

# Analisis Keamanan Website PPDB SMK XYZ Menggunakan Pendekatan OWASP Top 10 2017

1<sup>st</sup> Dandy Pratama Arisandy

*Sistem Informasi*

*Telkom University Surabaya*

Surabaya, Indonesia

[dandypratama@student.telkomuniversit  
y.ac.id](mailto:dandypratama@student.telkomuniversit<br/>y.ac.id)

2<sup>nd</sup> Muhammad Nasrullah

*Sistem Informasi*

*Telkom University Surabaya*

Surabaya, Indonesia

[emnasrul@telkomuniversity.ac.id](mailto:emnasrul@telkomuniversity.ac.id)

3<sup>rd</sup> Adzani Rachmadhi Putra

*Sistem Informasi*

*Telkom University Surabaya*

Surabaya, Indonesia

[adzrachmadhip@telkomuniversity.ac.id](mailto:adzrachmadhip@telkomuniversity.ac.id)

*Abstrak — Sistem informasi memiliki peran yang krusial dalam berbagai bidang kehidupan, termasuk dalam sektor pendidikan, khususnya dalam proses Penerimaan Peserta Didik Baru (PPDB). Meskipun memberikan kemudahan, sistem ini juga membawa potensi ancaman keamanan, seperti aksi peretasan. Website PPDB milik SMK XYZ menyimpan informasi penting calon siswa, seperti username, password, dan data pribadi lainnya, yang dapat menjadi target serangan jika keamanannya tidak memadai. Oleh karena itu, penelitian ini dilakukan untuk mengevaluasi dan mengkaji tingkat keamanan dari website PPDB tersebut. Metode yang digunakan dalam penelitian ini adalah Penetration Testing dengan pendekatan dari OWASP Top 10 tahun 2017 guna mengidentifikasi kemungkinan celah keamanan. Temuan dari pengujian ini akan digunakan sebagai dasar untuk memberikan rekomendasi dan solusi perbaikan guna membantu pihak pengelola meningkatkan keamanan sistem secara optimal.*

**Kata kunci— OWASP Framework, Keamanan Sistem Informasi, Penetration Testing, Website PPDB**

## I. PENDAHULUAN

Peningkatan jumlah pengguna internet di Indonesia, yang mencapai 221,5 juta jiwa atau 79,5% dari total populasi pada tahun 2024, telah mendorong tingginya permintaan akan website untuk berbagai kebutuhan. Seiring dengan pertumbuhan ini, isu keamanan menjadi sangat krusial[1]. Aplikasi berbasis web, yang sangat bergantung pada internet, rentan terhadap serangan siber dari pihak tidak bertanggung jawab yang bertujuan mencuri informasi sensitif untuk merugikan pihak lain. Oleh karena itu, analisis keamanan untuk mengukur tingkat kualitas proteksi sebuah website menjadi sebuah keharusan[2].

Salah satu contoh entitas yang sangat bergantung pada website adalah SMK XYZ, yang menggunakan situsnya untuk menyajikan informasi sekolah hingga proses Penerimaan Peserta Didik Baru (PPDB) secara online. Website PPDB ini mengelola data-data penting dan sensitif, seperti data pribadi calon siswa dan orang tua wali. Keamanan yang lemah pada sistem ini dapat menjadi celah bagi peretas untuk mencuri data krusial, seperti detail login dan informasi pribadi, yang dapat disalahgunakan dan

merugikan banyak pihak. Untuk mengatasi potensi risiko tersebut, penelitian ini akan melakukan analisis keamanan pada website PPDB SMK XYZ menggunakan metode *Penetration Testing* (*Pentest*) dengan kerangka kerja dari *Open Web Application Security Project* (OWASP). Metode ini dipilih karena komprehensif, mudah digunakan, dan akurat dalam mengidentifikasi kerentanan. Hasil pengujian, yang berfokus pada parameter seperti autentikasi dan enkripsi data, akan digunakan untuk memberikan rekomendasi dan solusi konkret guna meningkatkan keamanan website PPDB SMK XYZ.

## II. KAJIAN TEORI

### A. Penetration Testing

*Penetration testing* (uji penetrasi atau *pentest*) adalah sebuah proses evaluasi keamanan sistem atau jaringan yang dilakukan dengan mensimulasikan serangan siber. Tujuannya adalah untuk menemukan kerentanan yang dapat dieksplorasi oleh penyerang sebelum kerentanan tersebut benar-benar dimanfaatkan[3]. Proses ini mencakup berbagai teknik seperti pemindaian (*scanning*), enumerasi, dan audit keamanan. Terdapat beberapa kerangka kerja yang dapat digunakan untuk melakukan *pentest*, di antaranya adalah *Information System Security Assesment Framework* (ISSAF), *Penetration Testing Execution Standard* (PTES), dan *Open Web Application Security Project* (OWASP)[4].

### B. Open Web Application Security Project (OWASP)

*OWASP* adalah organisasi nirlaba global yang berfokus pada peningkatan keamanan perangkat lunak. *OWASP* menyediakan berbagai sumber daya, dokumen, dan alat-alat gratis untuk membantu pengembang dan profesional keamanan dalam membangun dan memelihara aplikasi yang aman. Salah satu proyek utama dari *OWASP* adalah *OWASP Top 10*, sebuah dokumen standar yang berisi daftar sepuluh risiko keamanan aplikasi *web* paling kritis[5].

### C. Website

*Website* merupakan sebuah halaman informasi yang disediakan menggunakan jalur internet sehingga dapat diakses dimana saja selama perangkat yang digunakan terkoneksi dengan internet [6]. Penggunaan *website* ini

sangat sederhana karena dapat diakses kapan saja dan dimana saja selama memiliki akses internet, menjadikan *website* ini sebagai media penyampaian informasi yang terpercaya. Saat mengakses *website*, pengguna hanya perlu menggunakan *smartphone* atau perangkat komputer untuk mengaksesnya [7].

