaku file, bukan hanya berdasarkan signature tertentu. Fitur yang dirancang meliputi deteksi pola kode berbahaya (code pattern), identifikasi

hidden PHP, analisis payload Base64, pengukuran tingkat Shannon entropy untuk mendeteksi file yang mengalami obfuscation, pemeriksaan konfigurasi permission file dan direktori, identifikasi file PHP di luar direktori public_html, pendeteksian nama file dan folder mencurigakan, analisis konfigurasi .htaccess, serta pendeteksian kata kunci yang berkaitan dengan konten judi online. Selain proses deteksi, sistem juga menghasilkan laporan hasil pemindaian dalam bentuk log sebagai dasar pengambilan keputusan sebelum implementasi JemagoWEB.

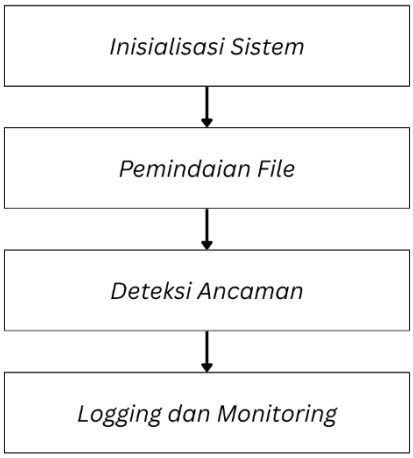
### 2.4 *Perancangan Sistem*

Tahap perancangan dilakukan dengan menyusun arsitektur JemagoScan beserta alur kerja sistem berdasarkan kebutuhan fungsional yang telah diidentifikasi. Sistem dibangun menggunakan pemrograman PHP sehingga dapat dijalankan pada lingkungan *shared hosting* tanpa memerlukan akses administratif terhadap server.

Perancangan sistem terdiri atas dua bagian utama, yaitu arsitektur sistem yang menjelaskan hubungan antar komponen penyusun aplikasi serta flowchart yang menggambarkan alur kerja JemagoScan.

### 2.5 *Arsitektur JemagoScan*

Arsitektur JemagoScan dirancang menggunakan pendekatan modular sehingga setiap komponen memiliki fungsi yang berbeda namun saling berhubungan. Aristektur sistem terdiri atas beberapa komponen utama, yaitu Inisialisasi Sistem, Pemindaian File, Deteksi Ancaman dan *Logging* dan *Monitoring*.




Gambar 2. Arsitektur JemagoScan

Penjelasan masing-masing komponen sebagai berikut :

1. Inisialisasi Sistem
   Komponen ini bertugas menyiapkan konfigurasi sistem sebelum proses pemindaian dimulai, seperti pengaturan parameter eksekusi, direktori target, file log, file progres, serta perhitungan jumlah file yang akan dipindai.
2. Pemindaian File
   Komponen ini melakukan penelusuran seluruh file dan direktori website secara rekursif menggunakan Recursive Directory Iterator sehingga seluruh file dapat diperiksa secara otomatis.
3. Deteksi Ancaman
   Komponen ini melakukan pemeriksaan keamanan terhadap setiap file menggunakan metode deteksi heuristik, meliputi analisis *permission*, *code pattern*, *hidden PHP*,

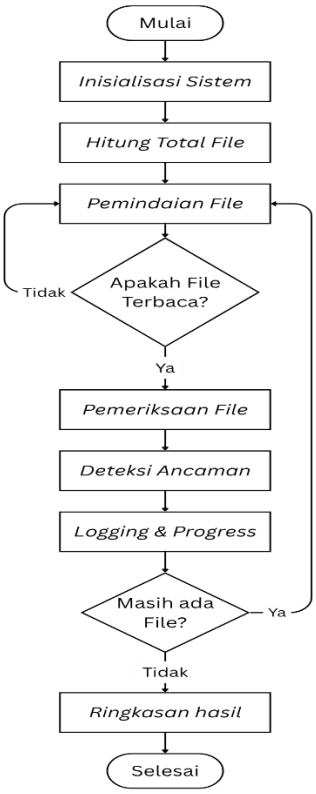
*payload Base64*, *Shannon entropy*, file PHP di luar *public_html*, nama file dan folder mencurigakan, konfigurasi *.htaccess*, serta kata kunci yang berkaitan dengan konten judi online.

