Volume 7 Nomor 2 Mei 2022

INFORMASI INTERAKTIF

JURNAL INFORMATIKA DAN TEKNOLOGI INFORMASI

PROGRAM STUDI INFORMATIKA – FAKULTAS TEKNIK -UNIVERSITAS JANABADRA

INFORMASI INTERAKTIF	Vol. 7	No. 2	Hal. 79 - 155	Yogyakarta Mei 2022	ISSN 2527-5240

INFORMASI INTERAKTIF

ISSN 2527-5240

DEWAN EDITORIAL

JURNAL INFORMASI INTERAKTIF merupakan media komunikasi hasil penelitian, studi kasus, dan ulasan ilmiah bagi ilmuwan dan praktisi dibidang Informatika. Diterbitkan oleh Program Studi Informatika Fakultas Teknik Universitas Janabadra di Yogyakarta, tiga kali setahun pada bulan Januari, Mei dan September.

INFORMASI INTERAKTIF

ISSN 2527-5240

DAFTAR ISI

ISSN 2527-5240

PENGANTAR REDAKSI

Puji syukur kami panjatkan kehadirat Allah Tuhan Yang Maha Kuasa atas terbitnya JURNAL INFORMASI INTERAKTIF Volume 7, Nomor 2, Edisi Mei 2022. Pada edisi kali ini memuat 10 (sepuluh) tulisan hasil penelitian dalam bidang informatika.

Harapan kami semoga naskah yang tersaji dalam JURNAL INFORMASI INTERAKTIF edisi Mei tahun 2022 dapat menambah pengetahuan dan wawasan di bidangnya masing-masing dan bagi penulis, jurnal ini diharapkan menjadi salah satu wadah untuk berbagi hasil-hasil penelitian yang telah dilakukan kepada seluruh akademisi maupun masyarakat pada umumnya.

Redaksi

MELINDUNGI SISTEM LOGIN PADA SITUS WEB DARI SERANGAN SQL INJECTION

Zajuli T Bisri[1], Chaerur Rozikin[2]

[1,2]Teknik Informatika, Universitas Singaperbangsa Karawang,
Jl. HS. Ronggo Waluyo, Telukjambe Timur, Karawang, Jawa Barat

Email : [1] zajuli.taupiq18198@student.unsika.ac.id

ABSTRACT

Currently, internet users are increasingly widespread, especially for those who use a website. Security is the main thing that must be considered for the convenience and security of users. It is the job of website developers to make this a challenge. Because basically there is no guarantee that a website is safe. Therefore, the author will test the security of the web server, namely by performing SQL Injection. SQL Injection is a vulnerability or risk threat that can occur when an attacker has the ability to carry out activities to affect an SQL query that goes through a website to the database itself. In this study, implementing a PHP function, namely mysqli real escape string, which works to provide a backslash against a unique character or a special character before sending a query to mysql which can harm data, including from the SQL Injection attack. The results of this study are as a security warning of a website from attacks that want to break into a website.

Keywords: *SQL injection, html, php, parameterized query, mysqli real escape string*

1. PENDAHULUAN

Teknologi internet dan World Wide Web (WWW) saat ini berkembang sangat pesat dan sangat berpengaruh pada kehidupan manusia. Manusia kini dapat bertukar informasi dengan mudah dan cepat. Perkembangannya juga ikut serta meningkatkan taraf hidup seseorang kearah yang lebih baik dari segala aspek. Namun, tak selamanya teknologi internet selalu menguntungkan manusia. Semakin banyaknya pengguna layanan internet terutama situs web maka semakin rentan keamanan suatu sistem pada situs web terhadap serangan. Untuk menghindari keadaan atau kondisi yang tidak diinginkan, perlu dilakukan pengamanan dan pengawasan Keamanan situs web merupakan tindakan atau upaya untuk melindungi serta menjaga dari virus maupun serangan peretas agar data dan kerahasiaan pengguna terjamin aman. Ada banyak jenis maupun tujuan kejahatan pada suatu situs web yang dilakukan oleh penyerang, salah satunya ialah SQL injection. Sehingga keamanan suatu situs web sangat penting demi menjaga data dan informasi didalamnya. Maka perlu adanya pengujian keamanan sebuah situs web terhadap serangan SQL Injection, dengan menerapkan fungsi PHP.

