Analisis Security Mitigation Terhadap Website Akademik Penunjang Pengajaran di Institusi XYZ Menggunakan Metode Penetration Testing Execution Standard (PTES)

1st Muhammad Hanafi Mu'amar Al Faridz
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
mhanafi@student.telkomuniversity.ac.id

2nd Umar Yunan Kurnia Septo Hediyanto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
umaryunan@telkomuniversity.ac.id

3rd Adityas Widjajarto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
adtwjrt@telkomuniversity.ac.id

***Abstrak*—Perkembangan pesat teknologi informasi sejalan dengan pertumbuhan penggunanya. Pemanfaatan teknologi informasi dalam bentuk website telah merambah berbagai pihak, termasuk Universitas XYZ yang menggunakannya untuk keperluan akademik, pembelajaran, dan kebutuhan internal. Salah satu website Universitas XYZ didesain untuk membantu pembayaran praktikum dan sertifikasi di Fakultas XYZ. Namun, keuntungan teknologi juga beriringan dengan peningkatan serangan keamanan. Ancaman beragam dengan tujuan merugikan pemilik dan pengguna website semakin meningkat. Oleh karena itu, diperlukan pengujian kerentanan untuk mengidentifikasi celah keamanan pada website Fakultas XYZ dengan menggunakan metode PTES. Pengujian melibatkan alat-alat seperti SQLMap, Burp Suite, jSQL Injection, dan Havij. Analisis kerentanan dilakukan untuk menentukan langkah mitigasi. Hasil analisis menunjukkan bahwa pengujian keamanan menggunakan metode SQL Injection pada website akademik di Institusi XYZ tidak menemukan celah yang mengindikasikan manipulasi data melalui serangan SQL Injection. Penelitian selanjutnya bisa menggunakan alat-alat yang beragam, akurat, dan luas untuk mengidentifikasi kerentanan dengan metode yang berbeda agar hasil lebih beragam. Pemeliharaan rutin website direkomendasikan untuk mengurangi celah keamanan.**

***Kata kunci*— Eksploitasi, Kerentanan, PTES, Mitigasi**

I. PENDAHULUAN

Perkembangan teknologi informasi kini tumbuh pesat sejalan dengan penggunaannya yang semakin meluas. Namun, ini juga membawa risiko penyalahgunaan informasi yang semakin tinggi. Oleh karena itu, keamanan informasi, terutama dalam konteks teknologi komputer dan jaringan, sangat penting [1]. Penggunaan sistem informasi, seperti di perguruan tinggi, bisa mendukung kesuksesan, tetapi juga memiliki risiko kelemahan keamanan yang bisa disalahgunakan. Salah satu contoh adalah website Akademik Penunjang Pengajaran di Institusi XYZ yang belum pernah diuji keamanannya. Hal ini meninggalkan celah yang dapat dieksploitasi oleh pihak luar, mengancam keberlangsungan sistem dan data. Solusinya adalah melakukan uji penetrasi untuk menemukan kerentanannya. Pengujian menggunakan beberapa alat seperti *SQLMap, Bupsuite, jSQL Injection*, dan *Havij*. Hasilnya akan membantu melindungi website dari ancaman serangan siber yang berpotensi merugikan.

II. KAJIAN TEORI

A. Keamanan Sistem Informasi

Menurut G. J. Simons, keamanan sistem informasi adalah cara mencegah atau mendeteksi penipuan dalam sistem berbasis informasi, di mana informasinya tidak berarti secara fisik [2]. Menurut [3], keamanan sistem informasi memiliki tiga aspek utama, dikenal sebagai CIA Triad:

1. *Confidentiality*, melibatkan langkah-langkah untuk menjaga privasi web app agar tidak diakses oleh orang yang tidak berwenang.
2. *Integrity*, Mengamankan keaslian data dengan memastikan hanya pengguna yang berwenang yang dapat mengubahnya melalui filter data dan filter pengguna.
3. *Availability*, Langkah-langkah untuk memastikan bahwa data tidak dapat diakses oleh orang yang tidak berwenang, memberikan jaminan autentikasi kepada pengguna.

B. *Security Mitigation*

Security mitigation merupakan strategi atau langkah-langkah yang diimplementasikan untuk mengurangi risiko serta melindungi sistem, jaringan, atau aplikasi dari kemungkinan ancaman keamanan. *Security mitigation* ini bertujuan untuk mencegah atau membatasi dampak yang dapat diakibatkan oleh potensi ancaman, seperti serangan siber, peretasan, atau celah keamanan yang dieksploitasi. Implementasi *security mitigation* melibatkan kebijakan keamanan yang ketat, penerapan teknologi keamanan yang canggih, pemantauan aktif terhadap ancaman yang baru muncul, serta meningkatkan kesadaran dan pelatihan keamanan bagi pengguna sistem.

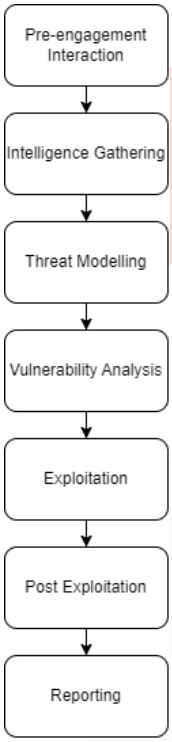
C. *Website*

Website adalah kumpulan dari halaman-halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di dalam internet [4]. Sedangkan, menurut [5], Pengertian

website adalah kumpulan halaman web yang tersedia di jaringan internet dan memiliki *Uniform Resource Locator* (URL), yang dapat diakses oleh setiap pengguna internet dengan mengetikan alamat domain tersebut.

D. *Penetration Testing Execution Standard* (PTES)

PTES adalah kerangka kerja pengujian penetrasi dengan panduan langkah demi langkah untuk menguji keamanan sistem, aplikasi, atau infrastruktur IT. Kelebihan PTES adalah kemudahannya. Tidak hanya menjelaskan tiap tahap, tetapi juga panduan teknis termasuk alat dan teknik penetrasi [6]. Langkah-langkah PTES seperti berikut:



Gambar II. 1
Gambar Tahapan PTES

1. *Pre-engagement Interaction*: Meminta izin dan merencanakan pengujian.
2. *Intelligence Gathering*: Mengumpulkan info target dari berbagai sumber.
3. *Threat Modeling*: Menentukan serangan berdasarkan pemahaman target.
4. *Vulnerability Analysis*: Cari kerentanan berdasarkan info sebelumnya.
5. *Exploitation*: Coba eksploitasi target, mungkin ada kejutan keamanan.
6. *Post-Exploitation*: Analisis infrastruktur setelah eksploitasi.
7. *Reporting*: Laporan penting untuk jelaskan hasil, risiko, dan perbaikan.

