

ANALISIS IMPLEMENTASI DESIGN PATTERN SECURITY PERANGKAT LUNAK (STUDI KASUS APLIKASI KULIAH ONLINE)

Hendra Ramdhan Amrullah¹, Dana Sulistyو Kusumo², Yanuar Firdaus A.w.³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Sekarang kebutuhan akan perangkat lunak semakin meningkat, tetapi hal tersebut tidak diikuti dengan tingkat kepercayaan pengguna terhadap perangkat lunak yaitu dengan masih rentannya perangkat lunak terhadap masalah keamanan. Hal ini dikarenakan developer masih belum memperhatikan aspek keamanan perangkat lunak dan terlalu fokus pada aspek fungsionalitas perangkat lunak. Design pattern merupakan solusi terhadap suatu permasalahan yang sering ditemukan pada object oriented programming (OOP). Design pattern memberikan kemudahan bagi developer dalam pengembangan perangkat lunak.

Design pattern Security merupakan solusi untuk mengatasi permasalahan pada keamanan perangkat lunak. Design pattern Security merupakan metode yang menggunakan pendekatan design pattern untuk mengatasi permasalahan yang sering terjadi pada keamanan informasi. Pada tugas akhir ini diimplementasikan design pattern Security pada suatu studi kasus (Aplikasi Kuliah Online), kemudian dianalisis keamanan perangkat lunak dan keuntungan penerapan design pattern Security terhadap pengembangan aspek keamanan perangkat lunak. Untuk melakukan analisis penulis membandingkannya dengan perangkat lunak tanpa menggunakan design pattern Security. Hasil yang didapat pada Tugas Akhir ini menunjukkan bahwa penerapan design pattern Security membuat perangkat lunak lebih aman terhadap celah keamanan (vulnerability) dibandingkan dengan tanpa design pattern Security. Selain itu juga penerapan design pattern Security memberikan beberapa keuntungan bagi pengembang yaitu dapat meningkatkan reusability dan mampu mengintegrasikan aspek keamanan ke dalam desain sistem.

Kata Kunci : OOP, design pattern, design pattern security, vulnerability, keamanan.

Abstract

Nowadays, the need of software has been appreciably increasing but unfortunately it isn't followed by the level of users' confidence. They consider that software still need developing in order to eliminate security problems. For your information, many developers still don't recognize how important the software security is, and they are too focused on software functionality aspect. Moreover, they left software security as a crucial aspect.

Design patterns were first introduced as a way of identifying and presenting solutions to reoccurring problems in object oriented programming (OOP). This solution would help developer on developing software. Security design pattern are method which to adapt design patterns approach for solution reoccurring information security problem.

In this final project, the writer would implement security design pattern to such a case study, the application of online learning. This project would be analyzed whether security design pattern would match within the development of software security. The first thing that the writer begins to analyze is comparing software which it hasn't been implemented. As a result, this project is going to convey that the implementation of security design pattern make software more secure to vulnerability. Furthermore, implementation Security design pattern have several advantage for developer for instance reusability improvement and enable to integrate security aspect into system design.

Keywords : OOP, design pattern, security design pattern, vulnerability, security.

1. Pendahuluan

1.1 Latar belakang

Perkembangan perangkat lunak sudah semakin jauh berkembang. Sehingga tidaklah heran apabila pemakaian perangkat lunak dalam kehidupan sehari-hari sudah semakin banyak digunakan untuk memudahkan dalam beraktifitas baik itu individual, lembaga atau perusahaan dll. Terkadang dalam informasi individual ataupun lembaga terdapat informasi yang rahasia dan tersedia jika dibutuhkan dan tidak boleh dimodifikasi oleh pihak yang tidak berkepentingan. Sehingga dibutuhkan sistem yang memberikan keamanan dan tingkat kepercayaan yang tinggi untuk penggunaanya.

Secara umum para pengembang perangkat lunak kurang memperhatikan aspek keamanan. Keamanan tidak diperhatikan baik itu pada tahap desain, implementasi, maupun operasional. Masalah keamanan baru diperhatikan jika sudah timbul masalah yang dapat merugikan finansial. Padahal memperbaiki perangkat lunak yang sudah jadi dan sudah dipakai jauh lebih mahal dan membutuhkan waktu yang lama dibandingkan desain atau implementasi.[13]

Rentannya keamanan perangkat lunak dikarenakan kurang baiknya para pengembang perangkat lunak dalam mendesain keamanan sebuah perangkat lunak. *Design pattern* merupakan suatu cara untuk mengidentifikasi dan memberikan solusi terhadap permasalahan desain yang berulang-ulang yang terjadi pada *object oriented programming*. *Design pattern Security* merupakan metode yang menggunakan pendekatan *design pattern* untuk mengatasi permasalahan yang sering terjadi pada keamanan informasi.

