**Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer** e-ISSN: 2548-964X
Vol. 9, No. 3, Maret 2025, hlm. x-x http://j-ptiik.ub.ac.id

# SISTEM KLASIFIKASI SERANGAN PADA WEBSITE BERBASIS WORDPRESS MENGGUNAKAN MACHINE LEARNING

**Adhiyaksa Ramadhana Purwidyantoro[1], Eko Sakti Pramukantoro[2]**

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]adhiyaksarp@student.ub.ac.id, [2]ekosakti@ub.ac.id

**Abstrak**

Penggunaan *website* yang semakin meluas, termasuk pada *platform* berbasis WordPress, telah membawa tantangan baru dalam menghadapi ancaman keamanan siber, seperti serangan *web defacement* dan penyisipan konten ilegal. WordPress, sebagai *Content Management System* (CMS) paling populer dengan pangsa pasar 62,8%, kerap menjadi target serangan yang memanfaatkan kerentanan plugin, kesalahan konfigurasi, dan celah keamanan lainnya. Serangan ini sering kali berujung pada penyisipan konten ilegal seperti promosi *judi online* melalui teknik *blackhat* SEO, yang merugikan berbagai institusi dan organisasi. Penelitian ini bertujuan mengembangkan metode berbasis *machine learning* untuk mendeteksi ancaman keamanan pada *website* berbasis WordPress melalui analisis log *web server*. Log *web server* merepresentasikan aktivitas pengguna dan menyimpan informasi penting untuk mendeteksi anomali, namun analisis manual banyak menghasilkan *false positive* dan kurang efisien. Dengan memanfaatkan *machine learning*, penelitian ini menawarkan pendekatan yang lebih akurat dan efisien dalam mendeteksi anomali dan mengidentifikasi jenis serangan yang dapat mengarah ke penyusupan konten judi *online*, seperti *Remote Code Execution*, SQL *Injection*, *File Inclusion*, *Brute Force Attacks*, *Unrestricted File Upload* dan *Cross-Site Scripting*. Hasil penelitian memperlihatkan model klasifikasi dengan Decision Tree mampu melakukan klasifikasi serangan yang ditentukan dengan memberikan hasil akurasi 87%, serta memiliki kinerja waktu klasifikasi di bawah 1 detik untuk log dengan 1000 transaksi.

**Kata kunci**: *wordpress, web, judi, ilegal, log, machine learning, klasifikasi, serangan, decision tree*

**Abstract**

*The increasing use of websites, including platforms based on WordPress, has introduced new challenges in addressing cybersecurity threats, such as web defacement and illegal content injection. WordPress, as the most popular Content Management System (CMS) with a market share of 62.8%, frequently becomes a target for attacks exploiting plugin vulnerabilities, misconfigurations, and other security flaws. These attacks often lead to the injection of illegal content, such as online gambling promotions, through blackhat SEO techniques, causing harm to various institutions and organizations. This study aims to develop a machine learning-based method to detect security threats on WordPress-based websites by analyzing web server logs. Web server logs represent user activities and contain critical information for anomaly detection. However, manual analysis often results in high false positive rates and inefficiencies. By leveraging machine learning, this study offers a more accurate and efficient approach to detecting anomalies and identifying attack types that could lead to illegal content injection, such as Remote Code Execution, SQL Injection, File Inclusion, Brute Force Attacks, Unrestricted File Upload, and Cross-Site Scripting. The findings demonstrate that the Decision Tree classification model performs effectively in categorizing predefined attack types, achieving an accuracy of 87%. Additionally, the model exhibits a classification time of less than 1 second for logs containing 1,000 transactions.*

**Keywords**: *wordpress, gambling, illegal, log, machine learning, classification, attacks, decision tree*

## 1. PENDAHULUAN

Kebutuhan akan teknologi dalam memenuhi layanan dan informasi secara cepat terus meningkat, salah satunya melalui penggunaan *website*. Hal ini mendorong pengembangan *website* yang lebih mudah dan cepat untuk memenuhi kebutuhan konsumen.

WordPress, sebagai salah satu *Content Management System* (CMS) terpopuler, menawarkan solusi pengembangan dan manajemen *website* yang efisien. Berdasarkan laporan W3Techs tahun 2022, WordPress menguasai 62,8% pangsa pasar CMS global, dengan 43,2% dari total *website* di internet menggunakan platform ini. Fleksibilitas dan adopsi luas WordPress membuatnya digunakan oleh berbagai sektor industri di seluruh dunia.

Namun, pesatnya adopsi teknologi berbasis WordPress juga disertai dengan meningkatnya ancaman keamanan, terutama terkait tren negatif seperti judi *online*. Laporan Pusat Pelaporan dan Analisis Transaksi Keuangan (PPATK) pada 2024 mengungkap perputaran uang senilai 283 triliun rupiah pada semester kedua, menunjukkan peningkatan signifikan aktivitas terkait judi *online*. Ancaman ini sering dikaitkan dengan praktik *blackhat search engine optimization* (SEO), yang menargetkan situs pemerintahan, organisasi, dan institusi melalui serangan seperti *defacing* dan penyusupan konten ilegal. Kerentanan dalam konfigurasi atau *plugin* WordPress sering dimanfaatkan untuk menyisipkan konten ilegal yang terindeks oleh mesin pencari.