E. Nmap

Nmap adalah perangkat lunak open-source yang digunakan untuk menemukan alamat IP yang bisa digunakan dalam kerangka kerja kita. Nmap akan membantu menemukan port dan layanan yang tersedia. Nmap juga akan digunakan untuk menemukan dan menyalahgunakan kerentanan dalam suatu system [7].

F. Nslookup

Name *Server Lookup (Nslookup)* adalah tools yang dapat digunakan untuk menemukan alamat IP *domain* yang terkait dengan nama domain yang diberikan untuk menemukan nama server. *Nslookup* memiliki dua mode: interaktif dan non interaktif. Dalam mode interaktif, dia dapat menggunakan argumen seperti program biasa. Dalam mode non interaktif, informasi yang dicari hanya didasarkan pada argumen yang telah disiapkan sebelumnya, yang sesuai dengan webserver yang akan ditujui [8].

G. SQLMap

SQLMap adalah aplikasi *open source* atau *tool* yang terdapat dalam Kali Linux. Aplikasi ini digunakan untuk mendeteksi dan mengeksploitasi kerentanan aplikasi web. Aplikasi ini mampu mengambil alih *server database*. Dengan menggunakan SQLMap penyerang atau tester dapat melakukan penyerangan pada *database SQL*, menjalankan perintah pada sistem operasi, mengambil detail struktur database, melihat atau menghapus data yang terdapat dalam databas dan bahkan mengakses sistem file dari server. SQLMap mendukung enam teknik injeksi SQL: *Boolean–based blind, Time–based blind, Error based, UNION–based, Inteferential*, dan *Out–of–band* [9].

H. Burp Suite

Burp Suite, sebuah aplikasi pengujian keamanan, berfungsi untuk mengidentifikasi dan melaporkan celah keamanan pada aplikasi *web* terkait *SQL Injection*. Aplikasi ini membantu mengidentifikasi input yang tidak terlindungi yang memungkinkan penyerang memasukkan kode SQL berbahaya. *Burp Suite* memberikan laporan dan detail terkait serangan, termasuk informasi tentang sumber kerentanannya, parameter yang terkena dampak, dan hasil eksekusi *query SQL* yang berpotensi merusak. Ini merupakan alat yang berguna bagi auditor keamanan, peneliti, dan penguji untuk menganalisis sistem dan mencari celah keamanan [10].

I. SQL Injection

SQL Injection adalah tindakan peretasan yang dilakukan pada aplikasi klien dengan memanipulasi perintah *SQL* yang ada di memori aplikasi klien. *SQL Injection* merupakan metode eksploitasi pada aplikasi web di mana perintah-perintah yang dimasukkan dapat diubah sesuai keinginan pengguna, bahkan jika pengguna tersebut tidak memiliki izin untuk melakukannya. Dampak yang ditimbulkan oleh pengguna yang melakukan *SQL Injection* dapat sangat fatal karena dapat menyebabkan kerusakan serius [11].

J. jSQL Injection

jSQL Injection adalah alat untuk menemukan dan mengeksploitasi kerentanan *SQL Injection* pada aplikasi *web*. Fungsinya adalah mengidentifikasi titik masukan yang rentan, menyisipkan kode *SQL* berbahaya, dan mencoba mendapatkan akses tidak sah ke basis data aplikasi target. Serangan *jSQL Injection* dapat menghasilkan data sensitif dari basis data, pesan kesalahan yang mengungkapkan struktur basis data, atau tidak ada hasil jika serangan tidak berhasil menemukan celah yang dapat dieksploitasi pada aplikasi *web* target.

K. Havii

Havij adalah tools penetrasi yang berfungsi untuk mendeteksi kerentanan *SQL Injection* pada situs *web*. Fungsi dari *tools Havij* adalah untuk mengidentifikasi dan menguji tingkat keamanan suatu situs *web* dengan mencari celah keamanan di mana serangan *SQL Injection* bisa dilakukan. *Output* yang diperoleh dari penggunaan *Havij* berupa laporan uji penetrasi yang mencakup informasi mengenai rentannya situs terhadap serangan *SQL Injection* dan apabila berhasil, data yang berhasil diekstraksi dari database, seperti tabel, kolom, dan isi lainnya.

L. Teknik *Penetration Testing*

Penetration testing adalah simulasi serangan siber untuk menemukan kelemahan keamanan pada sistem target. Ada tiga metode dalam pengujian penetration testing:

1. *Black box testing,* Pengujian dilakukan tanpa detail sistem target. Serangan dari luar, tanpa akses kode program atau konfigurasi.
2. *White box testing*, Pengujian dengan akses detail sistem. Serangan dari dalam, melihat kode dan konfigurasi seperti *administrator*.
3. *Grey box testing*, Pengujian dengan akses terbatas detail sistem. Serangan dari dalam dan luar, seperti *hacker* dan *administrator*. Akses terbatas mengakibatkan pengujian tidak menyeluruh.

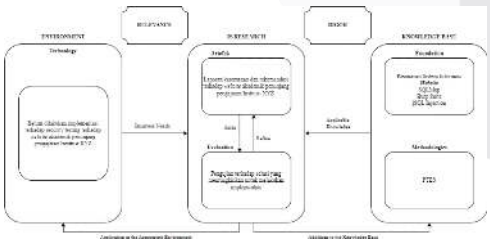
M. Penelitian Terdahulu

Pengerjaan tugas akhir ini dibuat mengacu pada referensi penelitian yang telah dilakukan sebelumnya yang terkait dengan *penetration testing* dengan referensi yang mengacu pada *Vulnerability Assessment* [12]-[15].

