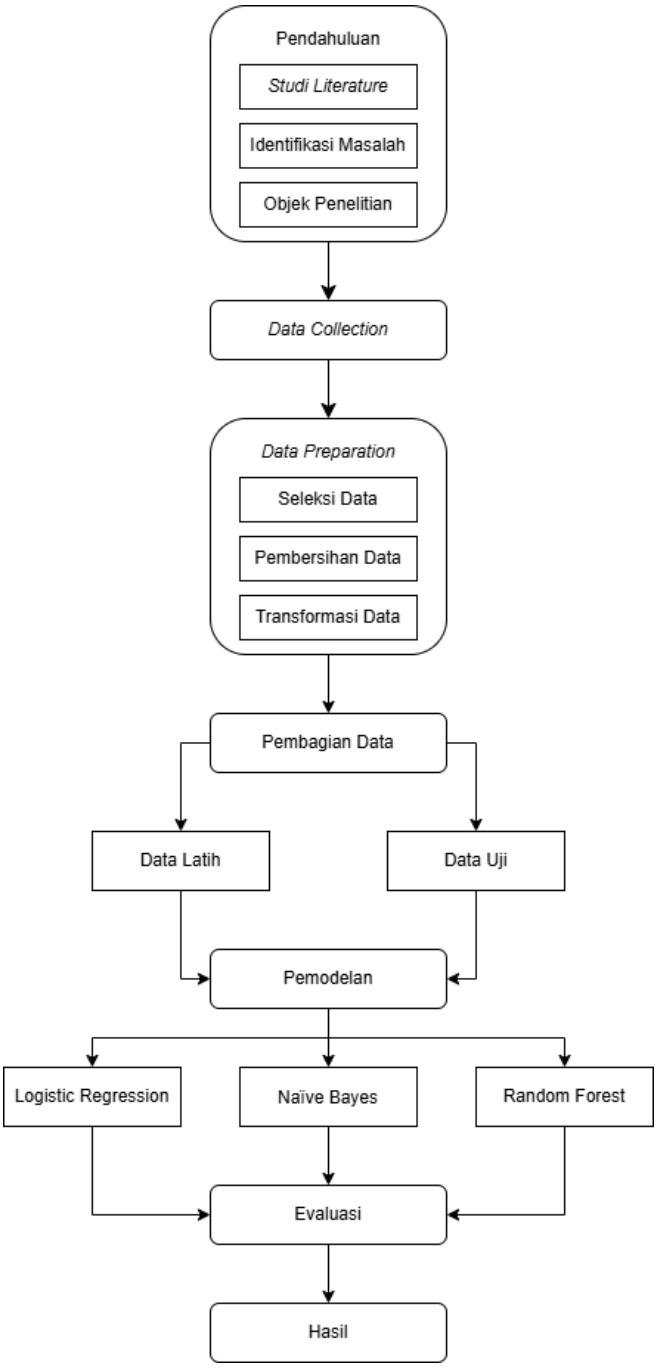
BAB III METODOLOGI

3.1. Tahapan Penelitian

Flowchart tahapan pada penelitian ini dilakukan nampak pada Gambar 3.1.



Gambar 3.1 Alur Tahapan Penelitian

3.2. Pendahuluan

3.2.1. Studi Literatur

Tahap ini berfokus pada penelusuran referensi serta pemahaman konsep yang relevan dengan permasalahan penelitian melalui jurnal ilmiah dan sumber daring. Literatur yang diperoleh digunakan sebagai landasan dan acuan dalam pelaksanaan penelitian ini, sebagaimana dirinci pada Sub BAB II.

3.2.2. Identifikasi Masalah

Implementasi *machine learning* masih relatif baru pada deteksi plugin rentan WordPress, hal tersebut juga menimbulkan ketergantungan terhadap model yang dipakai untuk proses prediksi. Penelitian komparatif algoritma *machine learning* menggunakan metadata yang mudah diperoleh menjadi relevan untuk menjawab ketergantungan tersebut, yaitu dengan menentukan model yang memiliki nilai prediksi kerentanan terbaik berdasarkan metode evaluasi yang dipakai.

3.2.3. Objek Penelitian

Objek penelitian yang dikaji adalah plugin rentan WordPress. Sebagian besar kerentanan keamanan WordPress dilaporkan berasal dari plugin pihak ketiga, yang dikembangkan dengan tingkat kualitas dan keamanan yang beragam serta memiliki akses langsung ke fungsi inti sistem. Penelitian-penelitian sebelumnya memberikan hasil bahwa deteksi keamanan yang biasa dipakai seperti plugin-plugin keamanan gratis tidak mampu untuk deteksi plugin rentan secara maksimal (Murphy dkk., 2021).

