# BAB 4
# HASIL PENELITIAN

## 4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini bertujuan untuk menganalisis kerentanan keamanan pada situs web SICAMA Universitas Jenderal Achmad Yani Yogyakarta dengan menggunakan berbagai alat pemindai keamanan seperti *Nikto Scanner, Helium Security*, dan ZAP (*Zed Attack Proxy*). Alat pemindai *Nikto Scanner* menemukan beberapa header keamanan yang diterapkan dengan baik (*x-content-type-options, x-xss-protection, strict-transport-security, content-security-policy, x-frame-options*), namun juga mengidentifikasi kebocoran *ETags* melalui file *`/f1wGtBPH.show`* yang dapat menyebabkan kebocoran informasi terkait *inode server file system*, serta file *`robots.txt`* yang mengandung *entry* yang harus diperiksa manual. *Cookies PHPSESSID* ditemukan tanpa *secure flag* dan *httponly flag*, membuatnya rentan terhadap *sniffing* dan serangan XSS. Selain itu, ditemukan penggunaan sertifikat *wildcard *.unjaya.ac.id* dan beberapa file dan direktori yang terekspos seperti *`/WEB-INF/web.xml`* dan *`/mail/`*. Pemindaian menggunakan *Helium Security* tidak menemukan kerentanan dengan kategori risiko tinggi (*high*), tetapi mengidentifikasi lima kerentanan dengan risiko sedang (*medium*), empat kerentanan dengan risiko rendah (*low*), dan enam kerentanan informatif (*informational*).

Hasil pemindaian menggunakan ZAP (*Zed Attack Proxy*) 2.14.0 menemukan adanya *SQL Injection* pada *endpoint `/proseslogin.php`*, serta kerentanan *CSP: Wildcard Directive, CSP: script-src unsafe-inline,* dan *CSP: style-src unsafe-inline*. Selain itu, ditemukan *library JavaScript* yang rentan (*moment.js* versi 2.21.0) pada *endpoint `/vendors/scripts/script.min.js`*, ketidakhadiran token Anti-CSRF pada formulir *login*, dan file tersembunyi seperti *`.hg`* yang dapat diakses. Hasil penelitian ini menunjukkan bahwa meskipun ada beberapa langkah keamanan yang telah diterapkan, masih terdapat beberapa kerentanan yang perlu diperbaiki untuk meningkatkan keamanan sistem secara keseluruhan. Rekomendasi

perbaikan mencakup penggunaan *header* keamanan yang lebih ketat, implementasi token Anti-CSRF, dan pembaruan *library* yang rentan. Dengan demikian, situs web SICAMA Universitas Jenderal Achmad Yani Yogyakarta dapat menjadi lebih aman dan dapat diandalkan dalam memberikan layanan informasi kepada penggunanya.

## 4.2 PERENCANAAN (*PLANNING*)

Tahap perencanaan (*planning*) merupakan langkah awal yang esensial untuk memastikan kelancaran dan keberhasilan penelitian penilaian kerentanan terhadap SICAMA Universitas Jenderal Achmad Yani Yogyakarta. Pada tahap ini, peneliti mempersiapkan semua bahan dan alat yang diperlukan, termasuk perangkat lunak pemindaian keamanan serta mengumpulkan dokumentasi teknis mengenai sistem SICAMA untuk memahami konfigurasi awal infrastruktur server. Peneliti kemudian merancang jadwal penelitian yang mencakup seluruh tahapan, dari perencanaan (*planing*), penemuan (*discovery*), pengujian (*attacking*), hingga pelaporan (*reporting*). Selanjutnya, koordinasi proyek dilakukan melalui komunikasi dan kolaborasi dengan pihak Pusat Sistem Informasi, serta penyiapan dokumen izin penelitian untuk diserahkan kepada Kepala Pusat Sistem Informasi Universitas Jenderal Achmad Yani Yogyakarta.

## 4.3 PENEMUAN (*DISCOVERY*)

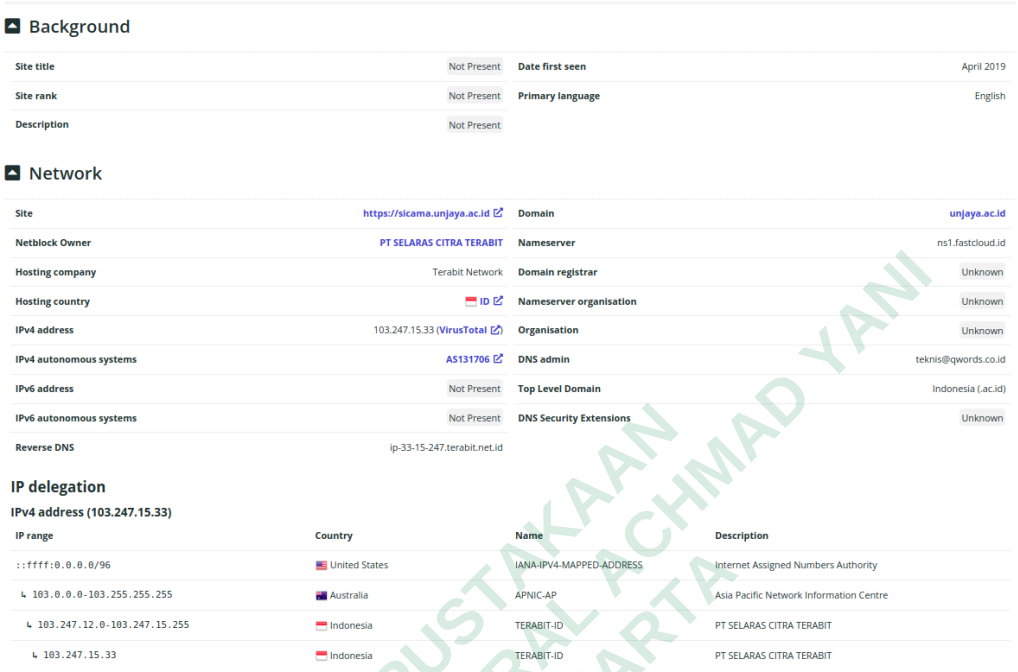
Tahap penemuan (*discovery*) dalam penelitian ini meliputi pengumpulan informasi *(information gathering)* dan pemindaian kerentanan *(vulnerability scanning)*. Pengumpulan informasi dilakukan menggunakan *Netcraft* untuk mengidentifikasi alamat IP dan konfigurasi *server*, serta Nmap untuk memetakan jaringan dan mengidentifikasi port yang terbuka. Pemindaian kerentanan dilakukan menggunakan Nikto untuk mendeteksi potensi kerentanan pada aplikasi web.

### 4.3.1 *Information Gathering*

1. *Netcraft*

Tahap awal penilaian kerentanan situs web adalah *information gathering*. Dalam penelitian ini, penulis menggunakan alat yang tersedia di

*website sitereport.netcraft.com*. *Netcraft* digunakan sebagai pemindai untuk menemukan informasi yang digambarkan pada Gambar 4.1.



Background

| | | | |
|---|---|---|---|
| Site title | Not Present | Date first seen | April 2019 |
| Site rank | Not Present | Primary language | English |
| Description | Not Present | | |

Network

| | | | |
|---|---|---|---|
| Site | https://sicama.unjaya.ac.id | Domain | unjaya.ac.id |
| Netblock Owner | PT SELARAS CITRA TERABIT | Nameserver | ns1.fastcloud.id |
| Hosting company | Terabit Network | Domain registrar | Unknown |
| Hosting country | ID | Nameserver organisation | Unknown |
| IPv4 address | 103.247.15.33 (VirusTotal) | Organisation | Unknown |
| IPv4 autonomous systems | AS131706 | DNS admin | teknis@qwords.co.id |
| IPv6 address | Not Present | Top Level Domain | Indonesia (.ac.id) |
| IPv6 autonomous systems | Not Present | DNS Security Extensions | Unknown |
| Reverse DNS | ip-33-15-247.terabit.net.id | | |

IP delegation

IPv4 address (103.247.15.33)

| IP range | Country | Name | Description |
|---|---|---|---|
| ::ffff:0.0.0.0/96 | United States | IANA-IPV4-MAPPED-ADDRESS | Internet Assigned Numbers Authority |
| ↳ 103.0.0.0-103.255.255.255 | Australia | APNIC-AP | Asia Pacific Network Information Centre |
| ↳ 103.247.12.0-103.247.15.255 | Indonesia | TERABIT-ID | PT SELARAS CITRA TERABIT |
| ↳ 103.247.15.33 | Indonesia | TERABIT-ID | PT SELARAS CITRA TERABIT |

**Gambar 4.1** Pemindaian dengan *Netcraft*

Setelah dilakukan pemindaian menggunakan *netcraft*, didapatkan hasil berupa kumpulan informasi detail mengenai alamat IP dan berbagai informasi umum terkait situs web Sicama Unjaya. Informasi yang terkumpul ini mencakup data teknis yang esensial untuk analisis keamanan dan performa situs web. Hasil pemindaian tersebut disajikan secara sistematis dalam bentuk tabel yang memberikan gambaran komprehensif tentang elemen-elemen penting dari situs web. Beberapa informasi penting yang diperoleh dari hasil pemindaian ini dapat dilihat pada Gambar 4.1.

**Tabel 4.1** Hasil *Information Gathering* menggunakan *Netcraft*

| No | Informasi | Hasil |
|---|---|---|
| 1 | *Site title* | *Not Present* |
| 2 | *Date first seen* | *April 2019* |

| No | Informasi | Hasil |
|---|---|---|
| 3 | *Site* | *https://sicama.unjaya.ac.id* |
| 4 | *Netblock Owner* | *PT SELARAS CITRA TERABIT* |
| 5 | *Hosting company* | *Terabit Network* |
| 6 | *Housting Country* | *ID* |
| 7 | *IPv4 address* | *103.247.15.33* |
| 8 | *Reverse DNS* | *ip-35-15-247.terabit.net.id* |
| 9 | *Main Domain* | *unjaya.ac.id* |
| 10 | *Nameserver* | *ns1.fastcloud.id* |
| 11 | *Top Level Domain* | *Indonesia (.ac.id)* |
| 12 | *DNS admin* | *teknis@qwords.co.id* |

Beberapa informasi yang ditemukan dari tahap *Information Gathering* seperti *Netblock Owner PT SELARAS CITRA TERABIT, Hosting company Terabit Network, IPv4 address 103.247.15.33, Reverse DNS ip-35-15-247.terabit.net.id, Main Domain unjaya.ac.id, Nameserver ns1.fastcloud.id, dan DNS admin teknis@qwords.co.id*.

*2. Network Mapper (NMAP)*

Untuk memulai proses evaluasi keamanan jaringan situs web target, langkah pertama yang dilakukan adalah mengecek koneksi ke *server* dengan menggunakan perintah *ping*. Tujuan dari langkah ini adalah untuk memastikan bahwa situs web target, yaitu *sicama.unjaya.ac.id*, dapat dijangkau dan merespons permintaan jaringan. Dalam proses ini, peneliti mengirimkan 5 paket data ke server menggunakan perintah *ping*, dan hasilnya menunjukkan bahwa semua paket berhasil terkirim tanpa adanya kehilangan paket, seperti yang ditunjukkan pada Gambar 4.2 berikut.

```
❯ ping sicama.unjaya.ac.id
PING sicama.unjaya.ac.id (103.247.15.33) 56(84) bytes of data.
64 bytes from ip-33-15-247.terabit.net.id (103.247.15.33): icmp_seq=1 ttl=51 time=49.5 ms
64 bytes from ip-33-15-247.terabit.net.id (103.247.15.33): icmp_seq=2 ttl=51 time=59.1 ms
64 bytes from ip-33-15-247.terabit.net.id (103.247.15.33): icmp_seq=3 ttl=51 time=65.2 ms
64 bytes from ip-33-15-247.terabit.net.id (103.247.15.33): icmp_seq=4 ttl=51 time=50.4 ms
64 bytes from ip-33-15-247.terabit.net.id (103.247.15.33): icmp_seq=5 ttl=51 time=61.9 ms
^C
--- sicama.unjaya.ac.id ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 49.480/57.217/65.170/6.241 ms
```

**Gambar 4.2** *Check* konektivitas dengan *server*

Tahap selanjutnya adalah tahap pemetaan jaringan (*network mapping*) dilakukan untuk mengetahui konfigurasi jaringan pada situs web target, dengan menggunakan alat *Nmap*. Informasi yang telah diperoleh pada tahap sebelumnya digunakan untuk mengidentifikasi topologi jaringan situs web tersebut. Selanjutnya, untuk proses pemindaian *port* (*port scanning*), peneliti menggunakan alat NMAP versi 7.93. Pemindaian dilakukan dengan beberapa perintah, yang pertama adalah perintah *nmap -sS -sV sicama.unjaya.ac.id*. Perintah ini bertujuan untuk mengidentifikasi port yang terbuka serta layanan dan versi yang berjenis *Transmission Control Protocol* (TCP).

```
root@hidayat:/# nmap -sS -sV sicama.unjaya.ac.id
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-26 22:19 WIB
Nmap scan report for sicama.unjaya.ac.id (103.247.15.33)
Host is up (0.059s latency).
rDNS record for 103.247.15.33: ip-33-15-247.terabit.net.id
Not shown: 998 filtered ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     nginx
443/tcp open  ssl/http nginx

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.35 seconds
```

**Gambar 4.3** Hasil TCP *Scan*

Berdasarkan Gambar 4.3, ditemukan bahwa terdapat port yang terbuka dan dapat diakses oleh pengguna. Informasi lebih rinci mengenai port-port yang terbuka ini dijabarkan pada Tabel 4.2.