#### D. Whois

*Whois* adalah layanan internet yang menampilkan informasi atau data *Whois* tentang sebuah domain. Data ini dikelola oleh perusahaan *registrar* dan *registry*, yang keduanya harus terakreditasi oleh *ICANN*. Secara sederhana, ketika Anda membeli domain, pendaftaran dilakukan melalui *registrar* dan *registry*. Kedua entitas ini bekerja langsung di bawah pengawasan *ICANN*, sebuah organisasi yang mengatur *database* internet untuk memastikan operasi jaringan yang aman dan stabil [8]. *Registrar* adalah perusahaan yang terakreditasi oleh *ICANN*, sementara *registry* adalah perusahaan yang berkontrak dengan *ICANN* untuk mengelola domain tertentu (seperti *.COM* atau *.NET*). Saat mendaftarkan domain, Anda diwajibkan untuk memberikan informasi pribadi, yang kemudian akan tersedia untuk umum sebagai bagian dari data *Whois*.

#### E. Wappalyzer

*Wappalyzer* adalah ekstensi *browser* yang berfungsi untuk mengidentifikasi teknologi yang digunakan pada suatu *website*. Ini berarti *Wappalyzer* bisa memberitahu Anda tentang *platform CMS* (seperti *Joomla* atau *WordPress*) hingga *library JavaScript* (seperti *jQuery* atau *React*) yang dipakai untuk membangun sebuah situs. *Ekstensi browser* sendiri adalah aplikasi yang terintegrasi dengan *browser* Anda (seperti *Google Chrome* atau *Firefox*) untuk menambah fungsionalitas dasarnya. Kebanyakan *browser* modern mendukung pemasangan ekstensi semacam ini [9].

#### F. Burpsuite

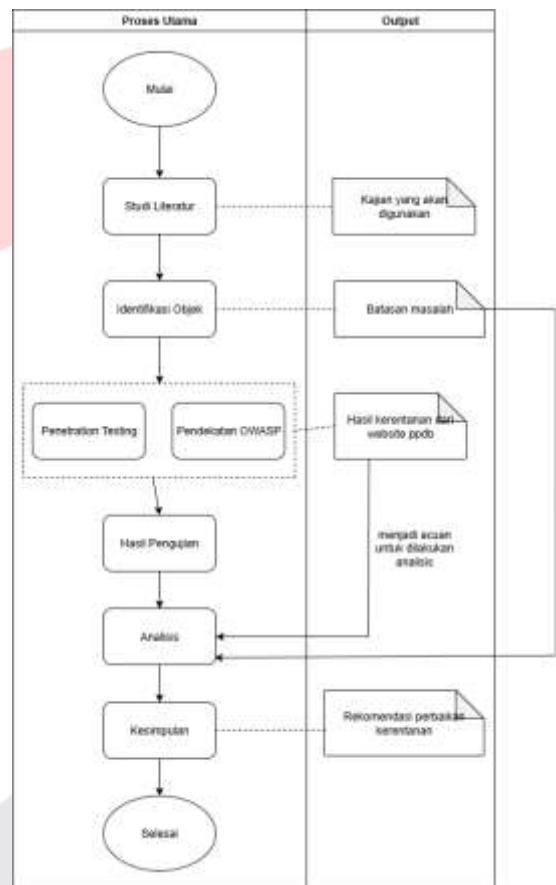
*Burp Suite* adalah alat yang digunakan untuk melakukan pengujian keamanan pada aplikasi *web*, dengan kemampuan untuk mengidentifikasi dan mengeksplorasi kerentanannya. Alat ini menyediakan berbagai fitur seperti *Intercepting Proxy*, *Scanner*, *Intruder*, dan *Repeater*, yang memungkinkan pengguna untuk menganalisis lalu lintas *HTTP/HTTPS*, mencari kerentanan otomatis, serta menguji ketahanan aplikasi terhadap input berbahaya. *Burp Suite* banyak digunakan oleh paraprofessional keamanan untuk memastikan bahwa aplikasi *web* yang diuji bebas dari celah yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab [10].

#### G. Kali Linux

*Kali Linux* adalah sistem operasi *Linux* yang didesain khusus untuk aplikasi *pentest*. *Kali Linux* menyediakan berbagai macam alat keamanan dan *pentest* yang dapat digunakan oleh tester keamanan untuk menguji keamanan sistem atau jaringan. *Kali Linux* juga didukung oleh berbagai macam fitur yang memungkinkan tester untuk melakukan berbagai macam tes keamanan [11]. *Kali Linux* juga dilengkapi dengan berbagai macam paket yang dapat membantu tester keamanan dalam menguji keamanan sistem atau jaringan.

### III. METODE

Dalam penelitian ini, metode yang digunakan untuk menganalisis keamanan *website* PPDB SMK XYZ adalah *Penetration Testing* dengan pendekatan *OWASP (Open Web Application Security Project) Top 10 - 2017*. Metode ini dipilih karena dapat mengidentifikasi dan mengevaluasi potensi kerentanan dalam sistem web secara sistematis dan komprehensif. Proses diawali dengan studi literatur serta identifikasi objek yang akan diuji. Pengujian dilakukan menggunakan *framework OWASP Top 10*, kemudian dilanjutkan dengan penetration testing untuk mendeteksi celah keamanan. Hasil pengujian kemudian dianalisis guna memahami tingkat risiko serta faktor penyebab kerentanan



GAMBAR 1  
(ALUR PENYELESAIAN PENELITIAN)

#### A. Studi Literatur

Pada tahap ini memiliki tujuan untuk menjelaskan kajian yang ada dari teori – teori penunjang yang mendukung untuk pelaksanaan penelitian ini. Kegiatan ini dilakukan dengan mencari referensi terkait melalui jurnal, artikel penelitian, dan melalui situs – situs yang ada pada internet.

