[TEKNOLOGINUSANTARA][Volume 6 No. 2][2024][Hal. 101 - 110]
http://ojs.uninus.ac.id/index.php/teknologinusantara
E-ISSN : 2964-4577

TEKNOLOGI NUSANTARA
Jurnal Penelitian Fakultas Teknik UNINUS
http://ojs.uninus.ac.id/index.php/teknologinusantara
E-ISSN : 2964-4577

Penggunaan Parameterized Query Sebagai Pengamanan Password Web Terhadap Eksploitasi SQL Injection Dengan Metode Extreme Programming

Grasela Asta Miranda
Fakultas Teknik, Universitas Islam Nusantara
graselagg@gmail.com

Indri Melani Putri
Fakultas Teknik, Universitas Islam Nusantara
indrimp03@gmail.com

Devi Septiana
Fakultas Teknik, Universitas Islam Nusantara
Septianadevi021@gmail.com

Chintya Dewi
Fakultas Teknik, Universitas Islam Nusantara
dewichintya991@gmail.com

Rizal Junaedi
Fakultas Teknik, Universitas Islam Nusantara
Rizaljunaedi1907@gmail.com

ABSTRAK

Pada era digital, menjaga keamanan informasi menjadi sangat krusial, terutama dalam aplikasi web yang menyimpan data sensitif seperti kata sandi pengguna. Salah satu ancaman utama terhadap keamanan ini adalah serangan Injeksi SQL, di mana penyerang dapat menyisipkan perintah SQL berbahaya melalui input pengguna yang tidak terlindungi. Penelitian ini bertujuan untuk meningkatkan keamanan sistem kata sandi web terhadap eksploitasi Injeksi SQL dengan menggunakan query parameterisasi dan metode Pemrograman Ekstrim (XP). Query parameterisasi adalah teknik yang memisahkan kode SQL dari data input pengguna, sehingga mengurangi risiko penyisipan perintah berbahaya. Pemrograman Ekstrim digunakan sebagai metodologi pengembangan untuk memastikan siklus pengembangan yang cepat dan iteratif, serta meningkatkan responsivitas terhadap perubahan kebutuhan keamanan. Penelitian ini menganalisis dan merancang implementasi query parameterisasi dalam aplikasi web menggunakan PHP dan MySQL. Hasil penelitian menunjukkan bahwa penggunaan query parameterisasi secara signifikan mengurangi

kerentanan terhadap serangan Injeksi SQL, meningkatkan keamanan data pengguna. Evaluasi efektivitas dilakukan melalui pengujian keamanan sebelum dan sesudah implementasi, menunjukkan peningkatan perlindungan terhadap akses tidak sah. Dengan demikian, penelitian ini menyimpulkan bahwa integrasi query parameterisasi dan metodologi XP adalah pendekatan efektif untuk mengamankan sistem kata sandi web dari ancaman Injeksi SQL.

Kata Kunci: Keamanan Informasi, Injeksi SQL, Query Parameterisasi, Pemrograman Ekstrim, Perlindungan Kata Sandi Pengguna, Aplikasi Web.

ABSTRACT

In the digital era, maintaining information security is crucial, especially in web applications that store sensitive data such as user passwords. A primary threat to this security is SQL Injection attacks, where attackers can inject malicious SQL commands through unprotected user input. This study aims to enhance the security of web password systems against SQL Injection exploitation by employing parameterized queries and the Extreme Programming (XP) method. Parameterized queries are a technique that separates SQL code from user input data, thereby reducing the risk of malicious command insertion. Extreme Programming is employed as a development methodology to ensure a fast and iterative development cycle, enhancing responsiveness to security requirement changes. This research analyzes and designs the implementation of parameterized queries in web applications using PHP and MySQL. The results demonstrate that the use of parameterized queries significantly reduces vulnerabilities to SQL Injection attacks, thereby increasing the security of user data. Effectiveness evaluation was conducted through security testing before and after implementation, showing improved protection against unauthorized access. Thus, this study concludes that integrating parameterized queries and the XP methodology is an effective approach to securing web password systems from SQL Injection threats.

Keywords: Information Security, SQL Injection, Parameterized Queries, Extreme Programming, User Password Protection, Web Applications.