**Tabel 4.2** Hasil TCP *Scan*

| ***Port*** | ***Status*** | ***Service*** | ***Version*** |
|---|---|---|---|
| *80/tcp* | *open* | *http* | *nginx* |
| *443/tcp* | *open* | *ssl/http* | *nginx* |

Setelah itu, penulis melakukan pengecekan jenis *User Datagram Protocol (*UDP) dengan perintah *command nmap –sU sicama.unjaya.ac.id*, yang memungkinkan untuk mengetahui status port yang terbuka. Karena ada port tambahan selain http dan https, oleh karena itu pengecekan jenis UDP diperlukan.

```
root@hidayat:/# nmap -sU sicama.unjaya.ac.id
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-26 22:21 WIB
Nmap scan report for sicama.unjaya.ac.id (103.247.15.33)
Host is up (0.24s latency).
rDNS record for 103.247.15.33: ip-33-15-247.terabit.net.id
Not shown: 995 closed ports
PORT    STATE         SERVICE
53/udp  open|filtered domain
67/udp  open|filtered dhcps
68/udp  open|filtered dhcpc
123/udp open          ntp
161/udp open          snmp

Nmap done: 1 IP address (1 host up) scanned in 56.05 seconds
```

**Gambar 4.4** Hasil UDP *Scan*

Hasil dari pemindaian UDP menunjukkan bahwa terdapat port yang berstatus terbuka (open) namun terfilter (filtered), yang mengindikasikan adanya firewall yang melindungi setiap port tersebut. Hal ini ditunjukkan oleh notifikasi yang menyatakan "*[port] open|filtered UDP ports*" pada hasil pemindaian. Informasi lebih rinci mengenai port-port yang terbuka ini dijabarkan pada Tabel 4.3.

**Tabel 4.3** Hasil UDP *Scan*

| ***Port*** | ***Status*** | ***Service*** |
|---|---|---|
| *53/ucp* | *open\|filtered* | *domain* |
| *67/udp* | *open\|filtered* | *dhcps* |

| *Port* | *Status* | *Service* |
|---|---|---|
| *68/udp* | *open\|filtered* | *dhcpc* |
| *123/udp* | *open* | *ntp* |
| *161/udp* | *open* | *snmp* |

Selanjutnya, proses *Maimon scan* dilakukan menggunakan perintah *nmap -sM sicama.unjaya.ac.id* yang bertujuan untuk memastikan apakah port tertentu terhalangi oleh firewall atau tidak. Perintah ini membantu mengidentifikasi adanya filter atau aturan firewall yang mungkin memblokir akses ke port yang seharusnya terbuka.

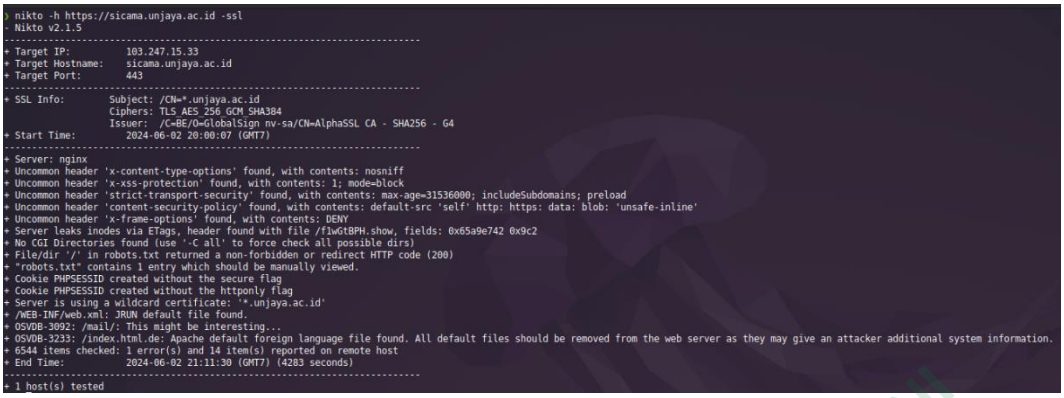
```
root@hidayat:/# nmap -sM sicama.unjaya.ac.id
Starting Nmap 7.80 ( https://nmap.org ) at 2024-05-26 22:46 WIB
Nmap scan report for sicama.unjaya.ac.id (103.247.15.33)
Host is up (0.15s latency).
rDNS record for 103.247.15.33: ip-33-15-247.terabit.net.id
All 1000 scanned ports on sicama.unjaya.ac.id (103.247.15.33) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 10.26 seconds
```

**Gambar 4.5** Hasil *Maimon Scan*

Hasil pemindaian menggunakan perintah *nmap –sM sicama.unjaya.ac.id* pada situs web target menunjukkan bahwa seluruh port berada dalam keadaan tertutup (*closed*). Temuan ini mengindikasikan bahwa tidak ada port yang terbuka atau terfilter pada situs web tersebut, sehingga meminimalisir potensi risiko dari akses yang tidak sah.

### 4.3.2 *Vulnerability Scanning*

Pemindaian kerentanan ini menggunakan *tools Nikto* bertujuan untuk mengumpulkan informasi celah kerentanan situs web target. *Nikto*, yang merupakan alat pemindai situs web, dijalankan pada sistem operasi *Linux*. Untuk memulai pemindaian, perintah yang digunakan adalah *nikto -h https://sicama.unjaya.ac.id/ -*ssl. *Nikto* akan melakukan pemindaian selama sekitar 60 menit, mengidentifikasi potensi kerentanan pada situs web tersebut.




**Gambar 4.6** Hasil pemindaian *Nikto Scanner*

Gambar 4.5 menunjukkan bahwa pemindaian menggunakan *tools Nikto* berhasil mengumpulkan data penting dari situs web Sicama Unjaya. Hasil pemindaian ini mengungkapkan bahwa *server* yang digunakan oleh situs web tersebut adalah *Nginx* yang berjalan pada sistem operasi *Ubuntu*. Alamat IP yang terkait dengan server ini adalah 103.247.15.33, dan layanan yang diakses berjalan pada *port* 443, yang merupakan *port* standar untuk komunikasi *HTTPS*. Informasi umum yang diperoleh dari hasil pemindaian ini telah dirangkum secara sistematis dalam Tabel 4.4.

**Tabel 4.4** Informasi umum hasil *Nikto Scanner*

| No | Jenis | Keterangan |
|---|---|---|
| 1 | *IP Address* | 103.247.15.33 |
| 2 | *Hostname* | *sicama.unjaya.ac.id* |
| 3 | *Port* | *443* |
| 4 | *Web Server* | *Nginx* |
| 5 | *SSL Info* | *Subject : /CN=*.unjaya.ac.id*<br>*Ciphers : TLS_AES_256_GCM_SHA384*<br>*Issuer : /C=BE/O=GlobalSign nv-sa/CN=AlphaSSL CA - SHA256 - G4* |
| 6 | *Start Time* | 2024-06-02 20:00:07 (GMT+7) |

| No | Jenis | Keterangan |
|---|---|---|
| 7 | *End Time* | 2024-06-02 21:11:30 (GMT+7) |

Selain informasi umum yang telah dikumpulkan, pemindaian menggunakan situs web *scanner* juga berhasil mengidentifikasi beberapa celah kerentanan yang ditemukan pada situs web Sicama Unjaya. Temuan kerentanan ini sangat penting untuk dievaluasi lebih lanjut dengan tujuan meningkatkan keamanan situs secara keseluruhan. Kerentanan-kerentanan ini mencakup berbagai aspek yang berpotensi mengekspos situs web terhadap ancaman keamanan siber yang serius. Rincian mengenai kerentanan yang ditemukan telah dirangkum secara sistematis dalam Tabel 4.5.

**Tabel 4.5** Hasil kerentanan *Nikto Scanner*

| No | Kerentanan | Keterangan |
|---|---|---|
| 1 | *Uncommon Header* | Beberapa header keamanan yang diterapkan dengan baik (*x-content-type-options, x-xss-protection, strict-transport-security, content-security-policy, x-frame-options*). Ini adalah langkah yang baik namun tetap perlu diverifikasi efektivitasnya. |
| 2 | *Server Leaks via ETags* | *ETags header* ditemukan dengan file */f1wGtBPH.show*, yang dapat menyebabkan kebocoran informasi terkait *inode server file system*. Hal ini bisa dimanfaatkan oleh penyerang untuk *fingerprinting server*. |
| 3 | *robots.txt File* | *robots.txt* mengandung *entry* yang harus diperiksa manual, karena file ini mengembalikan *HTTP code 200* yang dapat memberikan informasi tambahan kepada penyerang. |

| No | Kerentanan | Keterangan |
|---|---|---|
| 4 | *Cookie PHPSESSID Tanpa Secure dan HttpOnly Flags* | *Cookies PHPSESSID* dibuat tanpa *secure flag* dan *httponly flag*, yang membuatnya rentan terhadap *sniffing* jika transmisi tidak menggunakan *HTTPS* dan rentan terhadap serangan *XSS*. |
| 5 | *Wildcard Certificate* | Penggunaan sertifikat *wildcard *.unjaya.ac.id*. Meskipun memudahkan pengelolaan subdomain, penggunaannya harus dipastikan aman dan hanya pada subdomain yang tepercaya. |
| 6 | *Exposed Files* | - */WEB-INF/web.xml: JRUN default file* ditemukan.<*br*>- */mail/*: Direktori yang mungkin menarik.<*br*>- */index.html.de*: File default bahasa asing dari *Apache* ditemukan.<*br*> File dan direktori ini dapat memberikan informasi tambahan kepada penyerang. |
| 7 | *Server Leaks Inodes* | *Header* ditemukan dengan file */f1wGtBPH.show* menunjukkan kebocoran inode, yang bisa digunakan oleh penyerang untuk mendapatkan informasi lebih lanjut tentang struktur file *server*. |

Berdasarkan hasil pemindaian menggunakan *Nikto Scanner*, ditemukan beberapa kerentanan signifikan pada situs web SICAMA Universitas Jenderal Achmad Yani Yogyakarta. *Header* keamanan seperti *x-content-type-options* dan *x-xss-protection* telah diterapkan, namun perlu verifikasi lebih lanjut. Ditemukan *ETags header* yang dapat menyebabkan kebocoran informasi *inode* dan file *`robots.txt`* yang mengandung *entry* berisiko. *Cookies PHPSESSID* tanpa *secure*

dan *httponly flag* membuatnya rentan terhadap *sniffing* dan serangan XSS. Penggunaan sertifikat *wildcard *.unjaya.ac.id* dan beberapa file serta direktori terekspos perlu diperhatikan. Dua kerentanan utama adalah ketiadaan *header X-Frame-Options* yang membuat aplikasi rentan terhadap *clickjacking* dan konfigurasi *Content-Security-Policy* (CSP) yang tidak optimal, mengizinkan *`'unsafe-inline'`* yang melemahkan perlindungan terhadap XSS.

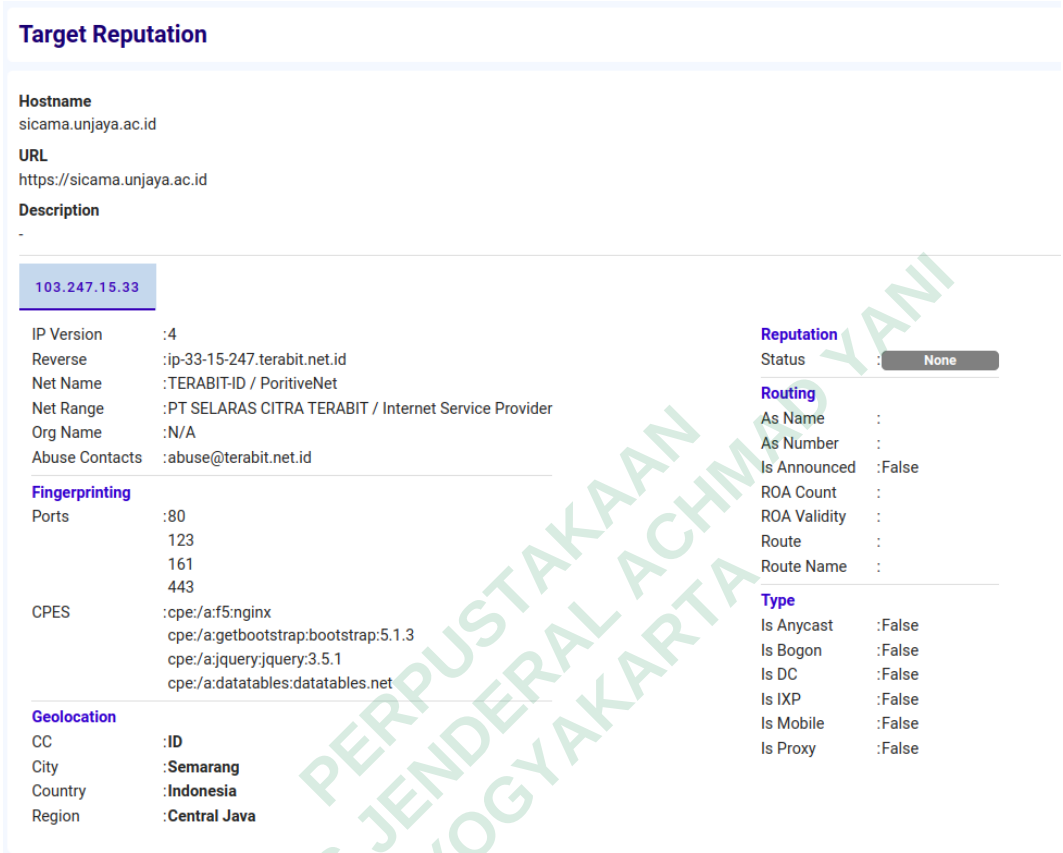
## 4.4 PENGUJIAN (*ATTACKING*)

Tahapan Pengujian (*Attacking*) merupakan langkah kritis dalam evaluasi keamanan jaringan dan aplikasi web, bertujuan untuk mengidentifikasi dan mengeksploitasi kelemahan yang ada. Dalam penelitian ini, digunakan dua alat utama yaitu *Helium Security* dan ZAP (*Zed Attack Proxy*). *Helium Security* berperan dalam pemetaan jaringan dan identifikasi konfigurasi serta topologi jaringan situs web target, dengan fokus pada pemindaian *port* dan layanan yang tersedia. Sementara itu, ZAP digunakan untuk pemindaian kerentanan web, termasuk deteksi potensi serangan seperti *SQL Injection* dan *Cross-Site Scripting* (XSS). Melalui penggunaan kedua alat ini, penelitian bertujuan untuk memberikan gambaran menyeluruh tentang status keamanan situs web, mengidentifikasi kerentanan kritis, dan memberikan rekomendasi mitigasi untuk meningkatkan ketahanan sistem terhadap serangan.

