DOI: https://doi.org/10.62017/tektonik

Pengujian Keamanan Aplikasi Perpustakaan Digital Desa Damai pada Sisi Autentikasi Menggunakan Serangan *SQL Injection*, *Session Hijacking,* dan *Brute Force*

Cahyo Prawiro *[1]
Jaroji [2]
Nurmi Hidayasari [3]

[1,2,3] Politeknik Negeri Bengkalis
*e-mail: cahyoprawiro787@gmail.com[1], jaroji@polbeng.ac.id[2], nurmihidayasari@polbeng.ac.id[3]

Abstrak

Perpustakaan digital Desa Damai menjadi aspek kunci dalam memberikan akses informasi, namun keamanan autentikasi belum diuji dengan baik. Penelitian ini mengidentifikasi kelemahan pada form login, termasuk kurangnya indikator password kuat dan kurangnya sanitasi karakter khusus. Uji keamanan dilakukan terhadap serangan SQL Injection, Session Hijacking, dan Brute Force dengan membandingkan hasil manual dan menggunakan standar OWASP. Hasil menunjukkan keberhasilan serangan Brute Force, menyoroti kebutuhan akan penguatan keamanan autentikasi. Penggunaan OWASP ZAP memberikan wawasan tambahan. Kesimpulan menekankan perlunya penguatan mekanisme login dan sesi, dengan rekomendasi seperti penggunaan token sesi dan CAPTCHA. Penelitian ini memberikan pemahaman lebih dalam tentang kerentanan keamanan aplikasi perpustakaan digital dan menggarisbawahi pentingnya kombinasi pengujian manual dan otomatis.

Kata Kunci: *Perpustakaan Digital, Keamanan Aplikasi, Autentikasi, SQL Injection, Session Hijacking, Brute Force, OWASP ZAP.*

Abstract

The Damai Village Digital Library serves as a key aspect in providing information access; however, the authentication security has not been thoroughly examined. This research identifies vulnerabilities in the login form, including the lack of strong password indicators and inadequate special character sanitization. Security testing is conducted against SQL Injection, Session Hijacking, and Brute Force attacks by comparing manual results with the OWASP standard. Results indicate successful Brute Force attacks, emphasizing the need for strengthening authentication security. The use of OWASP ZAP provides additional insights. Conclusions underscore the necessity to enhance login and session mechanisms, with recommendations such as the implementation of session tokens and CAPTCHA. This research offers a deeper understanding of security vulnerabilities in digital library applications, highlighting the importance of a combination of manual and automated testing.

Keywords: *Digital Library, Application Security, Authentication, SQL Injection, Session Hijacking, Brute Force, OWASP ZAP.*

PENDAHULUAN

Dalam era digital saat ini, perpustakaan digital menjadi alat penting untuk mengelola dan menyediakan akses informasi, memungkinkan pengguna untuk mengakses literatur dan buku elektronik secara online dengan mudah. Meski memberikan kemudahan, aspek keamanan, khususnya perlindungan data pengguna, menjadi perhatian yang perlu ditangani secara serius. Di Desa Damai, perpustakaan digital memainkan peran penting dalam meningkatkan akses informasi bagi warga tanpa harus datang ke perpustakaan fisik. Namun, keamanan aplikasi perpustakaan digital ini belum pernah diuji secara formal, sehingga potensi kerentanan, terutama pada sisi autentikasi, belum diketahui. Pengujian keamanan yang menyeluruh sangat diperlukan untuk memastikan aplikasi ini aman dari serangan yang mungkin terjadi.

Autentikasi merupakan elemen krusial dalam keamanan aplikasi perpustakaan digital, karena berfungsi untuk memverifikasi identitas pengguna. Observasi awal mengidentifikasi beberapa kekurangan, seperti tidak adanya indikator kekuatan kata sandi pada form registrasi, kurangnya sanitasi karakter khusus yang membuka celah untuk serangan *SQL Injection*, dan ketiadaan

DOI: https://doi.org/10.62017/tektonik

CAPTCHA yang meningkatkan risiko serangan *brute force* oleh bot. Celah keamanan ini memungkinkan serangan seperti *SQL Injection*, *Session Hijacking*, dan *Brute Force*, yang dapat menjadi ancaman serius terhadap keamanan autentikasi. *SQL Injection* melibatkan penyisipan kode *SQL* berbahaya dalam input pengguna, sementara *session hijacking* mengancam sesi yang sedang berjalan antara pengguna dan server, serta *brute force* mencoba berbagai kombinasi kata sandi untuk mendapatkan akses tidak sah.

Penelitian sebelumnya menunjukkan bahwa serangan-serangan ini tetap menjadi ancaman signifikan bagi aplikasi web, termasuk perpustakaan digital, yang dapat menyebabkan kebocoran data pengguna dan penyalahgunaan. Oleh karena itu, pengujian keamanan aplikasi perpustakaan digital di Desa Damai diperlukan untuk mengidentifikasi dan mengatasi kerentanan ini. Pengujian ini penting tidak hanya untuk melindungi data pengguna, tetapi juga untuk menjaga reputasi dan kepercayaan pengguna terhadap aplikasi tersebut. Standar keamanan internasional, seperti yang dikeluarkan oleh Open Web Application Security Project (OWASP), memberikan panduan penting dalam mengamankan aplikasi web. Pengujian keamanan yang dilakukan akan dibandingkan dengan standar OWASP untuk mengevaluasi sejauh mana validitas hasil pengujian dalam penelitian ini. Melalui pengujian keamanan ini, diharapkan bahwa kerentanan autentikasi dapat diidentifikasi dan diperbaiki, sehingga warga Desa Damai dapat menggunakan layanan perpustakaan digital dengan rasa aman. Selain itu, pengujian ini juga dapat memberikan rekomendasi yang berguna bagi pengembang aplikasi perpustakaan digital di berbagai daerah.