4. Logging dan Monitoring

   Komponen ini mencatat seluruh aktivitas dan hasil pemindaian ke dalam file log secara *real-time* serta memperbarui informasi progres pemindaian hingga seluruh proses selesai. File log yang dihasilkan menjadi dokumentasi hasil audit sekaligus dasar evaluasi keamanan website.

## 2.6 *Flowchart JemagoScan*

*Flowchart* atau diagram alir merupakan representasi visual yang menguraikan urutan proses dan langkah-langkah kerja suatu program secara sistematis dan berurutan [17].
Flowchart sistem digunakan untuk menggambarkan alur kerja JemagoScan dalam melakukan audit keamanan website secara otomatis. Alur ini disusun berdasarkan implementasi program sehingga merepresentasikan proses yang dijalankan sistem mulai dari inisialisasi hingga menghasilkan laporan hasil pemindaian.




Gambar 3. Flowchart JemagoScan

Berdasarkan Gambar 3, proses diawali dengan inisialisasi sistem, yaitu menyiapkan parameter yang dibutuhkan sebelum pemindaian dilakukan, seperti direktori target, file log, file progres, dan konfigurasi eksekusi program. Selanjutnya sistem menghitung jumlah file yang akan dipindai sebagai dasar untuk memantau perkembangan proses pemindaian.

Setelah proses inisialisasi selesai, sistem mulai mengambil setiap file yang terdapat pada direktori website untuk diperiksa. File yang valid kemudian menjalani proses pemeriksaan file, meliputi identifikasi karakteristik dasar seperti hak akses (*permission*), nama file, lokasi file, dan struktur direktori. Tahap berikutnya adalah deteksi ancaman, yaitu analisis heuristik terhadap isi file untuk mengidentifikasi berbagai indikasi ancaman keamanan, seperti *hidden PHP*, *code pattern* berbahaya, *payload Base64*, *Shannon entropy*, konfigurasi *.htaccess*, serta kata kunci yang berkaitan dengan konten judi online.

Hasil pemeriksaan selanjutnya dicatat secara otomatis ke dalam file log, disertai pembaruan informasi progres pemindaian. Proses tersebut dilakukan secara berulang untuk setiap file hingga seluruh file pada direktori website selesai diperiksa. Setelah proses pemindaian berakhir, sistem menghasilkan ringkasan hasil pemindaian yang menjadi dasar bagi administrator untuk melakukan evaluasi dan pembersihan hosting sebelum implementasi sistem perlindungan integritas file JemagoWEB.

## 3. HASIL DAN PEMBAHASAN

### *3.1 Implementasi Sistem*

JemagoScan diimplementasikan pada lingkungan *shared hosting* sebagai aplikasi audit keamanan website berbasis PHP yang dijalankan melalui *Command Line Interface* (CLI). Proses implementasi diawali dengan mengunggah file JemagoScan.php ke direktori utama akun hosting (/home/usernamecpanel/) melalui File Manager pada cPanel. Setelah file berhasil diunggah, hak akses (*permission*) diatur menjadi 700 agar aplikasi hanya dapat diakses dan dijalankan oleh pemilik akun. Tahapan tersebut dilakukan untuk menjaga keamanan aplikasi sekaligus menyesuaikan dengan karakteristik layanan *shared hosting* yang tidak memerlukan perubahan konfigurasi server maupun hak akses *root*.

Setelah proses konfigurasi selesai, aplikasi dijalankan melalui Terminal pada cPanel menggunakan perintah php /home/usernamecpanel/JemagoScan.php. Perintah tersebut mengeksekusi JemagoScan sehingga sistem secara otomatis melakukan pemindaian terhadap seluruh file dan direktori website menggunakan mekanisme *Recursive Directory Iterator*. Seluruh file yang ditemukan kemudian dianalisis menggunakan metode deteksi heuristik berdasarkan sejumlah indikator keamanan, meliputi deteksi *Hidden PHP*, *Code Pattern*, *Base64 Payload*, *High Entropy*, file PHP di luar direktori public_html, konfigurasi .htaccess, hak akses (*permission*), struktur direktori, serta kata kunci yang berkaitan dengan konten judi online. Proses implementasi ditunjukkan pada gambar 4.

```
[akfaralf@murai ~]$ php /home/akfaralf/JemagoScan.php
[2026-07-08 06:04:06] === SCAN START: 2026-07-08 06:04:06 ===
[2026-07-08 06:04:06] Pre-scan: menghitung total file (ini bisa memakan
waktu jika banyak)...
[2026-07-08 06:04:10] Pre-scan: total files = 78501
[2026-07-08 06:04:10] [PROGRESS] scanned=1 total=78501 percent=0% last=.
bash_logout elapsed=0s
[2026-07-08 06:04:10] [PROGRESS] scanned=50 total=78501 percent=0.06% la
st=.cpanel/nvdata/koalitySignupEmail elapsed=0s
[2026-07-08 06:04:10] [PROGRESS] scanned=100 total=78501 percent=0.13% l
ast=.cpanel/vcards/admin@sister.stikesalfatah.ac.id.vcf elapsed=0s
[2026-07-08 06:04:10] [PROGRESS] scanned=150 total=78501 percent=0.19% l
ast=public_html/jurnal/files/journals/1/articles/11/submission/original/
11-30-1-SM.docx elapsed=0s
[2026-07-08 06:04:10] [PROGRESS] scanned=200 total=78501 percent=0.25% l
ast=public_html/jurnal/files/journals/1/articles/133/submission/copyedit
/133-323-1-CE.txt elapsed=0s
```

