BAB V. KESIMPULAN DAN SARAN

5.1. Simpulan

Penelitian ini mengajukan dan mengimplementasikan dua metode, yaitu metode deteksi dan metode mitigasi. Pengujian kedua metode dilakukan secara independen, sesuai dengan karakteristik masing-masing metode. Pengujian secara independen berarti bahwa pengujian dilakukan secara komprehensif dan mendalam, secara satu per-satu, baik pada *parent* maupun *child method*. Pengujian metode deteksi dan mitigasi dilakukan berdasarkan teknik serangan yang telah ditentukan, sesuai dengan yang telah diungkapkan pada pembatasan penelitian.

Setelah kedua metode diuji secara independen, keduanya lalu diujikan secara terintegrasi atau kolaboratif. Model metode deteksi yang telah dihasilkan pada tahapan *machine learning*, kemudian diintegrasikan pada metode mitigasi lapisan kedua (*HTTP Middleware*). Tindakan integrasi ini dilakukan untuk mengetahui dua hal yaitu (1) tingkat akurasi metode deteksi dan (2) tingkat efektivitas metode mitigasi. Selain itu, mengintegrasikan metode deteksi pada metode mitigasi juga dapat semakin menguji kinerja metode deteksi, terutama ketika metode deteksi dilakukan *under presure*, karena langsung diujikan dengan aplikasi penyerang. Terkait dengan penelitian yang telah dilakukan, berikut ini adalah simpulan dari penelitian ini.

5.1.1. Tingkat Akurasi Metode Deteksi

Terdapat lima algoritma metode deteksi serangan siber yang diujikan pada penelitian ini yaitu (1) Naïve Bayes, (2) Logistic Regression, (3) Gradient Boosting, (4) Support Vector Machine, dan (5) K-Nearest Neighbor. Pemilihan kelima algori-

tma tersebut didasarkan pada penelitian sebelumnya. Merujuk pada penelitian sebelumnya, kelima algoritma tersebut terbukti mampu mencapai tingkat akurasi tinggi dalam hal mendeteksi serangan siber. Setelah dilakukan pengujian pada kelima algoritma tersebut, SVM menjadi algoritma yang mampu mendapatkan tingkat akurasi tertinggi. Oleh karena itu, SVM terpilih sebagai algoritma yang digunakan untuk mendeteksi tiga teknik serangan siber, yaitu XSS, SQLi, dan RCE. Berikut ini adalah tingkat akurasi SVM pada masing-masing teknik serangan.

Tabel 5.1. Tingkat Akurasi Metode Deteksi Berdasarkan Teknik Serangan

No	Teknik Serangan	Algoritma	Baris Dataset	Tingkat Akurasi	ToP/Query
1	Cross Site Scripting (XSS)	SVM	49.226	0,996546	0,009618
2	SQL Injection (SQLi)	SVM	30.609	0,997713	0,001008
3	Remote Code Execution (RCE)	SVM	47.177	0,987495	0,004356

Pada umumnya, kebanyakan metode deteksi serangan atau model prediksi berbasis *machine learning* hanya berpijak pada tingkat akurasi berdasarkan proses *data training* dan *data testing*. Saat tingkat akurasi mencapai target yang diinginkan, optimalisasi kinerja metode/model berhenti sampai tahap tersebut. Berbeda dengan pola penelitian seperti itu, penelitian ini melakukan tahapan pengujian kinerja (*performance testing*) dengan dataset yang berbeda, tidak berhenti pada tahapan optimalisasi kinerja (*performance evaluation*).

Tidak hanya itu, metode deteksi berbasis *machine learning* yang dihasilkan juga diujikan pada aplikasi penyerang untuk melihat apakah metode deteksi dapat bekerja secara optimal dan aplikatif. Pada tahapan ini, tingkat akurasi metode deteksi dan lama *time of process* setiap masukan juga dikalkulasikan. Hal ini dilakukan untuk memastikan bahwa metode deteksi tidak hanya mampu mendeteksi dengan

tingkat akurasi tinggi, tetapi juga dapat bekerja secara stabil, tanpa mendistorsi transaksi *request* dan *response* pada *website*. Dengan didukung oleh dua aplikasi penyerang, simulasi serangan dikontruksi agar serupa dengan situasi sebenarnya. Berikut ini adalah simpulan dari pengujian kinerja pada masing-masing metode deteksi.