METODE

Tahapan pertama dalam desain sistem ini adalah identifikasi masalah, di mana peneliti merumuskan masalah terkait kerentanan pada aplikasi perpustakaan digital di Desa Damai, yang perlu diuji untuk memastikan keamanannya. Selanjutnya, dilakukan studi literatur untuk mengumpulkan sumber yang relevan mengenai serangan *SQL Injection, Session Hijacking*, dan *Brute Force,* guna membangun dasar teoritis yang kuat. Persiapan alat dan bahan melibatkan perangkat keras seperti laptop dengan spesifikasi tertentu dan perangkat lunak yang mencakup sistem operasi Windows 10 dan *Kali Linux*, serta alat uji seperti *Burp Suite* dan *OWASP ZAP*. Dalam tahap pengujian keamanan, peneliti melaksanakan pengujian manual dan otomatis pada aplikasi perpustakaan, dimulai dengan serangan *SQL Injection* menggunakan *SQLmap* dan *Burp Suite,* diikuti dengan pengujian *session hijacking* menggunakan fitur *Intercept Proxy* di *Burp Suite,* dan diakhiri dengan pengujian *brute force* menggunakan *Burp Suite* untuk mencoba berbagai kombinasi kata sandi. Hasil dari masing-masing pengujian ini digunakan untuk mengevaluasi dan mengidentifikasi kerentanan dalam sistem autentikasi aplikasi perpustakaan digital di Desa Damai.

HASIL DAN PEMBAHASAN

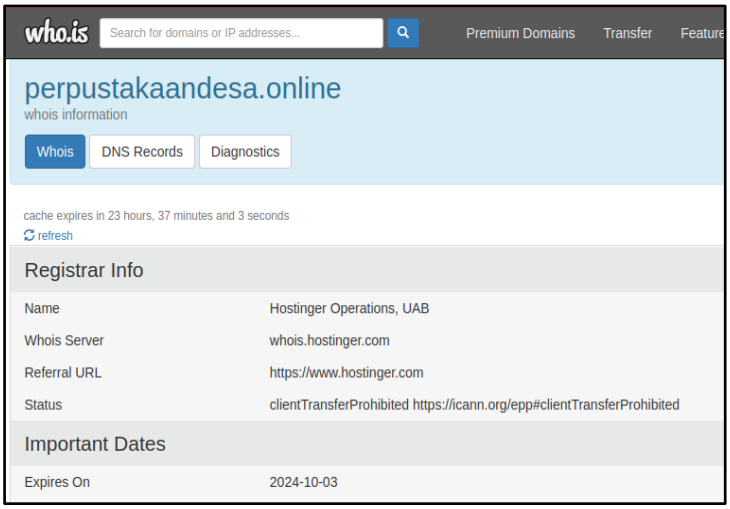
A. Pengujian Manual

Tahap awal dalam melakukan eksperimen penelitian membutuhkan perencanaan yang matang dan hati-hati. Langkah pertama yang dilakukan adalah pencarian dan pengumpulan informasi yang relevan dengan topik atau target penelitian yang sudah ditetapkan sebelumnya. Proses pengumpulan informasi ini harus dilakukan dengan teliti, karena informasi yang dikumpulkan pada tahap awal ini akan menjadi fondasi utama dalam merancang dan melaksanakan eksperimen yang lebih terarah. Dengan informasi yang akurat dan relevan, eksperimen dapat dikembangkan dengan baik sesuai dengan konteks dan tujuan penelitian, sehingga hasil yang diperoleh lebih valid dan sesuai dengan harapan penelitian yang telah dirumuskan sejak awal. Pendekatan yang sistematis dalam pencarian informasi ini juga membantu peneliti memastikan bahwa setiap langkah eksperimen didasarkan pada pemahaman yang kuat tentang materi yang sedang diteliti.

- **Target Pengujian**

DOI: https://doi.org/10.62017/tektonik

Gambar 1. Tampilan website perpustakaan digital



Gambar 2. Tampilan hasil *whois* website target

Dengan menggunakan WHOIS, diperoleh berbagai informasi penting terkait dengan situs web target yang akan diserang.

Tabel 1. Hasil *whois*

Nama	http://perpustakaandesa.online
Domain	185.27.134.167
Server	Hostinger.com
Registered On	2022-12-16
Expires On	2024-10-03
Status	Active

Situs web dengan nama domain "perpustakaandesa.online" di-host di server dengan alamat IP 185.27.134.167, yang dikelola oleh Hostinger.com. Domain ini didaftarkan pada tanggal 16 Desember 2022, menandai awal penggunaan alamat tersebut oleh pemiliknya. Saat ini, domain tersebut masih aktif dan dapat diakses, namun perlu diingat bahwa masa aktifnya akan berakhir pada 3 Oktober 2024. Oleh karena itu, pemilik domain harus memperbarui registrasinya sebelum tanggal tersebut agar tetap bisa menggunakan nama domain ini.

- Serangan *SQL Injection*

Dalam eksperimen ini, *Burp Suite* digunakan sebagai alat uji keamanan untuk melancarkan serangan *SQL Injection* pada formulir login aplikasi perpustakaan digital di Desa Damai. Peneliti akan mencoba menyisipkan kode SQL berbahaya ke dalam input yang dikirimkan ke aplikasi web,

DOI: https://doi.org/10.62017/tektonik

dengan tujuan menguji seberapa baik aplikasi tersebut dapat menahan serangan ini. Hasil dari serangan akan dicatat dengan teliti, termasuk apakah sistem mampu menangani dan mengatasi serangan SQL Injection atau justru rentan terhadapnya. Jika aplikasi gagal mengatasi serangan ini, hal ini menunjukkan adanya kelemahan serius dalam validasi input yang bisa dieksploitasi oleh penyerang. Ini menunjukkan pentingnya pengujian mendalam dalam memastikan bahwa aplikasi web memiliki mekanisme pertahanan yang cukup untuk melindungi data pengguna dari ancaman keamanan semacam ini. Sebaliknya, jika aplikasi berhasil menahan serangan, ini menunjukkan bahwa sistem telah dirancang dengan tingkat keamanan yang memadai.

