**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

**Nama** : **Yustiana Tri Wahyuni**

**NRP** : **5109 100 075**

Dosen Wali : Ary Mazharuddin Shiddiqi, S.Kom, M.Comp.Sc

1. **JUDUL TUGAS AKHIR**

Rancang Bangun Add-ons Deteksi XSS pada Peramban Mozila Firefox

1. **LATAR BELAKANG**

Saat ini aplikasi web dinamis memainkan peran penting dalam menyediakan manipulasi sumber daya dan interaksi antara klien dan server. Fitur yang saat ini berkembang didukung oleh browser seperti tag HTML, *script*, *hyperlink* dan fungsi-fungsi lainnya telah meningkatkan peluang bisnis yang ada serta menyediakan interaktivitas yang lebih beranekaragam dalam layanan berbasis web, seperti e-commerce, internet banking, jejaring sosial, blog, forum dan lain sebagainya.

Di sisi lain, fitur tersebut juga meningkatkan ancaman berbahaya yang memungkinkan eksploitasi. Menurut Wasserman dan Su[1], sebagian besar bahasa pemrograman web tidak menyediakan transfer data klien yang aman. Tidak adanya prosedur transfer data yang aman ini dapat menyebabkan salah satu serangan yang paling sering terjadi dalam aplikasi web, yaitu *Cross-Site Scripting*(XSS). XSS merupakan salah satu jenis serangan injeksi code(*code injection attack*)*.*XSS dilakukan oleh penyerang dengan cara memasukkan kode HTML atau *client script* code lainnya ke suatu situs dimana serangan seolah-olah datang dari situs tersebut. XSS dapat diklasifikasikan menjadi tiga jenis yaitu *Persistent, Reflective,* dan *DOM* based.

1. *Persistent* XSS

Celah *Persistent* XSS muncul ketika sebuah program web server menyimpan input pengguna dalam data persisten seperti database di mana data yang tersimpan tersebut diakses oleh program dan dapat dilihat oleh pengguna yang berbeda.

1. *Reflective* XSS

Pada *Reflective* XSS, celah muncul ketika program server merujuk pada suatu data yang tidak diperiksa terlebih dahulu dari parameter HTTP *request* pada suatu halaman web yang langsung dikirim ke pengguna.

1. DOM *based* XSS

Celah DOM *based* XSS muncul ketika pada sisi klien terdapat *script* yang merujuk pada input pengguna secara dinamis yang diperoleh dari struktur DOM(*Document Object Model*) tanpa validasi yang tepat sehingga *script* berbahaya dapat disisipkan kedalam DOM tersebut.

Akibat serangan ini antara lain penyerang dapat mem-*bypass* keamanan di sisi klien, mendapatkan informasi sensitive, atau menyimpan aplikasi berbahaya. Penelitian dari OWASP[2] menunjukkan bahwa XSS masih merupakan celah yang paling sering terjadi dalam aplikasi web.

Pada tugas akhir ini, dirancang dan diimplementasikan add-ons pada perambah *Mozilla Firefox* untuk mendeteksi XSS.Tahap pertama untuk melakukan deteksi XSS pada situs yaitu dengan melakukan *decoding* halaman web, dimana halaman web diubah ke dalam bentuk tertentu. Kemudian dari hasil *decoding* halaman web tersebut dilakukan tahap kedua kedua yakni ekstraksi fitur. Fitur-fitur yang akan digunakan untuk proses klasifikasi antara lain panjang url, jumlah domain, karakter spesial, keyword, skema tags dan cookie. Dari proses ekstraksi fitur pada tahap kedua dihasilkan vektor fitur, dimana dari hasil vektor fitur ini akan dilakukan proses deteksi tahap terakhir yakni proses klasifikasi untuk menentukan web terdeteksi XSS atau tidak.

