RANCANG BANGUN APLIKASI PENDETEKSIAN *VULNERABILITY STRUCTURED QUERY LANGUAGE (SQL) INJECTION* UNTUK KEAMANAN WEBSITE

Angela Merici Elu
Email: angela.merici.elu@gmail.com
Fakultas Sains dan Teknologi Program Teknik Informatika

ABSTRAKSI

Sebuah website memerlukan tingkat keamanan yang tinggi untuk mencegah kebocoran, manipulasi, penghapusan, perubahan, pencurian data dan berpindahnya hak pengelolahan yang diakibatkan oleh pihak-pihak yang tidak bertanggung jawab. Dengan perkembangan teknologi yang pesat, sebuah website memiliki banyak celah yang membuatnya rentan terhadap serangan penyusup, salah satu metode serangannya adalah SQL Injection. Kebutuhan akan adanya suatu aplikasi yang dapat mengenali celah serangan SQL Injection ini menjadi sesuatu yang penting bagi pengelola website untuk mempermudah pekerjaan mereka. Berdasarkan kebutuhan tersebut maka penulis membuat suatu penelitian dengan judul Rancang Bangun Aplikasi Pendeteksian Vulnerability SQL Injection Untuk Keamanan Website.

Aplikasi yang akan dibuat menggunakan metode POST supaya dapat memasukkan data berupa regular expression sebagai penguji kerentanan pada bagian input form yang ada pada sebuah website. Setelah regex dimasukkan ke dalam input form, maka aplikasi menunggu respon dan menganalisa pesan yang muncul, ada tidaknya muncul pesan eror atau pesan sql dari hasil respon terhadap data yang dimasukkan. Apabila muncul kedua pesan tersebut maka dari hasil analisa aplikasi, URL yang sedang dianalisa dan diuji dianggap rentan terhadap serangan SQL Injection.

Setelah menganalisa buku-buku ilmiah, pengumpulan data lapangan dan perancangan aplikasi serta ujicoba aplikasi maka dihasilkan suatu aplikasi yang mampu mendeteksi kerentanan website terhadap serangan SQL Injection. Dimana dengan adanya aplikasi ini mampu mencukupi kebutuhan para pengelola website untuk menguji website mereka apakah rentan terhadap serangan SQL Injection atau tidak.

Kata kunci: Keamanan Website, SQL Injection, Vulnerability

1. PENDAHULUAN

Perkembangan teknologi, terutama *internet* mendorong banyak pihak untuk ikut berpartisipasi di dalamnya baik sebagai *user* maupun *administrator*. Dengan dibangunnya suatu *website* mendorong banyak informasi yang dikoneksikan dengan *internet* mudah diakses oleh para *user*. Informasi yang berada pada suatu *website*, baik informasi yang terdapat pada halaman *interface* maupun *administrator* tidak secara otomatis dijamin aman dikarenakan adanya risiko terdapatnya celah pada lapisan keamanan yang memungkinkan disalahgunakan oleh pihak yang tidak berwewenang. Semakin tinggi nilai informasi pada suatu *website* semakin tinggi pula risiko yang akan ditanggung oleh *website* tersebut sehingga harus menciptakan suatu *website* yang memberikan jaminan keamanan baik bagi *user* dan terutama *administrator* demi kemudahan dalam tukar-menukar informasi, transaksi, yang membutuhkan mobilitas tinggi dan ketangguhan terhadap gangguan, kestabilan. Hal ini diperlukan untuk melindungi data supaya tidak rusak/hilang dikarenakan oleh virus komputer atau oleh serangan *intruder*.

Berbagai jenis serangan yang umumnya terjadi pada suatu *website* membuat pihak *administrator* senantiasa untuk melakukan pengujian dan upaya penyempurnaan *website* demi meningkatkan keamanan *website* tersebut. Salah satu ancaman yang sering terjadi pada suatu *website* adalah *SQL Injection* dimana ancaman ini dikategorikan dalam 10 peringkat teratas kerentanan *website* pada tahun 2010 oleh *Open Web Application Security project*. Serangan ini memanfaatkan celah keamanan pada *query* dengan memasukkan *SQL statement*. Seringkali para pemilik *website* tidak mengetahui celah tersebut, sehingga mengira bahwa *website* yang dimilikinya telah memiliki keamanan yang sempurna dan tidak mengetahui adanya celah ancaman keamanan dari "pintu belakang (*Backdoor*)" *website* tersebut. *SQL Injection* melakukan serangan dari celah tersebut yang kemudian berusaha untuk mengetahui dan mencuri data *administrator* dengan tujuan mengambil alih kepemilikan *website* tersebut. Hal tersebut terjadi karena input user tidak divalidasi sehingga muncul celah keamanan tersebut.