Laporan tahunan Badan Siber dan Sandi Negara (BSSN) pada 2022 mencatat 2.048 kasus web defacement, sebagian besar menyerang situs berbasis WordPress. *Plugin* WordPress yang bersifat open-source atau gratis menjadi salah satu faktor kerentanan, terutama akibat kesalahan konfigurasi, kurangnya pemahaman terhadap kode sumber, atau penggunaan versi *plugin* yang usang. Menurut penelitian Ramadhani (2024), pemindaian keamanan menggunakan WPScan mengidentifikasi banyak kerentanan pada situs WordPress yang dapat dieksploitasi untuk meningkatkan risiko serangan. Dalam konteks ini, analisis log *web server* menjadi salah satu pendekatan yang efektif untuk mendeteksi serangan sejak dini.

Log *web server* merepresentasikan data interaksi antara pengguna dan *website*, mencakup permintaan akses ke berbagai sumber daya. Menurut Nedelkoski et al. (2020), log membantu mendeteksi, melokalisasi, dan menyelesaikan masalah pada sistem IT. Namun, volume data log yang besar sering kali menyulitkan administrator untuk menganalisis secara manual. Ryciak et al. (2022) menyoroti pentingnya pengawasan log secara menyeluruh untuk mendeteksi anomali yang dapat mengancam ketersediaan layanan, integritas, dan kerahasiaan data. Pendekatan manual yang bergantung pada pencarian kata kunci seperti error sering menghasilkan hasil false positive dan dinilai kurang efektif.

Berdasarkan hal tersebut, penelitian ini berfokus pada pengembangan metode berbasis *machine learning* untuk mendeteksi dan menganalisis ancaman keamanan pada *website* berbasis WordPress melalui analisis log *web server*. Pendekatan ini bertujuan meningkatkan akurasi dan efisiensi dalam mendeteksi anomali atau serangan berbasis *machine learning*.

## 2. PENELITIAN TERKAIT

Penelitian oleh Ramezany et al. (2023) mengembangkan *framework* berbasis machine learning untuk mendeteksi dan mengklasifikasikan payload mencurigakan pada web. *Framework* ini dirancang untuk menangkap dan mengklasifikasikan berbagai serangan web dengan menggunakan *dataset* baru yang memiliki entitas dan variasi modern. Kategori serangan yang diidentifikasi meliputi *Remote Code Execution* (RCE), *Injection* (LDAP, SQL, dan NoSQL), *Local File Inclusion*, *Cross-Site Scripting*, *XML External Entity* (XXE), *Open Redirect*, *Carriage Return Line Feed* (CRLF) *Injection*, dan *Deserialization Attack*. Penelitian ini juga melakukan studi komparasi algoritma, seperti Support Vector Machine, Random Forest, dan Stochastic Gradient Descent, dalam penerapan pada framework tersebut. Berdasarkan hasil analisis, Random Forest menunjukkan performa terbaik dalam presisi, recall, dan F1-score pada sebagian besar jenis serangan, meskipun belum mampu mengungguli algoritma lain dalam mendeteksi serangan XXE.

Penelitian oleh Riera et al. (2022) menggunakan *dataset* SR-BH 2020 untuk mengidentifikasi normalitas permintaan web dan mengklasifikasikannya ke dalam kategori CAPEC. Penelitian ini mengusulkan metode konversi string alfanumerik dan simbol menjadi nilai numerik melalui perhitungan rata-rata nilai ASCII setiap karakter, yang memungkinkan ekstraksi fitur dan pelatihan model *machine learning* secara cepat dan efisien. Beberapa model klasifikasi multi-label dianalisis menggunakan pustaka scikit-learn dan scikit-multilearn, dengan algoritma LightGBM dan CatBoost sebagai model yang diuji. Kombinasi algoritma CatBoost dan model Two-phase MultiOutputClassifier dari scikit-learn terbukti memiliki performa terbaik dalam klasifikasi multi-label berdasarkan hasil eksperimen.

Penelitian oleh Li et al. (2019) memperkenalkan model deteksi anomali berbasis rekonstruksi error untuk mengidentifikasi permintaan berbahaya pada aplikasi web. Model ini menggunakan jaringan multi-head attention dan gated convolution network untuk menangkap pola permintaan normal, didukung metode segmentasi baru yang meningkatkan representasi struktural permintaan dengan menyisipkan permintaan dasar ke dalam matriks fitur. Hasil eksperimen menunjukkan model ini unggul

dalam membedakan permintaan normal dan abnormal, serta menunjukkan performa lebih baik dibandingkan metode Regularized Deep Autoencoder (RDA) dalam mendeteksi berbagai jenis serangan, termasuk Remote Code Execution, Remote Command Execution, SQL Injection, Local File Include, dan Cross-Site Scripting. Penggunaan jaringan multi-head attention dan gated convolution network, ditambah metode embedding baru, menghasilkan representasi data yang lebih efektif, sehingga meningkatkan akurasi deteksi anomali secara signifikan.