2. TINJAUAN PUSTAKA

Masalah keamanan merupakan masalah yang penting dan utama dalam sistem komputer yang terhubung dalam suatu jaringan. Data maupun informasi menjadi targetserangan oleh pihak-pihak yang tidak bertanggung jawab sehingga perlu untuk menjaga integritas data dan informasi. Dalam aplikasi web dibutuhkan mekanisme yang dapat melindungi data dari pengguna yang tidak berhak. Mekanisme ini dapat diimplementasikan dalam bentuk sebuah proses *login* yang biasanya terdiri dari tiga buah tahapan yaitu identifikasi, otentikasi dan otorisasi. Seiring banyaknya fasilitas *internet* yang membutuhkan akses masuk (*login*) seperti *email*, akses *web server* maupun *account* lainnya, maka *user* perlu lebih berhati-hati terutama jika *account* tersebut sangat rahasia dan berharga mengingat internet merupakan jaringan public [1].

Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Jika berbicara mengenai masalah keamanan yang berkaitan dengan penggunaan komputer, maka sulit memisahkannya dengan kriptografi. Kriptografi bertujuan untuk memberikan layanan keamanan, termasuk keamanan untuk menjaga *password*. Data *password* yang dimiliki harus dapat dijaga atau dilindungi kerahasiaanya.

Jangan sampai data *password* yang ada, jatuh ke tangan orang-orang yang tidak berhak atau berkepentingan [2].

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi akan tetapi masalah keamanan sering kali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting.mBeberapa hal penting yang perlu diperhatikan pada keamanan *web* dan menjadimasalah yang penuh kerentanan adalah *login* dan *database*. Sistem *login* yang menggunakan *database* sebagai autentikasi *user* dan *password* sangat rentan untuk diretas. SQL *Injection* adalah salah satu teknik serangan yang dapat digunakan oleh penyerang untuk mengeksploitasi aplikasi *web*, sebagai akibatnya penyerang bisa mendapatkan akses tidak sah ke *database* atau untuk mengambil informasi langsung dari *database* [3].

Perkembangan jaringan komputer di masa kini memungkinan kita untuk melakukan komunikasi atau pengiriman pesan melalui jaringan komputer. Salah satu bentuk komunikasi adalah dengan menggunakan tulisan. Ada banyak informasi yang dapat disampaikan melalui tulisan (teks) dan terkadang dalam teks tersebut terdapat informasi yang bersifat rahasia. Untuk menjaga keamanan pesan yang bersifat rahasia,terdapat beberapa cara dan teknik tertentu yang dapat digunakan. Salah satunya dengan kriptografi yang berfungsi untuk menyamarkan pesan menjadi bentuk pesan tersandi. *Caesar Cipher* merupakan salah satu algoritma *cipher* tertua dan merupakan salah satujenis *cipher* substitusi yang membentuk *cipher* dengan cara melakukan pergeseran terhadap semua karakter pada *plainteks* dengan nilai pergeseran yang sama. Kelemahan *Caesar Cipher* adalah kita bias memperoleh pesan asli dengan memanfaatkan metode*Brute Force* dan presentasi frekuensi huruf yang paling sering muncul dalam suatu kalimat. Oleh karena itu penulis tertarik untuk mengkombinasikan *Caesar Cipher* dengan *Affine Cipher* untuk meningkatkan keamanan dari pesan. *Affine Cipher* adalahperluasan dari metode *Caesar Cipher* yang mengalikan pesan asli (plainteks) dengan sebuah nilai *intege*r dan menambahkannya dengan sebuah pergeseran (dalam *integer*) dinyatakan dengan fungsi kongruen [4]

Sistem *login* merupakan suatu hal yang pasti ditemukan didalam dunia *internet*. Saat seseorang melakukan *login* pastinya akan memasukkan *password* dimana *password* tersebut bersifat privasi dan rahasia. Oleh karena itu, masalah keamanan menjadi masalah yang sangat penting mengingat *internet* merupakan jaringan publik yang saling terhubung dalam suatu jaringan dan akan sangat berbahaya jika *password* yang dimasukkan *user* tersebut tidak dienkripsi sebelum dikirim ke *server* melalui jaringan. Disitulah celah kesempatan bagi para *sniffer* atau pengendus dapat melacak *password* atau data *user*. Sistem *login* dibuat dengan pemrograman PHP kemudian dilakukan pengamanan dengan enkripsi menggunakan MD5 yang dikombinasikan dengan pengacak atau menggabungkan *password* asli dengan suatu *string* tertentu lalu dienkripsi. Isi pengacak serta format untuk enkripsi hanya yang membuat aplikasi yang mengetahuinya. Setelah dilakukan pengamanan pada sistem *login* kemudian dilakukan analisis keamanannya dengan menggunakan sebuah *software* yaitu *wireshark* dan dapat dideteksi mana *password* yang dienkripsi dan yang tidak dienkripsi. [5].