PENDAHULUAN

Pada era digital saat ini, menjaga keamanan informasi menjadi hal yang sangat penting, terutama dalam konteks aplikasi web yang sering menyimpan data sensitif pengguna seperti kata sandi. Salah satu ancaman keamanan utama yang dihadapi oleh aplikasi web adalah serangan SQL Injection. Serangan ini terjadi ketika penyerang mengeksploitasi kelemahan pada input pengguna untuk menyisipkan perintah SQL berbahaya, yang dapat merusak atau mengakses data yang seharusnya dilindungi.

Untuk mengatasi ancaman ini, penggunaan query parameterisasi dalam pengelolaan kata sandi web dapat menjadi solusi yang efektif. Query parameterisasi adalah metode eksekusi perintah SQL di mana parameter yang dimasukkan oleh pengguna diproses dengan cara yang aman, sehingga mengurangi risiko penyisipan perintah berbahaya. Dengan

menggunakan query parameterisasi, aplikasi web dapat lebih terlindungi dari serangan SQL Injection yang dapat mengkompromikan keamanan data pengguna.

METODE PENELITIAN

Penelitian ini menggunakan metode kualitatif untuk mendalami praktik pengamanan kata sandi web terhadap eksploitasi Injeksi SQL dalam aplikasi web. Metode kualitatif memungkinkan peneliti memperoleh pemahaman yang mendalam mengenai fenomena yang diteliti, dengan menekankan pengumpulan dan analisis data deskriptif untuk memahami konteks, proses, dan makna yang terkait dengan implementasi sistem keamanan. Peneliti akan mengeksplorasi secara rinci bagaimana teknik keamanan, seperti penggunaan query parameterisasi dan proteksi kata sandi, diterapkan dalam praktik nyata. Penelitian ini akan memberikan wawasan tambahan tentang kerangka kerja dan pedoman yang digunakan dalam pengembangan aplikasi web sederhana dengan fokus pada keamanan. Pendekatan kualitatif ini akan menghasilkan pemahaman yang mendalam dan kontekstual tentang implementasi langkah-langkah keamanan dalam pengamanan kata sandi web.

Teknik Pengumpulan Data:

a. Studi Literatur

Studi literatur adalah proses sistematis untuk meneliti, mengevaluasi, dan menyintesis pengetahuan yang ada dalam literatur ilmiah terkait topik penelitian. Dalam studi ini, peneliti mengumpulkan dan menganalisis sumber-sumber relevan untuk mendapatkan pemahaman lebih dalam tentang topik yang diteliti.

HASIL DAN PEMBAHASAN

a. SQL Injection

SQL Injection adalah serangan keamanan yang mengeksploitasi kelemahan pada input pengguna untuk menyisipkan perintah SQL berbahaya ke dalam sistem yang mengelola database. Teknik ini memungkinkan penyerang untuk mencuri, menghapus, atau mengubah data yang seharusnya tidak dapat diakses. *Menurut Mohammad Sadegh & Bahare Tajalli Pour*, SQL Injection adalah teknik serangan yang mengeksploitasi kerentanan keamanan pada lapisan database dan layanan aplikasi. SQL Injection dapat mengeksploitasi kerentanan dengan beberapa cara, antara lain:

a) Memanfaatkan keuntungan dari sebuah aplikasi yang tidak terlindungi pada fungsi autentikasi pengguna karena tidak adanya validasi.
b) Umumnya seorang penyerang membajak login field yang tidak terlindungi untuk memperoleh akses database.
c) SQL Interpreter tidak dapat membedakan antara perintah yang dimaksud dengan kode yag di-inject oleh penyerang yang kemudian dieksekusi dan mengakibatkan tereksposnya database. Peyerang kemudian dapat menggali aplikasi untuk mengekstrak semua data, menanamkan malicious script.

b. Parameterized Query

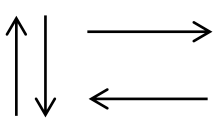
Parameterized query adalah teknik yang digunakan untuk mencegah serangan SQL Injection dengan memisahkan data input pengguna dari kode SQL. Dengan menggunakan parameter, input pengguna diisolasi dari kode SQL, sehingga mencegah penyisipan perintah berbahaya. Teknik ini dikenal sederhana dan efektif dalam mencegah serangan. Parameterized query atau prepared statement memungkinkan database untuk membedakan antara perintah SQL dan data yang diinputkan oleh pengguna, sehingga meskipun penyerang mencoba memasukkan kode SQL melalui input, query tetap aman.