## 3. METODE PENELITIAN

Penelitian ini , dilakukan untuk melakukan klasifikasi serangan atau anomali yang terjadi pada *website* WordPress melalui data yang berasal dari log *web server* berbasis *machine learning*.

### 3.1. Pembuatan Lingkungan Penelitian

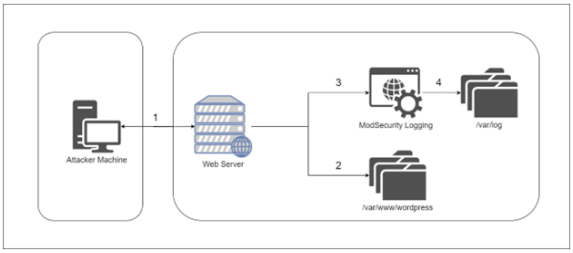
Penelitian ini dilakukan dengan membangun sebuah lingkungan uji yang dirancang untuk mensimulasikan serangan dalam jaringan lokal. Lingkungan ini terdiri dari dua komponen utama, yaitu lingkungan target dan lingkungan penyerang, yang terhubung dalam satu subnet IP.. Lingkungan uji ini merepresentasikan interaksi antara beberapa komponen utama, seperti *attacker machine*, *web server*, dan *system logging* yang ditunjukkan pada gambar 1 merupakan alur dari lingkungan penelitian, yang terdiri dari beberapa komponen, serta hubungan yang direpresentasikan dalam bentuk penomoran. Berikut ini penjelasan terkait hubungan antar komponen :

1. *Attacker Machine* melakukan interaksi dengan *website* WordPress, hubungan ini berupa aktivitas *requests* dari *attacker machine* kepada *web server*, dan nantinya *web server a*kan memberikan *response* sesuai dengan *request* yang diminta dari *attacker machine*

2. *Web Server* melakukan akses konten *website* yang tersimpan di bawah direktori "/var/www/wordpress", maupun dapat menyimpan data ke dalam direktori terseebut, sesuai dengan kebutuhan dan permintaan dari klien

3. Aktivitas *requests* dan *responses* dilakukan perekaman menggunakan ModSecurity *logging*.

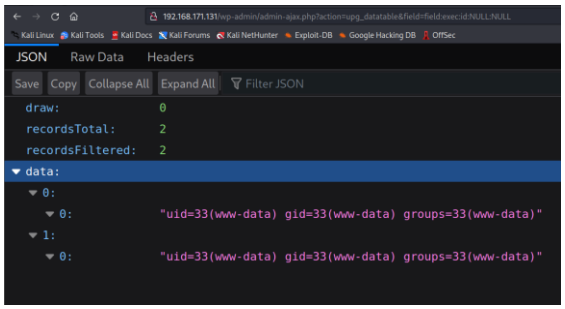
4. Hasil perekaman log akan disimpan secara langsung di bawah direktori "/var/log".




Gambar 1. Alur Lingkungan Penelitian

### 3.2. Pengambilan Data

Pada tahapan ini, dilakukan implementasi dari lingkungan penelitian yang berkaitan dengan serangan yang dilakukan oleh *attacker machine* terhadap target *machine* atau *web server* yang menjalankan layanan WordPress. Proses ini bertujuan untuk merepresentasikan interaksi antara penyerang dan target, sekaligus mendemonstrasikan cara pengambilan data dari serangan tersebut. Data yang diambil mencakup *requests* yang dikirimkan oleh *attacker machine* serta responses yang diterima dari *web server*. Sebagai contoh, salah satu serangan yang disimulasikan adalah *Unauthenticated Remote Code Execution* (CVE-2022-4060) yang ditinjau secara langsung pada gambar 2. Hasil log yang direkam kemudian ditampilkan pada Tabel 1 sebagai bentuk representasi data yang dihasilkan.




Gambar 2. Serangan *Unauthenticated Remote Code Execution* (CVE-2022-4060)

Tabel 1. ModSecurity Log

| |
|---|
| --dcc78318-A--<br>[11/Dec/2024:21:36:38 +0700]<br>Z1mjdcVRHCCQLuWJ39JhkwAAAAM 192.168.171.129<br>48892 192.168.171.131 80<br>--dcc78318-B--<br>GET /wp-admin/admin-<br>ajax.php?action=upg_datatable&field=field:exec:id:NULL:NU<br>LL HTTP/1.1<br>Host: 192.168.171.131<br>User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)<br>Gecko/20100101 Firefox/102.0<br>Accept:<br>text/html,application/xhtml+xml,application/xml;q=0.9,image/<br>avif,image/webp,*/*;q=0.8<br>Accept-Language: en-US,en;q=0.5<br>Accept-Encoding: gzip, deflate<br>Connection: keep-alive<br>Cookie:<br>_wsm_id_1_cc33=21f6c660b9acbb34.1721050086.3.1721065<br>129.1721053921; wp-settings-time-1=1731346250; wp-ps-<br>session=l78jrq2lf8ac7s2tqkqdpp8mon<br>Upgrade-Insecure-Requests: 1 |