### *4.4.1 Helium Security*

Tahap pengujian keamanan yang akan dilakukan menggunakan alat *Helium Security* bertujuan untuk mengidentifikasi dan menganalisis kerentanan pada aplikasi web target secara komprehensif. *Helium Security* merupakan alat pemindaian keamanan otomatis yang dirancang untuk mendeteksi berbagai jenis ancaman keamanan, *SQL Injection*, *cross-site scripting* (XSS), dan konfigurasi server yang rentan. Proses pengujian ini akan mencakup serangkaian pemindaian yang mendalam untuk mengevaluasi integritas, kerahasiaan, dan ketersediaan sistem, serta untuk memastikan bahwa mekanisme keamanan yang ada mampu melindungi aplikasi dari potensi serangan. Tahap pada fitur *targets Helium Security*

memberikan gambaran mendetail tentang konfigurasi yang dimiliki pada aplikasi web target, sebagaimana ditunjukkan pada Gambar 4.6.




**Gambar 4.7** Hasil pemindaian target *Helium Security*

Gambar 4.7 menunjukkan informasi yang dimiliki oleh situs web Sicama Unjaya. Informasi rinci yang didapatkan mencakup berbagai aspek penting, mulai dari nama *host*, *URL*, versi IP, hingga detail geolokasi dan *fingerprinting* perangkat lunak yang digunakan. Situs ini dihosting oleh PT SELARAS CITRA TERABIT dengan beberapa *port* penting yang terbuka seperti port 80, 123, 161, dan 443, serta menggunakan teknologi seperti *nginx*, *Bootstrap* versi 5.1.3, *jQuery* versi 3.5.1, dan *DataTables*. Data ini esensial untuk memahami konfigurasi dan potensi titik lemah yang dapat dimanfaatkan untuk meningkatkan keamanan situs web. Rincian hasil pemindaian tersebut dijabarkan dalam tabel 4.6.

**Tabel 4.6** Informasi Umum Hasil *Helium Security*

| No | Jenis | Keterangan |
|---|---|---|
| 1 | *Hostname* | *sicama.unjaya.ac.id* |
| 2 | *URL* | *https://sicama.unjaya.ac.id* |
| 3 | *IP Version* | *4* |
| 4 | *Reverse DNS* | *ip-33-15-247.terabit.net.id* |
| 5 | *Net Name* | *TERABIT-ID / PoritiveNet* |
| 6 | *Net Range* | *PT SELARAS CITRA TERABIT / Internet Service Provider* |
| 7 | *Abuse Contact* | *abuse@terabit.net.id* |
| 8 | *Fingerprinting* | *Ports : 80, 123, 161, 443*<br>*CPES :*<br>*cpe:/a:f5:nginx*<br>*cpe:/a:getbootstrap:bootstrap:5.1.3*<br>*cpe:/a:jquery:jquery:3.5.1*<br>*cpe:/a:datatables:datatables.net* |
| 9 | *Geolocation* | *CC : ID*<br>*City : Semarang*<br>*Country : Indonesia*<br>*Region : Central Java* |

Hasil pemindaian dasar menggunakan alat Keamanan *Helium Security* menunjukkan penilaian risiko sistem yang komprehensif. Opsi pemindaian yang digunakan dalam analisis ini adalah pemindaian dasar, yang memberikan pemahaman dasar tentang kerentanan sistem. Kategorisasi peringkat risiko adalah sebagai berikut:

1. Kategori risiko tinggi (*high*) menunjukkan bahwa tidak ada kerentanan kritis yang ditemukan, yang merupakan temuan positif;
2. Kategori risiko sedang (m*edium*) menunjukkan bahwa ada kerentanan sedang, yang membutuhkan perbaikan dan pemeliharaan untuk memastikan keamanan sistem; dan
3. Kategori risiko rendah (*low*) menunjukkan bahwa ada kerentanan kecil, yang dapat diperbaiki dengan pemeliharaan dan pembaruan rutin.
4. Kategori informasi (i*nformational*) memberikan wawasan tambahan tentang konfigurasi sistem dan masalah keamanan yang mungkin, yang bermanfaat untuk analisis dan pengembangan.

Secara keseluruhan, pemindaian dasar *Helium Security* memberikan penilaian menyeluruh terhadap kerentanan sistem dan tingkat risiko, yang memungkinkan pengambilan keputusan yang tepat tentang pengembangan dan pemeliharaan keamanan.

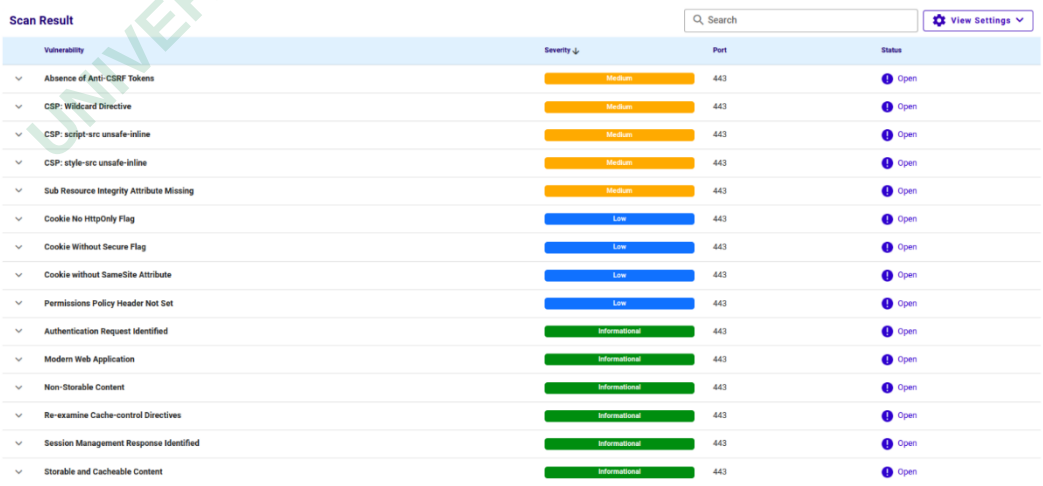

**Gambar 4.8** Hasil *Scan Helium Security*

Berdasarkan Gambar 4.8, ditemukan beberapa informasi dan hasil kerentanan secara umum yang memberikan gambaran mengenai keamanan sistem yang diuji. Hasil pemindaian ini mengidentifikasi berbagai tingkat kerentanan, mulai dari tingkat menengah hingga informatif, yang masing-masing menunjukkan potensi risiko terhadap integritas dan keamanan data. Untuk memberikan gambaran yang lebih jelas tentang jenis kerentanan yang ditemukan dan tingkat kerentanannya, data tersebut akan dirangkum dalam Tabel 4.7.

**Tabel 4.7** Informasi Umum Hasil *Helium Security*

| No | Jenis | Keterangan |
|---|---|---|
| 1 | *Task Name* | *Sicama Unjaya* |
| 2 | *Target* | *https://sicama.unjaya.ac.id* |
| 3 | *Scan Option* | *Basic Scan* |
| 4 | *Risk Rating* | *1. High : 0*<br>*2. Medium : 5*<br>*3. Low : 4*<br>*4. Informational : 6* |

Hasil selanjutnya dalam analisis kerentanan menyajikan gambaran *visual* yang memperinci temuan dari pemindaian. Melalui gambar hasil kerentanan, disajikan data yang mengilustrasikan distribusi risiko dalam konteks tingkat prioritas, serta memberikan insight terperinci terkait sifat dan potensi dampak dari masing-masing kerentanan yang teridentifikasi. Analisis visual ini memberikan landasan yang kokoh untuk pengambilan keputusan, memungkinkan pemangku kepentingan untuk mengalokasikan sumber daya secara efisien dalam mengatasi kerentanan yang paling kritis dan mendesak, sambil tetap memperhatikan konteks yang lebih luas dari sistem yang dievaluasi.




**Gambar 4.9** Hasil kerentanan *Helium Security*

Hasil kerentanan yang didapat dari pemindaian menggunakan *Helium Security* menunjukkan berbagai macam tingkat keparahan pada beberapa aspek keamanan situs web target. Dari hasil pemindaian, ditemukan bahwa situs ini memiliki beberapa kelemahan dengan tingkat keparahan sedang (*medium*), rendah (*low*), dan informatif (*informational*). Setiap kelemahan ini diidentifikasi pada *port* 443 dan statusnya *"Open"*, yang menunjukkan bahwa mereka memerlukan perhatian dan tindakan mitigasi untuk meningkatkan keamanan situs web tersebut.

**Tabel 4.8** Celah kerentanan hasil *Helium Security*

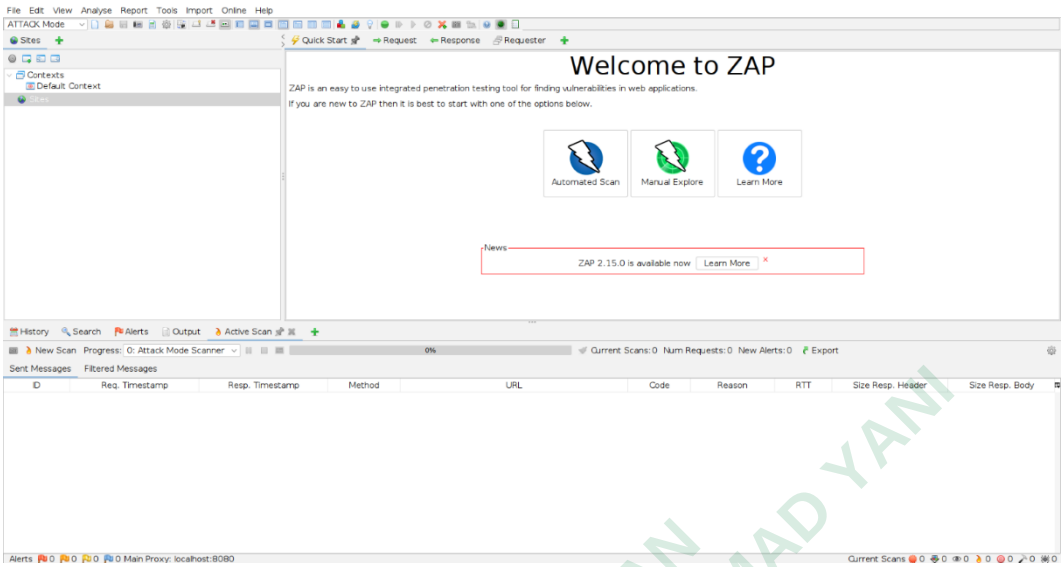
| No | Celah Kerentanan | Tingkat Kerentanan | *Port* | *Status* |
|---|---|---|---|---|
| *1* | *Absence of Anti-CSRF Tokens* | *Medium* | *443* | *Open* |
| *2* | *CSP: Wildcard Directive* | *Medium* | *443* | *Open* |
| *3* | *CSP: script-src unsafe-inline* | *Medium* | *443* | *Open* |
| *4* | *CSP: style-src unsafe-inline* | *Medium* | *443* | *Open* |
| *5* | *Sub Resource Integrity Attribute Missing* | *Medium* | *443* | *Open* |
| *6* | *Cookie No HttpOnly Flag* | *Low* | *443* | *Open* |
| *7* | *Cookie Without Secure Flag* | *Low* | *443* | *Open* |
| *8* | *Cookie without SameSite Attribute* | *Low* | *443* | *Open* |
| *9* | *Permissions Policy Header Not Set* | *Low* | *443* | *Open* |
| *10* | *Authentication Request Identified* | *Informational* | *443* | *Open* |
| *11* | *Modern Web Application* | *Informational* | *443* | *Open* |
| *12* | *Non-Storable Content* | *Informational* | *443* | *Open* |
| *13* | *Re-examine Cache-control Directives* | *Informational* | *443* | *Open* |

| No | Celah Kerentanan | Tingkat Kerentanan | *Port* | *Status* |
|---|---|---|---|---|
| *14* | *Session Management Response Identified* | *Informational* | *443* | *Open* |
| *15* | *Storable and Cacheable Content* | *Informational* | *443* | *Open* |

Hasil kerentanan yang dihasilkan menggunakan alat *Helium Security* memberikan gambaran yang komprehensif tentang potensi kerentanan dalam aplikasi atau sistem yang dianalisis. Berdasarkan hasil yang disajikan, terdapat beberapa kerentanan yang ditemukan dengan tingkat risiko sedang *(medium), rendah (low)* dan informatif (i*nformational).*

### **4.4.2** ZAP (*Zed Attack Proxy*) 2.14.0

Tahap pengujian menggunakan ZAP (*Zed Attack Proxy*) 2.14.0 dimulai dengan konfigurasi awal di mana target aplikasi web yang akan diuji ditentukan. Pertama, pemindaian pasif dilakukan saat aplikasi dijelajahi secara otomatis oleh ZAP, mengumpulkan informasi tanpa mengirimkan permintaan yang dapat mengubah data atau keadaan aplikasi. Pemindaian pasif ini membantu mengidentifikasi masalah dasar seperti *header* keamanan yang hilang dan konfigurasi yang tidak aman. Setelah pemindaian pasif selesai, dilanjutkan dengan pemindaian aktif yang lebih agresif, di mana ZAP mengirimkan berbagai permintaan yang dirancang untuk menguji kerentanan spesifik seperti *SQL Injection, cross-site scripting* (XSS), dan *command injection*. ZAP secara otomatis mencoba mengeksploitasi celah keamanan yang ditemukan untuk mengkonfirmasi keberadaannya. Setelah pemindaian aktif, hasilnya dianalisis untuk mengidentifikasi dan memprioritaskan kerentanan yang ditemukan, memberikan gambaran menyeluruh mengenai status keamanan aplikasi web yang diuji.