III. METODE PENELITIAN

A. Kerangka Berfikir

Menurut Hevner dan Chatterjee (2010), model konseptual adalah representasi visual atau deskriptif dari sistem atau proses yang menggambarkan komponen-komponen utama, hubungan antar komponen, serta dampak faktor *exsternal*.

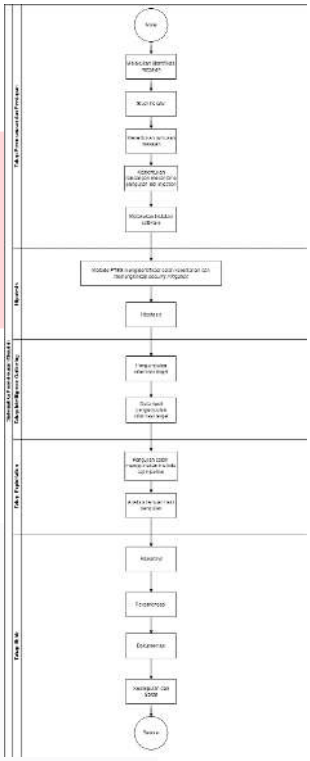


Gambar III. 1
Metode Konseptual

1. Penelitian ini mengaplikasikan teknologi untuk menguji kemampuan aplikasi dalam membatasi akses yang tidak sah pada *Website* Akademik Penunjang Pengajaran, karena kurangnya uji keamanan pada aplikasi tersebut.
2. *IS Research* membantu menyaring dan mengevaluasi solusi celah keamanan yang dikembangkan oleh pihak terkait melalui artefak (solusi) dan evaluasi (pengujian).
3. Penelitian ini menggunakan Knowledge Base yang mencakup Keamanan Sistem Informasi, *Website, SQLMap, Burp Suite*, serta metode PTES.

B. Sistematika Penyelesaian Masalah

Sistematika penyelesaian masalah melibatkan langkah-langkah terstruktur untuk menuntaskan masalah dalam penelitian. Ini terdiri dari tahap perencanaan, persiapan, *intelligence gathering*, eksploitasi, dan tahap akhir, seperti diilustrasikan dalam diagram di Gambar III.2.



Gambar III. 2 Sistematika Penyelesaian Masalah

Dalam tahap penelitian, langkah-langkah yang digunakan meliputi:

1. Tahap Perencanaan dan Persiapan: Identifikasi masalah, batasan, tujuan, dan rancangan pengujian *SQL Injection*. Instalasi perangkat lunak.
2. Hipotesis: Menggunakan metode PTES untuk identifikasi celah kerentanan dan tahap mitigasi keamanan.
3. Tahap Intelligence Gathering: Kumpulkan informasi target dan data dengan alat *Nmap* dan *Nslookup*.
4. Tahap Eksploitasi: Uji celah dengan SQL Injection menggunakan *SQLmap, Burp Suite, jSQL Injection, Havij*. Analisis dampak dan resiko, serta hasilnya.
5. Tahap Akhir: Laporan hasil penelitian, dokumentasi, rekomendasi, kesimpulan, dan saran untuk mengatasi celah keamanan di masa mendatang pada *Website* target.

C. Pengumpulan Data

Tahap pengumpulan data melalui *SQL Injection* melibatkan informasi Website Akademik Penunjang Pengajaran di Institusi XYZ yang akan diuji dan perencanaan penetration testing, dikenal sebagai *intelligence gathering* dalam metode PTES.

D. Pengolahan Data

Pada tahap pengolahan data, peneliti menganalisis hasil dari *intelligence gathering* untuk mengevaluasi celah kerentanan *SQL Injection* pada *Website* Akademik Penunjang Pengajaran Institusi XYZ.

E. Metode Evaluasi

Tahap evaluasi melibatkan pengujian celah kerentanan *SQL Injection* pada *Website* Akademik Penunjang Pengajaran Institusi XYZ yang telah diidentifikasi. Melalui eksploitasi menggunakan alat pengujian, tujuannya adalah memverifikasi efektivitas metode evaluasi pada *Website* tersebut.

F. Alasan Pemilihan Metode

Dalam penelitian ini, digunakan metode PTES yang merupakan standar dalam audit dan analisis keamanan sistem di perusahaan. PTES memberikan tahapan spesifik untuk memeriksa celah keamanan dan menganalisis ancaman. Tingkat evaluasi yang disediakan oleh PTES mudah dipahami oleh pengguna dengan berbagai tingkat keahlian dalam pengujian penetrasi.

IV. HASIL DAN PEMBAHASAN

A. Perencanaan dan Skenario

1. Perencanaan dan Persiapan

Dalam eksploitasi, penelitian ini memakai hardware dan software yang direncanakan sebelumnya. Teknik yang digunakan adalah *Black Box*, di mana pengujian dilakukan tanpa detail sistem dan jaringan yang diuji, dengan kerangka kerja PTES. Spesifikasi *hardware* tertera dalam tabel berikut,

Tabel IV. 1
Spesifikasi Hardware

Komponen	Spesifikasi	
Main OS	*Processor*	AMD Ryzen 5 5600H with Radeon Graphics (12 CPUs), ~3.3GHz
	Memory	16384 MB
	Hard Disk	942,67 GB
	System Type	64-bit
	Operating System	Windows 11 Home Single Language 64-bit (10.0, Build 22621)
Virtual Machine	*Processor*	AMD Ryzen 5 5600H with Radeon Graphics (12 CPUs), ~3.3GHz
	Memory	4096 MB
	Hard Disk	30 GB
	System Type	64-bit
	Operating System	Kali Linux

Selanjutnya terdapat spesifikasi *software* dalam table sebagai berikut,

Tabel IV. 2
Spesifikasi Software

Tipe	Informasi *Software*	*Version*
Operating System	Kali Linux	2023.2
Virtual Machine	VMware Workstation	17.0.0 build-20800
Exploit Tools	SQLMap	1.7.6
	Burp Suite	Community Editic v2023.5.4
	jSQL *Injection*	v0.85
	Havij	1.12 *free* edition
Intelligence Gathering Tools	Nmap	7. 94
	Nslookup	9.18.13-1-Debia

Hardware dan *software* ini nantinya akan digunakan dalam pengujian.