--dcc78318-F--
HTTP/1.1 200 OK
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Referrer-Policy: strict-origin-when-cross-origin
X-Frame-Options: SAMEORIGIN
Content-Length: 172
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json

--dcc78318-E--
{"draw":0,"recordsTotal":2,"recordsFiltered":2,"data":[["uid=33(www-data) gid=33(www-data) groups=33(www-data)"],["uid=33(www-data) gid=33(www-data) groups=33(www-data)"]]}
--dcc78318-H--
Message: Warning. Pattern match "^[\\d.:]+$" at REQUEST_HEADERS:Host. [file "/usr/share/modsecurity-crs/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "696"] [id "920350"] [msg "Host header is a numeric IP address"] [data "192.168.171.131"] [severity "WARNING"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-protocol"] [tag "OWASP_CRS"] [tag "OWASP_CRS/PROTOCOL_VIOLATION/IP_HOST"] [tag "WASCTC/WASC-21"] [tag "OWASP_TOP_10/A7"] [tag "PCI/6.5.10"]
Apache-Error: [file "apache2_util.c"] [line 271] [level 3] [client 192.168.171.129] ModSecurity: Warning. Pattern match "^[\\\\\\\\\d.:]+$" at REQUEST_HEADERS:Host. [file "/usr/share/modsecurity-crs/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "696"] [id "920350"] [msg "Host header is a numeric IP address"] [data "192.168.171.131"] [severity "WARNING"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-protocol"] [tag "OWASP_CRS"] [tag "OWASP_CRS/PROTOCOL_VIOLATION/IP_HOST"] [tag "WASCTC/WASC-21"] [tag "OWASP_TOP_10/A7"] [tag "PCI/6.5.10"] [hostname "192.168.171.131"] [uri "/wp-admin/admin-ajax.php"] [unique_id "Z1mjdcVRHCCQLuWJ39JhkwAAAAM"]
Apache-Error: [file "/build/php7.4-sXiaX2/php7.4-7.4.3/sapi/apache2handler/sapi_apache2.c"] [line 349] [level 5] "fbc_"
Apache-Handler: application/x-httpd-php
Stopwatch: 1733927797822425 743462 (- - -)
Stopwatch2: 1733927797822425 743462; combined=7753, p1=749, p2=5991, p3=126, p4=675, p5=212, sr=117, sw=0, l=0, gc=0
Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.9.3 (http://www.modsecurity.org/); OWASP_CRS/3.2.0.
Server: Apache/2.4.41 (Ubuntu)
Engine-Mode: "DETECTION_ONLY"

--dcc78318-Z--


Data *log* ini dijadikan sebagai bahan utama dari pembuatan *dataset* dalam membangun model klasifikasi.

### 3.3. *Preprocessing* Data

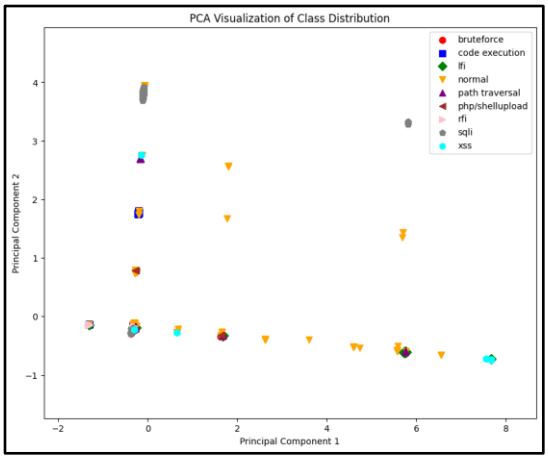
Tahap ini bertujuan untuk mengolah file log ModSecurity menjadi *dataset* yang siap digunakan dalam membangun model *machine learning*. File log awal diperiksa validitasnya, kemudian diproses melalui tahapan parsing untuk menghasilkan data dalam format JSON. Selanjutnya, dilakukan ekstraksi parameter dan nilai dari setiap transaksi, yang kemudian dimasukkan ke dalam Excel *workbook*. Hasilnya menjadi *dataset* dengan 26 fitur, termasuk satu label kelas.

*Feature Selection* dilakukan untuk memilah fitur yang relevan atau menjadi representasi dari serangan yang terjadi. Hasil dari pemilihan fitur disajikan dalam bentuk tabel yang telah dikelompokkan menjadi beberapa kategori berdasarkan teknik transformasi data yang digunakan, yaitu *Term Frequency - Inverse Document Frequency* (TF-IDF), dan *Label Encoding*,

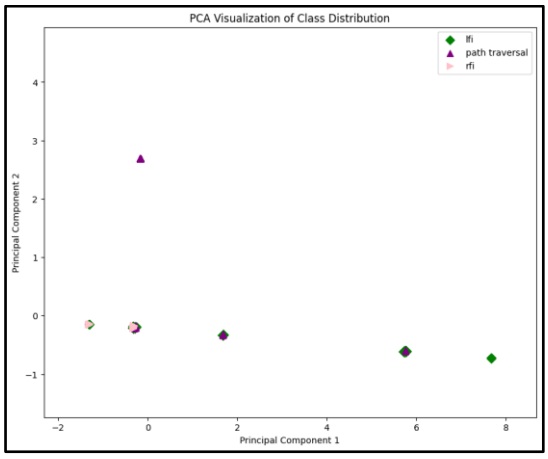
Tabel 2. Kategori Fitur Berdasarkan *Transformation* Data

| ***TF-IDF*** | ***Label Encoding*** |
|---|---|
| request_line_url | request_user_agent |
| request_body | request_line_method |
| response_body | response_status |

