JSIKA 2 (2013) 1-6

Jurnal Sistem Informasi

Situs Jurnal : http://jurnal.stikom.edu/index.php/jsika

RANCANG BANGUN APLIKASI BELAJAR *WEB HAKCING* BERBASIS JEJARING SOSIAL (*FACEBOOK*)

[1] Rio Astamal [2] Anjik Sukmaaji [3] Teguh Sutanto

1)Program Studi/Jurusan Sistem Informasi, STMIK STIKOM Surabaya,email: si06004@stikom.edu

2)Program Studi/Jurusan Sistem Informasi, STMIK STIKOM Surabaya, email: anjik@stikom.edu

3)Program Studi/Jurusan Sistem Informasi, STMIK STIKOM Surabaya, email: teguh@stikom.edu

Abstract:*Knowledge of various web-based application security needed to build web applications that do not potentially have security flaws. To assist web developers in understanding various web vulnerabilities need to be provided special learning media. Simulation-based learning is used as learning strategy so the users can test applications directly; the goal is to make users know what kind of mistake that trigger the security flaw. Learning Web Hacking Application is web-based application that contains a series of questions about web technology and the vulnerabilities of web applications. Users can obtain the answer by hacking the web application that has been provided. The success of this process depends on how much the user's understanding about the security of web application. Application also provides a feature called "Learning Center" which consist series of article to increase user knrowledge about web vulnerabilites and how to prevent it. Learning Web Hacking Application can be used to help the learning process on web application security and application can be integrated with Facebook and has social networking features so users or friends of the users in the circle of Facebook friends can be motivate to learn more about the security of web based applications.*

Keywords: *hacking*, *learning,* s*ecurity, social network, simulation-based, web vulnerabilities*

Pengetahuan tentang ancaman-ancaman keamanan web adalah salah satu hal yang wajib diketahui seorang *web developer*. Dari hasil penelitian dan pengujian yang dilakukan oleh Stuttard (2011) pada tahun 2007 sampai dengan 2011 terhadap 100 lebih aplikasi web, ternyata masih banyak yang memiliki celah keamanan. Celah-celah keamanan yang ditemukan dan prosentasenya yaitu: Kesalahan Otentikasi (62%), Kesalahan Akses Kontrol (71%), *SQL Injection* (32%), *Cross-site Scripting* – XSS (94%), Kebocoran Informasi (78%), dan *Cross-site Request Forgery* – CSRF (92%).

Pengetahuan yang minim tentang masalah tersebut dapat menyebabkan aplikasi yang dibuat rentan. Sehingga diperlukan pembelajaran untuk agar *web developer* memahami masalah tersebut. Strategi pembelajaran tambahan yang dapat digunakan adalah *simulation-based learning*, dimana celah-celah keamanan suatu aplikasi web direproduksi ke dalam lingkungan khusus yang telah disediakan untuk pembelajaran (simulator). Menurut Davidovitch (2006) simulasi menciptakan pemikiran yang kritis dan strategis, kemampuan merencanakan dan berpikir strategis tidak mudah dikembangkan dan keuntungan dari simulasi adalah menyediakan alat untuk membantu masalah tersebut.

Untuk membuat sesuatu yang aman Council (2003) memaparkan bahwa perancang dapat "berpikir sebagai pencuri", dimana mereka meletakkan diri meraka pada posisi pelanggar, antisipasi aksi mereka, ketahui *tools* yang mereka gunakan, pengetahuan dan kemampuan mereka dengan demikian dapat dikembangkan solusi yang dapat mencegah mereka tanpa membahayakan pengguna yang sah. Oleh karena itu, pada apikasi ini penulis menempatkan *web developer* sebagai *hacker* yang melakukan *hacking* pada suatu aplikasi *web* yang sudah disimulasikan. Menurut Erickson (2008) *Hacker*

adalah suatu istilah untuk orang atau kelompok yang menulis kode program dan juga yang mengeksploitasi kelemahannya. Suatu proses yang dilakukan *hacker* untuk menemukan sebuah solusi atas masalah dengan cara yang cerdas dan diluar intuisi inilah yang disebut dengan istilah *hacking*