2. *Intelligence Gathering*

Pada *Intelligence Gathering*, peneliti menjalankan eksplorasi informasi dengan tools,

a. *Nslookup* untuk mengidentifikasi IP address Website target (103.41.206.192) menggunakan *command nslookup* xxx-xxx-xxx.xxxxxxxxxx.id.

b. *Nmap* digunakan dengan command nmap -sT xxx-xxx-xxx.xxxxxxxxxx.id untuk pemindaian TCP Connect Scan.

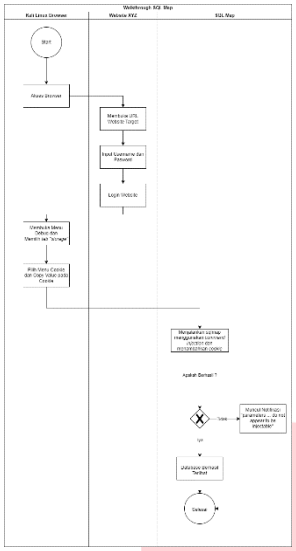
3. *Eksploitasi*

Eksploitasi adalah tahap menguji kerentanan. Pengujian dilakukan pada titik masuk kerentanan yang berpotensi diserang. Pertama, tentukan *tools* sesuai kerentanan setelah menganalisis celah keamanan pada website target. Jika berhasil, analisis hasil eksploitasi dilakukan.

Gambar IV. 1
Skenario Eksploitasi

a. Pengujian *SQL Injection* Menggunakan *Tools SQLMap*

Berikut merupakan langkah pengujian *SQL Injection*,

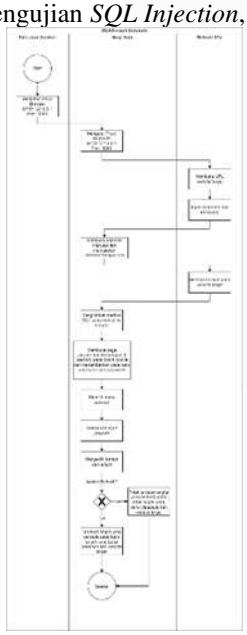


Gambar IV. 2
Langkah Pengujian SQL Injection Menggunakan Tools SQLMap

Penyerang pertama membuka browser di Kali Linux, mengakses URL target, dan *login*. Kemudian, dengan menu *debug*, penyerang mengambil *value cookie*. Setelah itu, menggunakan *SQLMap* dengan *cookie* yang diambil, mencoba masuk ke database target. Jika berhasil, penyerang bisa eksploitasi *database*. Jika tidak, akan ada *warning "parameters ... do not appear to be injectable"*.

b. Pengujian *SQL Injection* Menggunakan *Tools Burp Suite*

Berikut langkah pengujian *SQL Injection*,

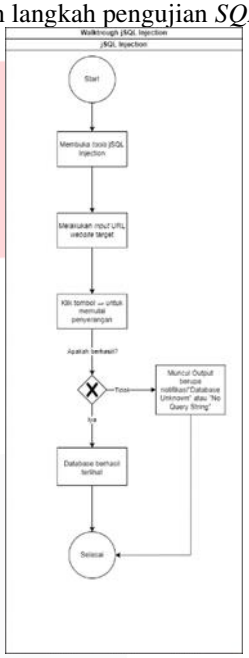


Gambar IV. 3
Langkah Pengujian SQL Injection Menggunakan Tools Burp Suite

Penyerang pertama membuka *browser* pada Kali Linux dan mengatur proxy dengan HTTP: 127.0.0.1 dan Port: 8080. Kemudian, dengan *tools Burp Suite* di Kali Linux, penyerang juga mengatur proxy dengan ketentuan yang sama. Setelah itu, penyerang membuka URL target pada *browser* Kali Linux, melakukan *login*, dan mengaktifkan *intercept* pada *Burp Suite*. Setelah berhasil *login*, penyerang mengirimkan *method GET* yang muncul pada halaman *intercept* ke *intruder* pada *Burp Suite*. Di halaman *intruder*, penyerang mengganti *cookie* dengan *username* dan *password* pada *payload*. Penyerang memilih *payload* yang disediakan, memulai penyerangan dengan mengklik *Start Attack*. Jika berhasil, akan terlihat angka yang berbeda pada baris *length*, menunjukkan *Database* dari *website* target. Jika gagal, tidak akan ada perbedaan *length* dan *Database* tidak ditemukan.

c. Pengujian SQL Injection Menggunakan *Tools jSQL Injection*

Berikut merupakan langkah pengujian *SQL Injection*,

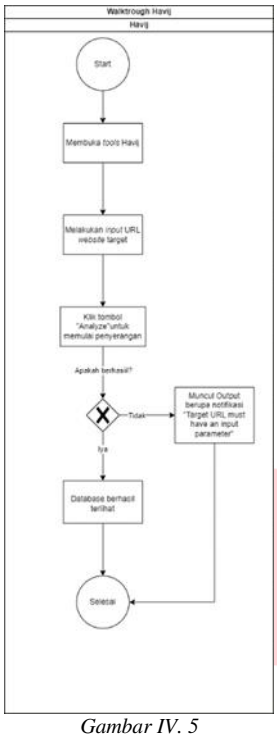


Gambar IV. 4
Langkah Pengujian SQL Injection Menggunakan Tools jSQL Injection

Penyerang memulai dengan membuka *tools jSQL Injection* pada Kali Linux. Setelah berhasil membuka *tools* tersebut, penyerang menginput URL *website* target dan mengklik tombol → untuk memulai penyerangan. Jika Injection berhasil, maka *database* dari *website* target akan terlihat. Namun, jika *Injection* tidak berhasil, akan muncul notifikasi *"Database unknown"* atau *"No query string"*.

d. Pengujian *SQL Injection* Menggunakan *Tools Havij*

Berikut merupakan langkah pengujian *SQL Injection*,



Gambar IV. 5
Langkah Pengujian SQL Injection Menggunakan Tools Havij

Penyerang memulai dengan membuka *tools Havij*. Setelah berhasil membuka *tools* tersebut, penyerang menginput URL *website* target dan mengklik tombol *"Analyze"* untuk memulai penyerangan. Jika *Injection* berhasil, maka *database* dari *website* target akan terlihat. Namun, jika *Injection* tidak berhasil, akan muncul notifikasi *"Database unknown"* atau *"Target URL must have an input parameter"*.