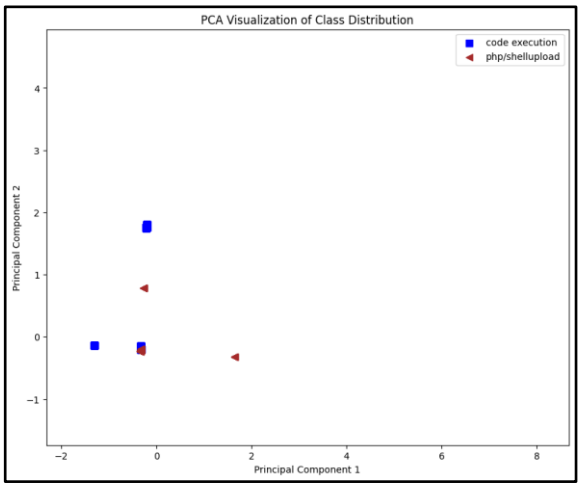
Data visualization dilakukan menggunakan Principal Component Analysis (PCA) untuk mereduksi dimensi data menjadi dua dimensi, mempermudah visualisasi distribusi kelas atau label. Diperlihatkan pada gambar 3 menampilkan persebaran seluruh label. Selain itu, visualisasi spesifik disajikan pada gambar lain, seperti distribusi kelas untuk *file inclusion attacks* ('lfi', 'rfi', 'path traversal') ditunjukkan pada gambar 4 dan *remote code execution* (RCE) *attacks* ('code execution', 'php/shell upload').ditunjukkan di gambar 5.




Gambar 3. Distribusi Keseluruhan Kelas




Gambar 4. Distribusi Antar Kelas *File Inclusion attacks*




Gambar 5. Distribusi Antar Kelas *Remote Code Execution*

Meninjau kembali dari hasil visualisasi data persebaran kelas, ditemukan adanya kelas yang saling *overlapping* atau saling bertindihan pada setiap kelas serangan *file inclusion attacks* ('lfi', 'rfi', 'path traversal'), serta ditemukan juga fenomena tersebut pada kelas serangan *remote code execution* (RCE) *attacks* ('code execution', 'php/shell upload').

### 3.4. *Feature Engineering*

Proses pelabelan *dataset* dilakukan berdasarkan analisis informasi dari setiap transaksi, seperti payload, URL path, dan pola perilaku serangan. Pelabelan ini menghasilkan enam kelas utama, yang terdiri dari lima kelas serangan dan satu kelas normal. Kelas serangan meliputi Bruteforce, Cross-Site Scripting (XSS), File Inclusion - Local Disclosure File (yang mencakup Remote File Inclusion, Local File Inclusion, dan Path Traversal), Remote Code Execution (mencakup Code Injection/Execution dan Unrestricted File Upload), serta SQL Injection. Kelas normal digunakan untuk transaksi yang tidak mengindikasikan serangan.

Langkah sistematis dilakukan dalam pelabelan *dataset* melalui beberapa langkah utama. Pertama, *Attack flow scenario* digunakan untuk meninjau alur eksekusi skenario serangan yang telah dirancang, sehingga setiap transaksi dapat diidentifikasi dengan tepat sesuai jenis serangan yang terjadi. Kedua, teknik *Regex Matching* diterapkan dengan memanfaatkan formula pada Excel untuk mendeteksi pola tertentu dalam payload atau URL yang menjadi indikator serangan spesifik. Terakhir, dilakukan *Manual Review* untuk memastikan akurasi pelabelan dengan memeriksa ulang transaksi secara manual.

### 3.5. Pembuatan Model Klasifikasi

Pada tahap perancangan model klasifikasi, penelitian ini menggunakan tiga model utama, yaitu Random Forest, Extreme Gradient Boosting, dan Decision Tree. Ketiga model ini dirancang dengan menetapkan parameter-parameter yang akan dioptimalkan melalui proses hyperparameter tuning menggunakan metode Randomized Search dengan ruang pencarian yang telah ditentukan.

Tabel 3. Parameter Masukan *Randomized Search*

| Random Forest | XGBoost | Decision Tree |
|---|---|---|
| n_estimators<br>max_depth<br>min_sample_split<br>min_sample_leaf<br>max_features' | max_depth<br>min_child_weight<br>learning_rate<br>n_estimators<br>gamma<br>subsample<br>colsample_bytree | criterion<br>max_depth<br>min_sample_split<br>min_sample_leaf<br>dan max_features |

Model *Random Forest*, dan *Decision Tree* dilatih menggunakan *library* Scikit-Learn, dan model *Extreme* dilatih dengan *library* xgboost, dimana keduanya dijalankan dalam bahasa pemrograman Python. Proses pelatihan model dilakukan menggunakan 80% dari total data yang tersedia (data latih), dengan data sisanya sebesar 20% digunakan untuk evaluasi. Ketiga model dilatih menggunakan parameter optimal yang diperoleh dari proses hyperparameter tuning.