1.2 Perumusan masalah

Berdasarkan latar belakang yang telah disebutkan, permasalahan yang menjadi fokus pada Tugas Akhir ini adalah :

- a. Bagaimana mendesain aspek keamanan dengan menggunakan *design pattern Security* dan mengimplementasikannya pada perangkat lunak (studi kasus aplikasi kuliah online) ?
- b. Bagaimana menganalisis keamanan perangkat lunak setelah menerapkan *design pattern Security* ?
- c. Apa saja keuntungan yang didapat dari penerapan *design pattern Security* pada pengembangan aspek keamanan perangkat lunak ?

1.3 Tujuan

Adapun Penyusunan Tugas Akhir ini bertujuan untuk :

- a. Mengimplementasikan *design pattern Security* pada Aplikasi Kuliah Online.
- b. Membandingkan keamanan antara perangkat lunak yang menerapkan *design pattern Security* dengan tanpa *design pattern Security*.
- c. Menganalisis keunggulan penerapan *design pattern Security* pada pengembangan aspek keamanan perangkat lunak.

1.4 Batasan masalah

Adapun yang menjadi batasan masalah pada Tugas Akhir ini adalah:

- a. Analisis keamanan dilakukan pada sisi keamanan perangkat lunak yang dibuat bukan pada sisi *networking* , *operating system* maupun DBMS.
- b. Hanya menerapkan 6 pattern dari 7 pattern yang ada pada *design pattern security* yaitu *single access point*, *checkpoint*, *roles*, *session security*, *limited view*, dan *full view with error*. Hal ini dikarenakan *secure access layer* merupakan pattern yang menjadi penghubung antara aplikasi dengan layer lain seperti network, operating system dll. Jika hal ini diterapkan maka analisis akan merambah pada sisi layer lain tersebut.
- c. Pengujian sistem keamanan untuk dianalisis terdiri dari serangan SQL *Injection*, *Cross Site Scripting (XSS)* , *brute force login*, *session hijacking* dan *broken access control*. Dengan tipe penyerangan tersebut diatas sudah dapat menguji pada *pattern-pattern Security* yang diterapkan.
- d. Tugas akhir ini tidak menganalisis sisi teknis dari pemrograman (kecepatan proses, pemakaian memori, dll).
- e. Bahasa pemrograman yang digunakan adalah PHP 5.0 .
- f. Studi kasus yang dipakai pada tugas akhir ini adalah Aplikasi Kuliah Online yang terdapat pada Moodle versi 1.9.1.
- g. Aplikasi yang dibangun berbasis web.
- h. Web Server yang digunakan Apache versi 2.2.8.
- i. DBMS yang digunakan adalah MySQL versi 5.0.51.

1.5 Metodologi penyelesaian masalah

Metodologi yang akan digunakan dalam merealisasikan Tugas Akhir ini adalah dengan menggunakan langkah-langkah berikut ini:

- a. Studi literatur dengan mempelajari literatur-literatur yang relevan dengan permasalahan yang menjadi bahasan tugas akhir ini, meliputi studi pustaka dan referensi tentang :
 - a) Konsep dari *design pattern Security*.
 - b) Konsep dari *design pattern Gang Of Four*.
 - c) Konsep dari *object oriented programming*.
 - d) Implementasi *object oriented programming* dengan menggunakan PHP.
 - e) Konsep *procedural programming*.
- b. Memahami struktur kode Moodle.
- c. Menganalisis kebutuhan dan desain perancangan perangkat lunak yang akan dibuat. Analisis dan perancangan ditekankan pada aspek keamanan perangkat lunak.
- d. Merekayasa ulang aspek keamanan Moodle sehingga sesuai dengan analisis dan perancangan.
- e. Mengimplementasikan hasil rancangan aplikasi menjadi aplikasi yang diinginkan (*coding*).

- f. Analisis hasil pengujian serangan terhadap perangkat lunak dengan menggunakan *design pattern Security* dan membandingkannya dengan hasil serangan pada Moodle untuk memperoleh kualitas keamanan perangkat lunak yang dibuat .
- g. Pengujian terhadap penambahan modul keamanan.
- h. Menyusun dokumentasi tahapan pengerjaan dan hasil tugas akhir berupa laporan tugas akhir.

1.6 Sistematika penulisan

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Bab ini berisi latar belakang, perumusan masalah, tujuan dan batasan masalah dari tugas akhir, metodologi yang digunakan dalam menyelesaikan tugas akhir ini serta sistematika penulisan buku tugas akhir.

