BAB I
PENDAHULUAN

1.1 Latar Belakang

Tercapainya kemajuan dalam bidang teknologi informasi telah mengubah wajah masyarakat, khususnya dalam sektor bisnis, di mana para pengusaha menggunakan teknologi ini untuk aktivitas bisnis mereka. Di era bisnis yang penuh kompetisi ini, teknologi informasi memberikan peluang untuk meningkatkan efisiensi operasional, meningkatkan produktivitas, dan memperluas jaringan pasar melalui media *online*[1]. Internet, sebagai salah satu produk teknologi informasi, telah menciptakan *platform* baru bagi dunia bisnis, dengan *website* sebagai salah satu komponen kunci. Pentingnya *website* dalam strategi bisnis terlihat jelas, terutama setelah masa pandemi Covid-19 yang memaksa bisnis-bisnis untuk beradaptasi dengan kehadiran *online* yang lebih kuat[2]. Ketergantungan bisnis pada aplikasi web meningkatkan risiko serangan siber. Namun, masih banyak bisnis yang tidak menyadari pentingnya keamanan siber, sehingga membuat mereka rentan terhadap ancaman[3]. Salah satu tantangan yang paling sering dihadapi adalah kurangnya kesadaran tentang keamanan siber itu sendiri [4]. Mengamankan aplikasi *web* sebuah bisnis semestinya menjadi salah satu perhatian utama dengan tujuan untuk melindungi bisnis dari akses tidak sah yang terjadi karena kesalahan pada konfigurasi. Pengembang aplikasi *web* harus lebih berhati-hati dalam memilih komponen pihak ketiga dan menulis kode program, karena kesalahan kecil dapat membuka celah keamanan kerap menjadi faktor yang mendasari munculnya kerentanan aplikasi *web* sebuah bisnis[3].

Apache Server adalah salah satu *web server* paling populer dan banyak digunakan dalam dunia pengembangan *web*. Dikembangkan oleh *Apache Software Foundation, Apache Server* menawarkan keunggulan dalam stabilitas, skalabilitas, dan keamanan. Dalam bahasa Indonesia, *Apache Server* lebih dikenal sebagai Layanan *Web Apache*. *Apache Server* berperan sebagai media penghubung antara *server* dan pengguna, memberikan respons permintaan web yang cepat dan efisien. Namun, perlu diingat bahwa konfigurasi yang salah pada *Apache Server* dapat

membuka celah keamanan dan membuat *server* rentan terhadap serangan siber. Oleh karena itu, penting untuk mengkonfigurasi *Apache Server* dengan benar dan memperbarui secara rutin untuk memastikan keamanan dan stabilitas *server*[5]. Dengan masifnya pengguna layanan *web server* Apache, maka penting untuk melakukan tindakan mitigasi atau remediasi keamanan yang ditemukan pada *web server* tersebut. Salah satu kerentanan keamanan yang paling umum adalah *Insecure Design (Misconfiguration)* atau kesalahan dalam mengkonfigurasi yang telah menjadi masalah serius dalam layanan web sejak beberapa dekade yang lalu. Selain itu, konfigurasi yang salah atau *Insecure Design (Misconfiguration)* pada *Apache Server* juga dapat membuka celah keamanan. Kerentanan ini dapat menyebabkan insiden pencurian data, yang mengakibatkan masalah privasi, dan hilangnya pendapatan suatu perusahaan. Untuk mencegah dan mendeteksi kerentanan ini, diperlukan kesadaran akan pentingnya keamanan aplikasi *web* dan keahlian dalam menyusun solusi sistematis. Oleh karena itu, penting untuk mengkonfigurasi *Apache Server* sesuai dengan saran keamanan yang diberikan oleh pengembang, memperbarui secara rutin, dan melakukan pengujian keamanan secara teratur untuk mengurangi risiko kerentanan keamanan[6].

Penganganan kerentanan *Insecure Design (Misconfiguration)* pada *web server Apache* masih memiliki banyak permasalahan. Mitigasi atau remediasi kerentanan *Insecure Design (Misconfiguration)* yang sudah ada, seperti pengamanan *Cross-Site Request Forgery (CSRF)* [7], masih sebatas implementasi dan konfigurasi alat atau aplikasi keamanan dan modifikasi kode sumber kerentanan secara manual. Hal ini membutuhkan banyak waktu dan keahlian di bidang keamanan siber untuk dapat menerapkannya. Penggunaan aplikasi pemblokiran otomatis terhadap akses mencurigakan seperti *Fail2ban* [8], juga masih memerlukan modifikasi konfigurasi yang dilakukan secara manual, sehingga memakan banyak waktu. Solusi pengamanan *web server Apache* terhadap *Insecure Design (Misconfiguration)* masih sangat sulit ditemukan, dan kebanyakan solusi pengamanan *web server Apache* hanya berfokus pada mitigasi *Denial of Service*. Kurangnya usulan solusi untuk remediasi *Insecure Design (Misconfiguration)* pada *web server Apache* juga didasari oleh dinamisme, kompleksitas, dan sulitnya

menggeneralisir penerapan remediasi dari *Insecure Design (Misconfiguration)* itu sendiri[6].

