BAB I

PENDAHULUAN

1.1 Latar Belakang

Framework Codeigniter merupakan suatu web aplikasi *opensource* yang digunakan untuk membangun website dinamis. *Codeigniter* adalah bagian dari *Application Development Framework* (ADF) untuk membangun situs web menggunakan bahasa php. *Codeigniter* mempunyai kelengkapan *library* untuk mengoperasikan aplikasi. Penelitian yang dilakukan oleh Destiningrum menggunakan metode *waterfall* pada sistem penjadwalan dokter yang berbasis *Framework codeigniter*. Adanya sistem tersebut dapat memberikan informasi jadwal dari praktik dokter dengan cepat, akurat, dan efisien. Dengan demikian penggunaan dari *Framework Codeigniter* dapat berkontribusi terhadap sistem penjadwalan. Dari penelitiann tersebut didapatkan kualitas web yang menggunakan *Framework Codeigniter* sebesar 87,87% (Destiningrum, 2017).

Codeigniter memiliki keunggulan lainnya yaitu *free, Light weight, Fast, Packs a Punch, Extensible,* dan *Thoroughly Documented*. *Codeigniter* mudah dalam dipelajari dan mempunyai sedikit konfigurasi. *Framework* ini juga memiliki ukuran yang relative kecil sehingga sangat banyak digunakan pada *webserver* umum. *Codeigniter* mempunyai dokumentasi yang lengkap sehingga dapat dimodifikasi oleh developer. Akan tetapi *Framework* ini mempunyai kelemahan dibidang keamanan.

Kelemahan keamanan yang paling umum dari *Framework Codeigniter* adalah serangan *Structured Query Language* (SQL) *Injection*. *Webserver* Sikamek yang menggunakan *Codeigniter* diprediksi memiliki celah SQL *Injection*. Serangan SQL *Injection* menjadi serangan paling sering yang dilakukan oleh *attacker* atau *tester*.

Penelitian yang dilakukan oleh Castillo pada halaman *login* website perpustakaan dengan melakukan metode *prepared statement* maka pengguna tidak sah (*attacker*) tidak dapat melakukan injeksi terhadap struktur kueri *Framework codeigniter*. *Attacker* juga tidak dapat melakuan injeksi komentar *line* dan *tautology* pada halaman *login* web. Penelitian ini membahas tentang pencegahan dari serangan SQL *Injection* pada halaman *login*. Hasil dari penelitian menunjukkan bahwa *illegal user* dapat melakukan serangan injeksi tipe *tautology* dan *comentline* pada halaman *login* pada website perpustakaan. Menerapkan teknik pernyataan (*prepared statement*) memiliki kemampuan dalam membuat halaman *login* memverifikasi skrip untuk memblokir serangan, dan menyaring *input* penyerang sebelum mengakses *database* (Castillo, *et al*. 2019).

Pada saat tertentu *attacker* juga dapat masuk kedalam sistem administrasi dari sebuah web dengan menggunakan *database* yang telah didapatkanya. *Attacker* dapat mengusai *webserver* dari kelemahan-kelemahan SQL, bahkan dengan *internet protocol* (IP) yang sama dengan situs. Melalui serangan SQL, *attacker* mungkin saja membuat celah lain dan menguasai situs lain yang se-*ip* dengan situs Sikamek. Serangan SQL *Injection* yang dilakukan oleh *attacker* biasanya terdapat pada alamat *login*. *Attacker* dapat mem*bypass* admin pada alamat *login*.

Metode *bypass admin* merupakan salah satu teknik injeksi kueri pada *form login*, misalnya menyuntikkan kode kueri '="or' pada *user* dan *password*. Kusuma melakukan penelitian tentang analisis serangan SQL *Injection* pada layanan situs web menggunakan metode kueri CANDID. Situs layanan web akan menjadi *error database* ketika ditambahkan tanda koma atas (‘) pada bagian url situs. Pengamanan dilakukan dengan menggunakan pendekatan respons dari waktu. Hasil penelitian mengungkapkan bahwa terdapat perbedaan waktu yang menutup celah dari SQL *Injection* walaupun ada perubahan pada aplikasi (Kusuma, 2018).