Gambar 4. Implementasi JemagoSan

Implementasi tersebut memungkinkan administrator melakukan pemeriksaan kondisi keamanan website secara rutin sebagai langkah awal sebelum menerapkan sistem JemagoWEB untuk melakukan *File Integrity Monitoring*, sehingga kondisi awal website dapat dipastikan telah melalui proses audit keamanan.

### *3.2 Hasil Pengujian Sistem*

Pengujian sistem dilakukan untuk mengevaluasi kemampuan JemagoScan dalam melakukan audit keamanan terhadap website pada lingkungan *shared hosting*. Setelah aplikasi dijalankan menggunakan perintah, sistem secara otomatis melakukan pemindaian terhadap seluruh file dan direktori website. Selama proses pemindaian berlangsung, seluruh aktivitas pemeriksaan serta hasil deteksi dicatat secara otomatis ke dalam file JemagoScan.log seperti terlihat pada gambar 4. Administrator dapat memantau proses tersebut secara langsung melalui Terminal menggunakan perintah tail -f /home/usernamecpanel/JemagoScan.log, sehingga setiap indikator ancaman yang berhasil dideteksi dapat diamati secara *real-time*.

```
[2025-11-06 02:44:28] [GAMBLING KEYWORD] tmp/awstats/awstats102025.stikesalfatah.
akfar-alfatah.ac.id.txt contains keyword 'spin'
[2025-11-06 02:44:28] [HIGH ENTROPY] tmp/awstats/awstats112025.pmb.stikesalfatah.ac.id.txt
entropy=4.6195143000449
[2025-11-06 02:44:28] [HIGH ENTROPY] tmp/awstats/awstats112025.stikesalfatah.akfar-alfatah.
ac.id.txt entropy=4.6453410558601
[2025-11-06 02:44:28] [GAMBLING KEYWORD] tmp/awstats/awstats112025.stikesalfatah.
akfar-alfatah.ac.id.txt contains keyword 'vip'
[2025-11-06 02:44:28] [HIGH ENTROPY] tmp/awstats/ssl/awstats102025.perpustakaan.
stikesalfatah.ac.id.txt entropy=4.6667127367661
[2025-11-06 02:44:28] [GAMBLING KEYWORD] tmp/awstats/ssl/awstats102025.perpustakaan.
stikesalfatah.ac.id.txt contains keyword 'slot'
[2025-11-06 02:44:28] [HIGH ENTROPY] tmp/awstats/ssl/awstats102025.stikesalfatah.
akfar-alfatah.ac.id.txt entropy=4.638415852887
[2025-11-06 02:44:28] [GAMBLING KEYWORD] tmp/awstats/ssl/awstats102025.stikesalfatah.
akfar-alfatah.ac.id.txt contains keyword 'agen'
[2025-11-06 02:44:28] [HIGH ENTROPY] tmp/awstats/ssl/awstats112025.jurnal.stikesalfatah.ac.
id.txt entropy=4.6841248200481
[2025-11-06 02:44:28] [GAMBLING KEYWORD] tmp/awstats/ssl/awstats112025.jurnal.stikesalfatah.
ac.id.txt contains keyword 'slot'
```

Gambar 5. Proses pemindaian file pada JemagoScan.log

File JemagoScan.log menjadi sumber data utama dalam proses evaluasi karena memuat informasi mengenai waktu pelaksanaan pemindaian, lokasi file yang diperiksa, jenis indikator ancaman yang terdeteksi, serta ringkasan hasil audit setelah seluruh proses selesai dilaksanakan. Berdasarkan hasil pengujian.