Tabel 2. *Queri SQL Input Password*

SELECT * FROM users WHERE username = 'input_username' AND password = 'input_password';

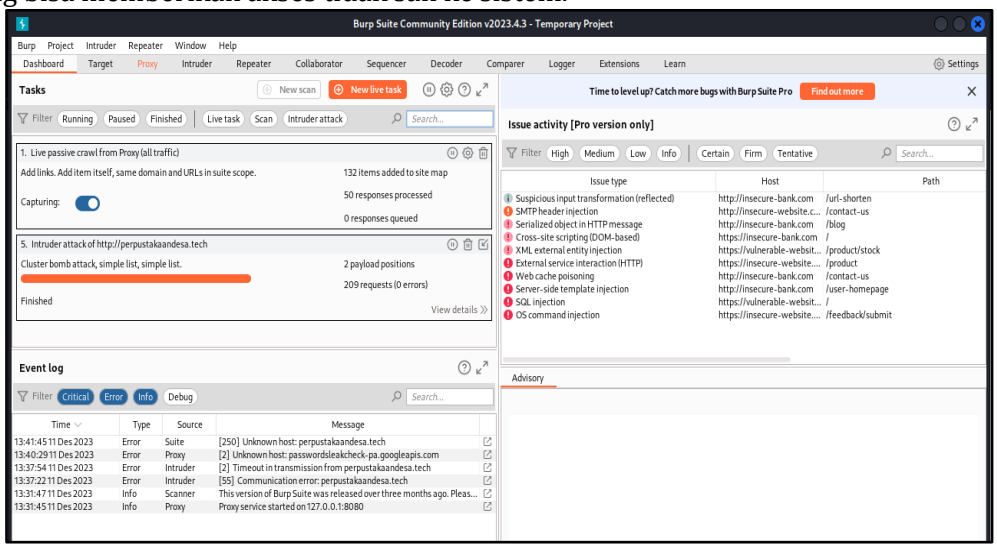
Kemudian penyerang akan memanipulasi atau menginjeksi pada inputan login seperti

' OR '1'='1'; –

Maka *script* menjadi

SELECT * FROM users WHERE username = '' OR '1'='1'; --' AND password = 'input_password';

Dua tanda hubung ganda (--) dalam *SQL* berfungsi sebagai komentar yang mengabaikan sisa kueri asli. Dalam skenario ini, penyerang mampu mengeksploitasi celah keamanan dengan menghindari mekanisme login. Hal ini terjadi karena kondisi seperti '1'='1' selalu bernilai benar, sehingga bagian sisa dari kueri tidak dieksekusi. Dengan pemahaman ini, jika login pada sebuah situs web memverifikasi kredensial pengguna melalui query SQL, maka penyerang dapat mencoba berbagai bentuk inputan injeksi yang diambil dari berbagai sumber. Eksploitasi semacam ini menunjukkan betapa pentingnya sanitasi input dan validasi kueri yang tepat untuk mencegah serangan injeksi SQL, yang bisa memberikan akses tidak sah ke sistem.

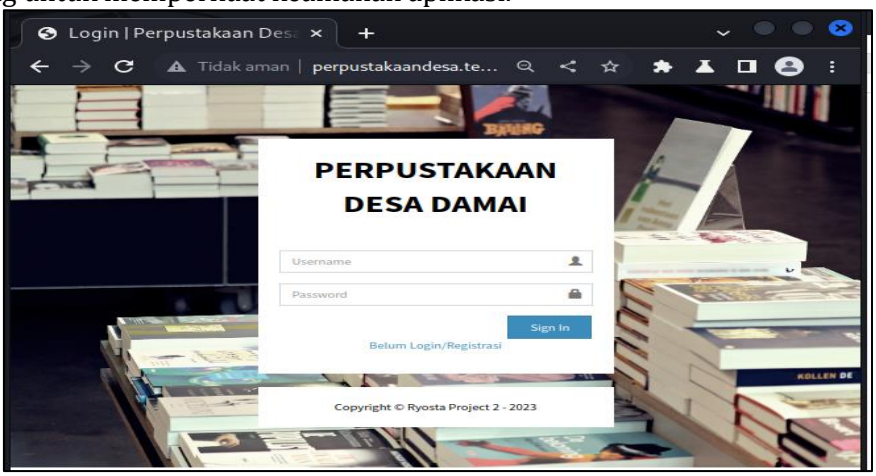


Gambar 3. Tampilan *Tool Burpsuite*

Pengujian keamanan pada aplikasi web dapat dilakukan menggunakan Burp Suite dengan membuka proxy dan menjalankan aplikasi perpustakaan digital melalui browser yang terhubung ke Burp Suite. Metode ini memungkinkan kita untuk memantau serta memodifikasi lalu lintas HTTP/HTTPS yang melewati aplikasi, memberikan kendali penuh atas analisis setiap permintaan dan respons yang dikirimkan. Dengan menjalankan perpustakaan digital melalui Burp Suite, seperti yang ditunjukkan pada Gambar 4, kita dapat secara efektif mengidentifikasi dan

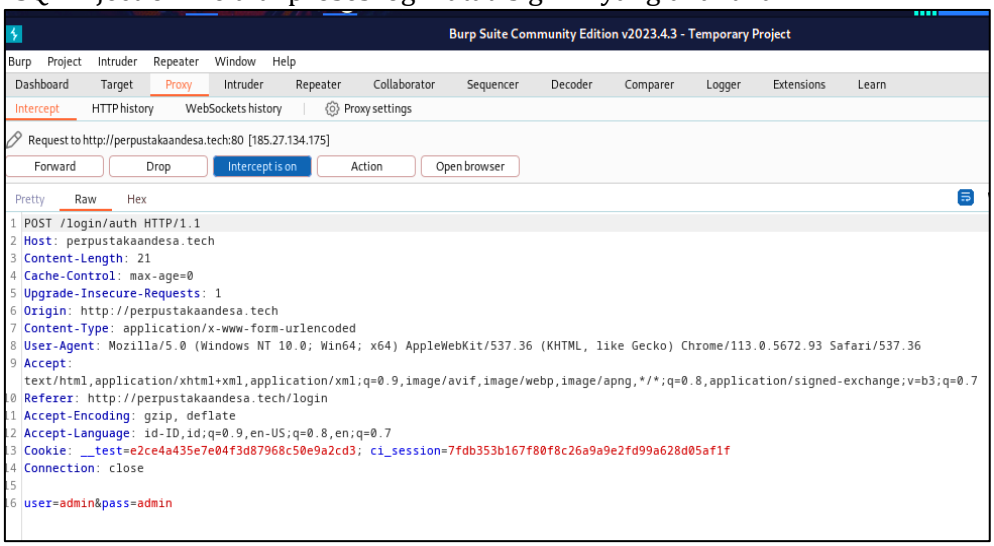
Vol. 1, No. 4 Juli 2024, Hal. 429-442
DOI: https://doi.org/10.62017/tektonik

mengeksploitasi potensi kerentanan yang mungkin tersembunyi dalam lalu lintas jaringan, yang sangat penting untuk memperkuat keamanan aplikasi.