Berdasarkan permasalahan tersebut, penelitian dengan judul "Otomasi Remediasi *Insecure Design* pada *Web Server Apache2* berbasis *Fail2Ban*" dilakukan untuk merancang solusi berupa sebuah sistem otomasi remediasi terhadap *Insecure Design* yang terdapat pada *web server Apache* disertai integrasi pengamanan akses melalui penyaringan *log* dengan aplikasi *Fail2ban*. Solusi pada sistem yang dirancang dalam penelitian ini berfokus untuk dapat melakukan remediasi terhadap *Insecure Design* berupa kerentanan terbukanya indeks opsi pada konfigurasi *web server Apache*, kerentanan izin akses yang tidak semestinya, akses menebak *URL* yang berulang, dan percobaan eksekusi perintah pada isi dari *http request* secara otomatis, sehingga dapat membuat pengamanan kerentanan *Insecure Design (Misconfiguration)* pada *web server Apache* menjadi lebih cepat dengan menurunkan kebutuhan waktu remediasi sebesar 43,98% dibandingkan dengan remediasi secara manual dan dengan rasio keberhasilan atau *success rate* remediasi sebesar 100%.

1.2 Rumusan Masalah

Berdasar latar belakang masalah seperti pada 1.1, rumusan masalah yang hendak diselesaikan dengan penelitian ini adalah bagaimana membuat sebuah aplikasi atau sistem yang dapat melakukan remediasi pada kerentanan secara otomatis pada *Insecure Design (Misconfiguration) Apache2* melalui integrasi aplikasi *Fail2Ban* tanpa perlu terlalu banyak interaksi dari pengguna ataupun keahlian khusus.

1.3 Batasan Masalah

Batasan masalah yang secara spesifik akan dibahas pada penelitian ini meliputi:

1. Fokus utama penelitian penelitian ini adalah otomasi terhadap remediasi kerentanan *Insecure Design (Misconfiguration)* pada *web server Apache2* versi 2.4.59 dengan aplikasi *Fail2ban* versi 0.11.1.

2. Remediasi kerentanan *Insecure Design (Misconfiguration)* pada konfigurasi *web server Apache2* hanya meliputi izin akses pada direktori aset dan konfigurasi *Apache2* serta konfigurasi situs.
3. Kerentanan *Insecure Design (Misconfiguration)* yang dibahas dan dilakukan remediasi secara spesifik meliputi kerentanan pada konfigurasi *web server Apache2* dan kerentanan dari percobaan menebak akses navigasi langsung *URL*.
4. Remediasi kerentanan *Insecure Design (Misconfiguration)* dalam percobaan menebak akses navigasi langsung *URL* diremediasi menggunakan aplikasi *Fail2Ban* dengan menambahkan *log* akses *Apache2* ke dalam konfigurasi penjara (*jail*) dan menerapkan filter terhadap akses *URL* yang gagal serta akses yang memuat teks tertentu, meliputi kode eror 4xx dan 5xx, serta teks *shell*, *exec*, *cmd*, *wget*, dan *curl*.
5. Otomasi diterapkan menggunakan bahasa pemrograman *python* (versi 3.11) dan C.
6. Otomasi dilakukan dengan melakukan kloning kode sumber dari repositori github milik penulis, kemudian menjalankan berkas "main.py" dari *server* yang akan diremediasi.
7. Keseluruhan kode sumber bersifat portabel, sehingga hasil laporan dari eksekusi otomasi juga diletakkan di dalam sub-direktori tersendiri dari direktori utama kode sumber tersebut.
8. Pengujian dilakukan menggunakan sistem operasi *Debian Server* versi *12.5.0* (*Codename : Bullseye*)
9. Penelitian ini tidak membahas terkait topologi jaringan dari *Virtual Private Server* (*VPS*) yang digunakan, melainkan hanya membahas kesinambungan dan peran dari setiap aplikasi yang terdampak langsung oleh sistem otomasi.