### *3.3 Hasil Deteksi Ancaman*

Setelah proses pemindaian selesai, JemagoScan menghasilkan ringkasan jumlah temuan berdasarkan masing-masing indikator deteksi yang digunakan selama proses audit. Ringkasan tersebut memberikan gambaran mengenai kondisi keamanan website serta menjadi dasar bagi administrator dalam menentukan file yang memerlukan pemeriksaan lebih lanjut. Hasil ringkasan deteksi ditunjukkan pada Tabel 1.

Tabel 1. Ringkasan Hasil Deteksi JemagoScan

| Indikator Deteksi | Jumlah Temuan |
|---|---|
| *RANDOM_NAME* | 853 |
| *SUSP_NAME* | 14 |

| | |
|---|---|
| *PHP_OUTSIDE* | 3328 |
| *HIDDEN_PHP* | 57 |
| *CODE_PATTERN* | 906 |
| *BASE64_PAYLOAD* | 28 |
| *HIGH_ENTROPY* | 22614 |
| *GAMBLING_KEYWORD* | 1898 |
| *HITACCESS_REDIRECT* | 0 |
| *PERMISSION_WARNING* | 20 |
| *SUSP_FOLDER* | 0 |
| *FOLDER_OUTSIDE* | 23 |
| *WARNING_UNREADABLE* | 2 |
| *ERRORS* | 0 |

Berdasarkan Tabel 1, indikator *High Entropy* menghasilkan jumlah temuan paling banyak, yaitu sebanyak 22.614 file. Temuan tersebut menunjukkan bahwa sistem mampu mengidentifikasi file yang memiliki tingkat entropi tinggi sebagai indikasi awal adanya proses *encoding*, kompresi, maupun *obfuscation*. Namun demikian, karakteristik tersebut tidak secara langsung menunjukkan keberadaan malware karena beberapa *library*, *framework*, maupun file hasil kompresi yang sah juga memiliki nilai entropi yang tinggi.

Selain itu, JemagoScan berhasil mendeteksi 906 file yang mengandung Code Pattern, 57 file dengan karakteristik Hidden PHP, 28 file yang mengandung Base64 Payload, serta 3.328 file PHP yang berada di luar direktori public_html. Sistem juga menemukan 20 file dengan konfigurasi *permission* yang memerlukan perhatian administrator dan 23 direktori yang berada di luar struktur direktori website. Temuan-temuan tersebut menunjukkan bahwa pendekatan deteksi heuristik yang diterapkan mampu mengidentifikasi berbagai karakteristik file yang berpotensi dimanfaatkan sebagai *web shell*, *backdoor*, maupun ancaman lainnya pada website berbasis PHP.

Indikator *Gambling Keyword* menghasilkan 1.898 temuan yang menunjukkan adanya file yang mengandung kata kunci tertentu yang berkaitan dengan konten judi online. Temuan tersebut berfungsi sebagai indikator awal (*early warning*) sehingga setiap file yang terdeteksi masih memerlukan proses verifikasi lebih lanjut sebelum dapat disimpulkan sebagai penyisipan konten judi online.

### ***3.4*** *Analisis Hasil Pengujian*

Hasil pengujian menunjukkan bahwa seluruh mekanisme yang dirancang pada JemagoScan berhasil diimplementasikan dengan baik. Sistem mampu melakukan audit terhadap 63.530 file tanpa mengalami kegagalan proses (Errors = 0), sehingga menunjukkan bahwa mekanisme pemindaian dapat berjalan secara stabil pada lingkungan *shared hosting*. Pendekatan deteksi heuristik yang diterapkan juga mampu mengidentifikasi berbagai karakteristik file yang berpotensi mengandung ancaman keamanan tanpa bergantung pada basis data *signature* malware.

Berdasarkan hasil evaluasi, indikator Code Pattern, Hidden PHP, Base64 Payload, PHP Outside Public_html, serta Permission Warning mampu membantu administrator mengidentifikasi file yang memerlukan pemeriksaan lebih lanjut. Sementara itu, indikator *High Entropy* berfungsi sebagai mekanisme pendukung untuk mendeteksi file yang memiliki karakteristik *obfuscation*. Oleh karena itu, hasil deteksi pada indikator tersebut tidak dapat langsung digunakan sebagai dasar untuk menyatakan bahwa suatu file merupakan malware, tetapi perlu dianalisis kembali berdasarkan isi dan fungsi file yang bersangkutan.

