**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# **IDENTITAS PENGUSUL**

**Nama : Virky Wiradiatma Saputra**

**NRP : 5109 100 190**

**Dosen Wali : Dwi Sunaryo, S.Kom., M.T.**

1. **JUDUL TUGAS AKHIR**

Rancang Bangun Add-ons Deteksi Web Phishing pada Mozilla Firefox dengan Analisa Konten dan Tautan Situs

1. **LATAR BELAKANG**

Penetrasi pengguna internet semakin meningkat dari tahun ke tahun. Berbagai transaksi dan/atau aktifitas yang semula dilakukan secara *offline* mulai dapat diakses dari mana saja secara *online*. Disisi lain, kemudahan ini dapat menjadi celah keamanan bagi para pengguna internet yang masih awam mengenai keamanan bertransaksi di dunia maya. Secara tidak sadar, mereka dapat memberikan informasi rahasia seperti data pribadi, kata sandi surat elektronik, bahkan informasi finansial seperti data kartu kredit dan *online banking*. Celah keamanan ini lebih populer dengan istilah *web phishing*.

*Web Phishing* merupakan teknik mengelabui korbannya dengan membuat korban seolah-olah sedang mengakses halaman situs dari sumber yang sah. Tampilan situs akan dibuat semirip mungkin dengan aslinya agar pengguna awam yakin sedang berada pada situs yang benar. Jika korban berhasil dikelabui dan memasukkan informasi yang diminta, penipu dapat dengan mudah menggunakan informasi tersebut pada situs yang sah.

*Web phishing* kebanyakan dilakukan pada situs transaksi finansial dan perbankan online. Situs populer yang menjadi sasaran empuk di antara adalah PayPal dan eBay [2]. Di dalam negeri, umumnya yang menjadi sasaran adalah situs *online banking* dari bank-bank ternama.

Berdasarkan data yang diperoleh dari Anti Phishing Working Group, pada kuartal pertama tahun 2012 tercatat 56.859 situs phising terdeteksi pada bulan Februari 2012 yang merupakan pencapaian tertinggi hingga saat itu. Jumlah target produk yang menjadi sasaran meningkat hingga 392 produk yang artinya meningkat 8 persen dari bulan Desember 2011. Situs yang menjadi target favorit phishing sendiri adalah situs yang berkaitan dengan layanan finansial. Hal ini dapat menyebabkan menurunkan kepercayaan pengguna bertransaksi melalui *e-commerce* yang sedang berkembang saat ini [6].

Untuk mengantisipasi serangan phishing, beberapa perusahaan mencoba mengembangkan *anti-phishing toolbar*, diantara Netcraft dan Google, Inc [2]. Namun hal ini dirasa kurang cukup karena toolbar ini tidak mampu mendeteksi situs *phishing* yang benar-benar baru, sehingga penggunanya masih mungkin terjebak sebelum situs phishing tersebut masuk dalam daftar hitam.

Pada tugas akhir ini, dirancang dan diimplementasikan add-ons pada perambah *Mozilla Firefox* untuk mendeteksi situs *web phishing*. Deteksi situs dilakukan dengan tahapan membuat struktur *DOM (Document Object Model) tree* dari laman situs lebih dulu. Kemudian setelah DOM tree terbentuk, dilakukan tahapan identity extraction yang menghasilkan Identity Set dan URL Identity. Hasil tersebut akan digunakan untuk melakukan *feature generation* yang menghasilkan *feature vector*. Tahapan terakhir yaitu melakukan pelabelan pada situs menggunakan metode klasifikasi SVM (Support Vector Machine) dari *feature vector.*

1. **RUMUSAN MASALAH**

Berdasarkan latar belakang yang telah diuraikan sebelumnya, terdapat beberapa permasalahan yang dapat dirumuskan, yaitu :

1. Bagaimana membentuk *DOM Tree* dari laman situs?
2. Bagaimana melakukan *identity extraction* dari *DOM Tree*?
3. Bagaimana mendapatkan *feature vector* dari hasil *identity extraction*?
4. Fitur apa saja yang perlu digunakan sebagai *feature vector*?
5. Bagaimana mengklasifikan situs dengan SVM menggunakan data *feature vector*?
6. Bagaimana membuat *add-ons* yang kompatibel dengan *Mozilla Firefox*?
7. **BATASAN MASALAH**