Sasaran serangan SQL *Injection* yang dilakukan oleh *attacker* pada *Framework Codeigniter* terdapat pada kondisi dari url-url. Alamat atau url dari situs yang memakai *Framework codeigniter*, biasanya memiliki celah SQL *Injection*. *Attacker* dapat memasukkan perintah-perintah SQL melalui url untuk dieksekusi oleh *database* . Penyebab utama dari celah ini adalah *variable* yang kurang tersaring. Banyak cara yang dilakukan dalam menemukan celah SQL untuk *Framework codeigniter*. Halib melakukan penelitian tentang teknik *hacking* SQL *Injection* pada *webserver* menggunakan metode eksperimen kuantitatif. Penelitian tersebut

dilakukan secara langsung pada target kemudian menghasilkan kemungkinan-kemungkinan celah SQL pada *webserver*. Dari celah SQL ini didapatkan password admin yaitu adminpass. Pengumpulan dari celah-celah *webserver* memudahkan administrator untuk memeriksanya. Dari penelitian tersebut peneliti akan mengumpulkan celah dari SQL *Injection* kemudian membuat laporan yang akan diberitahukan kepada admin dari situs Sikamek (Halib, *et al*. 2017).

Identifikasi kelemahan SQL *Injection* pada *Framework Codeigniter* peneliti menggunakan model forensik. Model forensik adalah suatu representasi tentang keamanan komputer terkait penyelidikan dalam menentukan sumber serangan melalui identifikasi, pengujian, analisis dan pelaporan. Sebuah serangan terhadap *webserver* dapat diidentifikasikan dengan menggunakan model forensik. Kelebihan model forensik adalah menganalisis dan mendapatkan kembali suatu fakta tersembunyi dan kejadian dari lingkungan. Fadlil melakukan penelitian tentang sistem pengamanan jaringan menggunakan analisis forensik dengan menggunakan software Winbox RouterOS v3,6. Hasil penelitian menunjukkan bahwa tercatatnya *ip* baru dari *attacker* pada *software router* di port 80. Menu *torch* dalam *software router* memberikan informasi detail terhadap serangan seperti *ip address*, jumlah data, dan waktu yang dilakukan oleh *attacker*. Setelah pengamanan terhadap sistem komputer dibuat *attacker* tidak mampu lagi melakukan penyerangan (Fadlil, *et al*. 2017).

Model forensik pada umumnya memiliki empat buah tahap-tahap yaitu deteksi, pemeriksaan, analisis, dan pelaporan. Umar melakukan penelitian tentang analisis eksperimen dari web *database* menggunakan model forensik langsung. Kerangka kerja dari model forensik langsung digunakan untuk menguji fitur privasi dari *database Firefox* dan *Incognito* secara eksperimental. Pada proses langsung tersebut melalui memori forensik ditemukan informasi kegiatan seperti jejak email dan id *facebook* bahkan setelah *database* ditutup. Eksperimen tersebut membuktikan bahwa *vendor* privasi dari *database* belum tentu mengamankan penelusuran pengguna. Dari penelitian ini didapatkan dengan menggunakan metode forensik langsung dapat mencari bukti digital lain terkait dengan web *database* (Umar, *et al*. 2017).

Implementasi dari model forensik dari web *database* juga diteliti oleh Faiz. Penelitian ini memahas tentang perbandingan dari beberapa *database* dalam keamanan email. Data penelitian berasal email *gmail, yahoo,* dan *outlook* kemudian *database* yang digunakan adalah *Internet Explorer*, *Mozilla Firefox*, dan *Chrome*. Penelitian ini menggunakan model forensik langsung

dan menghasilkan perbandingan keamanan dari ketiga *database* tersebut. Melalui model forensik langsung *database* yang paling lemah tingkat keamanannya adalah *Internet Explorer* sedangkan *Chrome* merupakan yang paling kuat untuk diserang (Faiz, 2017). Melalui model forensik ini peneliti akan mengidentifikasi celah SQL *Injection* studi kasus server Sikamek Provinsi Sumatera Barat.

Situs Sikamek merupakan bagian dari layanan pemerintahan sumatera barat dibidang pariwisata. Situs ini menyediakan informasi tentang hotel, resor, penerbangan, sewa liburan, serta paket perjalanan khusus daerah Sumatera Barat. Situs Sikamek menyajikan menu *search* yang dapat menelusuri kabupaten dan kota se Sumatera Barat. Setelah melakukan pengamatan bahwa situs Sikamek menggunakan *Framework Codeigniter* sebagai konten manajemenya. Struktur dari situs Sikamek sudah dimodifikasi dan memiliki fitur-fitur yang berbeda dengan *Codeigniter* lainnya. Walaupun memakai sistem manajemen konten *codeigniter*, situs Sikamek memiliki kelemahan dibidang keamanan. Kelemahan dari situs ini akan diidentifikasi peneliti dengan menggunakan model forensik.