1. **RUMUSAN MASALAH**

Berdasarkan latar belakang yang telah diuraikan diatas, terdapat beberapa permasalahan yang dapat dirumuskan sebagai berikut:

1. Bagaimana server XSS melakukan *decoding* pada halaman web?
2. Bagaimana server XSS melakukan ekstraksi fitur dari hasil decoding halaman web?
3. Fitur-fitur apa saja yang digunakan untuk mengklasifikasi celah *Cross Site Scripting*(XSS) pada halaman web?
4. Bagaimana server XSS melakukan klasifikasi dengan fitur-fitur yang didapatkan?
5. Bagaimana *add-ons* dapat berjalan dengan baik pada *Mozilla Firefox*?
6. **BATASAN MASALAH**

Batasan masalah dari aplikasi yang akan dibangun adalah sebagai berikut:

* + - 1. Bahasa pemrograman yang digunakan yaitu Java.
      2. Aplikasi berjalan pada peramban Mozilla Firefox.
      3. Aplikasi berjalan pada peramban berbasis dekstop.
      4. Dataset yang digunakan untuk melakukan uji coba berasal dari XSSed database dan Dmoz database.

1. **TUJUAN DAN MANFAAT TUGAS AKHIR**

Manfaat dari pembuatan tugas akhir ini adalah untuk memberikan keamanan kepada pengguna dari serangan XSS pada suatu situs web.Aplikasi ini akan selalu mendeteksi halaman web setiap kali pengguna membuka suatu halaman web baru. Sehingga dapat memberikan peringatan dan mencegah terjadinya serangan XSS.

Adapun tujuan yang diinginkan adalah membuat mekanisme pengamanan web dari serangan XSS*,* sehingga dapat menyelamatkan data pengguna.

1. **DASAR TEORI**
   1. **Web Service**

*Web service* adalah sebuah sistem yang dirancang untuk dapat mendukung interaksi antar mesin-mesin pada suatu jaringan dengan menggunakan format data standar XML yang tidak berbantung pada sistem operasi maupun bahasa pemrograman. *Web service* disediakan oleh suatu *website* sebagai fasilitas untuk menyediakan layanan kepada sistem lain, sehingga melalui layanan-layanan yang dibuat, sistem lain dapat berinteraksi dengan sistem tersebut.

Pada gambar 2 di bawah, ditunjukkan arsitektur dari web service yang terdiri dari *processes , description, invocation,* dan *transport.*



Gambar 1. Arsitektur *web service*

* *Prosesses*: Bagian dari arsitektur yang umumnya melibatkan lebih dari satu layanan web. Sebagai contoh, penemuan suatu layanan tertentu di antara sekian banyak layanan web yang tersedia.
* *Description*: Salah satu fitur yang disediakan oleh*web services* adalah *self-describing*. Web servicedapatmenggambarkan dirinya sendiri dan memberitahu operasi apa saja yang disediakan dan bagaimana cara untuk meminta operasi tersebut, yang biasanya ditangani oleh Bahasa Deskripsi Layanan Web(WSDL).
* *Invocation*: Meminjam Web Service dengan melibatkan pesan melewati antara klien dan server. SOAP(Simple Object Access Protocol) menentukan bagaimana kita harus memformat permintaan ke server, dan bagaimana server harus memformat responnya.
* *Transport*: Semua pesan harus dikirimkan antara server dan klien. Protokol yang biasanya digunakan untuk bagian ini adalah HTTP(HyperText Transfer Protocol), protokol yang sama digunakan untuk mengakses halaman web konvensional di Internet.
  1. **Mozilla Firefox**
     1. **Deskripsi Mozilla Firefox**

Mozilla Firefox adalah sebuah peramban web lintas platform yang bersifat gratis dan open source. Mozilla Firefox dikembangkan oleh Mozilla Foundation beserta ratusan sukarelawan[6]. Mozilla Firefox versi 1.0 pertama kali dirilis pada tanggal 9 November 2004 dan mendapat tanggapan bagus dari pihak media. Hingga 24 November 2004 tercatat lebih dari 6 juta orang men*download* Mozilla Firefox versi 1.0 ini[6]. Mozilla Firefox sendiri dapat dijalankan pada berbagai macam sistem operasi seperti Microsoft Windows, Linux, Mac OS dan FreeBSD.