Gambar 4. Tampilan Halaman Login

Selanjutnya, pada halaman login, username dan password dimasukkan secara acak, dan proses login akan ditangkap oleh Burp Suite dengan mengaktifkan fitur intercept. Setelah intercept diaktifkan, Burp Suite akan menangkap data login yang dimasukkan, memungkinkan eksekusi serangan SQL Injection melalui proses login atau sign-in yang dilakukan.



Gambar 5. Tampilan Halaman *Proxy Brupsuite Intersept on*

Tabel 3. Percobaan Bypass SQL Injection

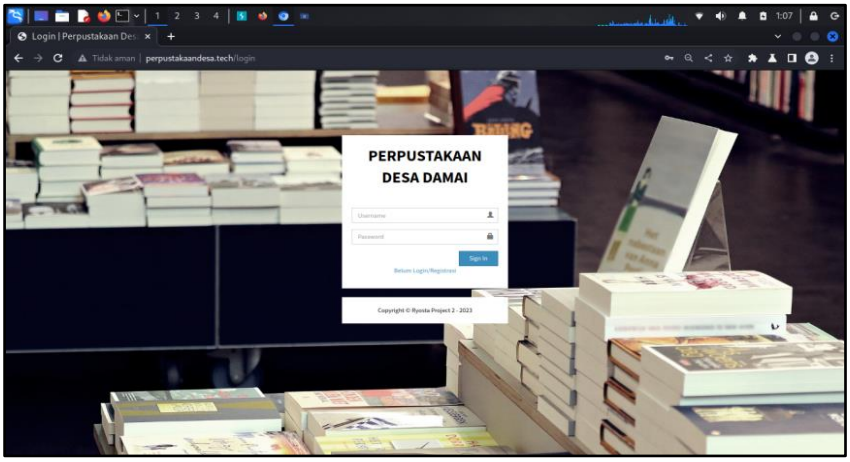
Percobaan Bypass SQL Injection
or 1=1–
' or 1=1--
' or '1'='1
' or '1'='1'--
' or '1'='1'/*
‘or '1'='1'#

Vol. 1, No. 4 Juli 2024, Hal. 429-442
DOI: https://doi.org/10.62017/tektonik

' or '1′='1
' or 1=1
' or 1=1 --
' or 1=1 –
' or 1=1;#
' or 1=1/*
' or 1=1#
' or 1=1–
') or '1'='1
') or '1'='1--
') or '1'='1'--
') or '1'='1'/*
') or '1'='1'#
') or ('1'='1
') or ('1'='1--
') or ('1'='1'--
') or ('1'='1'--
') or ('1'='1'/*
'or'1=1
'or'1=1′
" or "1"="1

Script Sql injection pada tabel 3 akan digunakan dalam implementasi serangan *Sql injection* perpustakaan digital dengan bantuan alat *burpsuite* agar proses lebih cepat. Setelah itu, masukkan script ini ke dalam kolom username dan password, lalu nonaktifkan intercept agar input *SQL Injection* dapat dijalankan dan memperoleh respons dari aplikasi perpustakaan digital. Hasilnya, tampilan halaman akan kembali ke halaman login, seperti yang terlihat pada Gambar 6, menunjukkan bahwa input script SQL Injection tidak berhasil.

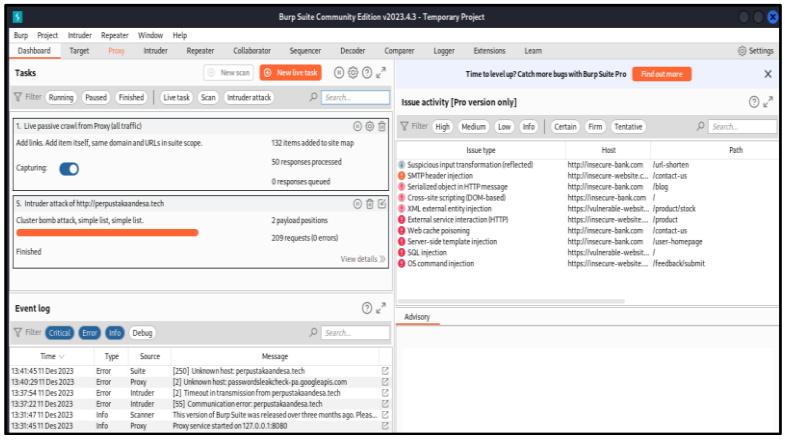
Vol. 1, No. 4 Juli 2024, Hal. 429-442
DOI: https://doi.org/10.62017/tektonik



Gambar 6. Tampilan Hasil Pengujian SQL Injection

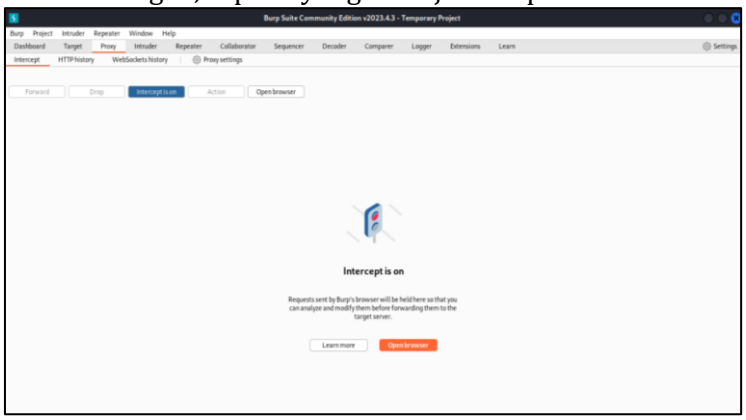
- Serangan Brute Froce

Penggunaan tool Burp Suite dilakukan dengan membuka proxy dan menjalankan aplikasi perpustakaan digital melalui browser yang terhubung ke Burp Suite. Proses ini dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Aplikasi Burpsuite

Pada halaman login, masukkan *username* dan *password* sembarangan, lalu aktifkan fitur intercept di *Burpsuite* untuk menangkap sesi login. Setelah intercept dihidupkan, dan proses sign-in dilakukan, *Burpsuite* akan menangkap data login dari halaman tersebut. Kemudian, aktifkan fitur Intruder untuk memulai serangan, seperti yang ditunjukkan pada Gambar 8.