2. TUJUAN DAN MANFAAT PENELITIAN

2.1 Tujuan Manfaat Penelitian

Tujuan dari penyusunan skripsi ini adalah merancang dan membangun sebuah sistem aplikasi yang nantinya berfungsi untuk membantu dan mempermudah para *administrator* pengelola *website* untuk menguji tingkat keamanan *website* mereka terhadap serangan *SQL Injection.* Menghasilkan suatu aplikasi yang mampu memenuhi kebutuhan *admistrator* pengelola *website* untuk mencegah terjadinya serangan *SQL Injection*.

2.2 Manfaat Penelitian

Adapun metode pemecahan masalah yang diambil yaitu :

1. Bagi mahasiswa

   Agar mahasiswa dapat menerapkan ilmu yang didapat selama perkuliahan, menambah wawasan dan pengalaman sebagai bekal dalam memasuki dunia kerja
2. Bagi Masyarakat

   Dapat memenuhi kebutuhan akan adanya suatu aplikasi yang dapat mendeteksi kerentanan *website* terhadap *SQL Injection*.
3. Manfaat Bagi Perguruan Tinggi

   Dapat berfungsi sebagai bahan evaluasi dalam menilai kemampuan seorang mahasiswa dalam menerapkan ilmu yang didapat dari lembaga pendidikan dan peningkatan mutu dan masa depan.

3. METODE PENELITIAN

a. Metode yang digunakan dalam pemecahan ini adalah sebagai berikut:
b. Studi Literatur

c. Pada tahap ini dilakukan studi literatur dari beberapa referensi baik itu dari buku-buku, karya ilmiah dan artikel-artikel yang dapat ditemukan di internet.
d. Perancangan Aplikasi
e. Pada tahap ini dilakukan perancangan aplikasi yang akan dibangun, meliputi perancangan alur aplikasi, perancangan aplikasi dan pembuatan tampilan aplikasi.
f. Persiapan Data
g. Data-data penunjang yang didapatkan berupa suatu kesimpulan, fakta-fakta dan aturan yang mengatur proses pencarian data yang saling berhubungan satu sama lain.

4. LANDASAN TEORI

4.1 Pengertian *Website*

Website merupakan suatu layanan sajian informasi yang menggunakan konsep tautan (*hyperlink*) yang mempermudah pengguna *internet* untuk melakukan pencarian informasi. Suatu *website* merupakan kumpulan halaman *web* yang saling terhubung dan file-filenya saling terkait yang dapat disertai dengan *file* gambar, video ataupun *file-file* yang lain baik bersifat statis maupun dinamis. Informasi atau *file-file* yang yang terdapat dalam *website* tersimpan di dalam sebuah *server web* yang secara umum ditulis dengan format HTML atau *Hypertext Markup Language*. *Hyperlink* adalah sebuah acuan dalam dokumen *hypertext* ke dokumen yang lain yang dapat berbentuk grafis ataupun teks dalam dokumen tersebut yang biasanya ditandai dengan adanya garis bawah dan atau berwarna biru. Pada suatu *web* secara umum terdiri dari *page* atau halaman yang ditempatkan pada suatu *server web* yang dapat diakses melalui jaringan *internet* ataupun jaringan wilayah lokal yang merupakan *file* teks yang berisi *tag-tag* dengan format HTML

4.2 Kejahatan Dunia Maya

Istilah keamanan biasa digunakan untuk hal-hal yang berhubungan dengan kejahatan, segala bentuk pencurian, dan lain-lain. Keamanan Website adalah suatu cabang teknologi yang dikenal dengan nama keamanan informasi yang diterapkan pada website. Sasaran keamanan website antara lain adalah sebagai perlindungan terhadap informasi/data. Semakin maju sebuah teknologi yang diiringi dengan bertambahnya nilai informasi maka akan memicu juga munculnya jenis kejahatan baru. Secara khusus, kejahatan yang mengancam jaringan internet atau yang menggunakan jaringan internet untuk melakukan kejahatan disebut kejahatan dunia maya. Kejahatan dunia maya merupakan kejahatan yang dilakukan oleh seseorang dengan menggunakan fasilitas internet yang bersifat melintasi batas negara, dilakukan secara ilegal, menggunakan peralatan komputer dan internet, menyebabkan kerugian, dan sulit dibuktikan secara hukum. Kejahatan dunia maya (cybercrime) ini menjadi ancaman tersendiri bagi keamanan sebuah website.