#### B. Identifikasi Objek

Pada tahap ini penulis melakukan identifikasi objek penelitian yaitu *website* PPDB SMK XYZ. *Website* tersebut dibuat untuk memenuhi kebutuhan dalam melakukan pendaftaran calon peserta didik baru melalui media online. Pihak SMK XYZ dapat mengelola dokumen pendaftaran calon peserta didik melalui *website* PPDB tersebut.

#### C. Penetration Testing

Pada penelitian kali ini penulis melakukan pengujian berupa *penetration testing* menggunakan salah satu

pendekatan untuk mengetahui tingkat kerentanan sebuah website yaitu pendekatan *Open Web Application Security Project (OWASP)*. Penulis menggunakan *OWASP Top 10* sebagai standart acuan dalam mengetahui kerentanan website.

#### D. Hasil Pengujian

Pada tahap ini setelah penulis melakukan pengujian *penetration testing* maka penulis dapat memperoleh hasil dari pengujian terhadap keamanan website PPDB SMK XYZ. Hasil tersebut akan menjadi acuan dalam melakukan tahap analisis kemudian.

#### E. Analisis

Penulis melakukan analisis dari hasil pengujian *penetration testing* yang sudah dilakukan sebelumnya. Pada tahap ini akan diperoleh bagaimana kualitas keamanan dari website PPDB SMK XYZ yang telah diuji.

#### F. Kesimpulan

Pada tahap ini penulis akan menguraikan hasil yang diperoleh dari pengujian dan analisis tingkat keamanan dari website SMK XYZ. Tahap ini juga dapat digunakan untuk rekomendasi kepada pihak pengelola website SMK XYZ untuk meningkatkan kualitas dari keamanan website tersebut.

### IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari analisis keamanan pada website PPDB SMK XYZ. Proses analisis dilakukan melalui dua tahap utama: pengumpulan informasi (*information gathering*) untuk memetakan teknologi dan konfigurasi website , serta pengujian penetrasi (*penetration testing*) yang berfokus pada 10 risiko keamanan teratas menurut *OWASP Top 10 2017*.

#### A. Information Gathering

##### 1. Wappalyzer

Hasil *information gathering* menggunakan *Wappalyzer* menampilkan teknologi yang digunakan oleh sebuah website. Berikut hasil *information gathering* dengan menggunakan tools *Wappalyzer*.

TABEL 1  
(HASIL SCANNING WAPPALYZER)

<i>IP Address</i>	-
<i>Web Server</i>	<i>Nginx</i>
<i>UI Frameworks</i>	<i>Bootstrap</i>
<i>Web Frameworks</i>	<i>CodeIgniter</i>
<i>Font scripts</i>	<i>Google Font API</i>
	<i>Bootstrap Icons</i>
<i>Programming Languages</i>	<i>PHP</i>
<i>Javascript Libraries</i>	<i>Swiper</i>
	<i>Isotope</i>
	<i>AOS</i>

	<i>Lightbox</i>
	<i>JQuery 3.3.1</i>

Tabel 1 menjelaskan teknologi yang digunakan pada website yang diuji, dimulai dengan IP Address. Server web yang digunakan adalah Nginx, dan untuk kerangka kerja antarmuka pengguna (UI), website ini memanfaatkan Bootstrap. Di sisi server, website ini dibangun menggunakan web framework CodeIgniter, dengan PHP sebagai bahasa pemrogramannya. Untuk font, website ini mengintegrasikan Google Font API dan Bootstrap Icons. Berbagai library JavaScript juga digunakan, termasuk Swiper, Isotope, AOS, Lightbox, dan JQuery 3.3.1.

##### 2. Whois

Hasil *whois* memberikan informasi yang cukup lengkap mengenai seseorang atau organisasi yang bertanggung jawab atas sebuah *IP* atau domain, termasuk informasi kontak dan detail registrasi. Berikut hasil *information gathering* menggunakan tools *whois*.



```

person:
address:
address:
country:
phone:
e-mail:
nic-hdl:
mnt-by:
fax-no:
last-modified: 2021-03-09T04:50:43Z
source: IIONIC
  
```

GAMBAR 2  
(HASIL SCANNING DENGAN TOOLS WHOIS)

Hasil *whois lookup* yang ditampilkan di terminal berisi banyak informasi seperti detail kontak sebuah entitas, menampilkan hasil dari query informasi registrasi jaringan. Dalam hasil ini, terlihat bahwa data mencakup informasi pribadi seperti nama orang yang terdaftar *Adi S\*\*\*\*\**, alamat lengkap yang menunjukkan lokasi fisik di wilayah Kabupaten Malang, Indonesia (disensor sebagian untuk menjaga privasi), serta informasi kontak seperti nomor telepon, faksimile, dan alamat email (semuanya disamarkan). Selain itu, juga terdapat kode NIC handle (AS3093-AP) yang berfungsi sebagai identifikasi unik, serta informasi maintainer jaringan (MAINT-ID-\*\*\*\*\*). Data ini terakhir kali diperbarui pada 9 Maret 2021, dan bersumber dari IDNIC, yaitu Indonesia Network Information Center. Informasi seperti ini umumnya digunakan untuk keperluan administratif dan teknis, misalnya saat terjadi pelanggaran atau penyalahgunaan jaringan, namun tetap harus digunakan secara etis karena mengandung data pribadi yang sensitif.