**Gambar 4.10** Tampilan awal ZAP *(Zed Attack Proxy)* 2.14.0

Tampilan halaman awal ZAP (*Zed Attack Proxy*) 2.14.0 menyajikan antarmuka yang intuitif dan *user-friendly* yang dirancang untuk memudahkan pengujian keamanan aplikasi web. Pada halaman utama, pengguna dapat melihat berbagai panel yang memberikan akses cepat ke fitur-fitur utama ZAP, termasuk *"Sites"*, *"Alerts"*, dan *"History"*. Fokus utama untuk pengujian kali ini adalah fitur *"Automatic Scan"*, yang dapat diakses melalui menu *"Quick Start"*. Fitur ini memungkinkan peneliti untuk secara otomatis memindai situs web target, dalam hal ini *https://sicama.unjaya.ac.id*, dengan hanya memasukkan URL *target* dan memulai pemindaian. ZAP akan menjalankan serangkaian tes keamanan untuk mendeteksi kerentanan umum tanpa memerlukan interaksi manual yang intensif. Hasil pemindaian akan ditampilkan secara *real-time* dalam panel *"Alerts"*, memberikan ringkasan mengenai kerentanan yang ditemukan beserta tingkat keparahannya, sehingga memudahkan peneliti untuk segera menindaklanjuti dan melakukan mitigasi yang diperlukan.

## Report parameters

### Contexts

No contexts were selected, so all contexts were included by default.

### Sites

The following sites were included:

- https://sicama.unjaya.ac.id

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

### Risk levels

Included: High, Medium, Low, Informational

Excluded: None

### Confidence levels

Included: User Confirmed, High, Medium, Low

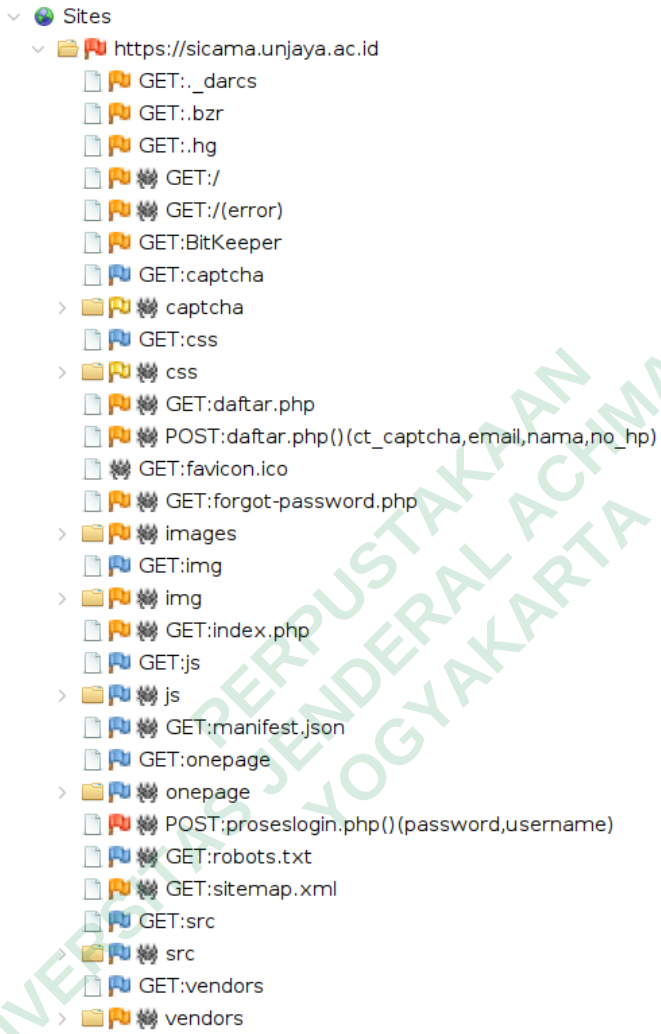
Excluded: User Confirmed, High, Medium, Low, False Positive



**Gambar 4.11** *Parameters* ZAP *(Zed Attack Proxy)* 2.14.0

Berdasarkan gambar 4.11, parameter yang digunakan dalam laporan pemindaian ZAP (*Zed Attack Proxy*) untuk situs Sicama Unjaya mencakup beberapa aspek penting. Pertama, tidak ada konteks spesifik yang dipilih, sehingga semua konteks disertakan secara default. Situs yang diuji adalah *https://sicama.unjaya.ac.id*, yang berarti pemindaian akan fokus sepenuhnya pada situs ini. Tingkat risiko yang disertakan dalam laporan meliputi risiko tinggi, sedang, rendah, dan informasional, memberikan gambaran lengkap tentang berbagai jenis kerentanan dari yang paling kritis hingga yang informatif. Tingkat kepercayaan yang disertakan mencakup kerentanan yang telah dikonfirmasi oleh pengguna serta yang memiliki tingkat kepercayaan tinggi, sedang, dan rendah. Tidak ada tingkat risiko atau kepercayaan yang dikecualikan dari laporan, sehingga

setiap potensi masalah keamanan teridentifikasi dan didokumentasikan dengan lengkap. Dengan parameter-parameter ini, pemindaian ZAP diharapkan dapat menganalisis dan memberikan laporan menyeluruh mengenai kerentanan yang ada.






**Gambar 4.12** *Result Scan* ZAP *(Zed Attack Proxy)* 2.14.0

Dari Gambar 4.12 di atas, dapat dilihat bahwa beberapa *flag* telah diidentifikasi oleh ZAP (*Zed Attack Proxy*). ZAP, sebagai alat pengujian keamanan aplikasi web yang kuat, telah mendeteksi sejumlah indikator kerentanan yang perlu diperhatikan lebih lanjut.

# Summaries

## Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

| | | Confidence | | | | |
|---|---|---|---|---|---|---|
| | | User Confirmed | High | Medium | Low | Total |
| Risk | High | 0 (0.0%) | 0 (0.0%) | 1 (5.9%) | 0 (0.0%) | 1 (5.9%) |
| | Medium | 0 (0.0%) | 3 (17.6%) | 1 (5.9%) | 2 (11.8%) | 6 (35.3%) |
| | Low | 0 (0.0%) | 0 (0.0%) | 3 (17.6%) | 1 (5.9%) | 4 (23.5%) |
| | Informational | 0 (0.0%) | 2 (11.8%) | 2 (11.8%) | 2 (11.8%) | 6 (35.3%) |
| | Total | 0 (0.0%) | 5 (29.4%) | 7 (41.2%) | 5 (29.4%) | 17 (100%) |



**Gambar 4.13** *Summary Scan* ZAP *(Zed Attack Proxy)* 2.14.0

Berdasarkan tabel ringkasan hasil pemindaian keamanan yang ditampilkan dalam Gambar 4.13, ZAP (*Zed Attack Proxy*) telah mengidentifikasi total 17 peringatan yang dikategorikan berdasarkan tingkat risiko dan tingkat kepercayaan. Dari total tersebut, terdapat 1 peringatan dengan risiko tinggi (5.9% dari total), yang memiliki tingkat kepercayaan sedang. Selain itu, 6 peringatan dengan risiko sedang (35.3% dari total) ditemukan, dengan rincian 3 peringatan memiliki tingkat kepercayaan tinggi, 1 peringatan memiliki tingkat kepercayaan sedang, dan 2 peringatan memiliki tingkat kepercayaan rendah. Selanjutnya, 4 peringatan dengan risiko rendah (23.5% dari total) semuanya berada pada tingkat kepercayaan sedang dan rendah. Terakhir, terdapat 6 peringatan informasional (35.3% dari total), dengan 2 peringatan memiliki tingkat kepercayaan tinggi dan 4 peringatan dengan tingkat kepercayaan rendah. Secara keseluruhan, dari 17 peringatan yang terdeteksi,

5 di antaranya berada pada tingkat kepercayaan tinggi (29.4%), 7 pada tingkat kepercayaan sedang (41.2%), dan 5 pada tingkat kepercayaan rendah (29.4%). Tidak ada peringatan yang dikonfirmasi secara manual oleh pengguna. Analisis lebih lanjut terhadap peringatan-peringatan ini penting untuk memahami potensi risiko keamanan yang ada dan menentukan tindakan mitigasi yang tepat guna memperbaiki kelemahan yang teridentifikasi.

**Alert counts by site and risk**

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

| | | Risk | | | |
|---|---|---|---|---|---|
| | | High (= High) | Medium (>= Medium) | Low (>= Low) | Informational (>= Informational) |
| Site | https://sicama.unjaya.ac.id | 1 (1) | 6 (7) | 4 (11) | 6 (17) |



**Gambar 4.14** *Alerts Counts* ZAP *(Zed Attack Proxy)* 2.14.0

Berdasarkan analisis pada gambar 4.13 dan 4.14, teridentifikasi adanya satu kerentanan dengan risiko kategori *High*, enam kerentanan dengan risiko kategori *Medium*, empat kerentanan dengan risiko kategori *Low*, dan enam kerentanan dengan risiko kategori *Informational*. Untuk meningkatkan efektivitas dan pendekatan ilmiah, daftar jenis kerentanan dapat dijabarkan dalam tabel yang mencakup informasi lebih rinci seperti jenis kerentanan, deskripsi singkat, lokasi kerentanan, tingkat keparahan, rekomendasi perbaikan, dan status tindak lanjut. Dengan menyediakan informasi yang lebih terperinci, tabel ini akan memberikan panduan yang lebih komprehensif untuk mengidentifikasi, menilai, dan mengatasi kerentanan keamanan pada situs web secara sistematis dan ilmiah.

| Alert type | Risk | Count |
|---|---|---|
| SQL Injection - SQLite | High | 1 (5.9%) |
| Absence of Anti-CSRF Tokens | Medium | 6 (35.3%) |
| CSP: Wildcard Directive | Medium | 556 (3,270.6%) |
| CSP: script-src unsafe-inline | Medium | 556 (3,270.6%) |
| CSP: style-src unsafe-inline | Medium | 556 (3,270.6%) |
| Hidden File Found | Medium | 4 (23.5%) |
| Vulnerable JS Library | Medium | 1 (5.9%) |
| Cookie No HttpOnly Flag | Low | 1 (5.9%) |
| Cookie Without Secure Flag | Low | 1 (5.9%) |
| Cookie without SameSite Attribute | Low | 1 (5.9%) |
| Timestamp Disclosure - Unix | Low | 2 (11.8%) |
| Authentication Request Identified | Informational | 1 (5.9%) |
| Information Disclosure - Suspicious Comments | Informational | 13 (76.5%) |
| Modern Web Application | Informational | 550 (3,235.3%) |
| Re-examine Cache-control Directives | Informational | 36 (211.8%) |
| Session Management Response Identified | Informational | 2 (11.8%) |
| User Agent Fuzzer | Informational | 264 (1,552.9%) |
| Total | | 17 |