4.3 Pengertian *Vulnerability*

Vulnerability adalah suatu titik lemah atau suatu kelemahan dalam suatu prosedur keamanan kontrol administratif, kontrol *internet* dan lain-lain sebagai yang dapat dieksploitasi melalui suatu trik untuk memperoleh akses yang tidak *valid* terhadap informasi atau untuk mengganggu proses secara kritis.

Vulnerability atau celah keamanan adalah suatu kelemahan yang mengancam nilai integrity *convidentiality* dan *avaibility* dari suatu *asset*.

Pada suatu sistem, *vulnerability* merupakan suatu keadaan, kelemahan karena tidak adanya prosedur keamanan, teknik, atau secara fisik atau suatu kendali yang lain yang dapat dieksploitasi oleh suatu *threat.* Dengan demikian maka dalam setiap jaringan berpotensi memiliki *vulnerability* demikian juga terhadap suatu *website,* tidak ada jaminan bahwa suatu website benar-benar *invunerability*.

Vulnerability merupakan sebuah cacat atau kelemahan dalam desain, inplementasi, atau operasi dan manajemen pada sistem jaringan aplikasi, atau protokol komputer yang dapat dimanfaatkan untuk melanggar kebijakan keamanan sistem.

Pada *website*, *vulnerability* terjadi dikarenakan adanya input yang belum divalidasi atau kesalahan pada validasi input.

4.4 SQL (Structured Query Language)

SQL merupakan singkatan *Structured Query Language*. Di dalam dunia *Database* istilah *query* dapat diartikan "Permintaan Data". SQL merupakan bahasa tingkat empat yang berfungsi menampilkan hasil atau melakukan sesuatu pada data yang tidak diinginkan. *SQL Query* terdiri dari satu atau beberapa *SQL Statements* yang secara efektif mengintruksikan tugas yang harus dilakukan oleh *server database*. Pada *SQL statements* juga dikenal *regular expressions* (*regex*) yang merupakan pola dari karakter-karakter yang sama atau gagal untuk disamakan, berbarengan dengan karakter lain di dalam teks (Andrew Watt, 2005). *Regular expressions* ini juga dapat digunakan untuk melakukan proses pengujian *vulnerability sql injection* pada suatu *website*.

4.5 SQL Injection

SQL Injection adalah sebuah teknik *hacking* yang bertujuan untuk menyusup ke dalam sistem sebuah *website* untuk mengetahui isi *database,* ataupun informasi-informasi yang terdapat di situs tersebut.

Teknik ini dapat terjadi dikarenakan adanya kode-kode program yang lemah dan adanya proteksi yang kurang aman dan lemah dari pengelola *website* tersebut atau yang disebut

administrator. *SQL Injection* merupakan salah satu teknik *hacking* yang hanya memerlukan *port* 80 meskipun sering dilakukan *patch* pada *server*. Kode SQL tersebut diinjeksikan pada *query SQL* sebagai input melalui halaman *website*. Hal ini memungkinkan *intruder* untuk memasuki halaman *web administrator* tanpa melakukan *scan port* yang terbuka, tidak terdeksi oleh *firewall* dan bahkan tidak menggunakan *tool* apapun.

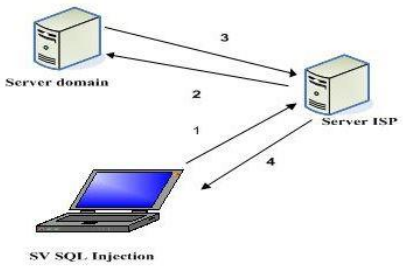
SQL Injection sendiri diperkirakan muncul atau telah ada semenjak *database* SQL dikoneksikan pertama kali dengan aplikasi *web* dan mulai menjadi perhatian publik semenjak tahun 1998. Semenjak itu para peneliti mulai membangun dan mengembangkan teknik untuk mengeksploitasi *SQL injection*.