Dalam pengerjaan tugas akhir ini terdapat beberapa batasan, yaitu :

1. Add-ons hanya bisa digunakan pada peramban Mozilla Firefox
2. Add-ons hanya bisa digunakan pada peramban versi desktop
3. Hanya 11 *feature* yang digunakan sebagai *feature vector*.
4. Penghitungan tf-idf (term frequency – inverse document frequency) menggunakan kakas bantu Apache Lucene.
5. Pengecekan konten situs dilakukan oleh server yang terhubung dengan add-ons melalui *web-service*.
6. Klasifikasi situs menggunakan Support Vector Machine (SVM) dengan kakas bantu LibSVM.
7. Situs yang bisa dideteksi dengan baik hanya situs berbahasa Inggris.
8. **TUJUAN DAN MANFAAT TUGAS AKHIR**

Manfaat pembuatan tugas akhir ini adalah untuk menghindarkan pengguna dari pencurian identitas rahasia. Penambahan add-ons pada peramban berfungsi sebagai peringatan dini bagi penggunanya agar tidak melakukan pengisian data pribadinya.

Tujuan dari pembuatan tugas akhir ini adalah dapat menjadi add-ons bisa menganalisa suatu situs secara real-time tanpa perlu membuat daftar situs phishing terlebih dulu.

1. **DASAR TEORI**

**7.1. Document Object Model**

Document Object Model (DOM) merupakan salah satu antarmuka pemograman standar yang bisa digunakan untuk dokumen HTML dan XML. DOM dapat mendefinisikan struktur dari dokumen dan cara sebuah dokumen diakses dan dimanipulasi. Dengan memanfaatkan DOM, programmer dapat membuat dokumen, mengetahui strukturnya, serta menambahkan, mengubah, atau menghapus elemen dan kontennya. Segala hal yang terdapat pada HTML atau dokumen XML dapat diakses, diganti, dihapus, atau ditambahkan dengan menggunakan DOM.

Dokumen HTML yang diproses menggunakan DOM dapat membentuk sebuah tree. Pada gambar 1 diberikan contoh dokumen HTML sederhana yang akan diproses menggunakan DOM untuk mengetahui strukturnya. Hasil dari pemrosesan tersebut direpresentan dalam bentuk tree seperti pada gambar 2.

<TABLE>

<TBODY>

<TR>

<TD>Shady Grove</TD>

<TD>Aeolian</TD>

</TR>

<TR>

<TD>Over the River, Charlie</TD>

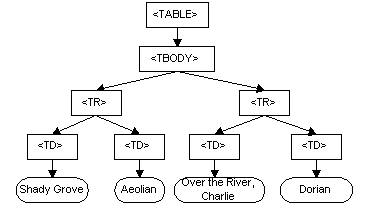
<TD>Dorian</TD>

</TR>

</TBODY>

</TABLE>

Gambar 1. Contoh halaman dengan kode HTML [3]



Gambar 2. Representasi DOM tree dari halaman HTML [3]

**7.2. Support Vector Machine**

Support Vector Machine pertama kali diperkenalkan pada tahun 1992 melalui paper pada konferensi COLT 1992 oleh Boser, Guyon, dan Vapnik. Kemudian pada tahun 1995 Cortes dan Vapnik memperkenalkna soft margin classifier [2]. SVM menjadi populer karena keberhasilannya dalam pengenalan pola pada tulisan tangan angka.

Support Vector Machine (SVM) merupakan salah satu permodelan supervised learning yang berasosiasi dengan algoritma machine learning yang dapat menganalisa data dan mengenali pola, yang kemudian digunakan untuk klasifikasi dan analisa regresi. Dasar dari SVM adalah mengembali sekumpulan data input dan memprediksi untuk tiap input, manakah output yang cocok diantara dua kelas yang ada.

**7.3. Mozilla Firefox Add-Ons**

**7.3.1. Tentang Mozilla Firefox**

Mozilla Firefox merupakan salah satu peramban gratis bersifat *open source* yang saat ini tersedia. Peramban ini dapat diinstalasi pada berbagai sistem operasi seperti Windows, Mac OS X, dan Linux. Pengembangan aplikasi sendiri dilakukan organisai nirlaba Mozilla Foundation.