**Gambar 4.15** Daftar hasil kerantanan ZAP *(Zed Attack Proxy)* 2.14.0

Hasil pemindaian kerentanan menggunakan ZAP (*Zed Attack Proxy*) versi 2.14.0 menunjukkan berbagai kerentanan yang ditemukan pada situs web target. Tabel berikut merangkum jenis-jenis kerentanan yang terdeteksi, tingkat risiko yang terkait, serta jumlah insiden yang ditemukan untuk masing-masing jenis kerentanan. Kerentanan dengan tingkat risiko tinggi, sedang, dan rendah serta informasi tambahan tentang praktik keamanan yang dapat diperbaiki diuraikan secara rinci dalam tabel ini. Dari hasil pemindaian, ditemukan 17 insiden dengan rincian sebagai berikut: 1 insiden dengan risiko tinggi (*SQL Injection - SQLite*), 6 insiden dengan risiko sedang (termasuk *Absence of Anti-CSRF Tokens*, *Hidden File Found*, dan *Vulnerable library JS*), dan 4 insiden dengan risiko rendah (terkait

pengaturan *cookie* dan pengungkapan *timestamp*). Selain itu, terdapat sejumlah besar informasi tambahan yang dicatat, termasuk pengungkapan komentar mencurigakan, penggunaan aplikasi web modern, dan kebutuhan untuk memeriksa kembali direktif *cache control*. Tabel berikut memberikan rincian lengkap dari hasil pemindaian ini:

**Tabel 4.9** Hasil Kerentanan ZAP (*Zed Attack Proxy*)

| **Jenis Kerentanan** | **Tingkat Kerentanan** | **Count** |
|---|---|---|
| *SQL Injection - SQLite* | *High* | 1 (5.9%) |
| *Absence of Anti-CSRF Tokens* | *Medium* | *6 (35.3%)* |
| *CSP: Wildcard Directive* | *Medium* | *556 (3,278.6%)* |
| *CSP: script-src unsafe-inline* | *Medium* | *556 (3,278.6%)* |
| *CSP: style-src unsafe-inline* | *Medium* | *556 (3,278.6%)* |
| *Hidden File Found* | *Medium* | *4 (23.5%)* |
| *Vulnerable JS Library* | *Medium* | *1 (5.9%)* |
| *Cookie No HttpOnly Flag* | *Low* | *1 (5.9%)* |
| *Cookie Without Secure Flag* | *Low* | *1 (5.9%)* |
| *Cookie without SameSite Attribute* | *Low* | *A 1 (5.9%)* |
| *Timestamp Disclosure - Unix* | *Low* | *2 (11.8%)* |
| *Authentication Request Identified* | *Informational* | *1 (5.9%)* |
| *Information Disclosure - Suspicious Comments* | *Informational* | *13 (76.5%)* |
| *Modern Web Application* | *Informational* | *550 (3,235.3%)* |
| *Re-examine Cache-control Directives* | *Informational* | *36 (211.8%)* |

| Jenis Kerentanan | Tingkat Kerentanan | Count |
|---|---|---|
| *Session Management Response Identified* | *Informational* | *2 (11.8%)* |
| *User Agent Fuzzer* | *Informational* | *264 (1,552.9%)* |
| **Total** | | **17** |

Dari hasil tersebut, dapat dilihat bahwa meskipun ada beberapa kerentanan dengan tingkat risiko tinggi dan sedang, sebagian besar peringatan adalah informasional yang menekankan pada perbaikan praktik keamanan untuk meningkatkan ketahanan sistem terhadap potensi serangan. Pemetaan dan analisis lebih lanjut diperlukan untuk mengatasi kerentanan yang ditemukan dan memperkuat keamanan sistem secara keseluruhan.

## 4.5 PEMBAHASAN

Pada tahapan pelaporan (*reporting*) dalam metode NIST SP 800-115, peneliti akan mengumpulkan dan mendokumentasikan semua temuan yang diperoleh selama proses pengujian. Laporan ini akan mencakup identifikasi kerentanan, analisis dampak, dan rekomendasi mitigasi untuk meningkatkan keamanan sistem. Setiap temuan akan dikategorikan dan dijelaskan dengan detail, serta disusun menggunakan acuan OWASP Top 10 untuk memastikan relevansi dan kepatuhan terhadap standar keamanan yang diakui secara global. Hal ini membantu dalam memberikan gambaran yang komprehensif dan terstruktur mengenai kondisi keamanan sistem yang diuji, serta langkah-langkah yang perlu diambil untuk mengatasi kerentanan yang ditemukan.

### *4.5.1 Nikto Scanner*

Pembahasan hasil dari *Nikto Scanner* berfokus pada analisis mendetail terhadap kerentanan yang ditemukan selama proses pemindaian, dengan penekanan khusus pada relevansi kerentanan tersebut terhadap kategori OWASP Top 10. *Nikto*, sebagai alat pemindai *server* web yang andal, mengidentifikasi berbagai jenis

kerentanan seperti kelemahan konfigurasi *server*, informasi sensitif yang terungkap, dan potensi *exploit*. Berikut hasil kerentanan yang didapat dengan menggunakan *Nikto Scanner* :

1. *Uncommon Header*

Beberapa *header* keamanan yang diterapkan dengan baik (*x-content-type-options, x-xss-protection, strict-transport-security, content-security-policy, x-frame-options*). Ini adalah langkah yang baik namun tetap perlu diverifikasi efektivitasnya.

2. *Server Leaks via ETags*

*ETags header* ditemukan dengan file */f1wGtBPH.show*, yang dapat menyebabkan kebocoran informasi terkait *inode server file system*. *Inode*, yang merupakan struktur data yang menyimpan informasi tentang sebuah file atau direktori dalam sistem berkas UNIX, dapat memberikan petunjuk penting tentang konfigurasi dan struktur *server*. Dalam konteks ini, *ETags header*, yang menyediakan tag identifikasi unik untuk versi spesifik dari file, dapat memungkinkan penyerang untuk melakukan *fingerprinting server* dengan mengumpulkan informasi *inode*.

*3. File robots.txt*

*File robots.txt* mengandung *entry* yang harus diperiksa manual, karena file ini mengembalikan *HTTP code 200* yang dapat memberikan informasi tambahan kepada penyerang.

4. *Cookie PHPSESSID Tanpa Secure dan HttpOnly Flags*

*Cookies PHPSESSID* dibuat tanpa *secure flag* dan *httponly flag*, yang membuatnya rentan terhadap *sniffing* jika transmisi tidak menggunakan HTTPS dan rentan terhadap serangan XSS.

5. *Wildcard Certificate*

Penggunaan sertifikat *wildcard *.unjaya.ac.id*. Meskipun memudahkan pengelolaan subdomain, penggunaannya harus dipastikan aman dan hanya pada subdomain yang tepercaya.

6. *Exposed Files*

*- /WEB-INF/web.xml: JRUN default file* ditemukan. <*br*>*- /mail/*: Direktori yang mungkin menarik. <*br*>*- /index.html.de*: File *default* bahasa asing dari *Apache* ditemukan. <*br*> File dan direktori ini dapat memberikan informasi tambahan kepada penyerang.

*7. Server Leaks Inodes*

*Header* ditemukan dengan file */f1wGtBPH.show* menunjukkan kebocoran *inode*, yang bisa digunakan oleh penyerang untuk mendapatkan informasi lebih lanjut tentang struktur file server.

**Tabel 4.10** Kerentanan *Nikto Scanner* dan acuan OWASP TOP 10

| No | Kerentanan | Tingkat Kerentanan | *Location* | Kategori OWASP TOP 10 |
|---|---|---|---|---|
| 1 | *Uncommon Header* | *Medium* | *Server Header* | *A05:2021 – Security Misconfiguration* |
| 2 | *Server Leaks via ETags* | *Medium* | *HTTP Response Header* | *A05:2021 – Security Misconfiguration* |
| 3 | *robots.txt File* | *Low* | *Web Root Directory* | *A05:2021 – Security Misconfiguration* |
| 4 | *Cookie PHPSESSID Tanpa Secure dan HttpOnly Flags* | *Medium* | *HTTP Cookies* | *A02:2021 – Cryptographic Failures* |
| 5 | *Wildcard Certificate* | *Medium* | *SSL/TLS Configuration* | *A05:2021 – Security Misconfiguration* |
| 6 | *Exposed Files* | *Medium* | *Web Directory* | *A05:2021 – Security Misconfiguration* |

| No | Kerentanan | Tingkat Kerentanan | *Location* | Kategori OWASP TOP 10 |
|---|---|---|---|---|
| 7 | *Server Leaks Inodes* | *Low* | *HTTP Response Header* | *A05:2021 – Security Misconfiguration* |

### *4.5.2 Helium Security*

Pemindaian keamanan menggunakan alat *Helium Security* terhadap situs Sicama Unjaya bertujuan untuk mengidentifikasi dan menganalisis kerentanan pada aplikasi web secara komprehensif. Hasil pemindaian ini sangat penting dalam memberikan gambaran menyeluruh tentang tingkat keamanan situs dan potensi risiko yang mungkin dihadapi. Dalam analisis ini, *Helium Security* melakukan pemindaian dasar yang mencakup berbagai aspek penting seperti *hostname*, URL, versi IP, detail geolokasi, dan *fingerprinting* perangkat lunak yang digunakan. Data ini memberikan wawasan mendalam mengenai konfigurasi sistem, yang mencakup teknologi seperti *Nginx*, *Bootstrap* 5.1.3, *jQuery* 3.5.1, dan *DataTables*, serta beberapa *port* yang terbuka seperti *port* 80, 123, 161, dan 443. Hasil pemindaian *Helium Security* menunjukkan bahwa tidak ada kerentanan dengan kategori risiko tinggi yang ditemukan, yang merupakan indikasi positif bahwa tidak ada ancaman kritis terhadap sistem. Namun, ditemukan lima kerentanan dengan risiko sedang, empat kerentanan dengan risiko rendah, dan enam kerentanan informatif. Kerentanan dengan risiko sedang membutuhkan perhatian segera untuk memastikan bahwa tidak ada celah keamanan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab. Sedangkan kerentanan dengan risiko rendah dan informatif dapat ditangani dengan pemeliharaan dan pembaruan rutin untuk memperkuat keamanan sistem. Berikut kategori kerentanan yang ditemukan dengan menampilkan kategori Medium :

1. *Absence of Anti-CSRF Tokens*

*No Anti-CSRF tokens* yang ditemukan dalam formulir pengiriman HTML, sehingga aplikasi terkena serangan pemalsuan permintaan lintas situs (CSRF).

CSRF mengeksploitasi kepercayaan yang dimiliki situs web terhadap *browser* pengguna dengan memaksa pengguna melakukan tindakan yang tidak diinginkan, seperti mengirimkan formulir atau melakukan transaksi, tanpa sepengetahuan mereka. Hal ini dicapai melalui tindakan URL/bentuk yang dapat diprediksi dan dapat diulang. Serangan CSRF efektif ketika korban memiliki sesi aktif, diautentikasi melalui HTTP, atau berbagi jaringan lokal yang sama dengan situs target. Meskipun secara tradisional digunakan untuk melakukan tindakan menggunakan hak istimewa korban, teknik terbaru menunjukkan bahwa CSRF juga dapat mengungkapkan informasi sensitif, terutama bila dikombinasikan dengan *Cross Site Scripting*(XSS), yang dapat mengeksploitasi kebijakan asal yang sama untuk mengintensifkan serangan. Oleh karena itu, penerapan *token Anti-CSRF* sangat penting untuk mencegah kerentanan tersebut dan memastikan integritas dan keamanan aplikasi web. Berikut hasil kerentanan yang didapat :

**Tabel 4.11** *Absence of Anti-CSRF Tokens*

| No | Celah Kerentanan | HTTP Method | *Uniform Resource Locator (URL)* |
|---|---|---|---|
| 1 | *<form action="proseslogin.php" method="post">* | *GET* | *https://sicama.unjaya.ac.id /* |
| 2 | *<form action="/daftar.php" method="post" class="tm-contact-form" style="min-height:300px;width:100%;">* | *POST* | *https://sicama.unjaya.ac.id /daftar.php* |
| 3 | *<form action="proseslogin.php" method="post">* | *POST* | *https://sicama.unjaya.ac.id /index.php* |

2. *CSP: Wildcard Directive, CSP: script-src unsafe-inline, CSP: style-src unsafe-inline*

*Content Security Policy* (CSP) merupakan lapisan keamanan tambahan yang membantu mendeteksi dan memitigasi jenis serangan tertentu. Termasuk

*Cross Site Scripting* (XSS), dan *Data Injection*. Serangan-serangan ini digunakan untuk segala hal mulai dari pencurian data hingga merusak situs atau distribusi *malware*. CSP menyediakan serangkaian *header* HTTP standar yang memungkinkan pemilik situs web menyatakan sumber konten yang dapat dipercaya dan diizinkan untuk dimuat oleh *browser* di halaman tersebut, tipe yang tercakup adalah *JavaScript*, CSS, HTML, *font*, gambar, dan objek yang dapat disematkan seperti *applet Java*, *ActiveX*, file audio dan video. Hasil bukti yang didapat dengan menggunakan *tools Helium Security* menunjukkan bahwa CSP yang diterapkan pada situs tersebut tidak memadai karena menggunakan *wildcard directive 'unsafe-inline'*. Bukti-bukti yang ditemukan pada tabel XX antara lain sebagai berikut :

**Tabel 4.12** *CSP Wildcard Directive Vulnerabilities*

| No | Celah Kerentanan | ***HTTP Method*** | ***Params*** | ***Uniform Resource Locator (URL)*** |
|---|---|---|---|---|
| 1 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *GET* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/* |
| 2 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *POST* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/daftar.php* |
| 3 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *GET* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/daftar.php* |
| 4 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *GET* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/index.php* |
| 5 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *GET* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/forgot-password.php* |

| No | Celah Kerentanan | *HTTP Method* | *Params* | *Uniform Resource Locator (URL)* |
|---|---|---|---|---|
| 6 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *GET* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/sitemap.xml* |
| 7 | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *POST* | *Content-Security-Policy* | *https://sicama.unjaya.ac.id/proseslogin.php* |

*3. Sub Resource Integrity Attribute Missing*

*Sub Resource Integrity* (SRI) merupakan fitur keamanan yang memungkinkan *browser* memverifikasi bahwa sumber daya yang mereka ambil, seperti *script* atau *stylesheet*, dikirimkan tanpa manipulasi yang tidak terduga. Atribut integritas pada tag <*script*> atau <*link*> membantu memastikan bahwa situs web hanya mengeksekusi konten yang dipercaya. Jika atribut integritas tidak ada, penyerang yang mendapatkan akses ke *server* eksternal dapat menyuntikkan konten berbahaya ke sumber daya ini, sehingga membahayakan keamanan aplikasi. Kerentanan ini khususnya mengkhawatirkan untuk sumber daya yang diambil dari *server* eksternal, karena dapat menyebabkan pelanggaran keamanan yang signifikan jika tidak dikelola dengan baik.