##### 3. Nmap

Hasil pemindaian *port* jaringan yang terbuka dapat menggunakan tools *nmap* didapatkan bahwa website tersebut terbuka pada Port 21/tcp - *ftp* (*File Transfer Protocol*), Port 53/tcp - *domain* (*Domain Name System - DNS*), Port 80/tcp - *http* (*HyperText Transfer Protocol*), Port 110/tcp - *pop3* (*Post Office Protocol v3*), Port 143/tcp - *imap* (*Internet Message Access Protocol*), Port 443/tcp - *https* (*HyperText Transfer Protocol Secure*), Port 587/tcp - *submission* (*Email Message Submission*), Port 993/tcp - *imaps* (*IMAP Secure over SSL/TLS*), Port 995/tcp - *pop3s* (*POP3 Secure over SSL/TLS*).

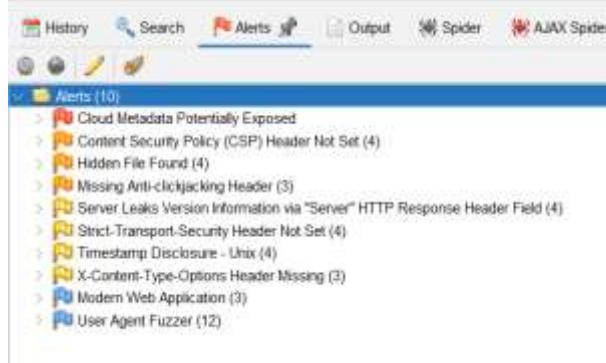
*SSL/TLS), dan Port 3306/tcp - mysql (MySQL database).* Setiap port tersebut dalam keadaan terbuka dengan status *syn-ack* dan *TTL 128*, menunjukkan bahwa layanan-layanan tersebut dapat diakses dari luar sistem.

TABEL 2  
(HASIL SCANNING NMAP)

Port	State	Service
21/tcp	open	<i>tcp</i>
53/tcp	open	<i>domain</i>
80/tcp	open	<i>http</i>
110/tcp	open	<i>pop3</i>
143/tcp	open	<i>imap</i>
443/tcp	open	<i>https</i>
587/tcp	open	<i>submission</i>
993/tcp	open	<i>imaps</i>
995/tcp	open	<i>pop3s</i>
3306/tcp	open	<i>mysql</i>

#### 4. Vulnerability Scanning

Proses scanning hasil *vulnerability* dilakukan dengan menggunakan *tools OWASP ZAP*. Hasil pemindaian menggunakan *tools OWASP ZAP* dapat dilihat pada gambar melalui proses *automated scan*.



GAMBAR 3

(HASIL VULNERABILITY SCANNING OWASP ZAP)

Hasil pemindaian keamanan aplikasi *web* menggunakan *tools OWASP ZAP*, telah mengidentifikasi sepuluh kategori peringatan yang menunjukkan adanya potensi celah keamanan. Peringatan yang paling kritis adalah "*Cloud Metadata Potentially Exposed*", yang mengindikasikan risiko tinggi di mana penyerang berpotensi mengakses data sensitif dari penyedia layanan *cloud* seperti kunci akses atau token keamanan, yang dapat berujung pada pengambilalihan infrastruktur *cloud*. Selain itu, ditemukan beberapa kerentanan signifikan terkait konfigurasi keamanan yang tidak memadai. Aplikasi ini tidak menerapkan *header Content Security Policy (CSP)*, sehingga lebih rentan terhadap serangan injeksi kode seperti *Cross-Site Scripting (XSS)*. Kerentanan terhadap serangan *Clickjacking* juga terdeteksi karena tidak adanya *header Anti-clickjacking (X-Frame-Options)*, yang memungkinkan penyerang menipu pengguna untuk melakukan aksi tanpa disadari. Pemindaian juga menemukan bahwa *header Strict-Transport-Security (HSTS)* tidak diatur, yang membuat pengguna rentan terhadap serangan *man-in-the-middle* saat terhubung melalui koneksi yang tidak aman. Ditemukannya "*Hidden File Found*" atau *file* tersembunyi juga menambah risiko, karena

*file* tersebut bisa saja berisi kode sumber atau kredensial yang seharusnya tidak dapat diakses publik. Pada tingkat risiko yang lebih rendah, pemindaian menemukan beberapa kebocoran informasi. *Server* secara terbuka mengungkapkan versi perangkat lunaknya ("*Server Leaks Version Information*") dan stempel waktu atau "*Timestamp Disclosure*", yang dapat memberikan petunjuk kepada penyerang untuk mengeksplorasi kerentanan yang telah diketahui pada versi tersebut. Peringatan "*X-Content-Type-Options Header Missing*" juga menunjukkan praktik keamanan yang kurang baik, meskipun risikonya tergolong rendah. Terakhir, beberapa temuan bersifat informasional, bukan celah keamanan. Peringatan "*Modern Web Application*" hanya mengonfirmasi bahwa target adalah aplikasi *web* modern, sementara "*User Agent Fuzzer*" menunjukkan bahwa alat pemindai telah melakukan pengujian fungsionalitas dengan berbagai tipe *User-Agent*, yang merupakan bagian dari proses analisis keamanan standar. Namun, hasil scanning *OWASP ZAP* yang memasuki kategori *OWASP Top 10* ditemukan sebanyak 8 kerentanan.

