

BAB I

PENDAHULUAN

1.1 Latar Belakang

SQL Injection adalah sebuah metode untuk menyisipkan perintah *SQL* sebagai input melalui sebuah aplikasi web untuk mendapatkan akses *database*^{[6][18]}. Serangan *SQL Injection* dapat dilakukan melalui input user maupun *url*^[17]. Berdasarkan informasi dari *OWASP (Open Web Application Security Project)* yang telah telah medata sejak tahun 2003 berbagai jenis serangan *hacking*, dimana *SQL Injection* masih menduduki peringkat pertama sebagai serangan *hacking* yang paling sering dilakukan^[12].

Pengamanan aplikasi *website* dapat dilakukan dengan memasang *firewall*, anti virus, atau software sejenis pada server. Namun yang lebih penting lagi adalah membuat kode-kode program yang bebas dari bugs karena *firewall* yang terbaik pun akan tidak berguna apabila kode-kode yang dibuat oleh developer tidak bebas dari kesalahan logika pemrograman.

PHP dan *Framework Codeigniter* merupakan contoh *scripting* yang sering digunakan para *developer* dalam membangun aplikasi *website*. Hal tersebut dipengaruhi beberapa faktor diantaranya mudah digunakan, ukuran relatif kecil dan cepat, kemudahan instalasi, terdapat *library* validasi, dokumentasi lengkap, mendukung *PHP 4* dan *PHP 5*, serta menyediakan banyak fungsi, diantaranya fungsi enkripsi, *session*, *cookies*, *xss filtering*, dan sebagainya^[5]. Perbedaan antara keduanya, *PHP* sendiri merupakan suatu bahasa pemrograman yang digunakan untuk membangun suatu web dinamis serta bekerja disisi *server* dan *Codeigniter* adalah *framework php* yang digunakan untuk membangun suatu web dinamis serta menggunakan konsep *MVC (Model-View-Controller)* dalam penulisan kode programnya. Pada penelitian sebelumnya pernah dilakukan mengenai analisis *SQL Injection* pada framework *Ruby on Rails* dan *PHP* yang mana memaparkan kasus-kasus serangan *SQL Injection* pada kedua *scripting* tersebut^[4]. Pada Tugas Akhir ini dilakukan analisis bagaimana teknik penanganan *SQL Injection* pada basis data *MySQL* dengan *framework Codeigniter* dan tanpa *framework (PHP)* sehingga diketahui kemampuan dan *vulnerability* kedua *scripting* terhadap *SQL Injection* beserta teknik penanganannya.

1.2 Tujuan

1. Melakukan analisis terhadap *framework codeigniter* dan tanpa *framework (PHP)* terutama dari sisi *vulnerability*, yang mendukung penanganan terhadap *SQL Injection*.
2. Melakukan analisis teknik untuk menangani *SQL Injection* pada *framework codeigniter* dan tanpa *framework (PHP)*.

1.3 Rumusan Masalah

Rumusan masalah dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Bagaimana kemampuan *PHP* dan *framework Codeigniter* dalam menangani *SQL Injection* sehingga bisa dianalisis kelemahan dan keunggulan kedua *scripting* tersebut.
2. Bagaimana teknik penanganan *SQL Injection* pada *framework Codeigniter* dan tanpa *framework (PHP)*.

1.4 Batasan Masalah

Batasan masalah yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Analisa *SQL Injection* dilakukan pada *PHP* versi 5 dengan menggunakan *database MySQL*.
2. Hanya digunakan satu jenis teknik untuk menangani *SQL Injection* pada masing-masing *PHP* dan *Codeigniter* dengan menggunakan *library* atau fungsi atau pola yang tersedia atau membutuhkan modul tambahan jika tidak terdapat fungsi pendukung pada kedua *scripting* tersebut.
3. Teknik yang digunakan untuk penanganan *SQL Injection* pada *PHP* dan *Codeigniter* bisa sama atau berbeda.
4. Tidak membahas keamanan dari sisi *network*.

1.5 Metodologi Penelitian

Metodologi penelitian yang dilakukan pada tugas akhir ini adalah sebagai berikut:

1. Studi Literatur
Tahap pengumpulan referensi yang berkaitan dengan topik tugas akhir ini seperti *SQL Injection, PHP, Codeigniter, MySQL*.
2. Penyusunan skenario *SQL Injection*

- Menyusun dan mengumpulkan skenario yang akan digunakan untuk melakukan pengujian *SQL Injection* pada aplikasi berbasis *PHP* dan *framework Codeigniter*.
3. Perancangan dan pembangunan
Merancang dan membangun aplikasi berbasis *PHP* dan *framework Codeigniter* sebagai media untuk melakukan pengujian yang akan dianalisis.
 4. Pengujian
Pengujian skenario yang telah dirancang pada aplikasi berbasis *PHP* dan *framework Codeigniter* pada database *MySQL*.
 5. Evaluasi
Mempelajari hasil pengujian untuk memecahkan permasalahan *SQL Injection* dan mengetahui teknik penanganan *SQL Injection* pada masing-masing aplikasi berbasis *PHP* dan *framework Codeigniter*.
 6. Analisis.
Melakukan analisis terhadap pengujian dan penanganan yang diberikan untuk menangani *SQL Injection* pada *PHP* dan *framework Codeigniter*.
 7. Penyusunan Laporan Tugas Akhir dan Penarikan kesimpulan.

1.6 Sistematika Penulisan

Sistematika penulisan proposal tugas akhir ini adalah sebagai berikut :

1. Bab I Pendahuluan
Bab ini menjelaskan mengenai latar belakang masalah, tujuan, rumusan dan batasan masalah, metodologi penelitian, dan sistematika penulisan.
2. Bab II Dasar Teori
Bab ini menjelaskan mengenai teori dasar yang digunakan pada pemodelan dan analisa pada tugas akhir.
3. Bab III Model Sistem
Bab ini menjelaskan perancangan yang akan dilakukan berdasarkan mekanisme dan batasan yang digunakan.
4. Bab IV Pengujian dan Analisa
Bab ini berisi hasil pengujian dan analisa yang telah didapat penulis terhadap sistem yang dirancang.
5. Bab V Penutup
Bab ini berisi kesimpulan yang dapat ditarik dari hasil pengujian serta saran bagi para pembaca untuk dapat mengembangkan tugas akhir ini kedepannya.