* + 1. **Add-ons pada Mozilla Firefox**

Add-ons adalah ekstensi tambahan yang disediakan oleh Mozillah Foundation untuk memberikan para pengembang kebebasan untuk membangun Firefox. Dengan adanya Add-ons dimungkinkan adanya penambahan fitur berupa ekstensi tambahan diluar Mozilla. Bahasa yang digunakan untuk pengembangan Add-ons yaitu JavaScript dengan menggunakan standar CSS.

1. **RINGKASAN TUGAS AKHIR**

Pada tugas akhir ini akan dibangun sebuah aplikasi yang secara umum memiliki fungsi untuk mendeteksi halaman web yang diduga terserang XSS. Ancaman serangan XSS pada suatu halaman web masih menjadi hal berbahaya. Oleh karena itu aplikasi berbasis web yang akan dibangun ini merupakan terobosan untuk mencegah terjadinya serangan XSS pada halaman web.

**server xss**

**web server**

3. request web page

4. respond HTML content

**add-ons**

1. take URL

**klien**

2. request URL

5. send detection result

6. send result

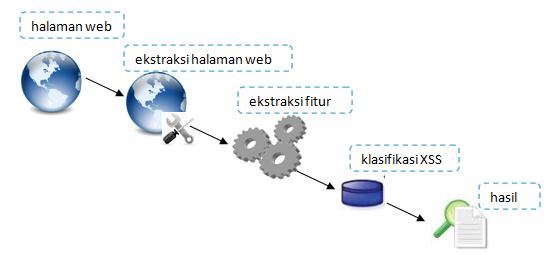
7. access web page

Gambar 2. Arsitektur jaringan

Dari gambar 3 diatas dapat dijelaskan sebagai berikut :

1. Aplikasi ini bekerja pada saat klien membuka suatu halaman web. Add-ons bekerja dengan mengambil URL halaman web yang akan diakses.
2. Kemudian *add-ons* mengirim data URL yang dibuka klien ke server XSS.
3. Server XSS mengirimkan permintaan ke web server untuk mengunduh dokumen web yang ingin diakses klien.
4. Lalu web server mengirimkan dokumen web yang diminta oleh server XSS kemudian server XSS melakukan klasifikasi terhadap dokumen web tersebut.
5. Server XSS mengirimkan kepada *add-ons* hasil klasifikasi berupa status halaman web yang akan dibuka.
6. Selanjutnya *add-ons* memutuskan tindakan apa yang harus dilakukan oleh peramban. Jika situs terdeteksi tidak terkena serangan XSS , maka *add-ons* akan mengijinkan klien untuk membuka halaman web. Sebaliknya, jika situs terdeteksi terkena serangan XSS, maka *add-ons* akan memberikan pencegahan berupa peringatan kepada klien untuk membuka halaman web tersebut.
7. Klien dapat memilih untuk melakukan akses ke halaman web yang dituju atau tidak.

Aplikasi ini memiliki tiga proses, yaitu proses *decoding* halaman web, ekstraksi fitur dan pengklasifikasian halaman seperti yang ditunjukkan pada gambar 4. Pada proses *decoding* halaman web, halaman web yang akan dituju diterjemahkan ke dalam format tertentu sehingga teks dapat dimengerti dan dimungkinkan untuk melakukan analisis dan ekstraksi fitur.



Gambar 3. Metode deteksi XSS

Proses selanjutnya dilakukan ekstraksi fitur yang diambil berdasarkan skema HTML/Javascript yang mencurigakan. Fitur-fitur yang akan digunakan pada aplikasi ini antara lain:

1. Panjang URL
2. Jumlah domain pada URL
3. Spesial karakter yang duplikat
4. Keyword
5. Skema HTML/JavaScript(Skema Tags)
6. Cookie domain

Terakhir dilakukan proses pengklasifikasian terhadap fitur yang didapatkan pada proses sebelumnya. Pada tahap terakhir ini akan didapatkan hasil berupa penanda XSS atau Non XSS.