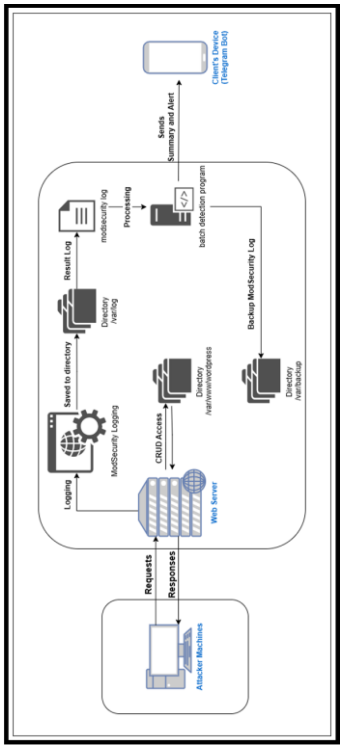
### 3.6. Skenario Pengujian

Pada tahap evaluasi model, tiga model klasifikasi yang telah dibangun, yaitu Random Forest, Extreme Gradient Boosting, dan Decision Tree, diuji menggunakan data uji yang diambil dari *dataset* dengan rasio 20% dari keseluruhan data. Evaluasi dilakukan dengan menggunakan metrik akurasi, presisi, recall, dan F1-score untuk menilai performa model dalam mengklasifikasikan serangan pada layanan *website* berbasis WordPress. Hasil dari evaluasi model yang telah dilakukan akan dipilih satu model dengan performa terbaik berdasarkan matrik evaluasi yang telah dilakukan.

Pada tahap perancangan inferensi, dirancang suatu sistem klasifikasi pada web service berbasis WordPress untuk mendeteksi anomali atau serangan berdasarkan jenis serangan yang telah ditetapkan.

Setiap model yang telah dibangun pada tahap sebelumnya, akan disimpan menggunakan *library* joblib dengan format joblib. Sehingga, nantinya pada program, model dapat dimuat dengan mudah. Program yang digunakan dalam inferensi ini dijalankan secara langsung pada *web server* dengan berbasis command line interface, dan berjalan secara batch detection, dimana program akan berjalan secara berkala dengan waktu tertentu. Berdasarkan alur sistem inferensi model pada gambar 6, terdapat alur kerja yang dengan penjelasan di bawah ini :

1. Attacker melakukan interaksi dengan *website* WordPress, dan diproses langsung oleh *web server*. Dimana pada proses ini, dapat diartikan attacker sebagai users melakukan *requests* atau permintaan kepada *web server*, kemudian, *web server* memberikan respon, berdasarkan permintaan klien, dalam hal ini attacker.
2. *Web server* dapat melakukan eksekusi CRUD (*create*, *read*, *update*, dan *delete*) sesuai dengan permintaan atau *request* dari klien.
3. Transaksi (set *request* dan *response*) dilakukan proses perekaman dan pencatatan (logging) dengan ModSecurity.
4. Hasil pencatatan transaksi disimpan di dalam direktori "/var/log" merupakan ModSecurity log.
5. ModSecurity log dilakukan pemrosesan dengan program batch detection program yang telah disiapkan. Alur pemrosesan dengan program tersebut ditunjukkan pada gambar 5.x .
6. Program batch detection dijalankan setiap n waktu, mengirimkan summary report dari hasil klasifikasi, dan alert ketika menemukan adanya transaksi yang terklasifikasi sebagai serangan.
7. Program batch detection akan melakukan backup dan truncate pada ModSecurity log.




Gambar 6 Alur sistem inferensi model

## 4. HASIL DAN PEMBAHASAN

### 4.1. Hasil *Training* Model

Proses pelatihan dan pengujian model yang telah dilakukan memberikan hasil evaluasi yang dapat dilihat pada Tabel 6.. Berdasarkan hasil tersebut, model Decision Tree memiliki nilai presisi dan akurasi tertinggi dibandingkan dengan Random Forest dan Extreme Gradient Boosting, dengan nilai presisi sebesar 0,99793 dan akurasi sebesar 0,99920. Di sisi lain, model Extreme Gradient Boosting memiliki nilai recall tertinggi sebesar 0,99788. Sedangkan untuk nilai f1-score, Random Forest dan Extreme Gradient Boosting memiliki nilai yang sama, yaitu sebesar 0,99787. Secara keseluruhan, nilai presisi, recall, f1-score, dan akurasi pada ketiga model memiliki selisih yang tidak signifikan, dengan margin di atas ambang batas bawah 0,99000.