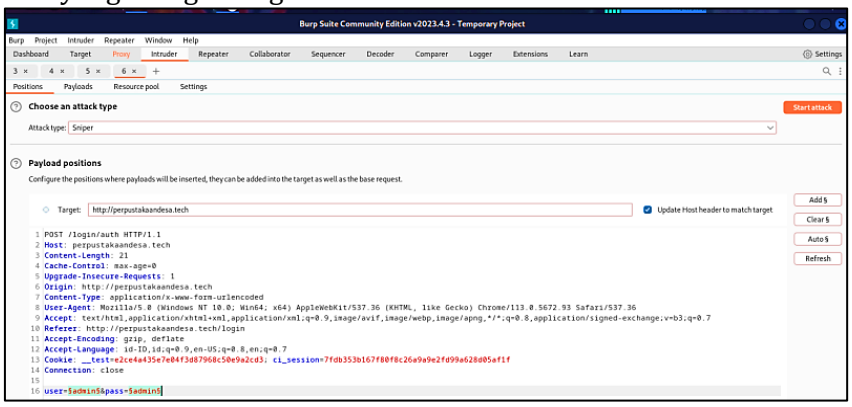


Gambar 8. Tampilan Menu Proxy Brupsuite Intercept

Pada menu *Intruder*, tentukan *username* sebagai payload 1 dan password sebagai payload 2 dengan menekan "add" setelah memblokir area yang relevan. Jenis serangan yang diterapkan

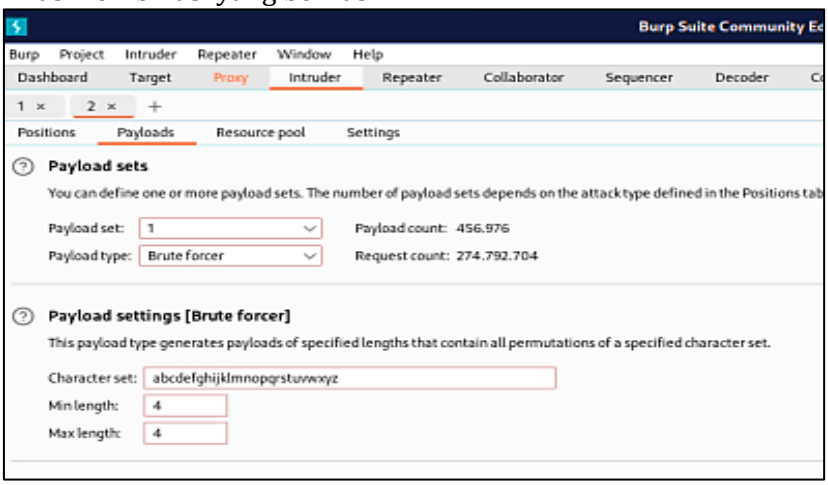
DOI: https://doi.org/10.62017/tektonik

adalah cluster bomb, memungkinkan *brute force* pada username dan password secara bersamaan. Di menu payload, masukkan daftar karakter yang mungkin digunakan untuk username di payload 1 dan atur jumlah karakter minimum yang perlu dicoba untuk login, sebagaimana tampak pada Gambar 9. Begitu pula, di payload 2, masukkan daftar karakter untuk password dan atur panjang minimal password yang mungkin digunakan.

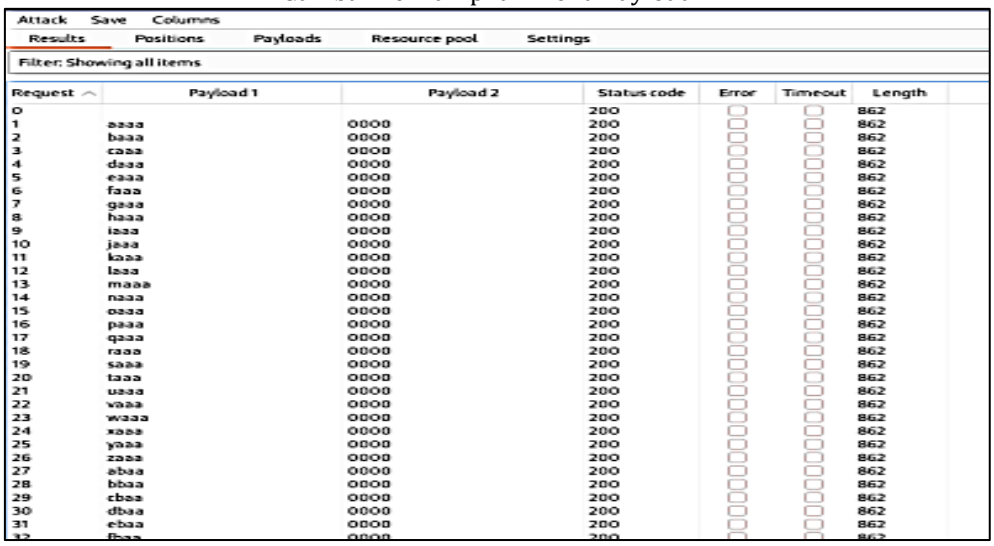


Gambar 9. Tampilan Menu Include menambah Kode

Ketika serangan brute force dijalankan, Burp Suite akan menampilkan berbagai kemungkinan kombinasi username dan password yang dapat digunakan untuk login, seperti pada Gambar 10. Proses ini memungkinkan penilaian terhadap kekuatan sistem dalam menghadapi serangan brute force dan identifikasi kombinasi yang berhasil.



Gambar 10. Tampilan Menu Payload 1



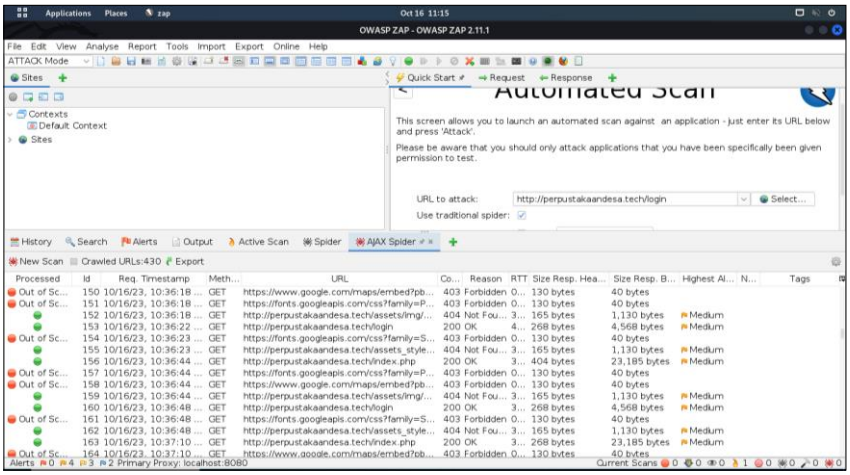
Gambar 11. Tampilan Serangan *Brute Force*

DOI: https://doi.org/10.62017/tektonik

Gambar 11 menunjukkan tampilan saat serangan *brute force* dilakukan. Pada tahap ini, *Burpsuite* akan mencoba berbagai kombinasi username dan password yang mungkin digunakan untuk login. Proses ini menghasilkan daftar kemungkinan kombinasi yang diuji secara otomatis. Dengan melakukan serangan *brute force*, *Burpsuite* dapat mengidentifikasi username dan password yang valid dengan menguji setiap kombinasi yang terdaftar dalam daftar. Analisis hasil serangan ini memungkinkan peneliti untuk mengevaluasi kekuatan sistem autentikasi terhadap serangan *brute force* dan mengidentifikasi potensi kerentanan dalam mekanisme login aplikasi.