5. PERANCANGAN SISTEM

5.1 Perancangan Sistem

Perencanaan antarmuka mengandung penjelasan tentang desain dan implementasi sistem yang digunakan dalam sistem yang kita buat.

Secara umum Alir perancangan sistem atau desain sistem yang akan dibangun :



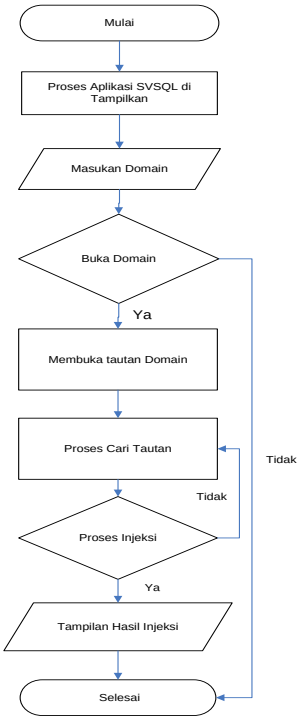
Gambar 3.1 Perancangan Sistem Secara Global

Dari gambar 3.1 dapat diketahui :

1. Aplikasi *SV SQL Injection* yang dibangun melakukan *request* ke *server* ISP berupa tautan yang diinputkan.
2. *Server* ISP meneruskan request dari *SV SQL Injection* ke *server* domain
3. *Server* domain merespon kembali ke *server* ISP berupa hasil injeksi HTML
4. Server ISP meneruskan respon hasil injeksi dari *server* domain ke Aplikasi *SV SQL Injection*.
5. Aplikasi *SV SQL Injection* memberi report hasil keseluruhan

5.2 Diagram Alir Aplikasi

Berikut diagram rancangan proses alir aplikasi SV SQL *Injection* yang akan dibangun:



Gambar 3.2 Diagram Alir Proses Aplikasi

6. IMPLEMENTASI APLIKASI

Pada bab berikut ini akan dijelaskan mengenai implementasi aplikasi yang disusun menurut desain *form* yang telah dibuat dan bagaimana cara menggunakan aplikasi pendeteksi *vulnerability sql injection* ini untuk sebuah *website.*

6.1 Antar Muka

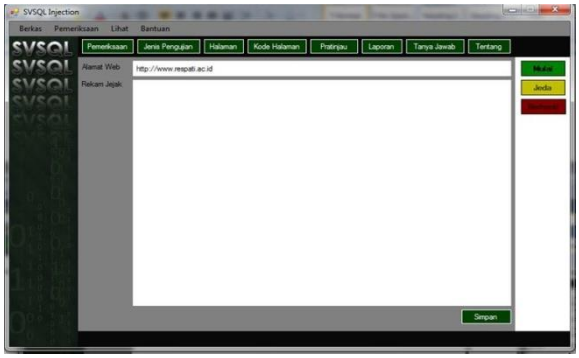
6.2 Tampilan Jendela Utama Aplikasi

Jendela utama merupakan tampilan yang muncul pertama kali pada saat aplikasi dijalankan. Pada tampilan ini *user* dapat melihat semua menu, *command button* dan *toolbar* yang ada pada aplikasi tersebut.

Gambar 4.1 Tampilan Jendela Utama

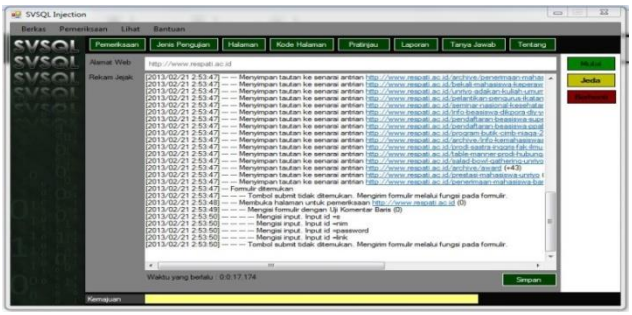
6.3 Tampilan Pemeriksaan

Tampilan Pemeriksaan yang juga tampil saat pertama kali aplikasi dijalankan.