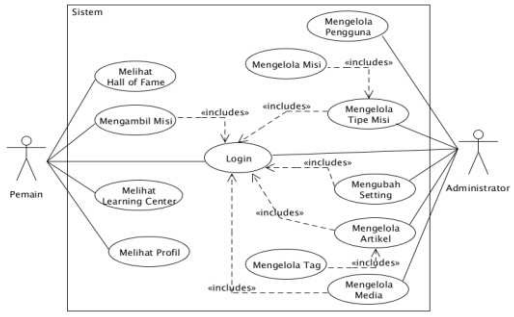
Aplikasi yang akan dibuat buat adalah sebuah aplikasi pembelajaran berbasis web yang didalamnya terdapat soal-soal yang menyangkut berbagai komponen teknologi web dan keamanannya. Setiap soal direpresentasikan oleh sebuah sub aplikasi yang didesain khusus untuk soal tersebut sesuai tujuan materi pembelajaran yang ingin dicapai. Untuk soal yang cukup kompleks maka soal tersebut dibuat dalam bentuk simulasi. Setiap soal yang berhasil diselesaikan maka pengguna akan mendapatkan poin. Semakin banyak poin yang diperoleh maka dapat diasumsikan pengetahuan *web developer* tersebut semakin baik.

Aplikasi ini menggunakan *platform* Facebook untuk proses autentikasi pengguna sehingga proses registrasi menjadi sederhana. Selain itu penggunaan *platform* Facebook dimaksudkan untuk menambahkan fitur jejaring sosial dimana informasi-informasi dan aktivitas yang dilakukan pengguna pada aplikasi ini dapat dibagikan ke *timeline* Facebook pengguna.

Diharapkan dengan adanya Aplikasi Belajar *Web* Hacking Berbasis Jejaring Sosial (Facebook) ini para *web developer* terutama mereka yang pemula dapat memahami berbagai celah keamanan pada aplikasi web sehingga pada akhirnya mereka dapat membuat aplikasi yang lebih aman. Fitur jejaring sosial yang ada pada aplikasi diharapkan dapat memotivasi pengguna lain dalam dilingkaran pertemanan Facebook untuk mempelajari keamanan aplikasi berbasis web.

Metode pengembangan yang dilakukan untuk mengembangkan aplikasi adalah *Agile Model Driven Development* (AMDD). Menurut Ambler (2012) AMDD merupakan versi agile dari Model Driven Development (MDD). Perbedaan dengan AMDD adalah pada AMDD pembuatan model tidak dilakukan secara ekstensif melainkan cukup membuat model yang dapat membuat pengembangan segera dijalankan. Tahap-tahap dalam AMDD yaitu *envisioning*, iterasi pemodelan, *model storming* dan implementasi via Test-Driven Development (TDD).

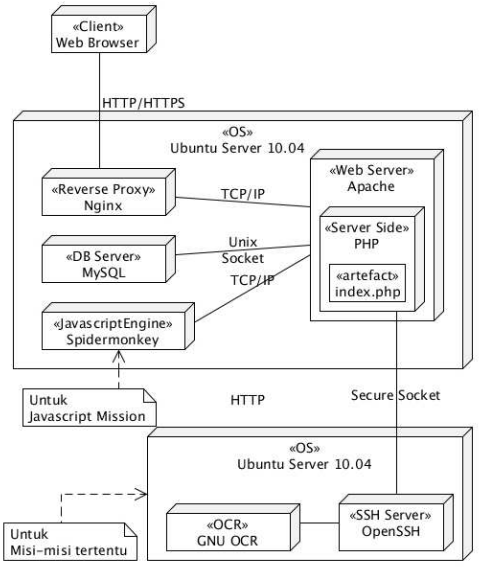
Makalah ini disusun dengan organisasi sebagai berikut: judul, abstrak, pendahuluan, perancangan sistem, hasil dan pembahasan, dan kesimpulan.



Gambar 1. Use Case aplikasi belajar Hacking

RANCANGAN SISTEM

Soal-soal yang dibuat pada aplikasi disebut dengan misi. Misi dibagi dalam tiga tipe yaitu *Basic Mission*, *Javascript Mission* dan *Realistic Mission*. Semua soal pada tipe *Realistic Mission* berbentuk simulasi dari sebuah aplikasi web yang memiliki celah keamanan. Setiap misi memiliki poin, poin dari tiap misi tergantung tingkat kesulitan dari misi tersebut.



Gambar 2. Deployment Diagram Aplikasi Belajar *Hacking*

1. Envisioning

Pada tahap ini kegiatan yang dilakukan adalah a) Pemodelan kebutuhan awal b) Pemodelan arsitektur awal.