- Pengujian dengan Standar Kemanan OWASP

Pengujian keamanan dilakukan dengan menggunakan standar OWASP ZAP, sebuah alat yang dirancang untuk melakukan pemindaian otomatis terhadap situs web aplikasi perpustakaan digital. Dengan memanfaatkan OWASP ZAP, proses pemindaian ini bertujuan untuk mengidentifikasi berbagai kerentanan yang mungkin ada dalam aplikasi. Alat ini secara sistematis memeriksa situs web untuk menemukan potensi masalah keamanan, seperti celah dalam konfigurasi, kerentanan dalam kode, dan masalah lainnya yang dapat dieksploitasi. Hasil dari pemindaian ini kemudian akan dicatat secara rinci untuk dianalisis lebih lanjut. Proses ini tidak hanya membantu dalam mendeteksi kerentanan, tetapi juga memberikan wawasan berharga untuk memperbaiki dan menguatkan sistem keamanan aplikasi perpustakaan digital secara keseluruhan.



Gambar 12. Tampilan Hasil Alert Owasp ZAP

B. Hasil Pengujian

Penelitian ini melibatkan serangkaian uji coba untuk mengevaluasi dampak berbagai jenis serangan terhadap sistem. Selama penelitian, berbagai serangan telah diterapkan untuk mengidentifikasi potensi kerentanan dan efeknya. Hasil dari serangkaian eksperimen ini menunjukkan bagaimana sistem merespons berbagai jenis ancaman, memberikan wawasan mendalam tentang efektivitas mekanisme keamanan yang ada dan area yang perlu diperbaiki. Melalui pengujian ini, dapat diidentifikasi aspek-aspek kritis dari sistem yang mungkin menjadi target serangan, serta langkah-langkah yang diperlukan untuk memperkuat pertahanan dan meningkatkan ketahanan sistem terhadap ancaman.

- Hasil Serangan SQL Injection

Peneliti akan mengevaluasi hasil dari serangan SQL Injection untuk menentukan apakah aplikasi dapat melindungi dirinya dengan efektif dari ancaman tersebut. Jika teridentifikasi adanya kerentanan, akan diberikan rekomendasi untuk perbaikan guna meningkatkan keamanan aplikasi.

Tabel 4. Hasil serangan SQL *Injection*

DOI: https://doi.org/10.62017/tektonik

Percobaan *Bypass SQL Injection*	**Hasil Serangan**
or 1=1–	Tidak Bisa
' or 1=1--	Tidak Bisa
' or '1'='1	Tidak Bisa
' or '1'='1'--	Tidak Bisa
' or '1'='1'/*	Tidak Bisa
‘or '1'='1'#	Tidak Bisa
' or '1′='1	Tidak Bisa
' or 1=1	Tidak Bisa
' or 1=1 --	Tidak Bisa
' or 1=1 –	Tidak Bisa
' or 1=1;#	Tidak Bisa
' or 1=1/*	Tidak Bisa
' or 1=1#	Tidak Bisa
' or 1=1–	Tidak Bisa
') or '1'='1	Tidak Bisa
') or '1'='1--	Tidak Bisa
') or '1'='1'--	Tidak Bisa
') or '1'='1'/*	Tidak Bisa
') or '1'='1'#	Tidak Bisa
') or ('1'='1	Tidak Bisa
') or ('1'='1--	Tidak Bisa
') or ('1'='1'--	Tidak Bisa
') or ('1'='1'--	Tidak Bisa
') or ('1'='1'/*	Tidak Bisa
'or'1=1	Tidak Bisa
'or'1=1′	Tidak Bisa
" or "1"="1	Tidak Bisa

Semua upaya untuk melakukan *bypass SQL Injection* tidak membuahkan hasil. Serangan yang diterapkan, seperti mencoba menyisipkan karakter khusus atau menggunakan kueri dengan kondisi selalu benar (1=1), tidak berhasil mengecoh sistem. Tampaknya, aplikasi perpustakaan

DOI: https://doi.org/10.62017/tektonik

digital telah dirancang dengan baik dan dilengkapi dengan perlindungan efektif terhadap serangan SQL Injection. Perlindungan ini kemungkinan melibatkan penggunaan kueri parameterized, pernyataan terpersiapkan, atau metode keamanan lainnya yang mencegah eksekusi kode SQL yang tidak diinginkan.

- Hasi; Serangan Session Hijacking

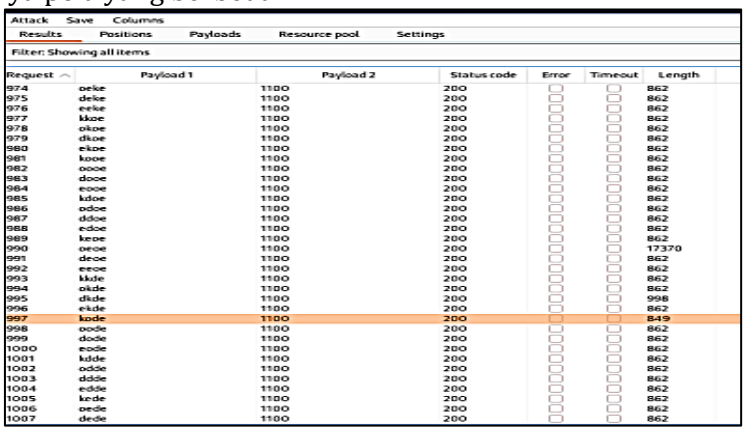
Hasil dari serangan session hijacking akan mencakup penilaian apakah aplikasi dapat melindungi dirinya dari jenis serangan ini atau tidak. Jika ditemukan kerentanan, rekomendasi untuk perbaikan akan disediakan.