Sistem Keamanan Toko Online berbasis Kriptografi AES merupakan aplikasi dalam bentuk website toko online yang dikembangkan untuk meningkatkan keamanan dan kenyamanan bagi pengguna. Sistem keamanan Toko Online diimplementasikan pada saat login dalam bentuk kode verifikasi. Sistem keamanan toko online ini membatasi akses website hanya pada satu komputer tertentu sehingga data menjadi lebih aman dari pihak yang tidak bertanggung jawab.Sistem Keamanan Toko Online berbasis Kriptografi AES dikembangkan menggunakan *webeditorAdobe Dreamweaver* dengan *web promgramming PHP* dan *database server MySQL*. Desain web berupa header, banner dan komponen web lainnya diedit menggunakan program aplikasi pengolah gambar *Adobe Photoshop*. Proses pengolahan data menggunakan sintaks SQL.Hasil penelitian dan pengujian menunjukan bahwa Sistem Keamanan Toko Online berbasis Kriptografi AES dapat menampilkan informasi dengan baik dan bersifat *userfriendly* baik untuk pengguna maupun admin serta juga dilengkapi dengan sistem keamanan yang dapat meningkatkan kenyaman pengguna dalam mengakses toko [6]).

Pengamanan login untuk mengakses Sistem Informasi Akademik berbasis WEB, berupa pengamanan menggunakan OTP (*One Time*

Password) yang di bangkitkan dengan *Hash* MD5 yang menghasilkan sebuah kode lewat SMS untuk otentikasi. Aplikasi OTP menggunakan masukan untuk *hash* MD5 dari tabel mahasiswa yang diambil adalah field NIM, No telp, dan waktu akses. Hasil dari fungsi *hash* tersebut menghasilkan 32 digit bilangan hexadesimal, kemudian menggantidengan angka bila ditemukan huruf di dalamnya.

Selanjutnya diambil enam digit dari bilangan tersebut. Enam angka tersebut yang dikirimkan sebagai OTP dengan layanan aplikasi Gammu berupa SMS dan juga disimpan dalam tabel. OTP yang dikirimkan kepada pengguna akan dicocokkan dengan yang tersimpan dalam tabel untuk mengecek validitasnya. Apabila cocok antara OTP yang dikirimkan dengan yang tersimpan dalam tabel, maka pengguna baru bisa mengakses Sistem Informasi Akademik. (SIAKAD). OTP yang dihasilkan adalah untuk otentifikasi pengamanan akun pengguna SIAKAD setelah Login dengan memasukkan *username* dan *password*.Waktu aktif untuk pengamanan login dengan OTP berbasis SMS selama tiga menit, pembatasan tersebut adalah untuk mempersempit waktu hacker untuk menyadap dan menyusup. Selain itu juga sesuai dengan uji coba yang telah dilakukan dengan beberapa layanan operator selular di Indonesia [7].

Perkembangan teknologi informasi berkembang pesat, menyebabkan keamanan data membutuhkan keamanan yang cukup baik. Sekarang setiap orang dapatdengan mudah bertukar informasi dalam hal apapun, termasuk diantaranya adalah berbagi pengetahuan untuk mengakses data secara ilegal. Gudang data dalam tabel *database* telah dipasang sistem login dengan pasword, begitu pula dengan sistem inventori TB Mita. Namun orang jahat mencari cara lain untuk mengakses data tersebut dengan cara mengakses langsung pada tabel *database* tanpa melalui sistem aplikasi tersebut. Dengan kemungkinan dari akses data ilegal yang mengakses langsung pada tabel *database* tersebut, diperlukan keamanan yang lebih baik terhadap *database* sisteminventori TB Mita. Ada banyak cara yang bisa dilakukan untuk meningkatkan keamanan. Dalam penelitian ini akan menggunakan cara dengan mengenkripsi database dengan algoritma Caesar cipher dan Hill cipher. Caesar cipher dan Hill ciphermerupakan bagian dari algoritma simetris, yang artinya pada proses enkripsi dan dekripsi memiliki kunci yang sama. Proses enkripsi dan dekripsi pada algoritma Caesarcipher dan Hill cipher masing-masing memiliki satu kunci, gabungan dari kedua algoritma ini menghasilkan dua kunci sehingga menjadi lebih kuat [8].