Berdasarkan data yang diterbitkan dari berbagai sumber pada bulan Oktober 2012, Firefox merupakan peramban terbesar ketiga yang paling banyak digunakan. Pengguna Firefox diperkirakan sekitar 20%-24% dari seluruh pengguna browser dunia. Peramban ini sendiri paling sukses melakukan penetrasi di Indonesia yaitu sebesar 65% pengguna [5].

Saat ini pengembang firefox juga telah mempublikasikan peramban versi *mobile*. Peramban ini diberi nama Firefox for Mobile dengan kode nama Fennec. Aplikasi pertama dibuat untuk sistem operasi Nokia Maemo dan dirilis pada 28 Januari 2010. Perkembangan terakhir dari aplikasi ini sampai dengan versi 16.0 dan dapat diinstalasi pada perangkat dengan sistem operasi android.

**7.3.2. Fitur Add-Ons pada Firefox**

Sebagai peramban berbasis open source, Mozilla Foundation membuka diri bagi para pengembang untuk membantu membangun Firefox. Salah satunya melalui penambahan fitur pada peramban firefox menggunaan Add-Ons. Add-Ons merupakan ekstensi tambahan yang memungkinkan pengembangan aplikasi memberikan fitur-fitur tambahan pada peramban firefox diluar fitur standar yang sudah dimilikinya. Bahasa standar untuk pengembangan aplikasi ini menggunakan JavaScript dan desain menggunakan standar CSS.

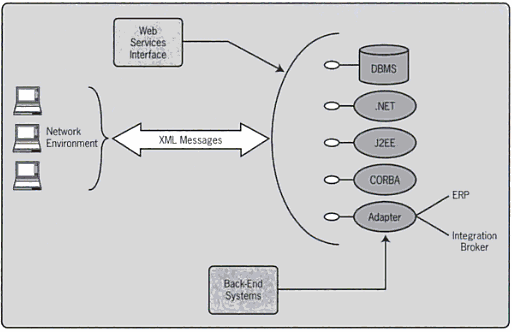
**7.4. Web Service**

**7.4.1. Tentang Web Service**

Web service merupakan suatu aplikasi Extensible Markup Language (XML) yang dipetakan ke dalam program, objek, basis data, atau suatu prses bisnis. Dengan menggunakan pesan yang dibentuk ke dalam format dokumen XML, suatu program dapat melakukukan permintaan ke sebuah Web Service melalui jaringan dan juga menerima balasan dalam bentuk dokumen XML juga.

Teknologi web service dapat diterapkan pada berbagai bidang. Web service dapat berjalan pada desktop menangani aplikasi pada sisi klien yang mengakses aplikasi internet, contohnya seperti sistem reservasi atau sistem pelacakan pengiriman barang. Web service juga bisa digunakan untuk integrasi Business-to-Business (B2B), yaitu dengan menghubungkan aplikasi-aplikasi yang dijalankan dari berbagai organisasi untuk keperluan pengolahan data [1].

Seperti ilustrasi pada gambar 3**,** web service berfungsi sebagai “pembungkus” yang terhubung langsung pada jaringan dan sebagai standar untur antarmuka perangkat lunak pada sistem back-end. Antarmuka web service akan menerima pesan dengan format XML standar dari jaringan, kemudian mengubahnya kedalam format yang dimengeti oleh perangkat lunak di sisi back-end. Implementasi web service bisa dibuat dengan menggunakan berbagai bahasa pemrograman, sistem operasi, dan sistem *middleware* [1].