Hasil	Laptop 1	Laptop 2
Ip Adress	192.168.25.187	192.168.43.145
Host	perpustakaandesa.online	perpustakaandesa.online
Cookie: __test	1e0b4b9839451aedb1d46b745d46f60f	5b074b0259451amkj1d46b758l46kl0l
ci_session	3b7c776974674b7c079708c48c6c1c8933bf9e32	69er765474672b7c598638c36k8c1c89344fgh25
user	admin	admin
password	admin	admin
hasil session hijacking	laptop 2 gagal menerapkan session hijacking	

Setelah mencoba serangan session hijacking, ternyata serangan ini tidak berhasil. Kemungkinan penyebabnya adalah aplikasi perpustakaan Desa Damai menggunakan token sesi yang kompleks dan sulit untuk diprediksi. Meskipun serangan ini dilakukan, aplikasi tetap aman karena tidak menerapkan sertifikat keamanan dan tidak mengimplementasikan Autentikasi Dua Faktor (2FA), yang seringkali meningkatkan risiko terhadap serangan semacam ini.

- Hasil Serangan *Brute Force*

Analisis hasil serangan brute force akan mengevaluasi sejauh mana kekuatan aplikasi dalam menghadapi metode serangan tersebut. Jika aplikasi terbukti rentan dan dapat diakses dengan mudah melalui brute force, rekomendasi perbaikan akan disarankan untuk meningkatkan keamanan. Selama serangan, beberapa kombinasi username dan password dengan panjang yang bervariasi berhasil teridentifikasi. Perbedaan panjang ini menjadi indikasi bahwa kombinasi tertentu mungkin valid, karena panjang respons yang diterima dari aplikasi perpustakaan digital menunjukkan adanya pola yang berbeda.



Gambar 13. Tampilan Hasil Serangan Brute Force

Dari hasil tersebut, username dan password yang memiliki panjang respons yang berbeda dibandingkan dengan yang lainnya diperiksa lebih lanjut. Pada pemeriksaan ini, ditemukan

DOI: https://doi.org/10.62017/tektonik

bahwa salah satu kombinasi username dan password yang diuji menghasilkan respons yang menunjukkan bahwa kredensial tersebut benar dan dapat digunakan untuk login. Temuan ini menandakan bahwa aplikasi mungkin memiliki kerentanan yang perlu diperbaiki untuk mencegah akses tidak sah melalui *brute force.*

Tabel 5. Hasil serangan *brute force*

Hasil Serangan Brute Force		
Target	http://perpustakaandesa.online	
attack type	*cluster bomp*	
List Payload 1	abcdefghijklmnopqrstuvwxyz	
List Payload 2	1234567890	
Payload type	BruteForce	
Minimal Character	4	
Maximal Character	5	
Time	--	
Length	862, 870, 1006, 849,	
Jumlah kemungkinan	1,357,218,720,000	
Hasil Yang Didapat	Username	Password
	kode admin	1100 admin

Serangan brute force pada situs web "perpustakaandesa.online" menggunakan metode cluster bomb melibatkan kombinasi payload dari karakter abjad dan alfanumerik. Dalam serangan ini, penyerang menguji berbagai kombinasi dengan panjang username antara 4 hingga 5 karakter dan password dengan karakter numerik. Untuk menghitung jumlah kemungkinan kombinasi, jika List Payload 1 (abcdefghijklmnopqrstuvwxyz) digunakan sebagai username dan List Payload 2 (1234567890) digunakan sebagai password, dapat dilakukan dengan persamaan 4.1.

$$K = (P1^{MIN} + P1^{MAX}) \cdot (P2^{MIN} + P2^{MAX}) \quad \text{(Persamaan 4.1)}$$

$$K = (26^4 + 26^5) \cdot (10^4 + 10^5)$$
$$K = (456976 + 11881376) \cdot (10000 + 100000)$$
$$K = (12338352) \cdot (110000)$$
$$K = 1.357.218.720.000$$

Jumlah kemungkinan kombinasi username dan password yang diuji mencapai 1.357.218.720.000. Selama serangan, panjang respons yang diterima dari perpustakaan digital tercatat sebagai 862, 870, 1006, dan 849. Hasil serangan menunjukkan keberhasilan akses dengan dua kombinasi: yang pertama adalah username "admin" dengan password "admin", dan yang kedua adalah username "kode" dengan password "1100".

DOI: https://doi.org/10.62017/tektonik

- Hasil Pengujian dengan Standar OWASP ZAP

Tabel 6. Hasil Pengujuan Standar Kemanan Owasp

NO	Jenis *Alert*	Tingkat
1.	*Absence of Anti-CSRF Tokens*	Medium
2.	*Content Security Policy (CSP) Header Not Set*	Medium
3.	*Missing Anti-clickjacking Header*	Medium
4.	*Vulnerable JS Library*	Medium
5.	*Cookie without SameSite Attribute*	Low
6.	*Cross-Domain JavaScript Source File Inclusion*	Low
7.	*X-Content-Type-Options Header Missing*	Low
8.	*Content-Type Header Missing*	Informational
9.	*Information Disclosure - Suspicious Comments*	Informational

- Analisis Hasil Perbandingan dan Validasi Serangan

Setelah melakukan berbagai serangan, seperti *SQL Injection, Session Hijacking, dan Brute Force*, serta pencarian celah keamanan menggunakan OWASP ZAP, hasil dari serangan-serangan tersebut akan dibandingkan. Perbandingan ini bertujuan untuk menemukan rekomendasi perbaikan yang dapat meningkatkan keamanan sistem, sehingga lebih terlindungi dari ancaman siber yang tidak diinginkan.