#### B. Penetration Testing

Bagian ini menjelaskan bagaimana sistem diuji berdasarkan metode penetration testing dengan pendekatan *OWASP Framework*. Pengujian keamanan terhadap website PPDB SMK XYZ dilakukan berdasarkan pendekatan *OWASP Top 10 2017*, yang mencakup sepuluh jenis kerentanan paling umum pada aplikasi web. Setiap pengujian dilakukan secara sistematis menggunakan berbagai tools seperti *Burp Suite*, *OWASP ZAP*, *Dirsearch*, dan *cURL*.

##### 1. A01-2017 SQL Injection

Pengujian ini dilakukan dengan mencoba mengeksplorasi kelemahan dalam proses autentikasi menggunakan teknik *SQL Injection Authentication Bypass*. Pengujian ini menggunakan *Burp Suite Intruder* dengan payload dari *PayloadBox SQL Injection list*. Meskipun tidak berhasil masuk ke sistem secara tidak sah, ditemukan perbedaan signifikan dalam panjang respons server pada payload tertentu (misalnya '), yang menandakan bahwa kueri SQL di backend kemungkinan besar terganggu oleh input tersebut. Hal ini menjadi indikasi kuat adanya potensi celah *SQL Injection* yang dapat dieksplorasi lebih lanjut, terutama jika sistem tidak memvalidasi input dengan baik.

TABEL 3  
(HASIL PENGUJIAN MENGGUNAKAN PAYLOAD SQL)

Analisis	Deskripsi	Implikasi Keamanan (Indikasi)
Alat dan Tujuan	Pengujian dilakukan menggunakan <i>Burp suite</i> dan <i>SQL Injection</i> .	Ada upaya aktif untuk mengidentifikasi kelemahan pada aplikasi <i>web</i> yang dapat dieksplorasi oleh penyerang.
Proses Pengujian	Alat ini mengirimkan serangkaian <i>payload</i> (muatan)	Pengujian difokuskan pada satu parameter <i>input</i> yang

	berbahaya ke satu titik masukan ( <i>Position: 1</i> ). <i>Payload</i> yang digunakan adalah karakter dan perintah <i>SQL</i> umum, seperti ‘ (tanda kutip tunggal), -- (komentar <i>SQL</i> ), dan <i>WAITFOR DELAY</i> (perintah untuk menunda <i>respons</i> ).	dicurigai tidak memvalidasi masukan pengguna dengan benar sebelum meneruskannya ke <i>database query</i> .
Hasil Signifikan	Baris 83 yang disorot menunjukkan hasil yang anomali. Ketika <i>payload</i> yang dikirimkan adalah satu tanda kutip ('), <i>respons</i> dari server memiliki panjang ( <i>length</i> ) 873 byte. Ini berbeda secara signifikan dari <i>respons</i> untuk <i>payload</i> lain yang umumnya memiliki panjang sekitar 793 byte	Ini adalah indikator kuat adanya celah keamanan <i>SQL Injection</i> . Perbedaan panjang <i>respons</i> menunjukkan bahwa <i>server</i> memproses <i>input</i> tanda kutip tersebut secara berbeda, yang kemungkinan besar karena <i>input</i> tersebut merusak <i>query SQL</i> di <i>backend</i> .
Kesimpulan	Aplikasi <i>web</i> PPDB SMK XYZ sangat mungkin rentan terhadap serangan <i>SQL Injection</i> . Anomali pada panjang <i>respons</i> dan pesan eror yang spesifik Ketika dikirimkan <i>payload</i> (') merupakan bukti yang kuat adanya kerentanan tersebut.	Jika celah ini tidak segera diperbaiki, penyerang dapat mengeksloitasinya lebih lanjut untuk mencuri data sensitif, memodifikasi data, atau bahkan mengambil alih control atas <i>database server</i> .

## 2. A02-2017 Broken Authentication

Pengujian dilakukan dengan teknik *Brute Force Attack* terhadap halaman *login*. Teknik ini untuk mencoba masuk ke dalam sistem dengan mencoba berbagai kombinasi *username* dan *password* secara otomatis hingga menemukan yang benar. Teknik ini biasanya dilakukan dengan bantuan alat *Burpsuite*. Meski permintaan berhasil dikirimkan ke server, sistem menampilkan kode status *HTTP 505*, yang menandakan bahwa server menolak permintaan karena versi *HTTP* yang digunakan tidak didukung. Dalam hal ini, proteksi dari *Cloudflare* terdeteksi aktif dan memblokir permintaan otomatis dari *Burp Suite*. Walaupun sistem tampak cukup aman dari brute force, penambahan mekanisme *rate-limiting* serta *lockout* otomatis setelah percobaan *login* gagal tetap disarankan sebagai pertahanan tambahan.



GAMBAR 4  
(HASIL PERCOBAAN BRUTE FORCE HALAMAN LOGIN)

## 3. A03-2017 Sensitive Data Exposure

Dari hasil pengujian ini, ditemukan file-file konfigurasi penting seperti *.htpasswd*, *.gitignore*, dan *.idea/workspace.xml* melalui pemindaian direktori menggunakan *Dirsearch* yang dapat dilihat pada gambar 5. File-file ini dapat diakses secara publik, yang seharusnya tidak terjadi karena berisiko membocorkan informasi konfigurasi internal, termasuk struktur folder proyek dan potensi kredensial pengguna. Ini menunjukkan bahwa file sensitif belum diamankan dengan baik, dan penyerang dapat memanfaatkannya untuk melakukan eksploitasi lanjutan.