**Tabel 4.13** *Sub Resource Integrity Attribute Missing Vulnerabilities*

| No | Celah Kerentanan | *HTTP Method* | *Uniform Resource Locator (URL)* |
|---|---|---|---|
| 1 | *<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet">* | *GET* | *https://sicama.unjaya.ac.id/* |

| No | Celah Kerentanan | *HTTP Method* | *Uniform Resource Locator (URL)* |
|---|---|---|---|
| 2 | *<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet">* | *GET* | *https://sicama.unjaya.ac.id/?error=0* |
| 3 | *<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet">* | *POST* | *https://sicama.unjaya.ac.id/daftar.php* |
| 4 | *<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:300,400">* | *GET* | *https://sicama.unjaya.ac.id/forgot-password.php* |
| 5 | *<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet">* | *GET* | *https://sicama.unjaya.ac.id/index.php* |



Hasil pemindaian menggunakan *Helium Security* mengidentifikasi berbagai kerentanan pada situs web target, yang dikategorikan sesuai dengan OWASP Top 10. Analisis ini memberikan wawasan mendalam mengenai kelemahan keamanan yang perlu diperbaiki untuk meningkatkan integritas dan ketahanan sistem. Berikut hasil kerentanan yang ditemukan menggunakan *tools Helium Security* yang ditampilkan pada tabel berikut :

**Tabel 4.14** Kerentanan *Helium Security* dan acuan OWASP TOP 10

| No | Celah Kerentanan | Tingkat Kerentanan | *Location* | *Kategori OWASP TOP 10* |
|---|---|---|---|---|
| 1 | *Absence of Anti-CSRF Tokens* | *Medium* | *Login and Registration Forms* | *A01:2021 – Broken Access Control* |
| 2 | *CSP: Wildcard Directive* | *Medium* | *Main Page Configuration* | *A05:2021 – Security Misconfiguration* |
| 3 | *CSP: script-src unsafe-inline* | *Medium* | *Main Script Configuration* | *A05:2021 – Security Misconfiguration* |
| 4 | *CSP: style-src unsafe-inline* | *Medium* | *CSS Style Configuration* | *A05:2021 – Security Misconfiguration* |
| 5 | *Sub Resource Integrity Attribute Missing* | *Medium* | *External Scripts* | *A05:2021 – Security Misconfiguration* |
| 6 | *Cookie No HttpOnly Flag* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |
| 7 | *Cookie Without Secure Flag* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |
| 8 | *Cookie without SameSite Attribute* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |

| No | Celah Kerentanan | Tingkat Kerentanan | *Location* | *Kategori OWASP TOP 10* |
|---|---|---|---|---|
| 9 | *Permissions Policy Header Not Set* | *Low* | *Header Configuration* | *A05:2021 – Security Misconfiguration* |
| 10 | *Authentication Request Identified* | *Informational* | *Authentication Requests* | *A01:2021 – Broken Access Control* |
| 11 | *Modern Web Application* | *Informational* | *Modern Web Application* | *A01:2021 – Broken Access Control* |
| 12 | *Non-Storable Content* | *Informational* | *API and Dynamic Content* | *A05:2021 – Security Misconfiguration* |
| 13 | *Re-examine Cache-control Directives* | *Informational* | *Cache Configuration* | *A05:2021 – Security Misconfiguration* |
| 14 | *Session Management Response Identified* | *Informational* | *Session Management* | *A05:2021 – Security Misconfiguration* |
| 15 | *Storable and Cacheable Content* | *Informational* | *Cache Configuration* | *A05:2021 – Security Misconfiguration* |



### 4.5.3 ZAP (*Zed Attack Proxy*) 2.14.0

Hasil pengujian kerentanan menggunakan ZAP (*Zed Attack Proxy*) 2.14.0 menemukan beberapa jenis kerentanan dalam pengujian menggunakan ZAP (*Zed Attack Proxy*) versi 2.14.0. Fokus pembahasan adalah pada kerentanan dengan

kategori risiko tinggi (*high*) dan sedang (*medium*), yang paling mempengaruhi keamanan sistem secara keseluruhan. Berikut hasil kerentanan pada kategori *high* dan *medium* :

1. *SQL Injection - SQLite*

*SQL Injection* ditemukan pada *endpoint /proseslogin.php* di situs *https://sicama.unjaya.ac.id*. Serangan ini memungkinkan penyerang untuk mengendalikan waktu eksekusi *query* dengan memanipulasi parameter *password*, yang menunjukkan bahwa *query SQL* mungkin rentan terhadap manipulasi. ZAP mengidentifikasi bahwa dengan menggunakan nilai parameter seperti *case randomblob(10000000) when not null then 1 else 1 end*, waktu eksekusi *query* meningkat menjadi 416 milidetik, dan dengan nilai *case randomblob(100000000) when not null then 1 else 1 end*, waktu eksekusi meningkat lebih jauh menjadi 688 milidetik. Sebagai perbandingan, nilai asli ZAP hanya memerlukan 169 milidetik. Perbedaan waktu ini menunjukkan bahwa *query SQL* rentan terhadap injeksi, memungkinkan penyerang untuk menyisipkan logika tambahan dalam parameter input untuk mengontrol perilaku *query*.

HTTP *Request* :

*POST https://sicama.unjaya.ac.id/proseslogin.php HTTP/1.1*
*Host: sicama.unjaya.ac.id*
*User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36*
*Pragma: no-cache*
*Cache-Control: no-cache*
*Content-Type: application/x-www-form-urlencoded*
*Referer: https://sicama.unjaya.ac.id/*
*Content-Length: 84*
*Cookie: PHPSESSID=77cfnqihijsoq11ij9kl3qams7*
*username=ZAP&password=case+randomblob%2810000000%29+when+not+null+then+1+else+1+end+*

**Tabel 4.15** *SQL Injection*

| No | ***Parameters*** | ***Value Attack*** | **Hasil Pengujian** |
|---|---|---|---|
| 1 | *password* | *case randomblob(10000000) when not null then 1 else 1 end* | *The query time is controllable using parameter value, causing the request to take 416 milliseconds.* |
| 2 | *password* | *case randomblob(100000000) when not null then 1 else 1 end* | *The query time is controllable using parameter value, causing the request to take 688 milliseconds.* |
| 3 | *password* | *ZAP* | *The original unmodified query with value [ZAP] took 169 milliseconds.* |

2. *CSP: Wildcard Directive, CSP: script-src unsafe-inline, dan CSP: style-src unsafe-inline*

Serangan-serangan ini digunakan untuk berbagai tujuan, mulai dari mencuri data hingga merusak situs web atau menyebarkan malware. CSP menyediakan serangkaian header HTTP standar yang memungkinkan pemilik situs web mendeklarasikan sumber konten yang disetujui dan boleh dimuat oleh browser di halaman tersebut. Dalam kasus ini, ditemukan bahwa CSP yang diterapkan pada situs *https://sicama.unjaya.ac.id* menggunakan *wildcard directive* yang terlalu luas atau tidak didefinisikan dengan baik, yang dapat membuka celah bagi serangan XSS dan injeksi data. Direktif yang menggunakan *wildcard* atau tidak didefinisikan dengan benar mencakup *script-src, style-src, img-src, connect-src, frame-src, frame-ancestors, font-src, media-src, object-src, manifest-src, worker-src, dan form-action*. Penggunaan *wildcard, script-src* dan *style-src* ini dapat mengizinkan

pemuatan konten dari sumber yang tidak terverifikasi, sehingga meningkatkan risiko serangan.

HTTP *Request* :

*GET https://sicama.unjaya.ac.id HTTP/1.1*

*Host: sicama.unjaya.ac.id*

*User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36*

*Pragma: no-cache*

*Cache-Control: no-cache*

**Tabel 4.16** CSP *(Content Security Policy)*

| No | ***Parameters*** | ***Value Attack*** | ***URL*** |
|---|---|---|---|
| 1 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/* |
| 2 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/?error=0* |
| 3 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/daftar.php* |
| 4 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/forgot-password.php* |
| 5 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/index.php* |

| No | *Parameters* | *Value Attack* | *URL* |
|---|---|---|---|
| 6 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/sitemap.xml* |
| 7 | *Content-Security-Policy* | *default-src 'self' http: https: data: blob: 'unsafe-inline'* | *https://sicama.unjaya.ac.id/proseslogin.php* |

*3. Vulnerable JS Library*

Penggunaan *library JavaScript* yang rentan merupakan masalah keamanan yang signifikan dalam pengembangan aplikasi web. Dalam kasus ini, ditemukan bahwa *library moment.js* versi 2.21.0 yang digunakan oleh situs *https://sicama.unjaya.ac.id* dengan *endpoint /vendors/scripts/script.min.js* rentan terhadap beberapa kerentanan keamanan. Kerentanan ini diidentifikasi dengan referensi ke dua *Common Vulnerabilities and Exposures* (CVE) yaitu CVE-2022-31129 dan CVE-2022-24785. Kedua kerentanan ini dapat dieksploitasi oleh penyerang untuk menjalankan serangan berbahaya seperti pencurian data, manipulasi waktu, dan injeksi *script*. Versi rentan dari *library moment.js* memungkinkan penyerang untuk mengeksploitasi kelemahan dalam fungsi *library*, yang dapat menyebabkan perilaku aplikasi yang tidak terduga dan berpotensi berbahaya. Penggunaan *library* yang tidak diperbarui atau rentan menempatkan aplikasi dalam risiko tinggi karena memberikan pintu masuk bagi penyerang untuk melakukan serangan.

**Tabel 4.17** *Vulnerable JS Library*

| No | *Parameters* | Kerentanan | *URL* |
|---|---|---|---|
| 1 | *Library* | *moment.js version 2.21.0* | *https://sicama.unjaya.ac.id/vendors/scripts/script.min.js* |

| No | *Parameters* | Kerentanan | *URL* |
|---|---|---|---|
| 2 | *CVE* | *CVE-2022-31129* | *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-31129* |
| 3 | *CVE* | *CVE-2022-24785* | *https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-24785* |

*4. Absence of Anti-CSRF Tokens*

Kerentanan ini ditemukan pada situs *https://sicama.unjaya.ac.id*, di mana tidak ada *token* Anti-CSRF yang ditemukan dalam formulir HTML pada *endpoint /proseslogin.php*. CSRF mengeksploitasi kepercayaan yang dimiliki situs web terhadap pengguna. Dengan tidak adanya *token* Anti-CSRF, aplikasi rentan terhadap serangan di mana penyerang dapat memaksa pengguna yang terautentikasi untuk melakukan tindakan seperti perubahan kata sandi, transfer dana, atau tindakan berbahaya lainnya tanpa sepengetahuan mereka. Kerentanan ini sangat berisiko jika korban memiliki sesi aktif di situs target atau jika mereka diautentikasi melalui *HTTP auth*. Risiko ini meningkat secara dramatis jika situs target juga rentan terhadap XSS, karena XSS dapat digunakan sebagai platform untuk serangan CSRF, memungkinkan serangan tersebut beroperasi dalam batasan kebijakan asal yang sama (*same-origin policy*). Untuk mengatasi kerentanan ini, sangat penting untuk menerapkan *token* Anti-CSRF pada semua formulir yang mengirimkan data penting, serta memverifikasi *token* tersebut pada *server* untuk setiap permintaan.

**Tabel 4.18** *Absence of Anti-CSRF Tokens* - ZAP

| No | *Parameters* | Kerentanan | *URL* |
|---|---|---|---|
| 1 | *Form Action* | *<form action="proseslogin.php" method="post">* | *https://sicama.unjaya.ac.id/proseslogin.php* |

| No | *Parameters* | Kerentanan | *URL* |
|---|---|---|---|
| 2 | *Form Fields* | *"password" "username"* | *https://sicama.unjaya.ac.id/* |

5. *Hidden File Found*

Penemuan file tersembunyi yang dapat diakses adalah masalah keamanan yang signifikan dalam pengembangan dan pengelolaan aplikasi web. Dalam kasus ini, ditemukan bahwa file tersembunyi pada *endppoint /.hg* dapat diakses melalui situs *https://sicama.unjaya.ac.id*. File .hg biasanya terkait dengan sistem kontrol versi *Mercurial* dan dapat berisi informasi sensitif seperti konfigurasi administrasi, informasi kredensial, atau riwayat perubahan kode yang dapat dimanfaatkan oleh penyerang untuk menyerang sistem lebih lanjut atau melakukan upaya rekayasa sosial. Mengakses file tersembunyi seperti .hg dapat memberikan penyerang wawasan tentang struktur internal aplikasi, informasi konfigurasi, dan bahkan data sensitif yang dapat digunakan untuk mengeksploitasi kerentanan lebih lanjut dalam sistem. Oleh karena itu, penting untuk memastikan bahwa file-file tersembunyi ini tidak dapat diakses secara publik dan harus dilindungi atau dihapus dari direktori publik.