Tabel 7. Hasil Perbandingan Serangan

Perbandingan Serangan			
Serangan Manual		**OWASP ZAP**	
Serangan	**Celah keamanan**	**Serangan**	**Celah keamanan**
SQL Injection	Tidak ada	SQL Injection	Tidak ada
Session Hijacking	Tidak ada	Session Hijacking	Tidak ada
Brute Force	Berhasil	Brute Force	Tidak ada

Perbandingan antara serangan manual dan penggunaan OWASP ZAP dalam mengeksploitasi celah keamanan menunjukkan perbedaan yang signifikan dalam tingkat keberhasilan terhadap jenis celah tertentu. Serangan manual, yang melibatkan intervensi langsung dari penyerang, berhasil mengeksploitasi celah keamanan terkait serangan *Brute Force.* Dalam hal ini, serangan *Brute Force* memungkinkan penyerang untuk memperoleh akses dengan mencoba berbagai kombinasi kata sandi secara berulang. Sebaliknya, OWASP ZAP, sebagai alat pengujian keamanan otomatis, menunjukkan keterbatasan dalam beberapa jenis serangan. Meskipun alat ini mampu mengidentifikasi potensi celah keamanan, seperti pada SQL Injection dan Session Hijacking, eksploitasi yang berhasil tidak selalu terjadi. Kegagalan ini mungkin disebabkan oleh faktor-faktor seperti tingkat keamanan yang lebih tinggi pada sistem target atau kurangnya penyesuaian yang diperlukan pada alat tersebut.

Secara keseluruhan, meskipun serangan manual sering kali lebih berhasil dalam skenario tertentu, OWASP ZAP masih memiliki nilai dalam mendeteksi potensi celah. Kombinasi antara serangan manual dan penggunaan alat otomatis seperti OWASP ZAP dapat menjadi strategi yang lebih efektif dalam mengidentifikasi dan menutup celah keamanan pada suatu sistem. Pendekatan ini menawarkan keseimbangan antara kedalaman eksploitasi manual dan efisiensi deteksi otomatis, yang bersama-sama dapat memberikan perlindungan yang lebih komprehensif terhadap ancaman siber.

C. Rekomendasi Perbaikan

Hasil dari pengujian keamanan pada aplikasi Perpustakaan Desa Damai menunjukkan bahwa serangan *brute force* berhasil mendapatkan username dan password yang dapat dimanfaatkan oleh penyerang. Untuk mencegah serangan serupa di masa mendatang, beberapa rekomendasi perbaikan dapat diberikan. Pertama, penting untuk menerapkan pembatasan pada jumlah percobaan login yang dapat dilakukan, misalnya dengan mengunci akun setelah beberapa

kali gagal mencoba login atau menerapkan waktu jeda antara percobaan login. Kedua, menambahkan CAPTCHA atau mengaktifkan verifikasi dua faktor (2FA) pada proses login akan meningkatkan keamanan dan mencegah upaya brute force. Terakhir, monitoring secara aktif terhadap aktivitas login juga dianjurkan untuk mendeteksi pola yang mencurigakan atau upaya serangan brute force, sehingga tindakan cepat dapat diambil untuk mengurangi risiko..

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, beberapa kesimpulan penting dapat diambil. Meskipun serangan *SQL Injection* dan *Session Hijacking* tidak berhasil, hal ini menunjukkan adanya perlindungan yang kuat pada tingkat aplikasi. Namun, keberhasilan serangan *Brute Force* menyoroti kelemahan dalam manajemen sesi dan proses login, menekankan perlunya memperkuat mekanisme keamanan tersebut. Penggunaan *OWASP ZAP* dalam pengujian keamanan berhasil mengidentifikasi beberapa kerentanan, seperti kurangnya token *Anti-CSRF, header Content Security Policy (CSP),* dan perlindungan *Anti-clickjacking*, yang semuanya memerlukan perbaikan untuk meningkatkan keamanan aplikasi. Perbandingan antara serangan manual dan penggunaan alat otomatis seperti *OWASP ZAP* menunjukkan bahwa meskipun serangan manual lebih berhasil dalam beberapa kasus, alat otomatis tidak selalu mampu mengeksploitasi semua celah. Oleh karena itu, kombinasi antara serangan manual dan alat otomatis dianggap sebagai strategi yang paling efektif untuk mengidentifikasi dan mengatasi celah keamanan secara menyeluruh. Untuk meningkatkan keamanan, penting untuk menerapkan langkah-langkah seperti penggunaan token berbasis sesi, pembatasan percobaan login, serta penerapan CAPTCHA atau verifikasi dua faktor. Pengujian keamanan yang berkelanjutan dan pemantauan aktif terhadap teknologi keamanan juga sangat penting untuk menjaga aplikasi perpustakaan digital tetap aman.

DAFTAR PUSTAKA

Erickson, J. (2022). *Hacking: The Art of Exploitation*. https://doi.org/10.1515/9781474451109-011

Hayatuddiniyah. (2021). Perpustakaan digital berdasarkan perspektif Lucy A. Tedd dan Andrew Large (studi kasus di Perpustakaan Fakultas Teknik UGM Yogyakarta). *Pustaka Karya: Jurnal Ilmiah Ilmu Perpustakaan dan Informasi, 9*(1), 1. https://doi.org/10.18592/pk.v9i1.5141

Hwang, W. S., Shon, J. G., & Park, J. S. (2022). Web Session Hijacking Defense Technique Using User Information. *Human-centric Computing and Information Sciences, 12*. https://doi.org/10.22967/HCIS.2022.12.016

Jaiswal, A., Raj, G., & Singh, D. (2014). Security Testing of Web Applications: Issues and Challenges. *International Journal of Computer Applications, 88*(3), 26–32. https://doi.org/10.5120/15334-3667

Li, J. (2020). Vulnerabilities mapping based on OWASP-SANS: A survey for static application security testing (SAST). *Annals of Emerging Technologies in Computing, 4*(3), 1–8. https://doi.org/10.33166/AETiC.2020.03.001

Rizki, A. R., & Nunu, N. (2022). Rancang Bangun Aplikasi Analisis Standar Keamanan Website Dengan Metode Scanning Vulnerability Menggunakan Module Requests Python. *Seminar Teknologi Majalengka, 6*, 271–277. https://doi.org/10.31949/stima.v6i0.699

Schoenborn, J. M., & Althoff, K. D. (2020). Detecting SQL-injection and cross-site scripting attacks using case-based reasoning and SEASALT. *CEUR Workshop Proceedings, 2993*, 66–77.

Sosioteknologi, J. (2022). Section 1 Perspectives Of Open Science. *Jurnal Sosioteknologi, 21*(1), 120–124.

Zulkifli, Samsir, & Sirait, A. (2021). Implementasi Max Length dan Input Type Number Pada Form Login Website Untuk Mencegah Penetrasi SQL Injeksi Secara Paksa. *U-NET: Jurnal Teknik Informatika, 4*(1), 14–18. https://doi.org/10.52332/u-net.v4i1.223