4. Reporting

Reporting adalah langkah untuk melaporkan hasil dari *Intelligence Gathering* dan Eksploitasi terkait celah kerentanan yang ditemukan. Pada tahap ini, *developer* diberitahu tentang kerentanan berdasarkan eksploitasi. Jika diperbaiki, akan dilakukan pengujian ulang. Jika tidak, penelitian memberikan rekomendasi kepada *developer* terkait kerentanan tersebut.

B. Hasil dan Pembahasan

1. Hasil Eksploitasi dengan Pengujian SQL Injection

a. Hasil Eksploitasi dengan pengujian *SQL Injection* menggunakan *tool SQLMap*

Dengan menggunakan *SQLMap*, dilakukan serangan *SQL Injection* pada *website* target Akademik Penunjang Pengajaran Institusi XYZ untuk mengidentifikasi potensi celah kerentanan. Serangan menggunakan perintah:

sqlmap -u "http://xxx-xxx-xxx.xxxxxxxxxx.id" --cookie="ci_session=d87ba1a75ecd33027fc5372e61841c8454f8bc32" --dbms=mysql --random-agent --tamper=apostrophemask,apostrophenullencode --level=5 --risk=3

```
[04:53:29] [CRITICAL] all tested parameters do not appear to be injectable
[04:53:29] [WARNING] HTTP error codes detected during run:
404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times

[*] ending @ 04:53:29 /2023-07-25/
```

Gambar IV. 6
Hasil Injection Menggunakan Tool SQLMap

Pada Gambar diatas terlihat hasil serangan *SQL Injection* menggunakan *SQLMap* pada penelitian ini. Saat melakukan injeksi melalui *SQLmap*, serangan tidak berhasil menembus sistem target. *Output alert* menunjukkan *"critical all tested parameters do not appear to be injectable,"* dan terdapat juga *warning "HTTP error codes detected during run: 404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times."*

b. Hasil Eksploitasi dengan pengujian *SQL Injection* menggunakan *tools Burp Suite*

Penelitian ini mencoba serangan dengan *payload* terhadap *Website* Akademik Penunjang Pengajaran Institusi XYZ untuk melihat celah kerentanan *SQL Injection* yang mungkin ada.

```
12 Cookie: ci_session=721fdf12526d477e344ed89d70c59002578d85f0
```

Gambar IV. 7
Hasil Cookie yang didapatkan menggunakan tool Burp Suite

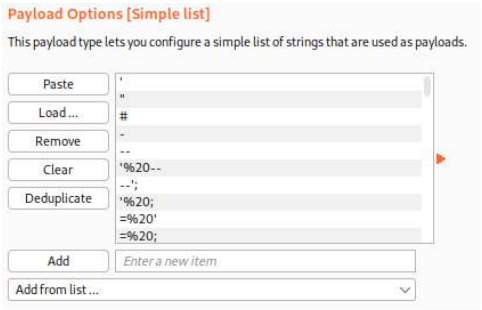
Gambar diatas menunjukkan hasil *cookie* yang diperoleh melalui penggunaan *Burp Suite*. Ini terjadi setelah peneliti mengaktifkan *intercept* dan *proxy* pada *tools Burp Suite* dan *browser Kali Linux* saat mengunjungi *website* akademik penunjang pengajaran di Institusi XYZ.

```
Cookie: ci_session=721fdf12526d477e344ed89d70c59002578d85f0
Upgrade-Insecure-Requests: 1

textUsername=§12021941235&textPassword=§123§
```

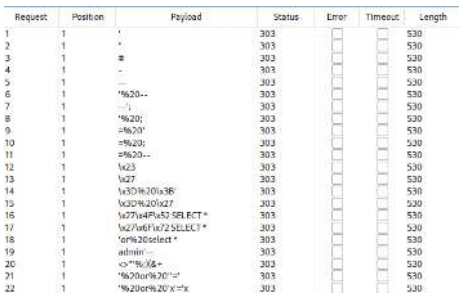
Gambar IV. 8
Hasil Penghapusan Section pada Cookie

Selanjutnya, *cookie* dikirim ke *intruder* dan *section* (§) pada *cookie* dihapus, sementara *section* (§) pada *"textUsername"* dan *"textPassword"* ditambahkan, seperti yang terlihat pada Gambar diatas. Section (§) pada *intruder* berfungsi untuk menentukan lokasi di mana *payload* akan dimasukkan dan dimodifikasi. Dengan menggunakan *section* (§), pengujian keamanan dapat dilakukan secara tepat dan terfokus pada area yang relevan tanpa mempengaruhi bagian lain dari permintaan yang tidak perlu diubah.



Gambar IV. 9
Menu Payload

Gambar diatas menunjukkan peneliti memilih *menu payload* dan memasukkan *payload* acak, yang digunakan untuk meluncurkan serangan, diikuti dengan mengeklik tombol *Start Attack.*

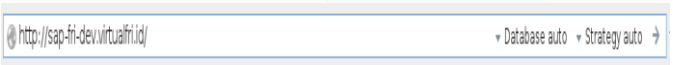


GAMBAR IV. 10
Hasil Penyerangan pada Tool Burp Suite menggunakan Payload

Gambar diatas menampilkan hasil penyerangan yang dilakukan. Namun, serangan tidak berhasil meretas target karena tidak ada perubahan dalam status *code* dan panjang data, mengindikasikan bahwa *website* target memiliki sistem keamanan yang kokoh terhadap serangan *SQL Injection*.

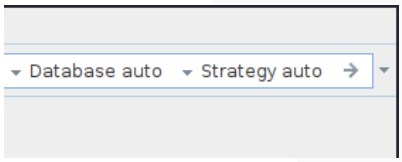
c. Hasil Eksploitasi dengan pengujian SQL *Injection* menggunakan *tools* jSQL *Injection*

Menggunakan *tools* jSQL Injection, serangan SQL Injection dilakukan pada *Website* Akademik Penunjang Pengajaran Institusi XYZ untuk mengidentifikasi celah kerentanan dalam bentuk *database* yang mungkin ada pada *website* target.