Evaluasi juga menunjukkan bahwa indikator *Gambling Keyword* masih memiliki keterbatasan karena menggunakan pendekatan berbasis pencocokan kata kunci (*keyword matching*). Berdasarkan hasil pengujian, beberapa kata kunci yang digunakan masih menghasilkan *false positive*, terutama pada kata "agen". Kata tersebut banyak ditemukan pada file yang tidak berkaitan dengan aktivitas perjudian daring sehingga tetap terdeteksi sebagai temuan

oleh sistem. Oleh karena itu, setiap file yang teridentifikasi melalui indikator Gambling Keyword perlu dilakukan pemeriksaan manual untuk memastikan apakah file tersebut benar-benar mengandung sisipan kode judi online, *web shell*, maupun malware lainnya. Dengan demikian, indikator ini lebih tepat digunakan sebagai mekanisme early warning dibandingkan sebagai penentu akhir keberadaan konten ilegal.

Secara keseluruhan, hasil penelitian menunjukkan bahwa JemagoScan mampu menjalankan fungsinya sebagai mekanisme audit keamanan awal (*pre-security assessment*) sebelum implementasi JemagoWEB. Melalui proses audit tersebut, administrator dapat memastikan kondisi hosting telah diperiksa dan melakukan pembersihan terhadap file yang terindikasi berbahaya sebelum proses pembentukan *baseline* pada JemagoWEB dilakukan. Dengan demikian, efektivitas mekanisme *File Integrity Monitoring* pada JemagoWEB dapat ditingkatkan karena proses pemantauan dilakukan pada kondisi website yang telah melalui tahap audit keamanan.

## 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut.

1. Penelitian ini berhasil mengembangkan JemagoScan sebagai aplikasi audit keamanan website berbasis PHP yang dapat diimplementasikan pada lingkungan shared hosting melalui Command Line Interface (CLI). Sistem mampu melakukan pemindaian secara otomatis terhadap 63.530 file menggunakan pendekatan deteksi heuristik untuk mengidentifikasi berbagai indikator ancaman, seperti *Hidden PHP*, *Code Pattern*, *Base64 Payload*, *High Entropy*, file PHP di luar direktori *public_html*, konfigurasi *permission*, struktur direktori yang mencurigakan, serta indikasi konten judi online berdasarkan kata kunci.
2. Berdasarkan hasil pengujian, JemagoScan berhasil menyelesaikan proses audit tanpa mengalami kegagalan (Errors = 0) serta menghasilkan informasi yang terdokumentasi dalam file log sebagai dasar evaluasi kondisi keamanan website. Hasil tersebut menunjukkan bahwa sistem mampu mendukung proses audit keamanan awal (*pre-security assessment*) sehingga administrator dapat mengidentifikasi potensi ancaman sebelum mekanisme File Integrity Monitoring pada JemagoWEB diimplementasikan.
3. Integrasi JemagoScan dengan JemagoWEB membentuk mekanisme keamanan yang saling melengkapi, yaitu JemagoScan berperan melakukan pemeriksaan kondisi awal hosting, sedangkan JemagoWEB berfungsi memantau perubahan integritas file setelah *baseline* terbentuk. Dengan demikian, proses pembentukan *baseline* dapat dilakukan pada kondisi hosting yang telah melalui proses audit keamanan sehingga efektivitas perlindungan integritas file website dapat ditingkatkan.

## 5. SARAN

Berdasarkan hasil evaluasi, mekanisme deteksi **Gambling Keyword** masih berpotensi menghasilkan *false positive*, terutama pada kata kunci yang bersifat umum seperti **"agen"**, sehingga setiap temuan masih memerlukan proses verifikasi manual untuk memastikan adanya penyisipan konten judi online maupun *malware*. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan metode deteksi yang lebih kontekstual, misalnya dengan mengombinasikan analisis kata kunci, struktur kode, serta mekanisme *risk scoring* sehingga tingkat akurasi deteksi dapat ditingkatkan dan potensi *false positive* dapat diminimalkan. Selain itu, pengembangan selanjutnya juga dapat mempertimbangkan penambahan mekanisme klasifikasi tingkat risiko pada setiap temuan agar proses analisis dan penanganan ancaman menjadi lebih efektif.

DAFTAR PUSTAKA

[1] J. Tahsinia, A. A. Zulfa, T. Ibrahim, and O. Arifudin, "PERAN SISTEM INFORMASI AKADEMIK BERBASIS WEB," vol. 6, no. 1, pp. 115–134, 2025.