GAMBAR 5  
(HASIL EKSEKUSI PERINTAH DIRSEARCH)

## 4. A04-2017 XML External Entities (XXE)

Pada hasil pemindaian kerentanan dari aplikasi *OWASP ZAP*, tidak ditemukan bahwa sistem dapat menerima input *XML* yang tidak tervalidasi dengan baik, hal tersebut membuat penyerang tidak dapat mengakses data sensitif yang ada pada *server website* PPDB SMK XYZ. Serangan *XXE* dapat terjadi ketika aplikasi atau sistem memproses *input XML* yang mengandung entitas eksternal, yang dapat merujuk ke sumber daya eksternal seperti file sistem, *URL*, atau bahkan sistem operasi. Hasil pengujian pada tabel menunjukkan bahwa respons dari request tersebut memberikan respons 200 (OK) yang berarti permintaan request pada *website* berhasil diproses, namun berisi eror dengan pesan “nomor tidak terdaftar pada whatsapp” yang mengindikasikan adanya validasi, dimana sistem melakukan pengecekan terhadap nomor telepon di *whatsapp* dan gagal. Hal tersebut dapat dianalisis bahwa tidak ditemukan adanya celah *XXE*. Respon dari pengujian

tidak mengandung entitas *XML* yang tidak valid atau data apapun yang mengindikasikan kegagalan pemrosesan *XML*.

```
curl: (35) [SSL: handshake failed] The TLS connection was non-properly terminated
```

GAMBAR 6  
(HASIL PENGUJIAN DENGAN PAYLOADS XXE)

#### 5. A05-2017 Broken Access Control

Percobaan ini dilakukan untuk mengetahui apakah *user* bisa bisa masuk ke halaman admin atau tidak. Kerentanan terdeteksi karena terdapat potongan kode *form* yang berisi berisi *form* cetak bukti menggunakan metode *POST* yang tidak mengaktifkan *Anti-CSRF*. Form ini tidak menerapkan token *Anti-CSRF*, yang memungkinkan penyerang memanfaatkan celah *Cross-Site Request Forgery* dengan mengirimkan request palsu atas nama pengguna aktif. Pengujian dengan tool Form CSRF di ZAP pada gambar mengonfirmasi bahwa sistem tidak memiliki perlindungan terhadap request palsu semacam ini, sehingga perlu penerapan token validasi untuk setiap aksi yang melibatkan otorisasi pengguna.



GAMBAR 7  
(HASIL TEMUAN DARI TOOLS CSRF)

#### 6. A06-2017 Security Misconfiguration

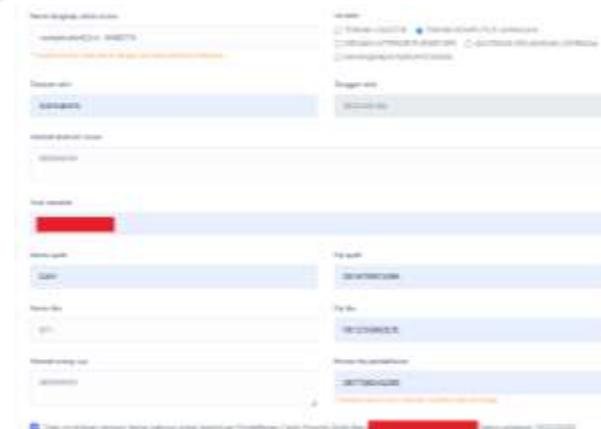
Beberapa kelemahan konfigurasi ditemukan, seperti *Content Security Policy (CSP)* yang tidak diterapkan, *Header Strict-Transport-Security (HSTS)* yang tidak diaktifkan, serta *cookie session* yang tidak memiliki *Secure Flag*, yang membuatnya rentan terhadap serangan *Man-in-the-Middle (MITM)*. Kerentanan *Cloud Metadata Potentially Exposed* terjadi ketika *server* atau aplikasi yang berjalan di layanan *cloud* memberikan akses tidak sah ke metadata *instance*. Dari hasil pengujian pada gambar ditemukan bahwa sistem belum mengaktifkan header keamanan penting seperti *Content Security Policy (CSP)*, *Strict-Transport-Security (HSTS)*, serta *X-Content-Type-Options*. Selain itu, *cookie* sesi tidak diberi atribut *Secure Flag*, yang membuatnya rentan terhadap serangan *man-in-the-middle (MITM)*. Percobaan untuk mengakses metadata *cloud* menggunakan *curl* pada endpoint */latest/meta-data/* gagal karena *TLS handshake error*, namun tetap menunjukkan bahwa sistem berpotensi salah konfigurasi. Hal ini harus diperbaiki untuk mencegah kebocoran data instance dari penyedia layanan *cloud* seperti AWS atau Azure.

```
curl: (35) [SSL: handshake failed] The TLS connection was non-properly terminated
```

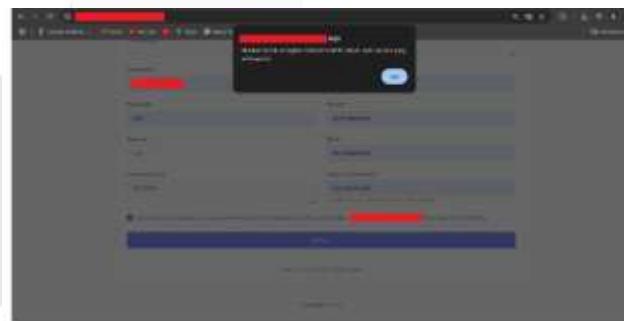
GAMBAR 8  
(HASIL PERINTAH cURL)

#### 7. A07-2017 Cross-Site Scripting (XSS)

*Cross-Site Scripting (XSS)* adalah kerentanan keamanan web dimana penyerang dapat menyisipkan kode berbahaya (umumnya *JavaScript*) ke dalam *website* yang kemudian dieksekusi di browser pengguna lain. Serangan ini terjadi ketika *website* tidak melakukan validasi yang tepat terhadap input pengguna. Melalui *XSS*, penyerang dapat mencuri data sensitif, mengubah tampilan *website*, atau mengambil alih akun pengguna karena skrip berbahaya tersebut akan berjalan dengan hak akses yang sama seperti *website* yang *legitimate*. Pengujian dilakukan dengan menyisipkan kode `<script>alert(1)</script>` pada form input pengguna. Hasilnya pada gambar beberapa input tidak melakukan validasi atau sanitasi dengan baik, sehingga script berhasil dijalankan di browser. Ini menunjukkan bahwa sistem masih rentan terhadap XSS yang dapat dimanfaatkan untuk mencuri cookie pengguna, melakukan *session hijacking*, atau memanipulasi tampilan halaman *web*.