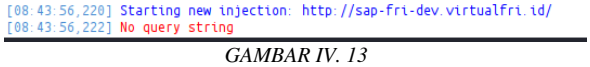
GAMBAR IV. 11
Input URL Tool jSQL Injection

Pada gambar diatas dilakukan *input* URL *Website* Akademik Penunjang Pengajaran Institusi XYZ yang telah di tentukan menjadi *website* target.



GAMBAR IV. 12
Tombol start Injection

Pada Gambar diatas, peneliti memulai injeksi dengan menekan tombol start Injection yang dilambangkan dengan simbol tanda panah. Proses ini menggunakan opsi *"Database auto"* dan *"Strategy auto"*. Opsi *"Database auto"* secara otomatis mendeteksi tipe basis data yang digunakan oleh website target. Setelah mendeteksi tipe basis data, opsi *"Strategy auto"* akan secara otomatis menentukan strategi serangan yang efektif terhadap target, dengan mencoba berbagai strategi serangan secara otomatis untuk menemukan celah pada website target.

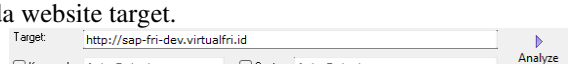


GAMBAR IV. 13
Hasil injeksi menggunakan Tool jSQL Injection

Pada gambar diatas terlihat hasil injeksi yang sudah di lakukan berupa output *No query string*.

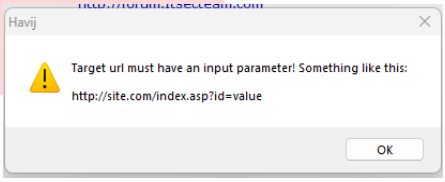
d. Hasil Eksploitasi dengan pengujian SQL *Injection* menggunakan *tools* Havij

Dengan tools Havij, serangan SQL Injection dilakukan pada Website Akademik Penunjang Pengajaran Institusi XYZ untuk mencari potensi celah kerentanan berupa database pada website target.



GAMBAR IV. 14
Input URL Tool Havij

Pada gambar diatas dilakukan *input* URL *Website* Akademik Penunjang Pengajaran Institusi XYZ yang telah di tentukan menjadi *website* target pada *tool* Havij dan melakukan klik tombol “Analyze” untuk mulai melakukan penyerangan.



GAMBAR IV. 15
Notifikasi Tool Havij

Pada gambar diatas terlihat hasil output berupa *pop-up* yang bertuliskan *“Target url must have an input parameter!”*

2. Hasil Output Tools Berdasarkan Eksploitasi dengan Pengujian SQL Injection

Berdasarkan hasil pengujian dan analisis yang dilakukan, berikut adalah output dari alat-alat yang digunakan:

TABEL IV. 3
Hasil Output Tools

Host	*Tools*	*Output*
Web akademik penunjang pengajaran Institusi XYZ	SQLMap	*Critical all tested parameters do not appear to be injectable, HTTP error codes detected during run: 404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times*
	Burp Suite	-
	jSQL *Injection*	*No query string*
	Havij	*pop-up* yang bertuliskan *"Target url must have an input parameter!"*

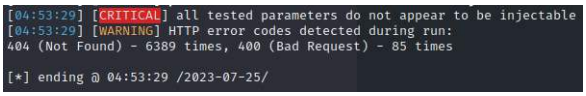
Hasil output dari eksploitasi adalah sebagai berikut berdasarkan tabel berikut:

a. Pada tools SQLMap, ditemukan output HTTP error codes detected during run: 404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times.
b. Pada tools Burp Suite, tidak ditemukan output karena tidak adanya length yang berbeda pada halaman intruder attack dari hasil pengujian menggunakan payload.
c. Pada tools jSQL Injection, output yang ditemukan adalah "No query string".
d. Pada tools Havij, muncul pop-up dengan tulisan "Target url must have an input parameter!".

3. Hasil Analisis Eksploitasi dengan Pengujian SQL Injection

a. Hasil Analisis Eksploitasi dengan pengujian SQL *Injection* menggunakan *tool* SQLMap

Hasil analisis eksploitasi dengan pengujian SQL Injection menggunakan tool SQLMap adalah sebagai berikut:

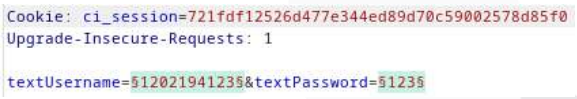


GAMBAR IV. 16
Analisis Hasil Output tool SQLMap

Selama 2 jam, dilakukan proses eksploitasi dengan menggunakan command injeksi yang sudah disiapkan. Pada Gambar berikut, hasil output menunjukkan "all tested parameters do not appear to be injectable" dan "HTTP error codes detected during run: 404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times". Output "all tested parameters do not appear to be injectable" menunjukkan bahwa SQLMap tidak menemukan kerentanan SQL Injection pada parameter yang diuji. Respons "404 Not Found" dan "400 Bad Request" dari server menandakan bahwa SQLMap gagal menemukan halaman yang berhubungan dengan SQL Injection pada sejumlah URL yang diuji. Dari analisis output, dapat disimpulkan bahwa website akademik penunjang pengajaran Institusi XYZ memiliki keamanan yang baik terhadap serangan SQL Injection.

b. Hasil Analisis Eksploitasi dengan pengujian SQL *Injection* menggunakan tool Burp Suite

Hasil analisis eksploitasi dengan pengujian SQL *Injection* menggunakan *tool* Burp Suite pada gambar dibawah ini adalah sebagai berikut:



GAMBAR IV. 17
Cookie yang didapatkan menggunakan tool Burp Suite

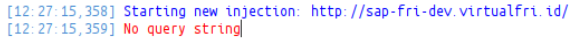
Proses eksploitasi berlangsung selama 10 menit dengan menggunakan payload yang telah disiapkan. Pada method GET, hasil intercept dari halaman diarahkan ke intruder Burp Suite untuk menghapus bagian tertentu dari cookie. Fungsi section pada Burp Suite digunakan untuk menentukan payload dalam uji coba serangan dan section (§) pada intruder menunjukkan area target di mana payload akan dimasukkan dan disesuaikan.