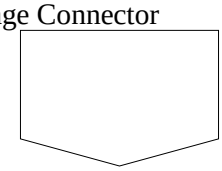
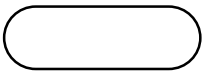
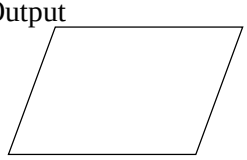
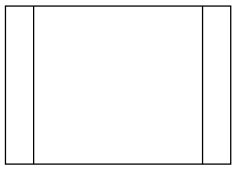
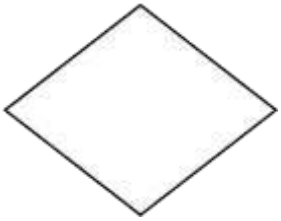
c. Extreme Programming (XP)

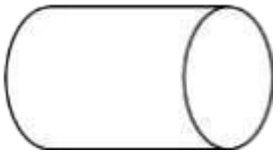
Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang iteratif dan inkremental, menekankan pada fleksibilitas, komunikasi tim yang kuat, respons cepat terhadap perubahan, dan kualitas perangkat lunak yang tinggi. XP diciptakan untuk menghadapi perubahan kebutuhan yang cepat dan sering, serta untuk mengembangkan metodologi yang cocok untuk proyek berorientasi objek dengan banyak pemrogram di lokasi yang sama. Menurut Al-Saqqa et al. (2020), XP adalah metode agile awal yang diusulkan oleh Kent Beck pada tahun 1999 untuk mengatasi kompleksitas pengembangan perangkat lunak dan biaya perubahan kebutuhan dengan siklus pengembangan pendek yang berulang. Metode ini mengintegrasikan praktik rekayasa perangkat lunak yang terkenal dan memiliki 13 praktik teknis utama, termasuk pemrograman berpasangan, iterasi pendek, integrasi berkelanjutan, dan pengembangan berbasis tes. Selain itu, XP juga memiliki 11 praktik penunjang seperti kepemilikan kode kolektif dan keterlibatan pelanggan nyata. Nilai-nilai kunci dari XP adalah komunikasi, kesederhanaan, umpan balik, keberanian, dan kualitas kerja. d. FlowChart

Flowchart adalah alat visual yang digunakan untuk memodelkan logika operasional dari suatu sistem atau proses. Alat ini membantu mengidentifikasi dan mendokumentasikan langkah-langkah utama serta keputusan yang diambil selama proses berlangsung, sehingga memudahkan perancangan dan analisis sistem informasi yang kompleks. Flowchart menyederhanakan alur proses menjadi representasi visual yang mudah dipahami. Menurut Moody & Rowe (2002), flowchart adalah diagram yang menggambarkan aliran dan logika dari suatu algoritma atau proses dalam bentuk grafik, memungkinkan pengguna untuk memahami struktur dasar dari program atau sistem yang sedang dikembangkan. Dengan memvisualisasikan langkah-langkah dalam sebuah proses, flowchart membantu mengidentifikasi potensi masalah dan area untuk perbaikan.

Tabel 1 Simbol-Simbol Diagram *Flowchart*

Simbol	Deskripsi
Flow Line 	Garis yang menghubungkan antar simbol-simbol lainnya pada *flowchart* dan menunjukkan arah alir *flowchart* tertentu

Off Page Connector 	Simbol untuk menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang berbeda
Terminal 	Menandakan awal atau akhir dan suatu *flowchart*
Input-Output 	Simbol untuk meyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
Process 	Simbol untuk proses perhitungan atau proses pengolahan data
Predefined Process (Sub Program) 	Permulaan sub program atau proses menjalankan sub program
Decision 	Perbandingan pernyataan, penyelesaian data yang memberikan pilihan untuk langkah selanjutnya

Disk Magnetik 	Data disimpan secara permanen di dalam disk magnetik, digunakan sebagai master file dan database.