Gambar 4.6 Tampilan Jendela Pemeriksaan

Setelah di klik maka akan muncul tampilan proses Rekam Jejak pengujian sebagai berikut:



Gambar 4.7 Tampilan Jendela Proses Rekam Jejak Berjalan

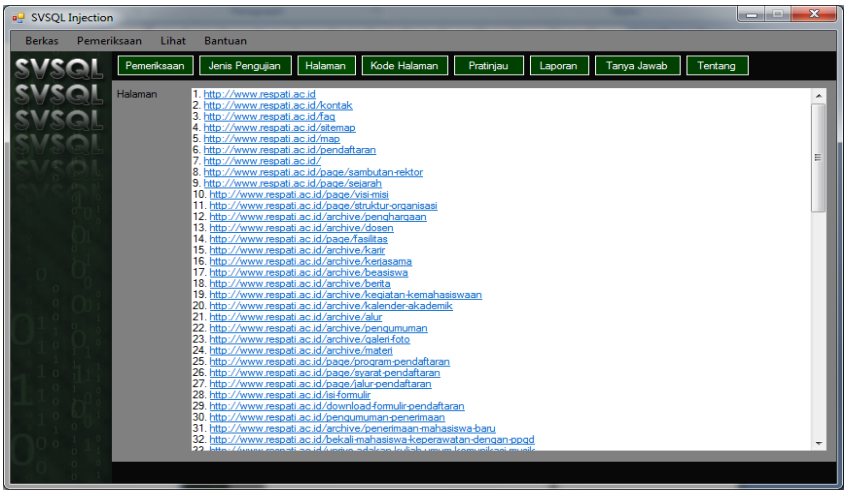
6.4 Tampilan Jenis Pengujian

Tampilan dari *Toolbar* Jenis Pengujian merupakan tampilan dimana *user* dapat memilih pada *checkbox* yang ada jenis pengujian apa saja yang akan diujikan pada alamat web yang diinputkan pada jendela Pemeriksaan.

Gambar 4.9 Tampilan Jendela Pilihan Jenis Pengujian

6.5 Tampilan Halaman

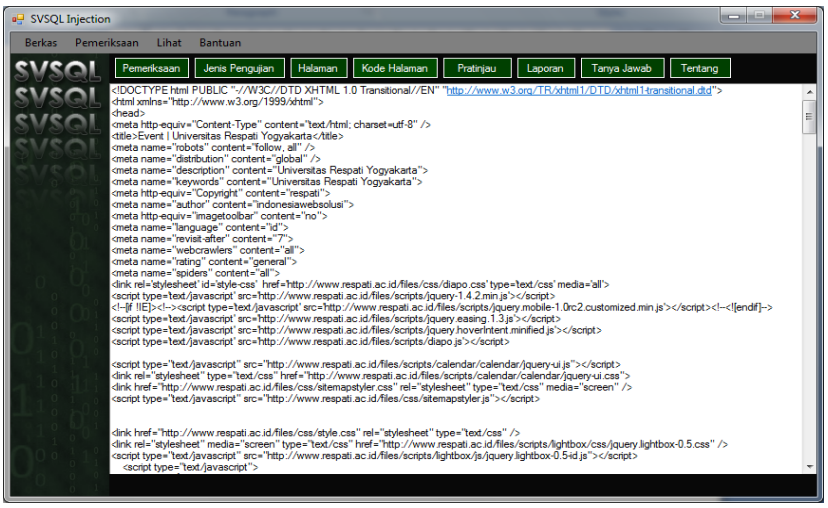
Pada jendela tampilan Halaman, tautan hasil *crawling* dari url/alamat web utama ditampilkan.



Gambar 4.10 Tampilan Jendela Halaman

6.6 Tampilan Kode Halaman

Pada jendela Kode Halaman selama proses pengujian akan ditampilkan kode HTML atau script *website* dari halaman yang sedang dianalisa.