1.4 Tujuan Penelitian

Tujuan yang akan dicapai oleh penulis dalam penelitian ini adalah merancang aplikasi atau sistem sederhana yang dapat digunakan untuk pengembang *web* untuk melakukan otomasi dan remediasi terhadap kerentanan *Insecure Design (Misconfiguration)* yang paling umum dan kerap terjadi tanpa disadari pada *web server Apache2*.

1.5 Manfaat Penelitian

Hasil penelitian ini diharap dapat memberikan manfaat bagi pengembang *web* untuk dapat menerapkan peningkatan keamanan situs *web* berbasis *web server* Apache2 pada sistem operasi *Linux* berbasis *Debian* yang dikembangkan. Pengembang *web* dapat menggunakan *tools* sistem otomasi dari penelitian ini untuk mengamankan server *web* dari kerentanan *Insecure Design (Misconfiguration)* yang paling umum ditemukan pada *web server Apache2* serta menerapkan *filter* pemblokiran akses mencurigakan berbasis *Fail2ban* dengan lebih cepat dan akurat. Selain pengembang *web*, hasil dari penelitian ini juga dapat dimanfaatkan sebagai acuan atau referensi dalam memperluas dan mengembangkan penilitan terkait solusi otomasi remediasi kerentanan keamanan siber lainnya.

1.6 Sistematika Penulisan

Sistematik penulisan berisikan garis besar atau gambaran secara umum penelitian ini sehingga mempermudah pemahaman alur isi. Adapun garis besar isi skripsi ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

   Bab pendahuluan memuat pemaparan terkait latar belakang penelitian, rumusan masalah penelitian, batasan masalah penelitian, tujuan penelitian, manfaat penelitian serta sistematika penulisan dari laporan penelitian. Pemaparan pada bab ini bertujuan untuk memberikan penjelasan dari permasalahan yang dibahas mulai dari bagaimana hal tersebut menjadi sebuah masalah, lingkup spesifik dari masalah tersebut,

usulan solusi terhadap masalah tersebut, serta struktur pelaporan dari penelitian ini.

2. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Bab tinjauan pustaka dan dasar teori memuat pemaparan sistematis terkait referensi dari penelitian lain yang relevan dengan pokok-pokok atau poin-poin penting yang dibahas dalam penelitian ini serta pemaparan teori dari poin-poin pemting tersebut. Pemaparan pada bab ini bertujuan untuk memberikan gambaran umum dari poin-poin penting yang dibahas dalam penelitian serta perbandingan sekaligus penjelasan dari perbedaan apa saja yang mendasari atau terdapat antara penelitian-penelitian lain yang relevan dengan penelitian pada penelitian ini.

3. BAB III METODE PENELITIAN

Bab metode penelitian memuat pemaparan terkait alat dan bahan yang digunakan, tahapan yang dilakukan, serta rancangan sistem atau solusi dari permasalahan yang dibahas pada penelitian ini. Pemaparan pada bab ini bertujuan untuk menjelaskan secara rinci terkait spesifikasi alat dan bahan, alur tahapan dalam menerapkan solusi, serta desain rancangan solusi guna memberikan gambaran yang dapat menjadi acuan dalam penggunaan kembali atau *reuse* dan pertimbangan dalam pengembangan lebih lanjut.

4. BAB IV HASIL DAN PEMBAHASAN

Bab hasil dan pembahasan memuat pemaparan terkait hasil-hasil yang didapatkan dalam implementasi dan pengujian dari rancangan sistem atau solusi pada penelitian penelitian ini. Pemaparan pada bab ini bertujuan untuk menyampaikan dan menjelaskan terkait apa saja yang dicapai serta hasil yang didapatkan dari implementasi dan pengujian sistem atau solusi yang diusulkan, sehingga dapat menjadi bahan atau materi pembelajaran serta menjadi acuan dalam pertimbangan untuk

pengembangan lebih lanjut atau untuk alternatif penyelesaian dari permasalahan yang serupa.

5. BAB V PENUTUP

Bab penutup memuat pemaparan terkait kesimpulan yang didapatkan dari pelaksanaan penelitian penelitian serta saran dari penulis terhadap hasil penelitian serta potensi pengembangan kedepannya. Pemaparan pada bab ini bertujuan untuk menyampaikan pokok dari hasil dan pembahasan pada penelitian secara singkat serta menyampaikan rekomendasi dari penulis untuk keberlanjutan, optimasi, atau alternatif dari penelitian penelitian yang sudah dilakukan.