**Tabel 4.19** *Hidden File Found*

| No | *Parameters* | *Request Headers* | *Response Headers* | *URL* |
|---|---|---|---|---|
| 1 | *File Path* | *GET https://sicama.unjaya.ac.id/.hg HTTP/1.1* | *HTTP/1.1 200 OK* | *https://sicama.unjaya.ac.id/.hg* |

Pengujian keamanan menggunakan ZAP (Zed Attack Proxy) 2.14.0 terhadap situs *https://sicama.unjaya.ac.id* mengungkapkan sejumlah kerentanan dengan berbagai tingkat risiko. Hasil pengujian ini diuraikan berdasarkan jenis kerentanan, tingkat keparahan, lokasi, dan kategori OWASP Top 10, memberikan

panduan yang komprehensif untuk mengidentifikasi dan memperbaiki kelemahan keamanan yang ada.

**Tabel 4.20** Hasil pemindaian ZAP dengan Kategori OWASP TOP 10

| No | Jenis Kerentanan | Tingkat Kerentanan | Location | Count |
|---|---|---|---|---|
| 1 | *SQL Injection - SQLite* | *High* | *https://sicama.unjaya.ac.id/proseslogin.php* | *A03:2021 – Injection* |
| 2 | *Absence of Anti-CSRF Tokens* | *Medium* | *Login and Registration Forms* | *A01:2021 – Broken Access Control* |
| 3 | *CSP: Wildcard Directive* | *Medium* | *Main Page Configuration* | *A05:2021 – Security Misconfiguration* |
| 4 | *CSP: script-src unsafe-inline* | *Medium* | *Main Script Configuration* | *A05:2021 – Security Misconfiguration* |
| 5 | *CSP: style-src unsafe-inline* | *Medium* | *CSS Style Configuration* | *A05:2021 – Security Misconfiguration* |
| 6 | *Hidden File Found* | *Medium* | *https://sicama.unjaya.ac.id/.hg* | *A05:2021 – Security Misconfiguration* |
| 7 | *Vulnerable JS Library* | *Medium* | *https://sicama.unjaya.ac.id/vendors/scripts/script.min.js* | *A06:2021 – Vulnerable and Outdated Components* |

| No | Jenis Kerentanan | Tingkat Kerentanan | Location | Count |
|---|---|---|---|---|
| 8 | *Cookie No HttpOnly Flag* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |
| 9 | *Cookie Without Secure Flag* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |
| 10 | *Cookie without SameSite Attribute* | *Low* | *Cookie Settings* | *A05:2021 – Security Misconfiguration* |
| 11 | *Timestamp Disclosure - Unix* | *Low* | *Multiple Locations* | *A05:2021 – Security Misconfiguration* |
| 12 | *Authentication Request Identified* | *Informational* | *Authentication Requests* | *A01:2021 – Broken Access Control* |
| 13 | *Information Disclosure - Suspicious Comments* | *Informational* | *Multiple Locations* | *A05:2021 – Security Misconfiguration* |
| 14 | *Modern Web Application* | *Informational* | *Overall Application* | *A05:2021 – Security Misconfiguration* |
| 15 | *Re-examine Cache-control Directives* | *Informational* | *Cache Configuration* | *A05:2021 – Security Misconfiguration* |

| No | Jenis Kerentanan | Tingkat Kerentanan | Location | Count |
|---|---|---|---|---|
| 16 | *Session Management Response Identified* | *Informational* | *Session Management* | *A05:2021 – Security Misconfiguration* |
| 17 | *User Agent Fuzzer* | *Informational* | *Multiple Locations* | *A05:2021 – Security Misconfiguration* |

### 4.5.4 *Report* OWASP TOP 10

Hasil pemindaian yang dilakukan dengan menggunakan tiga tools pengujian kerentanan menunjukkan bahwa situs *https://sicama.unjaya.ac.id* memiliki beberapa kelemahan dalam konfigurasi keamanan dan komponen yang digunakan. Kerentanan ditemukan di berbagai kategori OWASP Top 10, dengan beberapa di antaranya memiliki tingkat risiko tinggi dan sedang yang memerlukan perhatian segera. Implementasi langkah-langkah mitigasi seperti penerapan *token Anti-CSRF*, pembaruan pustaka yang rentan, dan pengaturan kebijakan keamanan yang lebih ketat sangat diperlukan untuk melindungi aplikasi dari potensi serangan. Berikut hasilnya.

**Tabel 4.21** Daftar Kerentanan berdasarkan Kategori OWASP TOP 10

| No | Kategori OWASP TOP 10 | Celah Kerentanan | Tools | Tingkat Kerentanan | *Location* |
|---|---|---|---|---|---|
| 1 | *A03:2021 – Injection* | *SQL Injection - SQLite* | *ZAP* | *High* | */proseslogin.php* |
| 2 | *A01:2021 – Broken Access Control* | *Absence of Anti-CSRF Tokens* | *Helium Security, ZAP* | *Medium* | *Login and Registration Forms* |
| 3 | *A01:2021 – Broken Access Control* | *Authentication Request Identified* | *Helium Security, ZAP* | *Informational* | *Authentication Requests* |
| 4 | *A01:2021 – Broken Access Control* | *Modern Web Application* | *Helium Security, ZAP* | *Informational* | *Modern Web Application* |
| 5 | *A02:2021 – Cryptographic Failures* | *Cookie PHPSESSID Tanpa Secure dan HttpOnly Flags* | *Nikto Scanner* | *Medium* | *HTTP Cookies* |
| 6 | *A05:2021 – Security Misconfiguration* | *Uncommon Header* | *Nikto Scanner* | *Medium* | *Server Header* |

| No | Kategori OWASP TOP 10 | Celah Kerentanan | Tools | Tingkat Kerentanan | *Location* |
|---|---|---|---|---|---|
| 7 | *A05:2021 – Security Misconfiguration* | *Server Leaks via ETags* | *Nikto Scanner* | *Medium* | *HTTP Response Header* |
| 8 | *A05:2021 – Security Misconfiguration* | *robots.txt File* | *Nikto Scanner* | *Low* | *Web Root Directory* |
| 9 | *A05:2021 – Security Misconfiguration* | *Wildcard Certificate* | *Nikto Scanner* | *Medium* | *SSL/TLS Configuration* |
| 10 | *A05:2021 – Security Misconfiguration* | *Exposed Files* | *Nikto Scanner* | *Medium* | *Web Directory* |
| 11 | *A05:2021 – Security Misconfiguration* | *Server Leaks Inodes* | *Nikto Scanner* | *Low* | *HTTP Response Header* |
| 12 | *A05:2021 – Security Misconfiguration* | *CSP: Wildcard Directive* | *Helium Security, ZAP* | *Medium* | *Main Page Configuration* |

| No | Kategori OWASP TOP 10 | Celah Kerentanan | Tools | Tingkat Kerentanan | *Location* |
|---|---|---|---|---|---|
| 13 | *A05:2021 – Security Misconfiguration* | *CSP: script-src unsafe-inline* | *Helium Security, ZAP* | *Medium* | *Main Script Configuration* |
| 14 | *A05:2021 – Security Misconfiguration* | *CSP: style-src unsafe-inline* | *Helium Security, ZAP* | *Medium* | *CSS Style Configuration* |
| 15 | *A05:2021 – Security Misconfiguration* | *Sub Resource Integrity Attribute Missing* | *Helium Security* | *Medium* | *External Scripts* |
| 16 | *A05:2021 – Security Misconfiguration* | *Cookie No HttpOnly Flag* | *Helium Security, ZAP* | *Low* | *Cookie Settings* |
| 17 | *A05:2021 – Security Misconfiguration* | *Cookie Without Secure Flag* | *Helium Security, ZAP* | *Low* | *Cookie Settings* |
| 18 | *A05:2021 – Security Misconfiguration* | *Cookie without SameSite Attribute* | *Helium Security, ZAP* | *Low* | *Cookie Settings* |

| No | Kategori OWASP TOP 10 | Celah Kerentanan | Tools | Tingkat Kerentanan | *Location* |
|---|---|---|---|---|---|
| 19 | *A05:2021 – Security Misconfiguration* | *Permissions Policy Header Not Set* | *Helium Security* | *Low* | *Header Configuration* |
| 20 | *A05:2021 – Security Misconfiguration* | *Non-Storable Content* | *Helium Security* | *Informational* | *API and Dynamic Content* |
| 21 | *A05:2021 – Security Misconfiguration* | *Re-examine Cache-control Directives* | *Helium Security, ZAP* | *Informational* | *Cache Configuration* |
| 22 | *A05:2021 – Security Misconfiguration* | *Session Management Response Identified* | *Helium Security, ZAP* | *Informational* | *Session Management* |
| 23 | *A05:2021 – Security Misconfiguration* | *Storable and Cacheable Content* | *Helium Security* | *Informational* | *Cache Configuration* |
| 24 | *A05:2021 – Security Misconfiguration* | *Information Disclosure - Suspicious Comments* | *ZAP* | *Informational* | *Multiple Locations* |

| No | Kategori OWASP TOP 10 | Celah Kerentanan | Tools | Tingkat Kerentanan | *Location* |
|---|---|---|---|---|---|
| 25 | *A05:2021 – Security Misconfiguration* | *User Agent Fuzzer* | *ZAP* | *Informational* | *Multiple Locations* |
| 26 | *A06:2021 – Vulnerable and Outdated Components* | *Vulnerable JS Library: moment.js 2.21.0* | *ZAP* | *Medium* | */vendors/scripts/script.min.js* |

Berikut adalah tabel hasil kerentanan dari daftar yang diberikan, yang mengelompokkan kategori OWASP Top 10 dan menunjukkan apakah celah kerentanan ditemukan atau tidak serta presentasenya:

**Tabel 4.22** Daftar Kerentanan berdasarkan Kategori OWASP TOP 10

| No | Kategori OWASP Top 10 | Celah Kerentanan | Presentase |
|---|---|---|---|
| 1 | *A01:2021 – Broken Access Control* | Ditemukan | 16.67 |
| 2 | *A02:2021 – Cryptographic Failures* | Ditemukan | 4.17 |
| 3 | *A03:2021 – Injection* | Ditemukan | 4.17 |
| 4 | *A05:2021 – Security Misconfiguration* | Ditemukan | 66.67 |
| 5 | *A06:2021 – Vulnerable and Outdated Components* | Ditemukan | 4.17 |
| 6 | *A04:2021 – Insecure Design* | Tidak Ditemukan | 0 |
| 7 | *A07:2021 – Identification and Authentication Failures* | Tidak Ditemukan | 0 |
| 8 | *A08:2021 – Software and Data Integrity Failures* | Tidak Ditemukan | 0 |
| 9 | *A09:2021 – Security Logging and Monitoring Failures* | Tidak Ditemukan | 0 |

| No | Kategori OWASP Top 10 | Celah Kerentanan | Presentase |
|---|---|---|---|
| 10 | *A10:2021 – Server-Side Request Forgery (SSRF)* | Tidak Ditemukan | 0 |

Setelah mengidentifikasi dan mengklasifikasikan kerentanan yang ditemukan berdasarkan kategori OWASP Top 10, pengguna dapat menganalisis persentase masing-masing tingkat kerentanan (*High, Medium, Low,* dan *Informational*) dari total kerentanan yang terdeteksi. Analisis ini memberikan gambaran umum tentang distribusi tingkat risiko yang terkait dengan situs *https://sicama.unjaya.ac.id* dan membantu dalam menentukan prioritas perbaikan keamanan. Berdasarkan hasil analisis dari berbagai *tools* pengujian keamanan yang digunakan (*Nikto Scanner, Helium Security, dan ZAP*), berikut adalah distribusi kerentanan berdasarkan tingkat risiko:

1. Kerentanan tinggi (*high*): Hanya satu kerentanan yang ditemukan dalam kategori ini, yaitu *SQL Injection* pada *endpoint /proseslogin.php*. Kerentanan ini mewakili 3.85% dari total kerentanan dan memerlukan perhatian segera karena dapat digunakan oleh penyerang untuk mendapatkan akses tidak sah ke sistem.
2. Kerentanan sedang (*medium*): Sebagian besar kerentanan (53.85%) jatuh dalam kategori ini. Kerentanan ini mencakup berbagai masalah seperti *Absence of Anti-CSRF Tokens*, konfigurasi CSP yang tidak memadai, dan penggunaan *library JavaScript* yang rentan. Meskipun tidak seberbahaya kerentanan tingkat tinggi, kerentanan ini tetap perlu ditangani segera untuk mencegah potensi eksploitasi.
3. Kerentanan rendah (*low*): Sekitar 26.92% dari total kerentanan berada pada tingkat risiko rendah. Ini termasuk masalah-masalah seperti *cookie* yang tidak dikonfigurasi dengan benar dan kebocoran informasi melalui header respons. Kerentanan ini memerlukan perhatian, tetapi biasanya tidak mendesak.
4. Kerentanan informasional (*informational*): 15.38% dari kerentanan yang ditemukan bersifat informasional, yang menunjukkan masalah potensial yang mungkin memerlukan pemantauan atau penyesuaian lebih lanjut.