Pengguna internet, biasanya menggunakan fasilitas internet untuk melakukan proses pengubahan informasi. Sehingga keamanan data sangatlah penting. Kebutuhan akan informasi menjadikan para pengembang website menyajikan berbagai macam layanan bagi para pengguna. Namun kebanyakan dari para pengembang website mengabaikan keamanan sistem pada website tersebut. Serangan yang paling banyak digunakan oleh para penyerang tersebut adalah serangan SQL Injection. Penelitian ini difokuskan pada pengamanan sistem menggunakan algoritma Rijndael untuk mengenkripsi data. Algoritma Rijndael terpilih sebagai algoritma kriptografi yang dapat melindungi informasi dengan baik serta efisien dalam implementasinya dan dinobatkan sebagai Advanced Encryption Standard (AES). Algoritma ini akan ditanamkan pada login sistem untuk melindungi akses yang tidak sah dari penyerang. Hasil dari penggunaan algoritma Rinjdael dapat melindungi sistem login dengan baik sehingga sistem dinyatakan aman dari para penyerang website [3].

3. LANDASAN TEORI

Penelitian ini didukung dengan teori-teori yang berkaitan sehingga memberikan solusi yang tepat.

a. SQL Injection

SQL Injection adalah teknik serangan yang digunakan untuk mengeksploitasi kode dengan mengubah pernyataan back-end melalui input manipulasi. Bentuk utama pada SQL Injection terdiri dari penyisipan secara langsung kode ke dalam baris parameter yang digabungkan dengan baris perintah SQL dan dieksekusi. Injection yaitu merupakan suatu ancaman terhadap keamanan di situs website. Yaitu berdasarkan sebuah laporan risiko ancaman dari komunitas yang didapat dari BSSN dari laporan resiko ancaman atau kerentanan yang didapat adalah merupakan resiko ancaman atau kerentanan SQL Injection itu sendiri. Tetapi berdasarkan pada laporan tersebut, tetapi hal yang masuk dalam resiko ancaman yang tidak sulit untuk pencegahan, pada saat ini selalu menjadi suatu resiko ancaman atau kerentanan pada situs tersebut, yang biasa ditemukan yaitu di internet serta beberapa organisasi atau situs website yang

memiliki resiko ancaman atau kerentanan terhadap potensinya kebocoran pada suatu data yang diakibatkan dari sebuah serangan SQL Injection itu sendiri. Serangan SQL injection atau Injeksi SQL itu sendiri yaitu sebuah tipe serangan kegiatan injeksi pada suatu kode yang dapat dimanfaatkan untuk suatu celah keamanan sistem yang biasa terjadi terhadap lapisan *database* dari sebuah situs.[9]Hal ini bisa saja terjadi karena disebabkan dari suatu data ketika diinputkan oleh beberapa *user* tidak melakukan pengecekan dan dimuat di dalam sebuah perintah kueri itu sendiri. Oleh karena itu menjadikan suatu bagian suatu data yang masukkan oleh *user* tersebut dibutuhkan sebagai kesatuan dari kode itu sendiri. [10]

Tujuan dan dampak dari SQL Injection antara lain: [11]

Dampak untuk sebuah tipe serangan terhadap SQL Injection sangat berbeda, macam-macam diantaranya adalah seperti berikut:

1. Otentikasi Bypass
   Pada tipe ini dapat memungkinkan *attacker* agar *login* ke sebuah situs website karena adanya *administrative privileges*, tanpa perlu memakai suatu nama_pengguna dan sandi yang sah.[12]
2. Mengedit Suatu Data
   Dengan teknik ini *attacker* bisa dapat mengerjakan kegiatan seperti perubahan untuk isi data tersebut yang ada di dalam *database* tersebut, serta juga memakai suatu rongga keamanan tersebut agar bisa menginputkan isi yang dapat mengancam ke dalam sebuah *page* web itu sendiri. Untuk sebuah kasus database yang telah diubah adalah sebuah basis data untuk perbankan, *attacker* bisa saja bisa mengerjakan kegiatan perubahan untuk sebuah transaksi dari pemilik, serta juga mengirim isi pengguna rekening pemilik ke alamat rekening berbeda dikendalikan olehnya yang dapat menyebabkan dampak keuangan bagi pemilik rekening tersebut.[13]

3. Keberadaan Suatu Data
   Mengerjakan kegiatan penghilangan untuk semua data tersebut yang telah ada di dalam *database* itu sendiri, Tipe ini dapat berpeluang bagi aspek keberadaan untuk sistem data base.[14]

4. Penngambilan Suatu Informasi
   Tipe jenis ini digunakan bagi penyerang untuk dapat mengerjakan kegiatan pengambilan data cukup sensitif yang telah disimpan di dalam suatu basis datanya.