1. Pemodelan Kebutuhan Awal

Pemodelan kebutuhan awal dilakukan untuk mengidentifikasi *hight-level requirements* dari aplikasi yang akan dibuat. Tahap ini juga

terdapat beberapa kegiatan yaitu: 1.a.) Membuat Usage Model 1.b.) Membuat Domain Model 1.c.) Membuat User Interface Model (UI).

a.1. Membuat Usage Model

Usage model menggambarkan bagaimana pengguna berinteraksi dengan sistem. *Usage model* dapat berupa *use case* atau *user stories* atau keduanya.

a.2. Membuat Domain Model

Domain model merupakan konseptual model yang *menggambarkan* entitas bisnis dan relasinya. Model ini sebaiknya dibuat dengan sangat sederhana asalkan dapat memuat informasi yang cukup untuk dapat menggambarkan proses yang ada.

a.3. Membuat User Interface Model

User interface model (UI) merupakan tahap pembuatan prototipe untuk desain antar muka dari aplikasi yang dibuat. Dapat digunakan sebuah sketsa untuk menggambarkan antar muka yang akan dibuat.

2. Pemodelan Arsitektur Awal

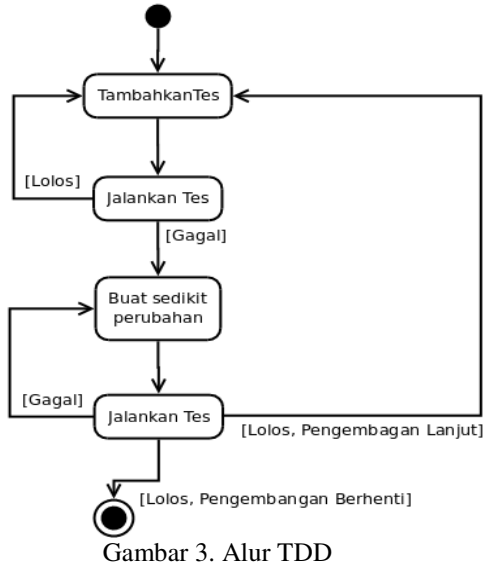
Menurut Ambler (2012) pada awal pengembangan, pengembang harus mempunyai gambaran seperti apa sistem yang akan dibangun. Pemodelan arsitektur berjalan paralel dengan tahap pemodelan kebutuhan awal. Sehingga pada tahap ini yang difokuskan adalah arsitektur perangkat keras dan perangkat lunak yang akan digunakan.

2) Iterasi Pemodelan

Pada tahap ini pengembang agile harus menyusun estimasi *jadwal* dan pekerjaan yang akan dilakukan pada setiap iterasi. Untuk melakukan estimasi yang tepat maka pengembang harus memahami pekerjaan seperti apa untuk mengimplementasikannya, dan itulah tujuan pemodelan pada tahap ini. Iterasi pemodelan ini akan dieksekusi pada *model storming* dan TDD.

3. Model Storming

Menurut Ambler (2012) model storming adalah *Just In Time* (JIT) *modeling* yang artinya pengembang mengidentifikasi masalah yang akan diselesaikan, jika dalam tim maka pengembang mengajak rekan yang dapat membantu, tim tersebut kemudian mengeksplorasi masalahdan kemudian masing-masing kembali melanjutkan pekerjaan seperti sebelumnya. Penulis melakukan kegiatan berikut dalam *model storming* yaitu: *flow-of-event*, *sequence diagram*, *class diagram* dan implementasi TDD.



Gambar 3. Alur TDD

4. Implementasi via Test-driven Development (TDD)

Menurut (Beck 2003; Astel 2003) dalam Ambler (2012) *Test*-Driven Development (TDD) adalah sebuah pendekatan evolusioner dalam pengembangan yang mengkombinasikan *test-first development* dimana pengembang menulis sebuah tes sebelum sepenuhnya menulis kode yang akan diperuntukkan untuk memenuhi tes dan *refactoring*. TDD dan *Model Storming* lebih sering berjalan paralel.

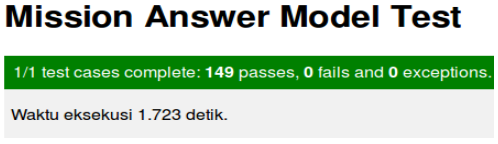
HASIL DAN PEMBAHASAN

Jumlah misi yang disediakan dalam aplikasi adalah 22 misi dengan perincian 7 *Basic Mission*, 8 *Javascript Mission*, dan 7 *Realistic Mission*. Total poin jika semua misi tersebut dapat diselesaikan oleh pengguna adalah 1000 poin.