BAB II Landasan Teori

Bab ini berisi uraian teori mengenai *object oriented programming, design pattern, design pattern security, moodle* dan keamanan.

BAB III Analisis dan Perancangan Perangkat Lunak

Bab ini berisi analisis kebutuhan sistem serta rancangan sistem secara terstruktur yang tertuang dalam bentuk *Unified Manipulation Language (UML)*.

BAB IV Implementasi dan Analisis Hasil Percobaan

Bab yang berisi hasil implementasi dan pengujian keamanan sistem, serta analisis dari hasil pengujian tersebut.

BAB V Kesimpulan dan Saran

Berisi tentang kesimpulan yang didapat dari pelaksanaan tugas akhir ini dan saran-saran yang diperlukan untuk perbaikan maupun pengembangannya lebih lanjut.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan penelitian, pengujian dan analisis penggunaan *design pattern security* untuk memodelkan aspek keamanan perangkat lunak dapat diambil kesimpulan sebagai berikut :

1. Desain aspek keamanan perangkat lunak yang dibuat dengan menerapkan *design pattern Security* mampu mencegah beberapa serangan yaitu *SQL injection*, *XSS*, *broken access control*, *session hijacking* dan *brute force login*.
2. Berdasarkan perbandingan hasil pengujian *black box testing* didapatkan bahwa perangkat lunak dengan menerapkan *design pattern security* mampu mencegah serangan *XSS* dan *session hijacking* dibandingkan dengan perangkat lunak tanpa menerapkan *design pattern Security*.
3. Penerapan *design pattern Security* memberikan beberapa keuntungan bagi *developer* antara lain :
 - ✓ Dapat mengintegrasikan aspek keamanan ke dalam desain sistem .
 - ✓ Dapat meningkatkan *reusability* aspek keamanan perangkat lunak.

5.2 Saran

Berikut ini saran-saran untuk pengembangan aplikasi selanjutnya :

1. Penerapan *design pattern Security* di layer lain misalnya pada *networking*, *database*, dll.
2. Untuk pengembangan selanjutnya dapat diteliti pengaruh struktur *source code* terhadap keamanan perangkat lunak.

Referensi

- [1] Alshanetsky Ilia , 2005, “Architect’s Guide to PHP Security”, Nanobooks.
- [2] Apvrille , Axelle and Pourzandi, Makan, 2005, “Secure Software Development by Example”, <http://www.computer.org/portal/site/security> didownload pada tanggal 12 desember 2008.
- [3] Barcalow Jeffrey, Yoder Joseph, , “Architectural Pattern for Enabling Application Security”, Journal of Computer Science.
- [4] Dowd Mark, John McDonald dan Justin Schuh, 2006, “The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities”, Addison-Wesley.
- [5] Fowler Martin, 2005, “UML Distilled Edisi 3 Panduan Singkat Bahasa Objek Standar”, ANDI: Yogyakarta. Weiple R. Edgar, 2005, Security in E-Learning, Springer.
- [6] Gamma, E., R. Helm, R. Johnson, J Vlissides, 2004, Design Pattern : Elements of Reusable Object Oriented Software, Addison-Wesley.
- [7] Hayder Hasin, 2007, Object-Oriented Programming with PHP5, Pack Publishing.
- [8] Helton Johenni dan Helton Rich, 2002, “Java Security Solutions”, Wiley Publishing.inc.
- [9] Heriyanto Bambang, 2004, “Rekayasa Sistem Berorientasi Objek, Informatika”. Informatika : Bandung.
- [10] McGraw Gary, 2006, “Software Security : Building Security In”, Addison Wesley.
- [11] Opdhal Andreas L, Sindre Guttorm, , “Capturing Security Requirement through Misuse Case”, journal of computer science.
- [12] Pinto Marcus, Stuttard Daffydd, 2008, “The Web Application Hacker Handbook: Discovering and Exploiting Security Flaw”, Willey Publishing: Indianapolis.
- [13] Raharjo Budi, , “Pembuatan Piranti Lunak Perlu Perhatikan Masalah Keamanan”, www.infolinux.web.id didownload pada tanggal 18 Mei 2008.
- [14] Rostad Lillian, , “An Extended Misuse Case : Including vulnerabilities and the Insider Threat”, Journal of Computer Science.
- [15] Schumacher Markus, et al, 2006, “Security Patterns Integrating Security and System Engineering”, John Wiley & sons.
- [16] Sweat E. Jason, 2005, “Architect’s Guide to PHP Design Patterns”, nanobooks.
- [17] <http://www.owasp.org/report>