Gambar 4.11Tampilan Jendela Kode Halaman

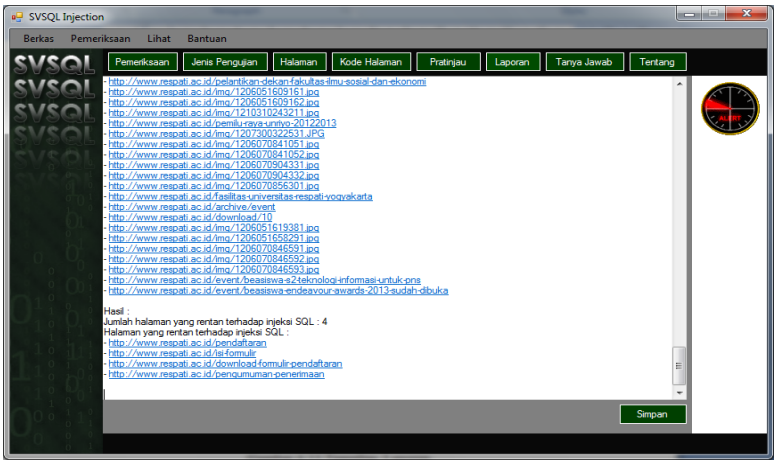
6.7 Tampilan Pratinjau

Pada jendela Pratinjau akan ditampilkan halaman *website* yang sedang dalam proses analisa.

Gambar 4.12 Tampilan Jendela Pratinjau

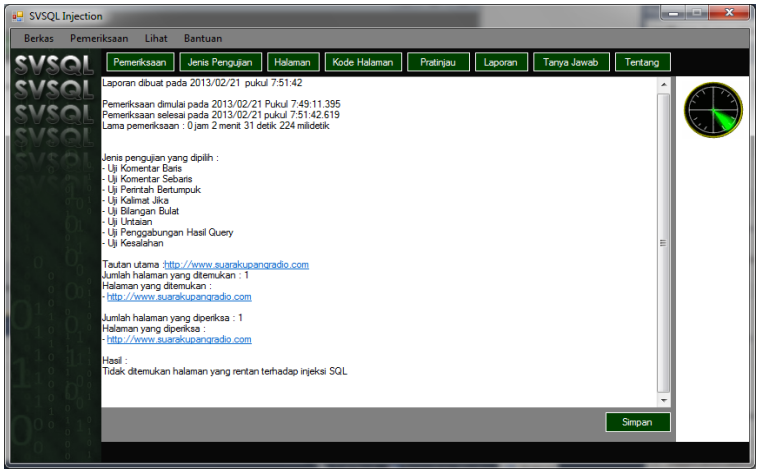
6.8 Tampilan Laporan

Toolbar Laporan berfungsi untuk menampilkan hasil akhir dari proses pengujian pada alamat web yang diuji.



Gambar 4.13 Tampilan Jendela Laporan Halaman Web Yang *Vulnerable*

Bilamana tidak ditemukan suatu halaman web yang *vulnerable* maka pada tampilan *report* dapat terlihat sebagaimana pada gambar di bawah ini:



Gambar 4.14 Tampilan Jendela Laporan Halaman Web Tidak *Vulnerable*

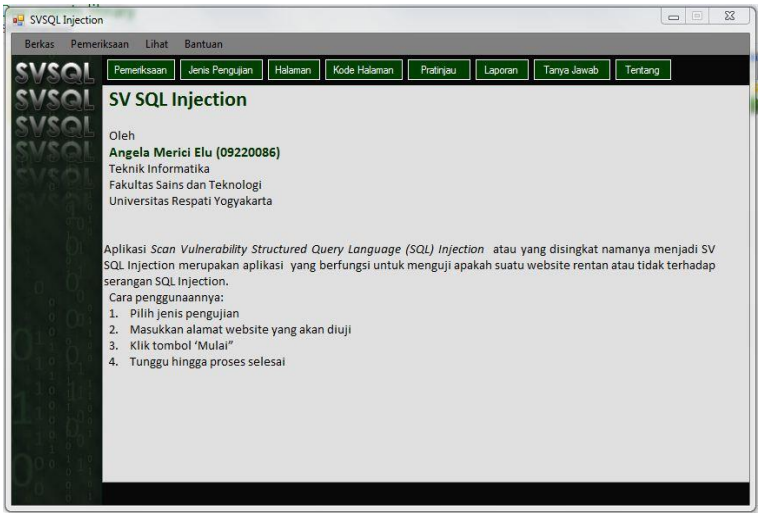
6.9. Tampilan Tanya Jawab

Pada tampilan jendela Tanya Jawab berisi informasi ringkas mengenai keamanan *Website* dan tentang *SQL Injection*.