Pengembangan aplikasi dimulai dengan melakukan pengkodean program sesuai dengan metode pengembangan yang digunakan. Kode-kode yang dibuat disimpan pada file dengan eksensi *.php. File-file tersebut kemudian dijalankan oleh Web Server Apache. Agar aplikasi dapat terhubung dengan Facebook dan dapat digunakan pengguna maka aplikasi perlu ditempatkan pada sebuah server yang terhubung ke internet dan *domain name* untuk memudahkan akses. Domain yang digunakan untuk mengakses aplikasi adalah *ta.rioastamal.net*.

Langkah-langkah agar aplikasi dapat diakses oleh pengguna adalah 1) membuat domain 2) mengarahkan domain ke server 3) memasang ubuntu server 4) memasang openSSH server 5) memasang MySQL dan membuat struktur database 6) memasang Nginx dan Apache 7) memasang ncat dan Spidermonkey 8) memasang git 9) membuat aplikasi pada Facebook.com untuk mendapatkan *App Secret* dan *App Key* yang digunakan untuk berkomunikasi dengan server Open Graph Facebook 10) mengupload file-file aplikasi ke server dengan *git push*.

Implementasi TDD pada aplikasi dimulai dengan melakukan unit testing terhadap class-class model yang digunakan pada aplikasi, gambar 4 menunjukkan output *unit testing* pada class *Mission_Answer_Test*.

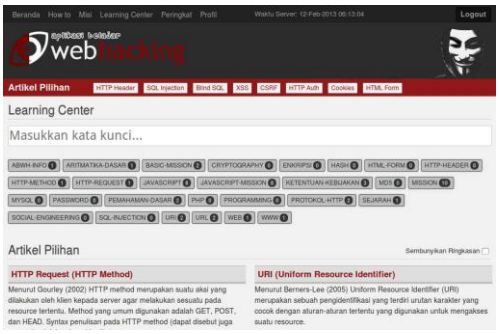


Gambar 4: Output Unit Testing pada Mission_Answer_Test

Gambar 5 menunjukkan output halaman depan dari aplikasi yang dibuat. Pada halaman tersebut pengguna dapat melihat keterangan tentang aplikasi dan daftar *Hall of Fame* yaitu pemain yang memiliki poin tertinggi.

Gambar 6 menunjukkan tampilan dari halaman *Learning Center.* Dimana pada halaman tersebut pengguna dapat menjelajah artikel-artikel seputar teknologi web dan keamanannya. Pemain dapat melakukan pencarian artikel berdasarkan kata kunci tertentu atau berdasarkan *tag* yang ada.

Gambar 5: Tampilan Hall of Fame pada Halaman Beranda

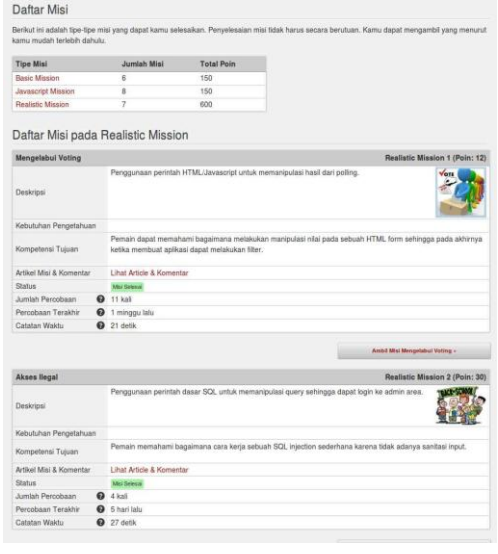


Gambar 6: Tampilan Learning Center

Gambar 7 menunjukkan contoh tampilan daftar misi yang disediakan oleh aplikasi. Pada halaman tersebut pemain dapat melihat deskripsi, tujuan dan status misi tersebut.

Gambar 8 menunjukkan tampilan halaman depan salah satu misi pada *Realistic Mission* yaitu *Realistic Mission 2* (RM2). Gambar 9 menujukkan proses *hacking* yang dilakukan dengan memanfaatkan celah *SQL Injection* pada aplikasi

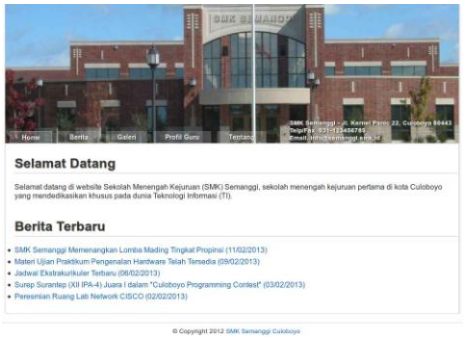
RM2. Teknik yang digunakan untuk mencari kelemahan aplikasi berbeda-beda. Pada contoh RM2 kelamahan SQL injection digunakan untuk masuk ke admin area dan mendapatkan jawaban yang diminta oleh misi.