3.3. *Data Collection*

3.3.1. Sumber Data

Data yang digunakan dalam penelitian ini berasal dari WordPress Plugin API, yaitu antarmuka pemrograman aplikasi resmi yang disediakan oleh WordPress.org untuk mengakses informasi plugin yang tersedia pada repositori publik. API ini menyediakan berbagai informasi non-kode (metadata) plugin, seperti jumlah unduhan, *rating* pengguna, jumlah ulasan, umur plugin, frekuensi pembaruan per tahun, dll. Data diperoleh secara otomatis melalui permintaan (*request*) API menggunakan bahasa pemrograman Python.

Plugin rentan WordPress dipilih dari data awal bersumber pada GitHub https://github.com/ihuzaifashoukat/wordpress-vulnerability-database sebagai data untuk plugin rentan. Ditemukan sekumpulan data (baris) plugin yang aktif dari https://github.com/jcmpagel/wordpress-dataset, kemudian dipakai untuk proses perbandingan dengan data di database WordFence untuk menemukan plugin yang tidak rentan.

Penggunaan metadata plugin dipilih karena data tersebut mudah diperoleh, bersifat publik, dan tidak memerlukan analisis kode sumber, sehingga lebih efisien untuk diterapkan dalam sistem prediksi kerentanan berbasis *machine learning*.

3.3.2. Distribusi Data

Dataset yang digunakan terdiri dari sekumpulan baris plugin WordPress yang masing-masing direpresentasikan oleh atribut metadata. Setiap baris data merepresentasikan satu plugin, sedangkan kolom merepresentasikan atribut atau

fitur data yang digunakan dalam proses klasifikasi. Secara umum, fitur data yang dikumpulkan meliputi:

1. Nama Slug
2. Total *Download*
3. Total *Reviewer*
4. Tahun *Update* Terakhir
5. Penilaian Pengguna
6. Umur Plugin dari Rilis
7. Frekuensi *Update* per Tahun
8. *Open Issue*
9. Jumlah *File* PHP
10. Jumlah *File*
11. *Line of Code*
12. Plugin *Size* (kb)
13. *Versi WordPress Terakhir Uji*
14. Versi Minimum WordPress

Selain atribut metadata, dataset juga memiliki label kelas, yaitu status plugin rentan atau tidak rentan, yang digunakan sebagai variabel target dalam proses pembelajaran terawasi (*supervised learning*).

3.3.3. Pemahaman Data

Berdasarkan eksplorasi awal, data yang dipakai memiliki karakteristik sebagai berikut:

1. Pelabelan data yang dipakai untuk prediksi adalah kelas rentan dan kelas tidak rentan, di mana plugin dengan label rentan dipahami sebagai plugin yang memiliki sejarah kerentanan pada *website* Wordfence.
2. Sebagian besar atribut bersifat numerik, namun memiliki skala yang berbeda-beda.
3. Terdapat kemungkinan nilai kosong (*missing values*).

3.4. Data Preparation

3.4.1. Seleksi Data

Pada tahap ini dilakukan pemilihan atribut metadata yang relevan dengan tujuan penelitian. Atribut yang memiliki tingkat *missing values* yang sangat tinggi dikeluarkan dari dataset dan dilakukan penyetaraan jumlah data. Proses ini bertujuan untuk meningkatkan kualitas data dan efisiensi pembelajaran model (Malhotra & Jain, 2021).

3.4.2. Pembersihan Data

Proses pembersihan data dilakukan untuk menangani:

1. Nilai kosong (*missing values*), yang ditangani dengan teknik imputasi seperti pengisian nilai rata-rata atau median untuk atribut numerik.
2. Data duplikat, yaitu plugin dengan identitas yang sama, yang dihapus agar tidak menyebabkan bias.
3. Nilai ekstrem (*outlier*) yang berpotensi memengaruhi model, terutama pada atribut jumlah unduhan.

3.4.3. Transformasi Data

Agar data dapat digunakan secara optimal oleh algoritma Logistic Regression, Naïve Bayes, dan Random Forest, dilakukan beberapa transformasi sebagai berikut:

1. Normalisasi atau standarisasi atribut numerik untuk mengurangi perbedaan skala antar fitur.
2. Konversi atribut versi WordPress ke dalam bentuk kategorikal agar dapat diproses oleh algoritma *machine learning*.
3. *Encoding* label kelas, di mana status plugin dikodekan menjadi nilai biner (misalnya: 1 untuk rentan dan 0 untuk tidak rentan).

3.5. Pembagian Data

Dataset yang telah diproses kemudian dibagi menjadi dua bagian, yaitu:

1. Data latih (*training set*) sebanyak 80% untuk membangun model.
2. Data uji (*testing set*) sebanyak 20% untuk mengevaluasi performa model.