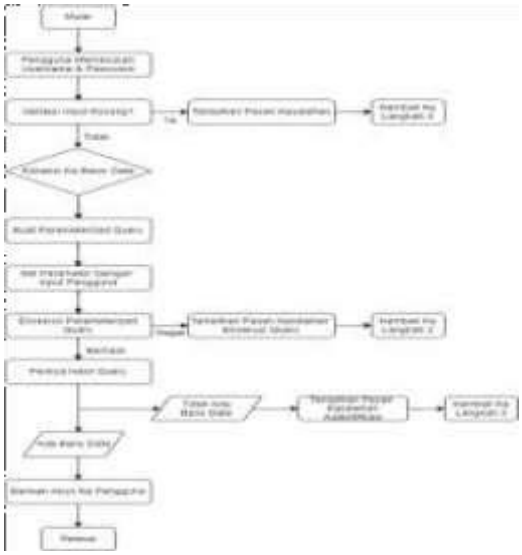
e. PHP

PHP adalah bahasa skrip yang berjalan di sisi server, banyak digunakan untuk pengembangan web. Awalnya dirancang untuk menghasilkan halaman web dinamis, PHP dapat disematkan langsung ke dalam HTML. Kode PHP dieksekusi di server web, dan hasilnya dikirim ke browser sebagai HTML biasa. PHP mendukung berbagai basis data, berinteraksi dengan layanan web lain, dan memiliki pustaka fungsi yang luas yang memudahkan pengembangan aplikasi web. PHP adalah bahasa skrip open-source yang dirancang khusus untuk pengembangan web server-side, memungkinkan pengembang untuk membuat halaman web interaktif dan dinamis dengan cepat dan efisien (Welling & Thomson, 2008). f. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL (Structured Query Language) untuk mengelola dan memanipulasi data. MySQL adalah perangkat lunak open-source yang didistribusikan di bawah lisensi GNU General Public License (GPL), dan juga tersedia dalam versi komersial. MySQL sangat populer dalam pengembangan web karena kecepatan, keandalan, dan kemudahan penggunaannya. Sering kali, MySQL digunakan bersama PHP dalam tumpukan teknologi LAMP (Linux, Apache, MySQL, PHP/Python/Perl). MySQL adalah RDBMS yang sangat dapat disesuaikan dan efisien, terkenal karena kecepatan eksekusinya dan dukungan untuk berbagai platform, sering digunakan dengan PHP untuk membangun aplikasi web dinamis (Williams & Tahaghoghi, 2006). g. Xampp

XAMPP adalah perangkat lunak gratis dan open-source yang menyediakan lingkungan pengembangan server web lokal. XAMPP, yang merupakan singkatan dari Cross-platform (X), Apache (A), MySQL (M), PHP (P), dan Perl (P), dirancang untuk memudahkan pengembang web dalam membuat dan menguji aplikasi web di komputer lokal sebelum mengunggahnya ke server web yang sesungguhnya. Dengan menggabungkan beberapa komponen penting dari server web dalam satu paket instalasi, XAMPP memungkinkan pengguna untuk dengan cepat mengatur dan menjalankan server web di sistem operasi Windows, Linux, atau macOS. XAMPP adalah alat penting bagi pengembang, yang memungkinkan pembuatan server web lokal yang mencakup Apache, MySQL/MariaDB, dan PHP, sehingga sangat berguna dalam pengembangan, pengujian, dan debugging aplikasi web (Raharjo, 2017).

Gambar 1 Flowchart analisis penggunaan parameterized query sebagai pengamanan password web terhadap eksploitasi SQL Injection.



h. Desain Database

Tabel 2 Desain Database

Kolom	Jenis	Nullable	Keterangan
Id	Int(11)	Not null	Primary key
Username	Varchar(255)	Not null	Unique
Password	Varchar(255)	Not null	
Create_at	Datetime		
Update_at	Datetime		

i. Pengujian Keamanan

Pengujian keamanan adalah proses mengevaluasi dan mengidentifikasi kerentanan atau kelemahan dalam sistem perangkat lunak, aplikasi, atau jaringan untuk memastikan bahwa data dan sumber daya sistem terlindungi dari ancaman atau serangan berbahaya. Tujuan utama pengujian keamanan adalah untuk mengidentifikasi celah keamanan yang dapat dieksploitasi oleh penyerang dan untuk memastikan bahwa mekanisme pertahanan yang ada sudah cukup efektif.