Gambar 4.15 Tampilan Jendela Tanya Jawab

6.10. Tampilan Tentang

Pada tampilan Tentang hanya sekedar menampilkan identitas pembuat Aplikasi.



Gambar 4.16 Tampilan Jendela Tentang

7. ANALISA APLIKASI

Aplikasi pendeteksi *SQL Injection* memang telah diuji secara sistem namun demikian karena aplikasi tersebut juga akan memiliki interaksi dengan *user* maka dibuat kuisioner untuk mengetahui respon *user* setelah mencoba aplikasi tersebut untuk menentukan layak tidaknya aplikasi di *'release'* kepada *user*. Untuk itu perlu ditetapkan standar nilai prosentase minimal persetujuan dari *user* sebagai respon persetujuan atas setiap pertanyaan yang diajukan dalam kuisioner. Dalam hal ini penulis menetapkan nilai standar sebesar 75% dari pernyataan persetujuan responden.

Kuisioner ini mengambil *sample* sebanyak 20 responden terdiri dari 3 orang pengelola *website,* 6 orang mahasiswa, dan 11 orang dari masyarakat umum. Setiap responden diajukan 5 pertanyaan dimana untuk setiap kolom pernyataan sikap diberikan nilai 5 (lima). Nilai 1 responden = 5 point untuk setiap pertanyaan

Perhitungan: pada pertanyaan pertama kolom pertama ada 14 pernyataan sangat setuju sehingga nilai dari pernyataan ini adalah 14 x 5 = 70.

Untuk mengetahui nilai prosentase dari setiap pertanyaan dan kolom pilihan dapat dilihat pada tabel berikut ini:

Dari tabel di bawah dapat diketahui bahwa prosentase responden yang menyatakan sangat setuju ada (200% : 500%) x 100%=40% dan yang menyataan setuju ada (290% : 500%) x 100%=58%. dari kedua pernyataan responden tersebut maka diperoleh total sebanyak 40%+58%=98% pernyaan persetujuan sehingga dapat disimpulkan bahwa aplikasi tersebut sudah layak untuk di'*release*'.

Hasil dari kuisioner tersebut setelah dirangkum dan dianalisa dapat dilihat pada tabel berikut ini:

Tabel 4.3 Tabel Hasil Perhitungan Kuisioner

No	Kriteria	SS	S	KS	TS	Total
1	Aplikasi mudah digunakan *user*	70	30	0	0	100
2	Tampilan aplikasi *user friendly*	45	50	5	0	100
3	*User* dapat melihat proses pengujian aplikasi	35	65	0	0	100
4	Fungsi *menu* dan *tool* mudah dipahami	15	80	5	0	100
5	Dengan adanya aplikasi ini memudahkan user untuk mengetahui celah website terhadap *sql injection*	35	65	0	0	100
Total		200	290	10	0	500
Prosentase		40 %	58%	2%	0%	100%

8. KESIMPULAN DAN SARAN

Dari hasil pengujian aplikasi terhadap beberapa alamat website secara acak maka penulis dapat menganalisa kemampuan aplikasi dan membuat kesimpulan dan saran sebagai berikut:

8.1 Kesimpulan

1. Dari aplikasi yang dibuat, untuk mengetahui apakah suatu *website vulnerable* atau tidak terhadap serangan *SQL Injection* yakni dengan cara melihat hasil pengujian pada *website* tersebut, bilamana dari hasil pengujian dengan aplikasi ditemukan laporan ada tautan yang *vulnerable,* maka *website* tersebut *vulnerable* terhadap serangan *SQL Injection*.
2. Aplikasi yang telah dibuat dapat mendeteksi *vulnerability* sebuah *website* terhadap serangan *SQL Injection* dengan menggunakan metode *POST* dimana hasil pengujian ditampilkan pada laporan yang dapat disimpan dalam format .txt.