3.6. Pemodelan

Tahap Pemodelan menggunakan Orange merupakan proses pembangunan *workflow* dan model *machine learning* yang digunakan untuk memprediksi status kerentanan plugin WordPress berdasarkan metadata yang telah dipersiapkan pada tahap sebelumnya. Penelitian ini menerapkan pendekatan *supervised learning* dengan tiga algoritma klasifikasi, yaitu Logistic Regression, Naïve Bayes, dan Random Forest, sehingga memungkinkan untuk dilakukan analisis perbandingan performa secara objektif.

3.6.1. Proses Pelatihan Model

Ketiga algoritma dilatih menggunakan dataset yang sama agar hasil perbandingan performa bersifat adil. Proses pelatihan dilakukan pada data latih, sedangkan evaluasi dilakukan menggunakan data uji atau teknik *k-fold cross-validation*. Parameter dasar (*hyperparameter*) masing-masing algoritma ditetapkan sesuai dengan konfigurasi standar untuk menjaga objektivitas perbandingan.

3.6.2. Output Model

Output dari setiap model berupa prediksi kelas plugin WordPress, yaitu rentan atau tidak rentan. Selain label kelas, beberapa model juga menghasilkan nilai probabilitas yang digunakan untuk analisis performa lebih lanjut. Hasil prediksi dari ketiga algoritma kemudian dibandingkan berdasarkan metrik evaluasi yang telah ditentukan pada tahap evaluasi model.

3.7. Evaluasi

Tahap Evaluasi bertujuan untuk menilai kinerja model *machine learning* yang telah dibangun pada tahap Pemodelan dalam memprediksi status kerentanan plugin WordPress. Evaluasi dilakukan untuk memastikan bahwa model yang dihasilkan memiliki tingkat akurasi dan keandalan yang memadai, serta untuk membandingkan performa algoritma Logistic Regression, Naïve Bayes, dan Random Forest secara objektif.

3.7.1. Metode Evaluasi

Evaluasi model dilakukan menggunakan data uji (*testing set*) yang tidak terlibat dalam proses pelatihan, sehingga hasil pengujian mencerminkan

kemampuan generalisasi model terhadap data baru. Selain itu, digunakan teknik *k-fold cross-validation* untuk mengurangi bias akibat pembagian data dan memperoleh hasil evaluasi yang lebih stabil.

3.7.2. *Confusion Matrix*

Kinerja model dianalisis menggunakan *confusion matrix* yang menggambarkan jumlah prediksi benar dan salah untuk setiap kelas, yaitu plugin rentan dan tidak rentan. *Confusion matrix* terdiri dari empat komponen utama, yaitu sebagai berikut:

1. *True Positive (TP)*,
2. *True Negative (TN)*,
3. *False Positive (FP)*, dan
4. *False Negative (FN)*.

Matriks ini digunakan sebagai dasar perhitungan matriks evaluasi pada tahapan selanjutnya.

3.7.3. Matriks Evaluasi

Proses penilaian performa klasifikasi secara komprehensif menggunakan beberapa metrik evaluasi sebagai berikut:

1. Akurasi

   Mengukur proporsi prediksi keseluruhan terhadap seluruh data uji.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

2. Presisi

   Mengukur tingkat ketepatan model dalam memprediksi plugin sebagai rentan.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3. *Recall*

   Mengukur kemampuan model dalam mendeteksi plugin yang benar-benar rentan.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

4. *F1-Score*

   Merupakan nilai harmonisasi antara presisi dan *recall* yang digunakan untuk menyeimbangkan kedua metrik tersebut.

$$F1 - Score = 2\ x\ \frac{Precison\ x\ Recall}{Precision + Recall} \tag{3.4}$$

3.7.4. Perbandingan Performa Model

Hasil evaluasi dari masing-masing algoritma dibandingkan berdasarkan nilai metrik yang diperoleh. Algoritma dengan nilai *F1-score* dan *recall* tertinggi dianggap lebih efektif dalam mendeteksi plugin rentan, sementara akurasi digunakan sebagai indikator performa umum model. Analisis perbandingan ini digunakan untuk menentukan algoritma yang paling sesuai untuk prediksi kerentanan plugin WordPress menggunakan metadata.

3.8. Interpretasi Hasil

Tahap akhir evaluasi difokuskan pada interpretasi hasil pengujian untuk menjawab tujuan penelitian. Hasil evaluasi digunakan untuk mengidentifikasi performa masing-masing algoritma, serta memberikan rekomendasi model yang paling optimal untuk diterapkan dalam sistem deteksi kerentanan plugin WordPress secara otomatis.