Gambar 3. *Web service* dengan berbagai aplikasi dan bahasa pemrograman pada sistem back-end [1]

**7.4.2. Teknologi Standar pada Web Service**

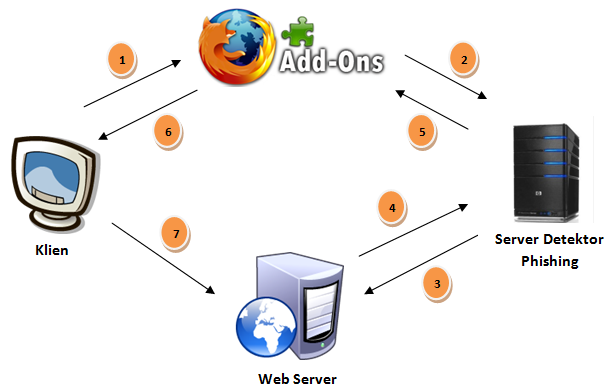
Web service memerlukan beberapa teknologi yang berkaitan dengan XML untuk mentrasmisikan dang mengubah format data pada suatu program atau basis data. Berikut adalah beberapa teknologi tersebut.

1. XML (Extensible Markup Language)  
   XML merupakan pondasi dasar pada pembuatan web service. XML digunakan untuk merepresikan data yang dikomunikasikan pada melalui jaringan. Standar XML telah dipublikasikan oleh World Wide Web Consortium (W3C)
2. WSDL (Web Service Description Language)  
   WSDL dibuat berdasarkan teknologi XML. Fungsi WSDL adalah mendefinisikan antarmuka web service, format data dan tipe pesan, pola interaksi dan pemetaan protokol yang digunakan.
3. SOAP (Simple Object Access Protocol)  
   SOAP digunakan untuk mendefinisikan *envelope* (pembungkus yang mendefinisikan awal dan akhir pesan) pada komunikasi service dan menyediakan serialisasi format untuk mentransmisikan dokumen XML melalui jaringan dan ketentuan representasi interasi RPC (Remote Procedure Call).
4. UDDI (Universal Description, Discovery, and Integration)

UDDI berfungsi mempublikasikan layanan yang tersedia pada suatu web service dan mencari layanan yang disediakan web service lainnya yang mungkin dapat untuk suatu aplikasi.

1. **RINGKASAN TUGAS AKHIR**

Add-ons yang dibuat pada tugas akhir ini diharapkan nantinya mampu mengklasifikan situs *phishing* dengan melakukan identifikasi terhadap konten dari yang sedang dijelajahi. Identifikasi dilakukan melalui beberapa tahapan seperti yang ditunjukkan pada gambar 4. Hasil akhir dari identifikasi ini adalah pelabelan terhadap situs yang dikunjungi apakah termasuk pada kategori phishing atau merupakan situs yang sah.

****

Gambar 4. Arsitektur Jaringan.

Berdasarkan gambar 4, alur kerja *add-ons* ini adalah sebagai berikut :

1. Klien mengakses alamat yang dituju menggunakan peramban *Mozilla Firefox*. Kemudian *add-ons* mengambil data URL yang diakses klien.
2. *Add-ons* menghubungi server detektor untuk menginformasikan URL yang diakses melalui *web service*.
3. Server detektor melakukan permintaan untuk mengunduh konten halaman situs yang dituju klien.
4. Web server tujuan memberikan balasan berupa konten html yang kemudian diunduh oleh server detektor. Setelah itu server akan menganalisai konten kemudian memberikan pelabelan terhadap situs tersebut.
5. Server detektor memberikan balasan kepada *add-ons* berupa hasil pelabelan situs yang akan dikunjungi klien.
6. *Add-ons* yang telah mendapatkan informasi hasil analisa situs kemudian menentukan tindakan selanjutnya yang akan diberikan pada peramban. Jika situs terdeteksi sebagai situs yang legal, maka add-ons akan mengijinkan peramban untuk menjelajah situs yang dituju. Jika situs terdeteksi sebagai *phishing* add-ons akan mengintervensi peramban untuk mengaksesnya dan memberikan peringatan kepada pengguna.
7. Peramban dapat mengakses halaman yang dituju secara normal jika situs yang dituju legal atau pengguna tetap ingin mengakses situs tujuan meskipun sudah mendapat peringatan dari *add-ons*.

Server detektor akan menganalisa konten situs melalui beberapa tahapan. Tahapan pertama yang dilakukan server detektor adalah membentuk DOM tree. Pembentukan DOM tree dimaksudkan untuk mengetahui konten dari situs yang dikunjungi. Pada tahap ini akan dicek apakah situs memiliki minimal satu *input text*. Jika ya, maka proses deteksi phishing akan dilanjutkan ke tahap berikutnya.