5. Menjalankan Suatu Perintah dari Jarak yang Jauh
   Pada teknik ini *attacker* juga bisa mengerjakan kegiatan *taks* eksekusi dengan basisdata tersebut dikarenakan agar *attacker* untuk mendapatkan kendali target.

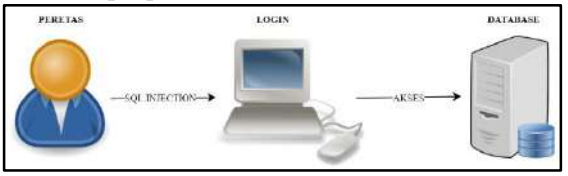
6. Menirukan Pemilik
   Attacker dapat mengerjakan serangan tipe ini untuk bisa mengerjakan kegiatan meniru dengan memanfaatkan rekening korban yang masih aktif di dalam data tersebut.

Tujuan :

1. Kerusakan atau kehilangan data.
2. Membocorkan data ke pihak yang tidak berkepentingan.
3. Penggunaan hak akses yang tidak sah.
4. Hilangnya kemampuan akses oleh pihak berwenang.
5. Pembajakan atau penipuan situs web.
6. Menghapus data korban yang memungkinkan untuk hacker atau penyerang menghapus seluruh data yang telah disimpan di dalam suatu data base.[15]

b. Cara Kerja SQL Injection

Kerentanan SQL Injection sering terjadi saat aplikasi web tidak memastikan bahwa nilai yang diterima dari formulir web, cookie, input parameter, dan sebagainya divalidasi atau dienkode sebelum meneruskannya ke SQL kueri yang akan dieksekusi di server basis data. Sebagai contoh, seorang peretas akan memanfaatkan kotak masuk yang tidak dilindungi agar untuk mendapatkan suatu akses basis data. [10]



Gambar 1. Proses *SQL Injection*

Gambar 1 menunjukan bagaimana seorang atau

pengguna peretas melakukan serangan terhadap situs web dengan metode SQL Injection melalui halaman login. Setelah melewati halaman login, peretas akan mendapatkan akses untuk melihat dan mengubah catatan atau berpotensi bertindak sebagai administrator basis data.[16]

c. Metode Menangkal SQL Injection

Bahasa queri terstruktur bisa dicegah dari sisi server, yaitu melalui beberapa cara seperti: [17]

1. Penggunaan *Parameterized Query*, yang nantinya akan mentransfer per parameter ke dalam *layer* kueri sesudah seluruh *code* bahasa kueri terstruktur sudah didefinisikan. Basis data dapat melakukan perbedaan apapun masukan dari pengguna antara kode dan data. Seorang atau pengguna yang melakukan penyerangan tidak mampu merubah tujuan kueri, meskipun bahasa kueri terstruktur perintah telah dimasukan di dalamnya.
2. Melakukan pengesahan inputan atau memberi batasan yang sudah didefinisikan seperti tipe, filter dan panjang terhadap masukan dari pengguna.
3. Mematikan pesan eror yang keluar dari basis data.
4. Mengunci dan membatasi basis data dari pengguna yang tidak berkepentingan dengan menonaktifkan akses perintah insert,update, dan delete.
5. Memberi batas pada kotak *input*, agar lebih terjamin di dalam setiap beberapa kotak dibatasi jumlah inputannya, untuk *user* setidaknya dimasukkan beberapa karakter lalu disesuaikan seperti dengan suatu keperluan, sehingga ketika ada suatu kegiatan yang masuk akan langsung terbatas dengan adanya jumlah karakter yang ada[18].

d. PHP

PHP atau *Hypertext Preprocessor* adalah sebuah bahasa pemrograman berbasis web yang terdapat pada *server side*. Artinya semua sintaks sepenuhnya dijalankan pada sisi server sedangkan hasilnya akan ditampilkan pada komputer klien. PHP juga merupakan HTML *embedded*, yaitu sintaks PHP yang dapat dituliskan bersamaan dengan sintaks HTML.[19]

e. Fungsi *Mysqli* Real Escape String

Fungsi mysqli real escape string pada PHP merupakan fungsi PHP yang dapat mencegah SQL Injection.[20] Cara kerja untuk fungsi ini adalah memberikan kegiatan backslash terhadap suatu karakter yang bertipe unik atau karakter tipe khusus sebelum mengirimkan kueri ke SQL yang dapat beresiko membahayakan data termasuk dari sebuah serangan SQL Injection itu sendiri. [21]Tetapi saat kegiatan menyimpan menuju SQL, kode akan tetap dalam kondisi normal tanpa adanya backslash.