Add-ons akan meneruskan permintaan untuk mengakses situs apabila situs yang dibuka dianggap sebagai situs Non XSS. Sebaliknya, add-ons akan memberikan pencegahan berupa peringatan apabila situs yang dibuka dianggap sebagai situs XSS. Klien dapat memilih untuk meninggalkan halaman atau tetap membuka halaman dengan resiko web terkena serangan XSS.

1. **METODOLOGI**
2. Penyusunan Proposal Tugas Akhir

Pada tahap ini penulis menyusun proposal tugas akhir sebagai langkah awal dalam pengerjaan tugas akhir. Pada proposal ini penulis menggagas penulisan tugas akhir untuk mengimplementasikan add-ons pendeteksi XSS ini pada peramban Mozilla Firefox.

1. Studi Literatur

Pada tahapan ini akan dilakukan studi literatur mengenai caradan kakas bantu yang akan digunakan untuk pembuatan aplikasi.

1. Implementasi

Implementasi merupakan tahap pembangunan aplikasi yang dilakukan dalam beberapa tahapan, yaitu :

* + - * 1. Pembuatan fungsi *decoding* halaman web*.*
        2. Pembuatan fungsi ekstraksi fitur.
        3. Pembuatan fungsi untuk klasifikasi XSS.
        4. Pembangunan *web service*.
        5. Perancangan add-ons yang berkomunikasi dengan *web service*
        6. Instalasi dan integrasi add-ons pada peramban.

1. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah dibuat.Tujuan uji coba perangkat lunak adalah untuk menemukan kesalahan-kesalahan (*bug*) sedini mungkin sehingga dapat diperbaiki sesegera mungkin.

1. Penyusunan Buku Tugas Akhir

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perancangan dan pembuatan aplikasi yang telah dibuat. Secara garis besar, buku laporan tugas akhir ini terdiri atas beberapa bagian yaitu:

1. Pendahuluan
   1. Latar Belakang
   2. Permasalahan
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka
7. **JADWAL KEGIATAN TUGAS AKHIR**

Tugas akhir ini diharapkan bisa dikerjakan menurut jadwal sebagai berikut:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Kegiatan | Bulan | | | | | | |
| Maret | | April | | Mei | | Juni |
| 1 | Penyusunan Proposal Tugas Akhir |  | |  | |  | |  |
| 2 | Studi Literatur |  |  |  | |  | |  |
| 3 | Implementasi |  | |  |  |  | |  |
| 4 | Pengujian dan Evaluasi |  | |  | |  | |  |
| 5 | Penyusunan Buku Tugas Akhir |  | |  | |  |  |  |

1. **DAFTAR PUSTAKA**
2. Waserman,G. e Su Z, 2008. *Static Detection of Cross-Site Scripting Vulnerabilities Investigation,* 30th International Conference on Software Engineering
3. OWASP,Foundation.“OWASP Testing Guide”, 2008. V3.0. <http://www.owasp.org/images/5/56/OWASP\_Testing\_Guide\_v3.pdf> (Diakses: 30 Oktober 2012).
4. Nadji, Y., Saxena, P., Song, D. “Document Structure Integrity: A Robust Basis for Cross-Site Scripting Defense”. In: 16th Annual Network & Distributed System Security Symposium, NDSS Symposium, 2009.
5. Kirda, E., Kruegel, C., Vigna, G. e Jovanovic, N. “Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks”. In: 21th ACM Symposium on Applied Computing, ACM, 2006.
6. Javascript.2012 <https://developer.mozilla.org/id/docs/JavaScript> (Diakses : 30 Oktober 2012)
7. Wikipedia. Mozilla Firefox. < http://id.wikipedia.org/wiki/Mozilla\_Firefox> (Diakses : 2 Februari 2013)