Tahapan selanjutnya adalah melakukan identity extration. Terdapat dua jenis identity yang dihasilkan pada tahap ini, yaitu identity set dan URL identity. Identity set dibentuk dari ekstraksi teks pada DOM tree yang kemudian menggunakan metode tf-idf (term frequency – inverse document frequency) untuk menghitung kata yang masuk dalam lima besar dari konten web tersebut. Sedangkan URL identity diperoleh dengan menganalisa tautan (*hyperlinks*) dari situs web yang dikunjungi. Proses penghitungan tf-idf pada aplikasi ini akan menggunakan bantuan Apache Lucene

Proses selanjutnya adalah membentuk feature vector. Berikut adalah feature yang akan digunakan, yaitu :

1. Alamat situs yang mencurigakan
2. ID alamat situs
3. Tautan kosong
4. ID tautan asing
5. Tautan asing
6. ID permintaan asing
7. Permintaan asing
8. Cookie domain
9. Setifikat SSL (Secure Socket Layer)
10. Jumlah titik pada alamat situs
11. Jumlah titik pada semua tautan

Tahapan terakhir adalah klasifikasi dari feature yang telah dicek sebelumnya. Pengklasifikasian menggunakan metode support vector machine (SVM). Feature vector akan menjadi input dari SVM. Hasil akhir nantinya akan berupa dua kelas yaitu *phishing* (+1) dan *legitimate* (-1). Proses klasifikasi menggunakan metode SVM akan menggunakan kakas bantu LibSVM.

Setelah hasil akhir diketahui, add-ons akan memberikan peringatan bahwa situs yang dikunjungi terindikasi sebagai situs *phishing*. Pengguna akan diberikan pilihan untuk membatalkan akses ke situs tersebut atau tetap melanjutkan dengan resiko keamanan ditanggung oleh pengguna.

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 web phishing ini pada peramban Mozilla Firefox.

1. Studi Literatur

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

1. Implementasi

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

* + - * 1. Pembuatan fungsi *identity extraction.*
        2. Pembuatan fungsi *feature extraction.*
        3. Pembuatan fungsi pengklasifikasian data dengan SVM.
        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**

Dalam pelaksanaannya, tugas akhir ini diharapkan dapat diselesaikan dengan jadwal sebagai berikut :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Kegiatan | Bulan | | | | | |
| September | Oktober | | Nopember | | Desember |
| 1 | Penyusunan Proposal Tugas Akhir |  |  |  |  | |  |
| 2 | Studi Literatur |  |  | |  | |  |
| 3 | Implementasi |  |  |  |  | |  |
| 4 | Pengujian dan Evaluasi |  |  | |  | |  |
| 5 | Penyusunan Buku Tugas Akhir |  |  | |  |  |  |

1. **DAFTAR PUSTAKA**

# Newcomer, Eric. 2002. *Understanding Web Service: XML, WSDL, SOAP,and UDDI*. Indianapolis: Pearson Education

# He, M., Horng, S., Fan, P., Khan, M.K, Run, R., Lai, J., Chen, R., Sutanto, A. 2011. *An Efficient Phishing Webpage Detector*. In Experts System with Applications. Elsevier (pp 12018-12027)

# Hégaret, P.L.. 2000. *What is Document Object Model?*. < http://www.w3.org/TR/DOM-Level-2-Core/introduction.html > (Diakses : 10 Oktober 2012)

# Weston, J. *Support Vector Machine Tutorial*. < http://www.w3.org/TR/DOM-Level-2-Core/introduction.html > (Diakses : 10 Oktober 2012)

# StatsCounter. *Top 5 Browser in Indonesia from April 2011 to Aug 2012*. <http://gs.statcounter.com/#browser-ID-monthly-201104-201208 > (Diakses : 10 Oktober 2012)

# Anti Phishing Working Group. 2012. *Phishing Activoty Trends Reporit 1st Quarter 2012*. <http://www.antiphishing.org/reports/apwg\_trends\_report\_q1\_2012.pdf > (Diakses : 10 Oktober 2012)