4. METODE PENELITIAN

Metode penelitian ini dilakukan dalam Sejumlah eksperimen untuk mendapatkan pengaruh resiko tindakan tertentu pada yang berbeda dalam lingkungan yang bisa terkendali [22] Penelitian ini akan dilakukan melalui beberapa langkah yang gambarkan melalui Gambar 2.

Gambar 2. Langkah Penelitian

Gambar 2 menunjukan penelitian dibagi kedalam 4 langkah kegiatan, yaitu: Pengamatan, Studi Pustaka, Eksperimen, dan Kesimpulan. Berikut uraian yang diberikan:

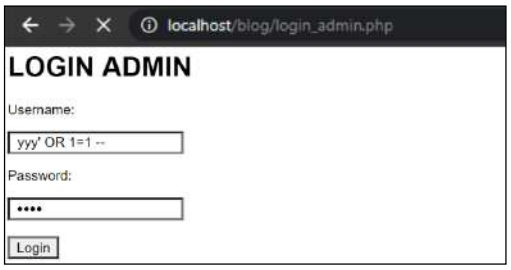
1. Pengamatan, bertujuan untuk mengidentifikasi masalah yang ada.
2. Studi Pustaka, ini berkaitan dengan sejumlah permasalahan yang tersedia untuk dapat menentukan metode penelitian terbaik agar bisa dijadikan pemecahan masalah.
3. Eksperimen, pada langkah ini pemecahan tersebut diterapkan dengan merangkai rangkaian suatu bahasa pemrograman yaitu menggunakan bahasa PHP.
4. Kesimpulan, menyimpulkan hasil dari eksperimen apakah mendapatkan hasil yang sesuai dengan tujuan utama yang akan dicapai.

5. ANALISIS DAN PERANCANGAN SISTEM

Halaman login merupakan gerbang masuk

pengamanan awal yang dimiliki bagi sebuah aplikasi situs web, biasanya berupa input username dan sandi. Kedua pada bagian ini memiliki suatu prinsip cara kerja logika AND yang berarti harus memenuhi nilai keduanya benar. Berikut tahapan eksperimen pada penelitian:

a. Login dengan *SQL Injection*.



Gambar 3. Login dengan *SQL Injection*

Gambar 3. Menunjukan *Username* dan *Password* diberi nilai sembarang. Namun untuk *Username* terdapat penambahan kalimat kueri yang berisi *SQL Injection* yaitu logika OR dan kueri komentar pada SQL (‘ OR 1=1 –).

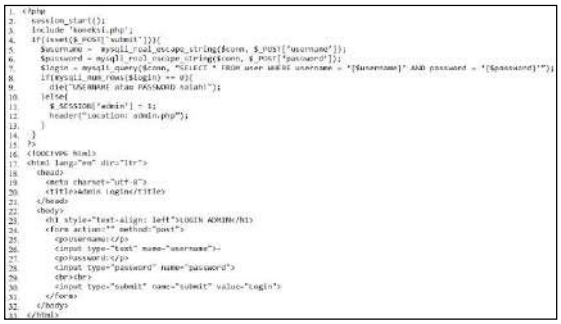
Tabel 1. Menerjemahkan Username dan Password dalam Queri SQL

Username	Password	Kueri SQL
‘yyy' OR 1=1 – ‘	‘xxxx’	SELECT * FROM pengguna WHERE nama_pengguna = 'yyy' OR 1=1 -- ' AND password = 'xxxx'

Tabel 1 menunjukan *Username* dan *Password* yang telah diinputkan akan diterjemahkan ke dalam kueri SQL menjadi *SELECT * FROM user WHERE username = 'yyy' OR 1=1 -- ' AND password = 'xxx'*. Karena pada *username* memasukan logika OR dan komentar dalam SQL (*AND password = 'xxxx'* akan menjadi komentar dan tidak akan dianggap sebagai kueri), maka kueri akan bernilai benar sehingga dapat masuk ke dalam situs web tersebut seperti pada Gambar 4.