GAMBAR 9  
(FORMULIR PENDAFTARAN SEKOLAH)



GAMBAR 10  
(HASIL PENGUJIAN KERENTANAN XSS)

#### 8. A08-2017 Insecure Deserialization

Kerentanan deserialization dalam *Java* terjadi ketika aplikasi menerima objek yang deserialisasi dari sumber yang tidak tepercaya dan mendeserialisasikannya tanpa validasi yang memadai. Pada pengujian ini digunakan Java Deserialization Scanner dari OWASP ZAP untuk mengidentifikasi apakah ada objek serialized yang tidak aman. Pengujian ini tidak menemukan adanya payload yang berhasil dieksekusi, atau tanda-tanda adanya sistem yang memproses deserialisasi objek dari klien yang ada pada gambar. Meski demikian, fitur ini harus diaudit

kembali apabila sistem mulai menerima upload data dalam format binary atau serialized di masa depan.

Results:

- Apache Commons Collection 3 (Sleep): NOT vulnerable. Response time: 2605 milliseconds
- Spring Alternate Payload (Sleep): NOT vulnerable. Response time: 832 milliseconds
- Apache Commons Collection 4 (Sleep): NOT vulnerable. Response time: 854 milliseconds
- JavaScriptNef (Sleep): NOT vulnerable. Response time: 836 milliseconds
- JSON (Sleep): NOT vulnerable. Response time: 840 milliseconds
- Apache Commons Collection 3 Alternate payload 2 (Sleep): NOT vulnerable. Response time: 634 milliseconds
- ROM (Sleep): NOT vulnerable. Response time: 821 milliseconds
- Mozilla Rhino 19 (Sleep): NOT vulnerable. Response time: 835 milliseconds
- Apache Commons Collection 4 Alternate payload (Sleep): NOT vulnerable. Response time: 870 milliseconds
- Java 8 Up to Java 10 (Sleep): NOT vulnerable. Response time: 842 milliseconds
- Java 9 and Java 11 Up to Java 17 (Sleep): NOT vulnerable. Response time: 853 milliseconds
- Apache Commons Collection 3 Alternate payload 4 (Sleep): NOT vulnerable. Response time: 848 milliseconds
- Hibernate 5 (Sleep): NOT vulnerable. Response time: 849 milliseconds
- Conversion Beans/JPA (Sleep): NOT vulnerable. Response time: 834 milliseconds
- Apache Commons Collection 3 Alternate payload 3 (Sleep): NOT vulnerable. Response time: 827 milliseconds
- iBatis Interceptor (Sleep): NOT vulnerable. Response time: 843 milliseconds
- Spring (Sleep): NOT vulnerable. Response time: 835 milliseconds
- Mozilla Rhino Alternate payload (Sleep): NOT vulnerable. Response time: 842 milliseconds
- Apache Commons Collection 3 Alternate payload 5 (Sleep): NOT vulnerable. Response time: 829 milliseconds

END

GAMBAR 11  
(HASIL DESERIALIZATION SCANNING)

## 9. A09-2017 Using Component With Known Vulnerabilities

Website PPDB XYZ menggunakan *jQuery* versi 3.3.1 *slim min* yang memiliki beberapa kerentanan serius. Pertama, terdapat kerentanan *Object Prototype Pollution* pada fungsi *jQuery.extend()* (*CVE-2019-11358*) yang dapat memungkinkan penyerang memodifikasi properti objek *JavaScript*. Kedua, terdapat kerentanan *XSS* pada *jQuery.htmlPrefilter* (*CVE-2020-11022*) yang dapat dieksloitasi untuk menginjeksi dan mengeksekusi kode *JavaScript* berbahaya. Ketiga, terdapat kerentanan pada pemrosesan elemen *<option>* (*CVE-2020-11023*) yang dapat menyebabkan eksekusi kode tidak terpercaya saat menggunakan metode *DOM manipulation*. Selain itu, website juga menggunakan *Bootstrap* versi 3.3.7 yang sudah *end-of-life* dan memiliki beberapa kerentanan *XSS*, antara lain pada atribut *data-viewport tooltip* (*CVE-2018-20676*), properti *data-container tooltip* (*CVE-2018-14042*), dan atribut *data-target* (*CVE-2016-10735*). *Library* ini digunakan di beberapa halaman termasuk *homepage*, halaman cetak bukti, dan halaman tamu. Hasil analisis menunjukkan bahwa situs ini menggunakan *jQuery v3.3.1 slim min* serta *Bootstrap v3.3.7*. Setelah mengidentifikasi versi *library* tersebut, dilakukan verifikasi terhadap *database CVE* (*Common Vulnerabilities and Exposures*) untuk mengetahui apakah versi yang digunakan memiliki kerentanan yang dapat dieksloitasi. Jika ditemukan celah keamanan dalam versi yang digunakan, maka aplikasi berisiko terhadap serangan seperti *XSS* (*Cross-Site Scripting*), *Prototype Pollution*, atau *Remote Code Execution (RCE)*, sehingga diperlukan pembaruan atau mitigasi untuk mencegah potensi eksloitasi.