GAMBAR IV. 18
Analisis Hasil Output tool Burp Suite

Pada gambar diatas, tidak ada output yang dihasilkan karena tidak ada perubahan pada length data dan status code setelah memasukkan payload dan memulai serangan. Hal ini disebabkan oleh adanya filter validasi yang ketat atau mekanisme keamanan pada website Institusi XYZ, yang mencegah payload mencapai server aplikasi dan menghindari penyisipan kode SQL berbahaya. Berdasarkan analisis output tersebut, dapat disimpulkan bahwa website akademik penunjang pengajaran Institusi XYZ memiliki keamanan yang baik dan aman dari serangan SQL Injection karena sulitnya menemukan celah yang memungkinkan pencurian database dan manipulasi data pengguna.

c. Hasil Analisis Eksploitasi dengan pengujian SQL *Injection* menggunakan tool jSQL *Injection*

Hasil analisis eksploitasi dengan pengujian SQL *Injection* menggunakan *tool* jSQL *Injection* adalah sebagai berikut:

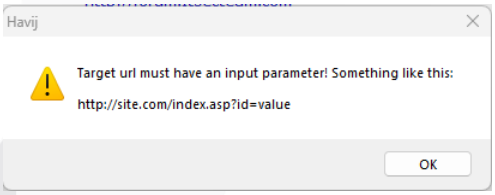


GAMBAR IV. 19
Analisis Hasil Output tool jSQL Injection

Proses eksploitasi berlangsung selama 3 menit dengan menggunakan URL website target yang telah disiapkan. Pada gambar diatas, terlihat output "No query string," menandakan ketiadaan parameter query string yang dapat dimanipulasi pada website target. Oleh karena itu, tools jSQL Injection tidak dapat menemukan celah untuk serangan SQL Injection melalui URL tersebut. Hasil analisis menunjukkan bahwa website tersebut memiliki tingkat keamanan yang baik, dan tidak rentan terhadap serangan SQL Injection pada parameter query string. Kesimpulannya, pengujian tidak menemukan celah kerentanan SQL Injection pada website tersebut.

d. Hasil Analisis Eksploitasi dengan pengujian SQL *Injection* menggunakan *tool* Havij

Hasil analisis eksploitasi dengan pengujian SQL *Injection* menggunakan *tool* jSQL *Injection* adalah sebagai berikut



GAMBAR IV. 20 A
nalisis hasil output tools Havij

Proses eksploitasi berlangsung selama 1 menit dengan menggunakan URL website target yang telah disiapkan. Pada gambar diatas, muncul output berupa pop-up yang menyatakan "Target url must have an input parameter," menunjukkan bahwa tidak ada parameter yang dapat diserang pada website target. Karena itu, tools Havij tidak berhasil menemukan celah untuk serangan SQL Injection pada website tersebut, dan menghasilkan output pop-up dengan pesan tersebut. Hasil analisis menunjukkan bahwa website tersebut memiliki tingkat keamanan yang baik dan tidak rentan terhadap serangan SQL Injection pada parameter query string. Kesimpulannya, pengujian tidak menemukan celah kerentanan SQL Injection pada website tersebut.

4. *Reporting*

Pada Website Akademik Penunjang Pengajaran di institusi XYZ, hasil pengujian menggunakan tools

SQLMap, Burp Suite, jSQL Injection, dan Havij menunjukkan adanya potensi resiko manipulasi data seperti mencuri informasi sensitive dari database. Sebelum pengujian dilakukan, terdapat celah resiko penyerangan dengan metode SQL Injection. Namun, hasil pengujian dengan tools SQLMap, Burp Suite, dan jSQL Injection menunjukkan bahwa celah resiko penyerangan dengan metode SQL Injection tidak ditemukan. Informasi lebih lanjut dapat dilihat pada Tabel dibawah yang menampilkan hasil eksploitasi dari pengujian SQL Injection menggunakan tools yang telah digunakan.

TABEL IV. 4
Informasi Hasil Eksploitasi

Tools	Eksperimen	*Output*	Keterangan
SQLMap	Melakukan inject menggunakan command: SQLmap -u "http://xxx-xxx-xxx.xxxxxxxxx.id" --cookie="ci_session=d87ba1a75ed330271c5372e61841c8454f8bc32" --dbms=mySQL --randomt-agent --tamper=apostrophemask,apostrophenullencode --level=5 --risk=3	*Critical all tested parameters do not appear to be injectable*, dan dan *"HTTP error codes detected during run: 404 (Not Found) - 6389 times, 400 (Bad Request) - 85 times"*	SQLMap telah menguji semua *paran* yang ditentukan namun tidak menem kerentanan SQL *Injection* yang terlih *parameter* tersebut.
Burp Suite	Melakukan inject menggunakan *payload* yang sudah disediakan pada halaman intruder	Tidak terdapat *status code* dan *length data* yang berbeda/berubah	Tidak ada *status code* dan *length* dan berbeda setelah menginputkan *payloa* memulai serangan, disebabkan oleh filter validasi yang ketat atau mekani keamanan yang menghalangi *payloa* mencapai server aplikasi.
jSQL *Injection*	Melakukan serangan dengan menginputkan *url website* target menggunakan opsi "*Database* auto" dan "Strategy auto"	*No query string*	Tidak ada *string query* yang terdeteks URL atau *parameter* yang diuji. Hal berarti bahwa jSQL tidak menemuka *parameter* yang dapat dieksploitasi u melakukan serangan SQL *Injection*.
Havij	Melakukan serangan dengan menginputkan *url website* target	*Pop-up* berupa *"Target url must have an input parameter"*	Tidak adanya *parameter* yang terdete dalam URL *website* target. Hal ini da berarti bahwa jSQL tidak menemuka *parameter* yang dapat dieksploitasi u melakukan serangan Havij.

Hasil pengujian SQL Injection menggunakan beberapa tools menunjukkan bahwa tidak ada celah kerentanan yang ditemukan pada Website Akademik Penunjang Pengajaran Institusi XYZ. Penggunaan tools seperti SQLMap, Burp Suite, jSQL Injection, dan Havij telah diuji, tetapi tidak ada tanda-tanda celah yang dapat dieksploitasi. Setiap tools memberikan hasil yang serupa: tidak ada parameter yang dapat dimanipulasi melalui URL, sehingga serangan SQL Injection tidak berhasil dilakukan. Oleh karena itu, dapat disimpulkan bahwa langkah-langkah pengujian eksploitasi SQL Injection yang dilakukan pada website tersebut telah mengonfirmasi penutupan celah kerentanan SQL Injection.