Tabel 5.2. Persentase Tingkat Akurasi pada Tahap *Performance Testing*

No	Teknik Serangan	*Valid*	*Invalid*	*Accuracy Percentage*
1	Cross Site Scripting (XSS)	7686	3231	70%
2	SQL Injection (SQLi)	33.514	213	99,37%
3	Remote Code Execution (RCE)	8480	13	99,85%

5.1.2. Tingkat Efektivitas Metode Mitigasi

Tingkat efektivitas metode mitigasi *multi-layer security* direpresentasikan dengan persentase dan diukur oleh dua indikator. Indikator pertama, tingkat efektivitas ditentukan oleh hasil perbandingan antara jumlah serangan yang dapat dilakukan pada tahap *implemented* dan jumlah serangan yang dapat dilakukan setelah metode mitigasi diimplementasikan (*implemented stage*). Indikator kedua, sebagai pendukung, tingkat efektivitas dilihat dari perbandingan jumlah serangan yang dapat dilakukan pada metode mitigasi penelitian sebelumnya dan jumlah serangan yang dapat dilakukan pada metode mitigasi penelitian ini. Berikut ini tingkat efektivitas metode miti-gasi serangan siber menggunakan *multi-layer security*.

Tabel 5.3. Metode Mitigasi Tahap *Implemented* dan *Unimplemented* Arachni

	Tahapan Serangan	XSS				SQLi	RCE
		DOM	Reflected	Stored	Total		
1	Unimplemented	137	37	14	188	15	66
2	Layer 1: OWASP ModSecurity	**0**	**0**	14	14	15	1

	Tahapan Serangan	XSS				SQLi	RCE
		DOM	Reflected	Stored	Total		
4	Layer 2: HTTP Middleware	**0**	**0**	14	14	0	1
5	Layer 3: Template Engine	**0**	**0**	**0**	**0**	**0**	1
6	Layer 4: Data Sanitizer	**0**	**0**	**0**	**0**	**0**	**0**
7	Layer 5: Framework	**0**	**0**	**0**	**0**	**0**	**0**

Tabel 5.4. Metode Mitigasi Tahap *Implemented* dan *Unimplemented* ZAP

	Tahapan Serangan	XSS				SQLi	RCE
		DOM	Reflected	Stored	Total		
1	Unimplemented	**0**	38	**0**	38	15	10
2	Layer 1: OWASP ModSecurity	**0**	**0**	**0**	**0**	**0**	**0**
4	Layer 2: HTTP Middleware	**0**	**0**	**0**	**0**	**0**	**0**
5	Layer 3: Template Engine	**0**	**0**	**0**	**0**	**0**	**0**
6	Layer 4: Data Sanitizer	**0**	**0**	**0**	**0**	**0**	**0**
7	Layer 5: Framework	**0**	**0**	**0**	**0**	**0**	**0**

Tabel 5.5. Efektivitas Metode Mitigasi

	Tahapan Serangan	Arachni	ZAP
1	Unimplemented	269	63
2	Layer 1: OWASP ModSecurity	30	0
4	Layer 2: HTTP Middleware	15	0
5	Layer 3: Template Engine	1	0
6	Layer 4: Data Sanitizer	0	0
7	Layer 5: Framework	0	0

Berdasarkan tabel 5.3, 5.4, dan 5.5, serta perbandingan hasil mitigasi yang dijelaskan pada 4.2.3.2, tidak ada serangan yang dapat dilakukan setelah *multi-layer* security diimplementasikan. Hasil mitigasi ini menunjukan bahwa tingkat efektivitas metode mitigasi yang diusulkan mencapai **100%.**

5.1.3. Implementasi Metode Deteksi dan Metode Mitigasi

Penjelasan pada 5.1.3 ini terkait dengan pertanyaan rumusan masalah ketiga, yaitu bagaimana implementasi metode deteksi menggunakan *machine learning* dan mitigasi menggunakan *multi-layer security* untuk mengatasi *cyber attack* pada *multi-websites*. Adapun implementasinya adalah sebagai berikut:

1) model atau metode deteksi dihasilkan melalui proses *data training, data testing, performance optimalization,* dan *performance testing*;
2) dalam konteks *multi-websites*, mitigasi dilakukan oleh OWASP *ModSecurity* (lapisan pertama) yang cakupan kerjanya adalah *web server*;
3) metode deteksi OWASP *ModSecurity* dilakukan secara tradisional berdasarkan pada pengaturan *core rule set* yang diaktifkan*;*
4) model *machine learning* yang dihasilkan untuk metode deteksi dieksekusi pada lapisan kedua metode mitigasi *multi-layer security*;
5) pada tahap development, model *machine learning* dapat disematkan pada lapisan kedua (HTTP *middleware*) setelah disimpan dalam bentuk Python Object dengan format Pickle;
6) pada tahapan *production*, metode deteksi dapat dikemas dalam bentuk *executable* atau dengan memanfaatkan *library* Cython atau PyPy untuk semakin meningkatkan kecepatan eksekusi;
7) pada lapisan *HTTP Middleware*, setiap *request* yang dikirimkan ke *website* divalidasi oleh metode deteksi. Apabila *request* terdeteksi sebagai *payload* maka aplikasi akan mengirimkan kode status HTTP 404;
8) ketika terdapat serangan yang sukses atau berhasil melewati lapisan *HTTP*

Middleware, serangan tersebut akan dimitigasi oleh lapisan berikutnya;

9) untuk semakin meningkatkan tingkat akurasi metode deteksi, setiap *request* yang dikirimkan serta label yang diprediksi metode deteksi dapat dicatat pada lapisan *HTTP Middleware*, kemudian dilakukan validasi pada tiap *request* dan label, sehingga *supervised learning* lanjutan dapat dilakukan.

5.2. Saran

5.2.1. Penelitian Selanjutnya

Berpijak pada hasil dan pembahasan penelitian, peneliti merekomendasikan tiga hal apabila ingin melakukan penelitian dengan topik atau masalah yang sejenis, yaitu (1) gunakan algoritma dengan tingkat akurasi dan ToP yang lebih baik dan uji dengan aplikasi simulasi serangan yang termuktahir; (2) eksplorasi lebih jauh teknik serangan atau celah keamanan selain XSS, SQLi, dan RCE, misalnya *Cross Site Requests Forgery* (CSRF) atau *Insecure Direct Object Reference*; dan (3) gunakan alat dan bahan penelitian dengan spesifikasi yang lebih tinggi untuk semakin mempercepat proses menghasilkan metode deteksi yang akurat.

5.2.2. Praktisi Keamanan Siber

Bagi praktisi keamanan siber atau profesi-profesi yang berhubungan dengan keamanan siber, peneliti menyarankan beberapa hal yaitu (1) implementasikan metode deteksi dan mitigasi yang dihasilkan pada penelitian ini pada *server* atau *web-server*, terutama pada sistem yang digunakan oleh banyak pengguna; (2) *server* berbasis *shared hosting* merupakan salah satu arsitektur yang cocok untuk diterapkan metode deteksi dan mitigasi karena terdiri dari banyak tingkatan pengguna, dan ti-

dak semua penggunanya memahami cara membangun dinding keamanan siber. Metode deteksi dan mitigasi yang dihasilkan pada penelitian ini dapat dimanfaatkan untuk mengatasi serangan siber pada *shared hosting*; dan (3) pastikan memilih dan menggunakan metode deteksi dan mitigasi yang sesuai kebutuhan dan kemampuan. Keamanan multi lapisan memang dapat membuat *website* lebih aman, namun memerlukan *hardware resources* yang tidak sedikit.

5.2.3. Keterbatasan Penelitian

Pada pihak manapun yang tertarik dengan hasil penelitian ini, baik yang tertarik untuk melanjutkannya atau ingin mengimplementasikannya, penelitian ini tetap memiliki keterbatasan yang harus diperhatikan yaitu (1) dataset yang digunakan pada penelitian ini bisa saja tidak lagi relevan digunakan pada tahun-tahun berikutnya mengingat perkembangan teknologi serangan siber yang juga terus berkembang (2) XSS, SQLi, dan RCE memang termasuk pada celah keamanan tingkat *high severity* atau dengan tingkat kerusakan tinggi, namun masih terdapat cukup banyak celah keamanan lainnya yang juga memiliki risiko kerusakan tinggi dan menarik untuk diuji dan diteliti; (3) metode deteksi berbasis *machine learning* dapat mempengaruhi kecepatan *load time* sehingga untuk *server* dengan spesifikasi terbatas dapat didukung oleh metode deteksi tradisional untuk semakin meningkatkan tingkat keamanan, misalnya dengan memanfaatkan *regular expression*.