GAMBAR 12  
(SCRIPT PENGUJIAN JQUERY)

## 10. A10-2017 Insufficient Logging & Monitoring

*Insufficient Logging & Monitoring* adalah kerentanan keamanan yang terjadi ketika sistem atau aplikasi tidak memiliki sistem pencatatan dan pemantauan yang memadai terhadap aktivitas keamanan yang penting. Kondisi ini membuat organisasi tidak dapat mendeteksi, melacak, dan merespons serangan atau aktivitas mencurigakan secara efektif. Tanpa logging dan monitoring yang baik, penyerang dapat melakukan serangan, mengakses sistem, dan memanipulasi data tanpa terdeteksi, serta mempersulit proses investigasi forensik ketika terjadi insiden keamanan. Hasil dari pengujian yang ada pada gambar menunjukkan bahwa tidak ada pencatatan atau notifikasi keamanan yang muncul saat dilakukan brute-force login, percobaan injection, maupun permintaan tidak sah. Tidak adanya log aktivitas mencurigakan ini berarti jika terjadi serangan nyata, pihak admin tidak akan segera mengetahui atau meresponsnya, yang bisa memperparah dampak serangan. Oleh karena itu, implementasi sistem logging dan alert real-time sangat diperlukan untuk mendeteksi aktivitas anomali sedini mungkin.



GAMBAR 13  
(PENGUJIAN DENGAN INTRUDER BURP SUITE)

## V. KESIMPULAN

Berdasarkan hasil analisis dan pengujian keamanan pada website PPDB SMK XYZ menggunakan pendekatan OWASP Top 10 tahun 2017, dapat disimpulkan bahwa kualitas keamanan sistem masih tergolong rentan terhadap berbagai serangan siber. Dari sepuluh kategori kerentanan yang diuji, ditemukan tujuh kategori yang berisiko tinggi, yaitu *Injection*, *Sensitive Data Exposure*, *Broken Access Control*, *Security Misconfiguration*, *Cross-Site Scripting (XSS)*, penggunaan komponen yang memiliki kerentanan yang diketahui, serta *Insufficient Logging & Monitoring*. Beberapa temuan utama meliputi adanya celah *SQL Injection*, file sensitif yang dapat diakses publik, form tanpa perlindungan *Anti-CSRF*, kesalahan konfigurasi keamanan, serta penggunaan library usang seperti *jQuery v3.3.1* dan *Bootstrap v3.3.7*. Selain itu, sistem pencatatan dan pemantauan aktivitas juga belum optimal dalam mendeteksi ancaman. Penelitian ini berhasil merumuskan rekomendasi teknis yang dapat diterapkan untuk memperbaiki kelemahan tersebut, seperti penggunaan parameterized queries, pembaruan library, penerapan token *Anti-CSRF*, dan pengujian konfigurasi server. Dengan mplementasikan rekomendasi tersebut, diharapkan postur keamanan website PPDB SMK XYZ dapat meningkat secara signifikan, melindungi data pribadi calon peserta didik, dan menumbuhkan kepercayaan pengguna terhadap sistem yang digunakan.

## REFERENSI

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia, “Jumlah Pengguna Internet Di Indonesia Tembus 221 Juta Orang,” APJII. Accessed: Jun. 10, 2025. [Online]. Available: <https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang>
- [2] Y. Mulyanto and E. Haryanti, “Sumbawa Menggunakan Metode Vulnerability Asesment,” *JINTEKS*, vol. 3, no. 3, 2021, doi: 10.51401.
- [3] A. Elanda and R. Lintang Buana, “Analisis Keamanan Sistem Informasi Berbasis Website Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review,” 2020. [Online]. Available: [www.xyz.com](http://www.xyz.com)
- [4] I. O. Riandhanu, “Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi,” *Jurnal Informasi dan Teknologi*, Oct. 2022, doi: 10.37034/jidt.v4i3.236.
- [5] S. Hidayatulloh and D. Saptadiaji, “Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP),” Aug. 2021. [Online]. Available: <http://jurnal.itg.ac.id/>
- [6] CNBC, “7 Pengertian Website Menurut Ahli, Lengkap Jenis & Fungsinya.” Accessed: Jan. 19, 2019. [Online]. Available: <https://www.cnbcindonesia.com/tech/20220618152119-37-348229/7-pengertian-website-menurut-ahli-lengkap-jenis-fungsinya>
- [7] S. E. Prasetyo and N. Hassanah, “Analisis Keamanan Website Universitas Internasional Batam Menggunakan Metode ISSAF,” 2021.
- [8] Eriga Syifaudin, “Mengenal Tentang WHOIS: Pengertian, Jenis dan Cara Kerja,” Website. Accessed: Jul. 17, 2025. [Online]. Available: <https://www.exabytes.co.id/blog/apa-itu-whois-domain/>
- [9] Revou, “Apa itu Wappalyzer? Arti, Fungsi, Contoh, FAQs 2025 | Revou,” Website. Accessed: Jul. 17, 2025. [Online]. Available: <https://www.revou.co.id/kosakata/wappalyzer>
- [10] V. Dwi Agustina, T. Ariyadi, T. Syah Putra, A. Lega, P. Teknik Komputer, and U. Bina Darma Palembang Jl Jenderal Ahmad Yani No, “Teknik Pengujian Penetrasi Http Menggunakan Tools Burp Suite Pada Kali Linux,” vol. 4, no. 1, pp. 16–21, 2025, doi: 10.55123.
- [11] M. I. Rusdi and D. Prasti, “Penetration Testing Pada Jaringan Wifi Menggunakan Kali Linux,” 2019.