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil analisis potensi risiko keamanan pada Website Akademik Penunjang Pengajaran di Institusi XYZ dengan menggunakan alat seperti SQLMap, Burp Suite, dan jSQL Injection, kami menemukan hal berikut:

1. Hasil analisis menunjukkan bahwa pengujian keamanan dengan metode SQL Injection pada website akademik di Institusi XYZ menemukan sedikit atau bahkan tidak ada celah kerentanan. Hal ini mengindikasikan bahwa upaya manipulasi data melalui SQL Injection tidak terdeteksi.
2. Rekomendasi yang dihasilkan adalah untuk tetap menjaga tingkat keamanan website Institusi XYZ agar tidak rentan terhadap potensi celah keamanan SQL Injection. Meski demikian, penting untuk diingat bahwa kemungkinan celah keamanan masih ada, terutama dengan adanya pembaruan atau metode serangan baru. Oleh karena itu, para pengembang disarankan untuk melakukan perawatan keamanan secara rutin dan melakukan pemeriksaan keamanan setelah melakukan pembaruan guna melindungi website dari risiko potensial celah keamanan.

B. Saran

Berdasarkan hasil analisis resiko celah keamanan yang telah di lakukan terhadap Website Akademik Penunjang Pengajaran Institusi XYZ menggunakan tools Nmap, Whois, SQLMap, Burp Suite, jSQL. Berikut saran-saran yang dapat diberikan untuk membantu pengembangan penelitian selanjutnya, yaitu:

1. Pada penelitian selanjutnya agar mempertimbangkan pemilihan tool yang akan digunakan untuk mengidentifikasi kerentanan keamanan yang beragam, akurat, dan lebih luas dari penelitian ini.
2. Pada penelitian selanjutnya disarankan menggunakan metode yang berbeda untuk
3. mendapatkan hasil yang berbeda dari penelitian ini.
4. Pada penelitian selanjutnya disarankan menggunakan framework yang berbeda agar dapat memperoleh hasil pengujian celah kerentanan yang lebih baik.
5. Disarankan melakukan maintenance secara rutin terhadap website terkait untuk meminimalisir celah keamanan.

REFERENSI

[1] S. Nurul, S. Anggrainy, dan S. Aprelyani, "FAKTOR-FAKTOR YANG MEMPENGARUHI KEAMANAN SISTEM INFORMASI: KEAMANAN INFORMASI, TEKNOLOGI INFORMASI DAN NETWORK (LITERATURE REVIEW SIM)," vol. 3, no. 5, 2022, doi: 10.31933/jemsi.v3i5.

[2] A. Ramadhani, "KEAMANAN INFORMASI," 2018. doi: https://doi.org/10.30999/n-jils.v1i1.249.

[3] A. Hermawan, T. Hartati, dan Y. A. Wijaya, "Analisa Keamanan Data melalui Website Zahra Software Menggunakan Metode Keamanan Informasi CIA Triad," vol. 7, no. 3, 2022.

[4] A. Josi, K. Akuntansi, S. Prabumulih, J. L. Patra No, K. Sukaraja, dan K. P. Selatan, "STMIK-MUSIRAWAS LUBUKLINGGAU 50 PENERAPAN METODE PROTOTIPING DALAM PEMBANGUNAN WEBSITE DESA (STUDI KASUS DESA SUGIHAN KECAMATAN RAMBANG)," 2017.

[5] A. Ibrahim, A. Ambarita, dan P. Sains dan Teknologi Wiratama Maluku Utara, "INFORMATION SYSTEMS WATER CUSTOMERS COMPLAINTS WEB-BASED ON PDAM TERNATE CITY," 2018. doi: https://doi.org/10.36549/ijis.v3i1.37.

[6] S. U. Sunaringtyas, D. Surya Prayoga, J. K. Siber, P. Siber, dan S. Negara, "Edu Komputika Journal Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada Layanan Single Sign-On," 2021. [Daring]. Tersedia pada: http://journal.unnes.ac.id/sju/index.php/edukom

[7] K. Maha Vidyalaya, "Penetration Testing-Reconnaissance with NMAP Tool," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, [Daring]. Tersedia pada: www.ijarcs.info

[8] C. N. Shivayogimath, "AN OVERVIEW OF NETWORK PENETRATION TESTING." [Daring]. Tersedia pada: http://www.ijret.org

[9] S. Lika, R. Dwi, P. Halim, dan I. Verdian, "Positif : Jurnal Sistem dan Teknologi Informasi ANALISA SERANGAN SQL INJEKSI MENGGUNAKAN SQLMAP Implementation Of Online Accounting Software As Supporting Of Financial Statement," vol. 4, no. 2, 2018, doi: https://doi.org/10.31961/positif.v4i2.610.

[10] A. Subari, S. Manan, E. Ariyanto, dan A. Fauzi, "PEMANFAATAN METODE WAVS (WEB APPLICATION SECURITY SCANNERS) MENGGUNAKAN BURP SUITE TOOLS DALAM AUDIT TEKNIS KEAMANAN SISTEM INFORMASI SURAT TUGAS SEKOLAH VOKASI UNDIP," 2021, doi: 10.14710/gt.v21i4.46828.

[11] A. Putra Erriyanto dan A. Eviyanti, "Seminar Nasional & Call Paper Fakultas Sains dan Teknologi (SENASAINS 5 th," 2022.

[12] J. N. Goel dan B. M. Mehtre, "Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology," dalam *Procedia Computer Science*, Elsevier, 2015, hlm. 710–715. doi: 10.1016/j.procs.2015.07.458.

[13] *2019 4th International Conference on Information Systems and Computer Networks (ISCON).* IEEE.

[14] S. Widya Ningsih Nasir *dkk.*, "Vulnerability Testing Analysis of XYZ Regional Government Site Using PTES," vol. 8, no. 3, hlm. 1543–1556, 2021, [Daring]. Tersedia pada: http://jurnal.mdp.ac.id

[15] M. Orisa dan M. Ardita, "VULNERABILITY ASSESMENT UNTUK MENINGKATKAN KUALITAS KEMANAN WEB," 2021. doi: https://doi.org/10.36040/mnemonic.v4i1.3213.