[2] J. Sirait, M. Y. Simanjuntak, and E. Sinaga, "Perancangan Website Penerimaan Mahasiswa Baru ( PMB ) dengan Menggunakan Php dan MySql untuk Memudahkan Pendaftar Melalui Online dalam Kepemilikan Kartu Peserta Ujian Universitas Audi Indonesia , Medan sistem yang dapat memproses data-data secara efektif ( O ' Neill , 2003 ).," no. 4, 2024.

[3] S. N. Oktaviana, V. Apriliani, and W. N. Novita, "Implementasi Sistem Informasi Akademik Dalam Meningkatkan Mutu Pelayanan Kampus," vol. 7, no. 1, pp. 53–63, 2024.

[4] R. Maulana and A. D. Wiranata, "ANALISIS SERANGAN SIBER DI WEBSITE UHAMKA DENGAN," vol. 9, pp. 113–123, 2026.

[5] J. Zhao, Y. Lu, X. Wang, and K. Zhu, "applied sciences WTA : A Static Taint Analysis Framework for PHP Webshell," pp. 1–21, 2021.

[6] A. Hannousse and S. Yahiouche, "Handling webshell attacks : A systematic," *Computers & Security*, vol. 108, p. 102366, 2021, doi: 10.1016/j.cose.2021.102366.

[7] H. Tahalli, R. Albar, M. D. Payana, and M. B. Wibawa, "PENGUJIAN KEAMANAN WEBSITE TERHADAP SERANGAN DEFACE DAN REDIRECT INJECTION MELALUI SIMULASI DENGAN OWASP ZAP ( STUDI KASUS : WEBSITE UNIVERSITAS UBUDIYAH INDONESIA ) WEBSITE SECURITY TESTING AGAINST DEFACE AND REDIRECT INJECTION ATTACKS THROUGH SIMULATION USING OWASP ZAP ( CASE STUDY : UNIVERSITAS UBUDIYAH INDONESIA WEBSITE )," vol. 11, no. 2, pp. 104–111, 2025.

[8] A. Bimandaru and A. Nugroho, "ANALISIS PENGUJIAN PENETRASI PADA LAYANAN HOSTING," vol. 13, no. 1, 2023.

[9] C. Della Permatasari, M. H. Putri, M. A. Shevchenko, and U. T. Yogyakarta, "ANALISIS KEAMANAN WEB SERVICE TERHADAP ANCAMAN," vol. 2, pp. 144–147, 2024.

[10] C. Kurniawan, A. Triayudi, I. Technology, and U. Nasional, "File Integrity Monitoring as a Method for Detecting and Preventing Web Defacement Attacks," vol. 9, no. 2, pp. 276–285, 2024, doi: 10.15575/join.v9i2.1326.

[11] F. H. Purwanto and F. Alfarisi, "Implementasi JemagoWeb untuk Perlindungan File Web dari Serangan Siber Judi Online Implementation of JemagoWeb for Web File Protection Against Online Gambling Cyber Attacks," vol. 16, no. 1, pp. 105–115, 2026.

[12] B. Supriyanto, Sumijan, and Yuhandri, "Jurnal Computer Science and Information Technology ( CoSciTech ) Audit Keamanan Website Menggunakan Acunetix Web Vulnerability ( Case Study At Muhammadiyah 3 Terpadu Vocational School Pekanbaru )," vol. 5, no. 1, pp. 134–143, 2024.

[13] Z. Yin and S. U. Lee, "Security Analysis of Web Open-Source Projects Based on Java," 2023.

[14] P. M. R. Gumilang and D. W. Chandra, “serangan berbasis website,” vol. 18, no. 1, pp. 54–68, 2021.
[15] F. Alaydrus, “Approaches in Determining Software Development Methods for Organizations : A Systematic Literature Review,” pp. 1–6, 2021.
[16] S. Gustiani, “RESEARCH AND DEVELOPMENT ( R & D ) METHOD AS A MODEL DESIGN IN EDUCATIONAL RESEARCH AND ITS ALTERNATIVES,” vol. 11, no. 2, pp. 12–22, 2019.
[17] K. I. Listyoningrum, D. Y. Fenida, and N. Hamidi, “Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan Sustainable Innovation in Business: Leverage Flowcharts to Optimize the Value of Corporate Waste,” vol. 1, no. 4, pp. 100–112, 2023.