8.2 Saran

1. Untuk melakukan pengujian terhadap suatu *website* yang memiliki banyak sub tautan, maka aplikasi ini memerlukan suatu koneksi terhadap jaringan *internet* yang stabil dan cepat supaya pengujian dapat dilakukan dengan cepat. Hal tersebut disebabkan karena aplikasi akan membuka *website* yang diuji dan melakukan pengujian di saat yang bersamaan.
2. Untuk tampilan dari aplikasi, pada antar muka (*interface*) aplikasi serta fungsi menu aplikasi, dari hasil respon kuisioner ada yang menyatakan kurang setuju. Hal ini disebabkan karena pada saat melakukan presentasi aplikasi untuk kuisioner, bahasa yang digunakan dalam aplikasi ialah bahasa Inggris, untuk itu dalam pambaharuan aplikasi yang dilakukan, bahasa antar muka aplikasi

telah diubah menjadi bahasa Indonesia sehingga *user* lebih mudah memahami fungsi dan cara kerja pada menu dan *tools* yang ada pada aplikasi.

DAFTAR PUSTAKA

Andreas, Jason. 2011. *The Basic of Information Security: Understanding the Fundamental of InfoSec in Theory And Practice*. Elsevier. Waltham. USA

Clarke, Justin. 2009. *SQL Injection Attack and Defense*. Elsevier Inc. Burlinton, USA

Ec-Council. 2010. *Computer Forensics: Investigating Network Intrusions And Cybercrime*. Cengage Learning Inc. New York. USA

Gourley, David. Brian Totti. 2002. *HTTP:The Definitive Guide*. O'Reilly Media Inc. California. USA

Hope, Paco Hope and Ben Walther. 2009. *Web Security Testing Cookbook*. O'Reilly Media Inc. California. USA

Jogiyanto, HM. 1993. *Analisis dan Desain Sistem Informasi Pendekatan Terstruktur*. Andi Offset. Yogyakarta. Indonesia

Kiddo. 2010. *Hacking Website*. Mediakita. Jakarta

Kizza, Joseph Migga. 2009. *Guide to Computer Network Security*. Springer-Verlag London Limited. New York. USA

Komputer, Wahana. 2010. *Membangun Website Tanpa Modal*. Penerbit Andi. Yogyakarta. Indonesia

Komputer, Wahana. 1998. *Desain Web Dengan Microsoft Frontpage 97*. Penerbit Andi. Yogyakarta

Martin, Eddie. 2006. *Computer Jargon: Dictionary and Thesaurus*. Specialist Computing Ltd. United Kingdom

Munir, Rinaldi. 2009. *Matematika Diskrit*. Penerbit Informatika Bandung. Bandung. Indonesia

O'Regan, Gerard. 2012. *A Brief History of Computing*. Springer-Verlag London Limited. New York. USA

Raharjo, Suwanto dan Jazi Eko Istiyanto. 2003. *Keamanan Akses Ke PostgreSQL Melalui PHP (menggunakan Apache WEB Server pada GNU/Linux)*. Penerbit Andi. Yogyakarta

Sanusi, Muzammil. 2008. *Teknik Membobol Data Dan Password*. PT Elex Media Komputindo. Jakarta

Shiflett, Chris. 2003. *HTTP Developer's Handbook*. Sams Publishing. Amerika Serikat

Simarmata, Janner. 2010. *Rekayasa Web*. Penerbit Andi. Yogyakarta. Indonesia

Sismoro, Heri. 2005. *Pengantar Logika Informatika, Algoritma dan Pemrograman Komputer*. Andi Offset. Yogyakarta. Indonesia

Stiawan, Deris. 2005. *Sistem Keamanan Komputer*. PT Elexmedia Komputindo. Jakarta. Indonesia

Sulianta, Feri. 2009. *Teknik Mengoptimalkan Password*. PT Elexmedia Komputindo. Jakarta. Indonesia.

Stuttard, Dafydd and Marcus Pinto. 2008. *The Web Application Hacker's Handbook: Discovering And Exploiting Security Flaws*. Wiley Publishing Inc. Indianapolis. Canada

Sunarto. 2006. *Teknologi Informasi Dan Komunikasi*. PT. Grasindo. Jakarta

Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Penerbit Andi. Yogyakarta

Tsihrintzis George A, et.al. 2010. *Multimedia Services In Intelligent Environment."Software Development Challenges And Solutions".* Springer-Verlag, Berlin. Jerman.

Watt, Andrew. 2005. *Beginning Regular Expressions*. John Wiley&Sons. Canada

Zam, Efvy. 2011. *Buku Sakti Hacker*. Mediakita. Jakarta. Indonesia