Gambar 4. Berhasil Login

b. Penambahan fungsi mysqli real escape string



Gambar 5. Penambahan fungsi pada program PHP

Gambar 5 menunjukan penambahan fungsi mysqli real escape string pada baris ke-5 dan ke-6 sehingga akan dilakukan proses pembersihan kueri dengan menyeleksi karakter yang tidak dianggap sebagai bagian dari kueri.

c. Melakukan serangan kedua kali melalui cara yang sama seperti cara pertama.

Gambar 6. Gagal Login

Gambar 6 membuktikan bahwa fungsi mysqli real escape string dapat mencegah serangan *SQL Injection* pada halaman login.

Tabel 2. Hasil dari kueri SQL setelah penambahan fungsi PHP

Username	Password	Kueri SQL
‘yyy' OR 1=1 – ‘	‘xxxx’	SELECT * FROM pengguna WHERE nama_pengguna = 'yyy\' OR 1=1

		-- ' AND password = 'xxxx'

Tabel 2 menunjukan fungsi mysqli real escape string akan menangani karakter unik seperti kutip dengan bantuan *backslash* sehingga kalimat kueri yang berisi *SQL Injection* akan menjadi sebuah *string* dan tidak dapat mengeksploitasi basis data.

d. Meminimalisir SQL Injection

Untuk menghindari atau meminimalisir kegiatan serangan SQL Injection cara yang biasanya digunakan yaitu memeriksa setiap karakter inputan yang dimasuk kedalam suatu data base menggunakan pernyataan SQL. Pada kasus diatas melakukan kegiatan pencegahan harus dilakukan pada karakter ' yang masuk dengan ". Contohnya :

```
<% option explicit %>
<%dim connstring,conn,recset
connstring =
"Provider=SQLOLEDB.1; Password=1234;
Persist
Security Info=True; User ID=sa;
Initial
Catalog=sqlinject; Data
Source=localhost"
set conn = server.creat eobject
("adodb.connection")
set recset =
server.createobject("adodb.recordset"
)
conn.open connstring
recset.open "select * from tbUser
where username =
'" &
replace(request.form("username"),"'",
"''") & "'
and password = '" &
replace(request.form("password"),"'",
"''") &
"'",conn,3,2
if not recset.eof then
response.write recset.recordcount
session("username") =
request.form("username")
response.redirect "secured_page.asp"
else
response.redirect "login.asp"
     end if
     %>
```

Perubahan kode diatas yaitu dilakukan terletak pada penginputan nama_pengguna dan password dapat dirubah dengan request.form("nama_pengguna") menjadi replace (request.form("nama_pengguna"),"","""), hal tersebut juga terjadi pada bagian password. Ada beberapa cara lain untuk menanggulanginya yaitu :

1. Untuk bisa melindungi QUERY SQL, dapat menggunakan jenis tipe teknik sainitasi yaitu jenis mengosongkan semua inptuan yang diperoleh dari request object ASP itu sendiri. Jenis teknik sanitasi ini bermanfaat untuk pemakaian RDBMS. Untuk pencegahan dan mengatasi resiko atauancaman terhadap SQL Injection, yaitu dengan menghindari penggunaan sebuah tanda petik tunggal (') dengan menggunakan kegiatan replace.
2. Pesan error dengan cara menonaktifkan atau merubah dan mengganti pesan error yang dapat mencegah penyerang untuk menelusuri alur database. Panjang inputan kotak yang dibatasi dapat membuat penyerang pemula menjadi bingung terhadap pemakaian code inject yang tidak berfungsi dikarenakan code yang terlalu panjang.

6. KESIMPULAN

Berdasarkan hasil eksperimen didapat sejumlah kesimpulan antara lain adalah: Hasil eksperimen membuktikan bahwa fungsi mysqli real escape string terbukti mampu menangani serangan SQL Injection, Fungsi mysqli real escape string terbilang efektif karena mudah diimplementasikan. Kenyamanan pengguna dan kecepatan proses adalah hal yang harus diperhatikan ketika ingin menentukan konsep keamanan pada lapisan awal suatu situs web.

DAFTAR PUSTAKA

[1] D. M. Khairina, “Analisis Keamanan Sistem Login,” Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer, vol. 6, no. 2, pp. 64–67, 2016.