Gambar 7: Tampilan Daftar Misi

Gambar 9 menunjukkan contoh informasi yang muncul pada Facebook ketika misi berhasil diselesaikan oleh pengguna. Pada informasi tersebut dicantumkan misi apa yang diselesaikan oleh pengguna dan terdapat keterangan singkat tentang misi tersebut. Proses pada gambar 10

adalah dilakukan otomatis oleh sistem.



Gambar 8: Tampilan Halaman Depan Realistic Mission 2

Gambar 9 : Tampilan saat dilakukan SQL Injection pada RM2

Gambar 10: Informasi penyelesaian misi muncul di Facebook

Berbeda dengan tampilan pada gambar 11. Pada gambar tersebut pengguna melakukan *share* sendiri hasil misi yang ia telah selesaikan. Proses share ini terjadi ketika pengguna berhasil menyelesaikan misi dan mengklik tombol "Lanjut" sehingga terdapat opsi yang muncul apakah hasil tersebut akan dishare ke Facebook atau tidak.

Gambar 11: Membagi hasil misi ke Facebook

Gambar 12 menunjukkan informasi yang muncul pada Facebook ketika seorang pemain menyalip perolehan poin dari pengguna lain. Gambar 13 menunjukkan tampilan halaman profil dari pengguna pada aplikasi. Pada halaman ini informasi yang disajikan adalah data diri singkat dari pemain dan informasi misi-misi yang diselesaikan pemsain serta tota poin yang didapat oleh pemain.

Gambar 13: Informasi Penyalipan Skor pada Facebook

Gambar 14: Tampilan halaman profil

KESIMPULAN

Setelah dilakukan uji coba dan evaluasi pada Aplikasi Belajar Web Hacking ini, maka dapat diambil kesimpulan bahwa aplikasi dapat digunakan sebagai pembelajaran tentang keamanan aplikasi berbasis web melalui serangkaian misi-misi yang disediakan dalam bentuk *simulation-based learning* dan dapat memberikan pen-skoran pada pengguna berdasarkan misi-misi yang berhasil diselesaikan. Aplikasi dapat terintegrasi dengan jejaring sosial Facebook dimana aktifitas yang dilakukan pada aplikasi dapat terlihat pada akun Facebook pengguna.

SARAN

Saran yang dapat diberikan untuk pengembangan aplikasi lebih lanjut adalah sebagai berikut:

1. Pemberian status tingkatan atas pencapaian yang dilakukan pengguna pada aplikasi. Sebagai contoh status tingkatan yang dapat diberikan adalah "Hacker", "Newbie", dan "Script

Kiddie". Salah satu cara untuk membuat status tingkatan dapat dilihat berdasarkan skor yang diperoleh atau berdasarkan materi penyusun soal-soal yang diselesaikan.

2. Materi pembelajaran keamanan aplikasi berbasis web yang dapat ditambahkan pada aplikasi adalah tentang *Cross-site Scripting* (XSS) dan *Cross-site Request Forgery* (CSRF). Dimana aplikasi harus dapat mensimulasikan hubungan antara tiga objek yaitu penyerang, korban, dan server.

3. Disediakan sebuah *mission creator* untuk membuat soal-soal baru tanpa harus melakukan upload file ke server secara manual.

DAFTAR PUSTAKA

Ambler, Scott W. 2011. *Introduction to Test Driven Development (TDD)*. http://www.agiledata.org/essays/tdd.html, diakses 04 Juni 2012.

Ambler, Scott W. 2012. *Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development*. http://www.agilemodeling.com/essays/amdd.htm, diakses 10 Oktober 2012.

Council, Design. 2003. *Think Thief: A Designer's Guide To Designing Out Crime* . http://www.designcouncil.org.uk/Documents/Documents/Publications/Think%20Thief_Design_Council.pdf, diakses 10 Oktober 2012.

Davidovitch, L., Parush A., Shtub, A. 2006. *Simulation-based Learning in Engineering Education: Performance and Transfer in Learning Project Management*. http://www.jee.org/2006/october/6.pdf, diakses 09 Nopember 2012.

Erickson, Jon. 2008. *Hacking: The Art Of Exploitation, 2nd Edition*. No Starch Press, Inc: San Fransisco.

Stuttard, Dafydd and Pinto, Marcus. 2011. *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Second Edition*. John Wiley & Sons, Inc: Indianapolis.