Tabel 4. Hasil Evaluasi Metrik Ketiga Model

| Model | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|
| Random Forest | 0,99110 | 0,99000 | 0,99043 | 0,99407 |
| Extreme Gradient Boosting | 0,99787 | 0,99788 | 0,99787 | 0,99917 |
| Decision Tree | 0,99793 | 0,99781 | 0,99787 | 0,99920 |

Tabel 5. Confusion Matrix Model Random Forest

| ***Web Attacks Classifier*** | **B** | **F** | **N** | **R** | **S** | **X** |
|---|---|---|---|---|---|---|
| **B** | 1462 | 0 | 4 | 0 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **F** | 0 | 1226 | 4 | 0 | 0 | 0 |
| **N** | 0 | 1 | 2540 | 5 | 0 | 1 |
| **R** | 0 | 0 | 18 | 806 | 0 | 0 |
| **S** | 0 | 0 | 53 | 0 | 13970 | 0 |
| **X** | 0 | 0 | 43 | 0 | 0 | 1632 |

Tabel 6. *Confusion Matrix* Model Extreme Gradient Boosting

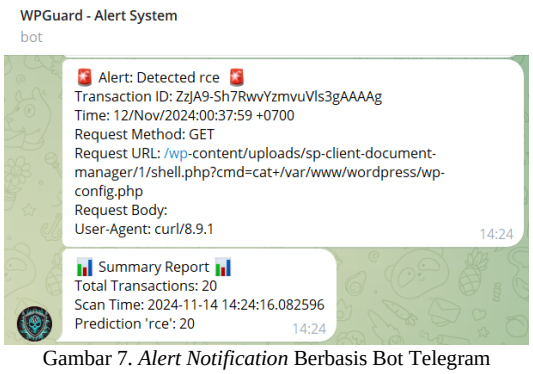
| ***Web Attacks Classifier*** | | | | | | |
|---|---|---|---|---|---|---|
| | **B** | **F** | **N** | **R** | **S** | **X** |
| **B** | 1462 | 0 | 4 | 0 | 0 | 0 |
| **F** | 0 | 1230 | 0 | 0 | 0 | 0 |
| **N** | 0 | 0 | 2540 | 7 | 0 | 0 |
| **R** | 0 | 0 | 5 | 819 | 0 | 0 |
| **S** | 0 | 0 | 0 | 0 | 14023 | 0 |
| **X** | 0 | 0 | 2 | 0 | 0 | 1673 |

Tabel 7. *Confusion Matrix* Model Decision Tree

| ***Web Attacks Classifier*** | | | | | | |
|---|---|---|---|---|---|---|
| | **B** | **F** | **N** | **R** | **S** | **X** |
| **B** | 1465 | 0 | 1 | 0 | 0 | 0 |
| **F** | 0 | 1230 | 0 | 0 | 0 | 0 |
| **N** | 0 | 0 | 2540 | 7 | 0 | 0 |
| **R** | 0 | 0 | 7 | 817 | 0 | 0 |
| **S** | 0 | 0 | 0 | 0 | 14023 | 0 |
| **X** | 0 | 0 | 2 | 0 | 0 | 1673 |

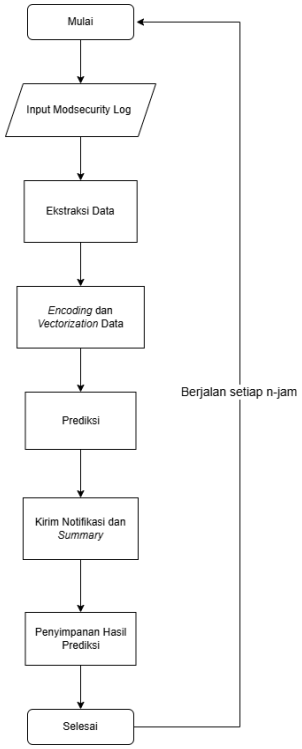
### 4.2. Implementasi Inferensi

Dalam penelitian ini, inferensi dilakukan implementasi dalam bentuk program Python yang dijalankan secara langsung pada *web server* setiap interval waktu yang ditentukan. Program juga ditambahkan implementasi *alert notification* melalui bot aplikasi Telegram dari klien, dimana akan mengirimkan notifikasi ketika hasil klasifikasi ditemukan label selain normal yang mengindikasikan adanya serangan. Implementasi *alert notification* ditunjukkan pada gambar 7.




Gambar 7. *Alert Notification* Berbasis Bot Telegram

Berdasarkan diagram alir yang ditampilkan pada gambar 8. Program inferensi ini akan melakukan analisis pada log yang telah berbentuk json dengan membaca keseluruhan data, kemudian dilakukan proses parsing, untuk mengambil data yang dibutuhkan sebagai masukan untuk model. Model akan menghasilkan prediksi berdasarkan enam kelas yang telah ditetapkan sebelumnya. Hasil prediksi akan ditampilkan secara langsung melalui notifikasi bot pada aplikasi Telegram, dan file dengan format json sebagai laporan detail yang tersimpan pada suatu folder di dalam *web server*.