[2] P. Dwi Agus, “Perancangan modernisasi migrasi jaringan dari kabel tembaga ke kabel serat optik studi kasus: di perumahan Dian Anugerah Regency Gambut Kabupaten Banjar, Kalimantan selatan,” 2014.

[3] E. A. Dharmawan, E. Yudaningtyas, and M. Sarosa, “Perlindungan Web pada Login Sistem Menggunakan Algoritma Rijndael,” Jurnal EECCIS, vol. 7, no. 1, pp. 77–84, 2013.

[4] D. Rachmawati and A. Candra, “Implementasi Kombinasi Caesar dan Affine Cipher untuk keamanan Data Teks,” JEPIN (Jurnal Edukasi dan Penelitian Informatika), vol. 1, no. 2, pp. 60–63, 2015.

[5] D. M. Khairina, “Analisis Keamanan Sistem Login,” Informatika Mulawarman: Jurnal Ilmiah Ilmu

Komputer, vol. 6, no. 2, pp. 64–67, 2016.
[6] B. Candra and J. Wahyudi, "Hermawansyah, 'PENGEMBANGAN SISTEM KEAMANAN UNTUK TOKO ONLINE BERBASIS KRIPTOGRAFI AES MENGGUNAKAN BAHASA PEMROGRAMAN PHP DAN MYSQL,' J," Media Infotama, vol. 11, no. 1, pp. 31–39, 2014.
[7] E. Sediyono and K. I. Santoso, "Suhartono Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS," in Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, pp. 1604–1608. 2013.
[8] E. D. Santosa, "Implementasi Algoritma Caesar Cipher dan Hill Cipher Pada Database Sistem Inventori TB Mita Jepara," Dokumen Karya Ilmiah Program Studi Teknik Informatika Universitas Dian Nuswantoro. Semarang, 2015.
[9] R. Parlika, H. Khariono, H. A. Kusuma, M. R. Abrori, and M. A. Rofik, "Implementasi Akses Mysql dan Web Server Lokal Melalui Jaringan Internet Menggunakan Ngrok," JIKO (Jurnal Informatika dan Komputer), vol. 3, no. 3, pp. 131–136, 2020.
[10] S. Benfano, G. Fergyanto E., and Frumentius. Hirzi, "Prevention Structured Query Language Injection Using Regular Expression and Escape String," ICCSCI, vol. 135, 2018.
[11] Y. Yulianingsih, "Menangkal Serangan SQL Injection Dengan Parameterized Query," JEPIN (Jurnal Edukasi dan Penelitian Informatika), vol. 2, no. 1, pp. 46–49, 2016.
[12] S. Herlambang and H. Tanuwijaya, "Sistem Informasi: konsep, teknologi, dan manajemen," Yogyakarta: Graha Ilmu, 2005.
[13] T. Wahyono, "Sistem Informasi," Yogyakarta: Graha Ilmu, 2004.
[14] M. F. Wali and M. Rehan, "Effective coding and performance evaluation of the Rijndael Algorithm (AES)," in 2005 Student Conference on Engineering Sciences and Technology, 2005, pp. 1–7.
[15] D. Stiawan, Sistem Keamanan Komputer. Elex Media Komputindo, 2005.
[16] I. R. Widiasari, "Combining advanced encryption standard (AES) and one time pad (OTP) encryption for data security," International Journal of Computer Applications, vol. 57, no. 20, 2012.
[17] H. Haviluddin, "Aplikasi Program PHP dan MySQL," 2016.
[18] K. M. S. Soyjaudah, M. A. Hosany, and A. Jamaloodeen, "Design and implementation of Rijndael algorithm for GSM encryption," in SympoTIC'04. Joint 1st Workshop on Mobile Future & Symposium on Trends In Communications (IEEE Cat. No. 04EX877), 2004, pp. 106–109.
[19] A. Kusuma, " Serangan, Metode, & Database," Teknik Informatika, (8053111074), 124., 2016.
[20] D. Surian, "Algoritma Kriptografi AES Rijndael," TESLA Jurnal Teknik Elektro UNTAR, vol. 8, no. 2, p. pp-97, 2009.
[21] J. Majumder and G. Saha, "Analysis of SQL Injection Attack," Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT), pp. 2231–5292, 2009.
[22] I. Mardianto, A. Sedyono, and A. Hafzan, "Analisa Kerentanan Sis. trisakti. ac. id Menggunakan Teknik Vulnerability Scan," Jetri: Jurnal Ilmiah Teknik Elektro, vol. 13, no. 1, 2015.