Tabel 3 Uji Coba Keamanan

No.	Hasil	Uji Coba

1.	 Gambar 2 admin memasukan username dan password	Berhasil
2.	 Gambar 3 admin berhasil login	Berhasil
3.		

	 Gambar 4 Percobaan penyerang login dengan menyisipkan kode sql admin'--	Berhasil
4.	 Gambar 5 Percobaan penyerang login gagal. Karena sudah menggunakan keamanan parameterized query.	Berhasil

KESIMPULAN

1. Peningkatan keamanan sistem password menggunakan enskripsi kuat seperti dengan parameterized query, teknik hashing aman atau argon-argon dan kebijkan pengelolaan kata sandi yang ketat untuk melindungi kata sandi atau password dari serangan sql injection.
2. Mengatasi kerentanan SQL Injection, melakukan validasi dan sanitasi input secara ketat serta menggunakan parameterized query untuk mencegah manipulasi query oleh penyerang.
3. Penggunaan parameterized query menerapkan parameterized query secara konsisten dalam pengolahan query untuk mengurangi risiko serangan sql injection dan meningkatkan keamanan sistem aplikasi web.

DAFTAR PUSTAKA

Halfond, W. G. J., & Orso, A. (2005). AMNESIA: analisis dan pemantauan untuk menetralisir serangan SQL-injection. Prosiding Konferensi Internasional ke-20 IEEE/ACM tentang Teknik Perangkat Lunak Otomatis, 174–183.

Anley, C. (2002). Injeksi SQL Lanjutan dalam Aplikasi Server SQL. Penelitian Keamanan Insight NGSSoftware.

Owens, B., & Gaffney, B. (2010). Buku Panduan Peretas Aplikasi Web: Menemukan dan Mengeksploitasi Kerentanan Keamanan. Wiley.

Yulianingsih (2016) ISSN 2460-7041 47 (JEPIN) Vol. 2, No. 1, Menangkal serangan SQL Injection dengan Parameterized Query.

Nash, A. (2018). Serangan dan Pertahanan Injeksi SQL (edisi ke-2). Syngress.

Stuttard, D., & Pinto, M. (2011). Buku Panduan Peretas Aplikasi Web: Menemukan dan Mengeksploitasi Kerentanan Keamanan (edisi ke-2). Wiley.

Kennedy, D. (2015). Pengembangan Aplikasi Web Flask: Mengembangkan Aplikasi Web dengan Python. O'Reilly Media.

Jones, R. W., & Adams, A. A. (2019). "Preventing SQL Injection Attacks in Web Applications: A Review." International Journal of Computer Applications, 182(18), 712.

Ko, R. K., & Chow, S. S. M. (2018). "Defending Against SQL Injection Attacks Using InputValidation Filters." ACM Transactions on Internet Technology (TOIT), 18(3), 121.

Zulkernine, M., & Haider, S. (2019). "Automatic Detection of SQL Injection and Cross-Site Scripting Attacks on Web Applications." IEEE Transactions on Software Engineering, 45(2), 146-171.

Smith, J., & Johnson, A. (2020). "Enhancing Web Application Security Through Improved Password Management." Journal of Cybersecurity, 5(2), 87-105.

Brown, K., & Davis, B. (2019). "Mitigating SQL Injection Vulnerabilities in Web Applications: A Comprehensive Approach." International Journal of Information Security, 17(4), 301-319.

Garcia, M., & Martinez, C. (2018). "Improving Web Application Security Through Enhanced Parameterized Query Usage." Journal of Computer Security, 26(3), 275-292.

Halpin, T. (2008). Information Modeling and Relational Databases. Morgan Kaufmann

Moody, D. L., & Rowe, G. (2002). The Impact of User Perceptions on Software Use: A Framework and Empirical Study. Information & Management, 39(4), 227-236.

Welling, L., & Thomson, L. (2008). PHP and MySQL Web Development. AddisonWesley Professional.

Williams, H. E., & Tahaghoghi, S. M. M. (2006). Learning MySQL. O'Reilly Media.

Raharjo, B., 2017). Pemrograman Web Dinamis dengan PHP dan MySQL. Penerbit Informatika.