Gambar 8. Diagram Alir Program Inferensi

### 4.3. Hasil Pengujian

Hasil pengujian performa model Decision Tree dari segi waktu klasifikasi disajikan pada tabel 8. Berdasarkan hasil tersebut, model Decision Tree menunjukkan kemampuan klasifikasi dengan waktu eksekusi kurang dari 1 detik untuk setiap jumlah transaksi yang diuji. Evaluasi performa inferensi dilakukan terhadap masing-masing kelas serangan, yaitu *Remote Code Execution* (RCE), *Cross-Site Scripting* (XSS), *Brute Force*, SQL *Injection* (SQLi), *File Inclusion-Local Directory Traversal* (FI-LDF), serta kombinasi dari seluruh kelas serangan.

Tabel 8. Pengujian Performa Waktu Klasifikasi

| **Jumlah Transaksi** | **Waktu (*seconds*)** |
|---|---|
| 10 | 0.005 |
| 100 | 0.049 |
| 1000 | 0.535 |

Hasil evaluasi inferensi ditampilkan pada tabel 9. Menunjukkan model Decision Tree mampu mencapai akurasi sempurna dalam klasifikasi serangan dengan label RCE, XSS, dan Brute Force.

Namun, pada label serangan SQLi, model mencatat performa akurasi terendah dibandingkan kelas serangan lainnya, yaitu sebesar 50%. Pada evaluasi kombinasi serangan, yang mencakup seluruh skenario serangan, model menunjukkan kemampuan klasifikasi yang baik dengan akurasi keseluruhan sebesar 87%.

| Serangan | TP | FN | Akurasi |
|---|---|---|---|
| **rce** | 20 | 0 | 100% |
| **xss** | 20 | 0 | 100% |
| **bruteforce** | 20 | 0 | 100% |
| **sqli** | 10 | 10 | 50% |
| **fi-ldf** | 17 | 3 | 85% |
| **combination** | 87 | 13 | 87% |

Tabel 9. Hasil Inferensi Model Decision Tree

**8.**

## 5. KESIMPULAN

Berdasarkan hasil penelitian, ModSecurity terbukti mampu menghasilkan data log yang relevan berdasarkan serangan yang terjadi pada *website* WordPress. Data log tersebut memenuhi kebutuhan dalam pembuatan *dataset* untuk membangun model *machine learning*. Selanjutnya, data log yang dihasilkan dapat diolah menjadi *dataset* dalam format XLSX dengan melakukan transformasi dari string menjadi representasi numerik menggunakan teknik Term Frequency-Inverse Document Frequency (TF-IDF) dan Label Encoding. Proses pelabelan *dataset* juga didukung oleh Regex Matching untuk meningkatkan akurasi klasifikasi.

Model klasifikasi berbasis Decision Tree yang dikembangkan menunjukkan performa yang sangat baik dalam mengklasifikasikan serangan, dengan nilai presisi, recall, F1-Score, dan akurasi masing-masing sebesar 0.99793, 0.99781, 0.99787, dan 0.99920. Pada tahap inferensi, model ini memberikan waktu eksekusi yang efisien, dengan rata-rata waktu sekitar 0.5 ms per transaksi untuk data berjumlah 10, 100, dan 1000 transaksi. Model juga berhasil mencapai akurasi 100% pada label serangan "rce" (remote code execution/injection dan unrestricted file upload), "xss" (*cross-site scripting*), dan "bruteforce" (*brute force attacks*) pada data sampel sebanyak 20 transaksi untuk setiap label serangan. Namun, pada label serangan "sqli" (SQL Injection), model menghasilkan akurasi yang kurang baik, yaitu sebesar 50%, yang mengindikasikan perlunya pengembangan lebih lanjut untuk meningkatkan performa pada label ini.

Secara keseluruhan, penelitian ini menunjukkan bahwa pendekatan berbasis *machine learning* dengan model Decision Tree dapat diandalkan untuk mengklasifikasikan serangan terhadap layanan web WordPress dengan hasil yang sangat baik, meskipun masih terdapat tantangan pada beberapa label serangan tertentu yang memerlukan perbaikan lebih lanjut.

## 6. DAFTAR PUSTAKA

RAMEZANY, S., SETTHAWONG, R. AND TANPRASERT, T., 2022. A machine learning-based malicious payload detection and classification framework for new web attacks. In: 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), May 2022, pp. 1–4. IEEE.

LI, J., FU, Y., XU, J., REN, C., XIANG, X. AND GUO, J., 2020. Web application attack detection based on attention and gated convolution networks. IEEE Access, 8, pp. 20717–20724. DOI: 10.1109/ACCESS.2019.2955674.

RYCIAK, P., WASIELEWSKA, K. AND JANICKI, A., 2022. Anomaly detection in log files using selected natural language processing methods. Applied Sciences, 12(10), p. 5089. DOI: https://doi.org/10.3390/app12105089.

NEDELKOSKI, S., BOGATINOVSKI, J., ACKER, A., CARDOSO, J. AND KAO, O., 2020. Self-attentive classification-based anomaly detection in unstructured logs. In: 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 2020, pp. 1196–1201. DOI: 10.1109/ICDM50108.2020.00148.

RIERA, T. S., HIGUERA, J. R. B., HIGUERA, J. B., HERRAIZ, J. J. M. AND MONTALVO, J. A. S., 2022. A new multi-label dataset for web attacks CAPEC classification using machine learning techniques. Computers & Security, 120, p. 102788.