

ABSTRAK

SQL Injection adalah sebuah metode untuk menyisipkan perintah *SQL* sebagai input melalui sebuah aplikasi web untuk mendapatkan akses *database*. Pengamanan aplikasi web 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*, dan menyediakan banyak fungsi, diantaranya enkripsi, *session*, *cookies*, *xss filtering*. Pada Tugas Akhir ini, dilakukan analisis mengenai teknik penanganan *SQL Injection* pada basis data *MySQL* dengan *framework Codeigniter* dan tanpa *framework (PHP)* yang mana digunakan 4 skenario dalam pengujian *SQL Injection* diantaranya pengujian karakter yang memiliki makna dalam *query*, pengujian modifikasi input *login* ketika username diketahui, pengujian modifikasi input *login* ketika username dan password tidak diketahui, dan pengujian injeksi *url*.

Dari hasil pengujian dan analisa didapat bahwa serangan *SQL Injection* pada *PHP* dan *Codeigniter* dapat ditangani dengan baik menggunakan fungsi *mysql_real_escape_string*(untuk *PHP*) dan penggunaan *library active record* , *uri segment*(untuk *Codeigniter*) yang mana dari hasil pengujian didapat bahwa 0% *error query* dan 0% terinjeksi(pada 4 skenario pengujian) dari 104 vektor serangan.

Kata Kunci : *Code Igniter* , *MySQL*, *PHP*, *SQL Injection*.