Berdasarkan analisis dari penelitian-penelitian yang telah dilakukan dengan menggunakan model forensik serta serangan SQL *Injection*. Peneliti akan melakukan identifikasi kelemahan pada situs atau subdomain Sikamek pemerintahan Provinsi Sumatera Barat bidang pariwisata. Situs ini menggunakan *Framework Codeigniter* dalam mengelola sistem manajemennya. Kelemahan yang akan diidentifikasi adalah serangan SQL *Injection*, kemudian peneliti menggunakan model forensik untuk membantu dalam mengumpulkan, menganalisa, dan melaporkan data kelemahan dari situs Sikamek. Sehingga penelitian ini dapat membantu pihak keamanan pemerintahan Provinsi Sumatera Barat. Dengan demikian penulis mengusulkan penelitian berjudul **Identifikasi Kelemahan *Framework Codeigniter* pada Server terhadap Serangan SQL *Injection* Menggunakan Model Forensik (Studi Kasus di Pemerintahan Provinsi Sumatera Barat).**

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan sebelumnya, maka penulis dapat merumuskan beberapa detail permasalahan sebagai berikut:

1. Bagaimana cara mengidentifikasi kelemahan aplikasi *Codeigniter* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection* menggunakan model forensik?
2. Bagaimana cara menggunakan model forensik dalam merepresentasikan celah keamanan situs Sikamek?

1.3 Batasan Masalah

Agar penulisan tidak keluar dari permasalahan yang ada dan hasil penelitian dapat diperoleh dengan baik, maka penulis membatasi ruang lingkup pembahasan sebagai berikut:

1. Mengidentifikasi kelemahan aplikasi *Codeigniter* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection*.
2. Menerapkan model forensik untuk menemukan kelemahan SQL *Injection* dari *webserver* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection*.

1.4 Tujuan Penelitian

Dalam penelitian ini dan pelaksanaannya ada beberapa tujuan yang hendak dicapai, diantaranya:

1. Mengetahui kelemahan aplikasi *Codeigniter* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection.*
2. Mengetahui tahap-tahap model forensik dalam menemukan kelemahan *Framework Codeigniter* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection* menggunakan model forensik.
3. Memberikan hasil identifikasi kelemahan *Framework Codeigniter* pada subdomain Sikamek pemerintahan Provinsi Sumatera Barat terhadap serangan SQL *Injection* kepada administrator.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini sebagai berikut:

1. Dengan penelitian ini diharapkan memberikan informasi tentang celah SQL *Injection* kepada administrator pemerintahan Provinsi Sumatera Barat sehingga menjadi pedoman untuk memperbaiki keamanan sistemnya.
2. Manfaat bagi penulis sendiri adalah untuk mengetahui url-url yang *vulnerable* sehingga meningkatkan cara mencari celah terbaru untuk situs lainnya.
3. Memberikan masukan kepada administrator web lain tentang cara menganalisa sistem aplikasi webnya.

1.6 Sistematika Penulisan

Sistematika disesuaikan dengan template yang diatur dalam tata penulisan program studi masing-masing. Seperti:

Bab I : Pendahuluan

Berisi Latar Belakang, Perumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Sistematika Penelitian.

Bab II : Landasan Teori

Menguraikan teori-teori dan penerapan model forensik yang digunakan dalam tahap-tahap penyelesaian masalah sesuai dengan topik penelitian. Teori tentang serangan SQL *Injection.*

Bab III : Metodologi Penelitian

Bagian ini menjelaskan jenis penelitian yang dilakukan, pendekatan yang digunakan, sumber data, lokasi penelitan, metode dan alat pengumpulan data serta teknik pengolahan dan analisa.

Bab IV : Analisa dan Perancangan Sistem

Bagian ini menjelaskan tentang proses model forensik dan hasil yang didapatkan.

Bab V : Implementasi dan Hasil

Bagian ini membahas yang telah didapatkan yaitu berupa laporan celah SQL.

Bab VI : Kesimpulan dan Saran

Bab ini membuat kesimpulan dan hasil penelitian keunggulan model forensik dan mengidentifikasi celah SQL *Injection* dari sebuah situs.