Meskipun tidak kritis, informasi ini membantu dalam meningkatkan keamanan keseluruhan sistem.

**Tabel 4.23** Persentase Kerentanan yang ditemukan

| No | Tingkat Kerentanan | Jumlah Kerentanan | Persentase (%) |
|---|---|---|---|
| 1 | *High* | 1 | 3.85% |
| 2 | *Medium* | 14 | 53.85% |
| 3 | *Low* | 7 | 26.92% |
| 4 | *Informational* | 4 | 15.38% |

### *4.5.5* Pencegahan *(Preventing)* berdasarkan OWASP TOP 10

Setelah mengidentifikasi dan menganalisis kerentanan yang ada, langkah selanjutnya dalam meningkatkan keamanan aplikasi *web* adalah implementasi tindakan pencegahan berdasarkan rekomendasi OWASP Top 10. Pencegahan ini bertujuan untuk mengurangi risiko yang terkait dengan berbagai jenis kerentanan dan memastikan bahwa aplikasi web tetap aman dari potensi serangan. Berikut adalah langkah-langkah pencegahan yang diambil untuk setiap kategori kerentanan yang telah diidentifikasi :

1. *A03:2021 – Injection*
   a. *Prepared Statements (with Parameterized Queries)*

      Pendekatan ini mengharuskan pengembang untuk mendefinisikan seluruh *query* SQL terlebih dahulu dan kemudian meneruskan setiap parameter ke dalam *query* pada tahap berikutnya. Metode ini memastikan bahwa basis data dapat membedakan antara *query* dan data, terlepas dari masukan pengguna, sehingga mencegah penyerang mengubah maksud dari *query* tersebut. Sebagai contoh, saat melakukan *query* ke basis data, penggunaan *prepared statements* memastikan bahwa meskipun penyerang mencoba menyisipkan perintah SQL berbahaya, perintah tersebut tidak akan dieksekusi sebagai bagian dari *query*.

      *PHP Prepared Statement Example :*

```
<?php
// This should REALLY be validated too
$custname = $_GET['customerName'];
// Perform input validation to detect attacks
$conn = new mysqli($servername, $username, $password, $dbname);
 // Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
 $query = "SELECT account_balance FROM user_data WHERE user_name =
?";
$stmt = $conn->prepare($query);
$stmt->bind_param("s", $custname);
$stmt->execute();
$result = $stmt->get_result();
 while ($row = $result->fetch_assoc()) {
    echo $row['account_balance'];
}
 $stmt->close();
$conn->close();
?>
```

b. *Stored Procedures*

Pendekatan ini mirip dengan penggunaan *query* berparameter, dengan syarat prosedur tersimpan diterapkan dengan aman, yang umumnya merupakan norma. Dalam konteks ini, prosedur tersimpan melibatkan pendefinisian *query* SQL di dalam *database* itu sendiri dan memanggilnya dari aplikasi, sehingga memisahkan *query* dari masukan pengguna dan mengurangi risiko injeksi SQL. Prosedur tersimpan efektif dalam mencegah injeksi SQL dengan memastikan bahwa *query* SQL diberi parameter. Hal ini memastikan bahwa masukan pengguna diperlakukan sebagai data, bukan sebagai bagian dari perintah SQL, sehingga mencegah penyerang mengubah maksud dari *query* tersebut.

*Stored Procedure in PHP*

```
SQL Command :
DELIMITER //
CREATE PROCEDURE sp_getAccountBalance(IN customerName
VARCHAR(255))
BEGIN
SELECT account_balance FROM user_data WHERE user_name =
customerName;
END //
```

```
DELIMITER ;
```

*PHP Code :*

```
<?php
$servername = "your_server";
$username = "your_username";
$password = "your_password";
$dbname = "your_dbname";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// This should REALLY be validated
$custname = $_GET['customerName'];
try {
    // Prepare and execute stored procedure
    $stmt = $conn->prepare("CALL sp_getAccountBalance(?)");
    $stmt->bind_param("s", $custname);
    $stmt->execute();
    $result = $stmt->get_result();
    while ($row = $result->fetch_assoc()) {
        echo $row['account_balance'];
    }
     $stmt->close();
    $conn->close();
} catch (mysqli_sql_exception $e) {
    // Logging and error handling
    error_log($e->getMessage());
    echo "Error: " . $e->getMessage();
}
?>
```

c. *Allow-list Input Validation*

Menerapkan validasi masukan daftar yang diizinkan sebagai mekanisme pertahanan penting ketika bagian dari *query* SQL, seperti nama tabel atau kolom serta indikator urutan pengurutan (ASC atau DESC), tidak

dapat menggunakan variabel pengikat. Pendekatan ini melibatkan validasi masukan pengguna terhadap daftar nilai yang dapat diterima yang telah ditentukan sebelumnya, memastikan bahwa hanya masukan legal dan yang diharapkan diperbolehkan dalam *query* SQL. Idealnya, nilai-nilai ini harus berasal dari *query* itu sendiri, bukan dari parameter pengguna. Validasi masukan daftar yang diizinkan membantu menjaga integritas *query* SQL dengan memastikan bahwa masukan pengguna yang tidak divalidasi tidak dimasukkan ke dalam query.

*Safe Table Name Validation in PHP :*

```
<?php
// Define the allowed table names
$allowedTables = array(
    "students" => "studentsTable",
    "courses" => "coursesTable",
    "enrollments" => "enrollmentsTable"
);
 // Function to validate table name
function validateTableName($input) {
    global $allowedTables;
    if (array_key_exists($input, $allowedTables)) {
        return $allowedTables[$input];
    } else {
        throw new Exception("Unexpected value provided for table
name");
    }
}
// Example usage
try {
    // Assuming the user input is received from a GET parameter
    $userInput = $_GET['tableName'];
        // Validate the table name
    $tableName = validateTableName($userInput);
        // Use the validated table name in the SQL query
    $query = "SELECT * FROM " . $tableName;
    echo "Safe query: " . $query;
```

```
} catch (Exception $e) {
    // Handle the exception
    echo "Error: " . $e->getMessage();
}
?>
```

2. *A01:2021 – Broken Access Control*

   a. *Token Based Mitigation*

Metode ini melibatkan pembuatan token unik, rahasia, dan acak di sisi server yang divalidasi dengan setiap permintaan pengguna. Token ini dapat dibuat per sesi atau per permintaan, di mana token per permintaan menawarkan keamanan lebih tinggi tetapi dapat mengganggu kegunaan seperti pada tombol "Kembali" di *browser*. Saat klien membuat permintaan, *server* memeriksa keberadaan dan validitas token CSRF dengan membandingkannya dengan token yang disimpan dalam sesi pengguna. Jika token tidak valid atau hilang, permintaan ditolak dan dicatat sebagai potensi serangan. Token tidak boleh dikirimkan melalui *cookie* atau URL untuk mencegah paparan kepada penyerang. Token CSRF harus bersifat unik per sesi, rahasia, dan acak. Token ini disertakan dalam respons HTML atau JSON dan dikirim kembali ke *server* dalam bidang formulir tersembunyi atau header khusus dalam permintaan AJAX, sehingga penyerang tidak dapat membuat permintaan yang valid tanpa mengetahui token tersebut.

Contoh :

*Generate the CSRF Token*

```
session_start();
 // Generate a CSRF token if it doesn't exist
if (empty($_SESSION['csrf_token'])) {
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}
Validating the CSRF Token on Form Submission :
<?php
session_start();
 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Check if CSRF token is set and valid
```

```
    if (!isset($_POST['csrf_token']) || $_POST['csrf_token'] !==
$_SESSION['csrf_token']) {
        // Invalid token, handle the error
        die("CSRF token validation failed");
    }
     // Process the form data
    // ...
     // Optionally regenerate the token to prevent reuse
    unset($_SESSION['csrf_token']);
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}
?>
```

b. *Disallowing simple requests*

Permintaan "sederhana" yang dikirimkan melalui *tag <form>* berisiko terhadap serangan CSRF karena browser mengizinkan permintaan tersebut dikirim ke asal mana pun tanpa *preflight*. Oleh karena itu, penting untuk melindungi permintaan ini menggunakan metode seperti token CSRF. Untuk mengurangi risiko, *server* atau API harus melarang tipe konten sederhana seperti *application/x-www-form-urlencoded, multipart/form-data*, atau *text/plain*. Banyak aplikasi web modern menggunakan API JSON dan memerlukan CORS, tetapi mereka mungkin tetap rentan terhadap CSRF jika menerima tipe konten sederhana ini. Dengan melarang tipe konten sederhana, aplikasi web dapat meningkatkan keamanan terhadap serangan CSRF.

Contoh konfigurasi *Cross-Origin Requests (*CORS*):*

```
Access-Control-Allow-Origin=http://website-name.com
Access-Control-Allow-Credentials=true
```

c. *SameSite (Cookie Attribute)*

Mengonfigurasi atribut *SameSite* dalam *cookie* untuk meningkatkan keamanan dan melindungi dari serangan CSRF. Nilai *Strict (Set-Cookie: JSESSIONID=xxxx; SameSite=Strict)* mencegah pengiriman *cookie* dalam semua konteks lintas situs. Nilai *Lax (Set-Cookie: JSESSIONID=xxxx; SameSite=Lax)* memberikan keseimbangan antara keamanan dan kegunaan,

memungkinkan pengiriman cookie saat mengikuti tautan eksternal namun memblokir metode rentan seperti POST. Nilai *None (Set-Cookie: JSESSIONID=xxxx; SameSite=None; Secure)* mengizinkan pengiriman *cookie* dalam semua permintaan lintas situs, namun memerlukan *flag Secure* untuk tambahan keamanan. Implementasi ini memastikan *cookie* hanya dikirim dalam konteks yang sesuai, melindungi aplikasi web dari serangan CSRF.

3. *A02:2021 – Cryptographic Failures*
    a. Enkripsi semua data dalam transit dengan protokol aman seperti TLS dengan *cipher forward secrecy* (FS), prioritas *cipher* oleh *server*, dan parameter aman. Terapkan enkripsi menggunakan arahan seperti HTTP *Strict Transport Security* (HSTS).
    b. Implementasi HSTS sangat disarankan bagi pemilik situs yang ingin meningkatkan keamanan transportasi data di situs web mereka. Untuk memasukkan *domain* dalam daftar pramuat HSTS yang dikelola oleh *Chrome* dan digunakan oleh *browser* lain seperti *Firefox* dan *Safari*, pemilik situs harus menggunakan *header* khusus:
    *Strict-Transport-Security: max-age=31536000; includeSubDomains; preload.*
4. *A05:2021 – Security Misconfiguration*
    a. Proses pengerasan yang berulang membuatnya cepat dan mudah untuk menerapkan lingkungan lain yang dikunci dengan tepat. Lingkungan pengembangan, QA, dan produksi harus dikonfigurasi secara identik, dengan kredensial berbeda yang digunakan di setiap lingkungan. Proses ini harus diotomatisasi untuk meminimalkan upaya yang diperlukan untuk menyiapkan lingkungan baru yang aman.
    b. *Platform* minimal tanpa fitur, komponen, dokumentasi, dan sampel yang tidak diperlukan. Hapus atau jangan instal fitur dan kerangka kerja yang tidak digunakan.

c. Arsitektur aplikasi yang tersegmentasi memberikan pemisahan yang efektif dan aman antar komponen atau penyewa, dengan segmentasi, containerisasi, atau *Cloud Security Group*(ACL).

d. Menggunakan versi PHP yang stable dengan konfigurasi yang sesuai dengan sistem aplikasi yang dibutuhkan. Dengan memperhatikan beberapa konfigurasi dibawah ini :

*Error handling* :

```
expose_php = Off
error_reporting = E_ALL
display_errors = Off # Matikan pada server produksi
display_startup_errors = Off
log_errors = On
error_log = /valid_path/PHP-logs/php_error.log
ignore_repeated_errors = Off
```

*Session handling :*

```
session.save_path = /path/PHP-session/
session.name = myPHPSESSID
session.auto_start = Off
session.use_trans_sid = 0
session.cookie_domain = full.qualified.domain.name
session.use_strict_mode = 1
session.use_cookies = 1
session.use_only_cookies = 1
session.cookie_lifetime = 14400 # 4 hours
session.cookie_secure = 1
session.cookie_httponly = 1
session.cookie_samesite = Strict
session.cache_expire = 30
session.sid_length = 256
session.sid_bits_per_character = 6 # PHP 7.2+
session.hash_function = 1 # PHP 7.0-7.1
session.hash_bits_per_character = 6 # PHP 7.0-7.1
```

5. *A06:2021 – Vulnerable and Outdated Components*

a. Hapus dependensi yang tidak digunakan, fitur, komponen, file, dan dokumentasi yang tidak diperlukan.

b. Pantau pustaka dan komponen yang tidak dipelihara atau tidak membuat patch keamanan untuk versi yang lebih lama. Jika patching tidak memungkinkan, pertimbangkan untuk menerapkan patch virtual untuk memantau, mendeteksi, atau melindungi terhadap masalah yang ditemukan.
c. Untuk memastikan keamanan komponen sisi klien dan sisi *server*, inventarisasikan versi kerangka kerja, pustaka, dan dependensinya menggunakan alat seperti *Version*, OWASP *Dependency-Check*, dan *retire.js*. Terus pantau sumber seperti *Common Vulnerabilities and Exposures* (CVE) dan *National Vulnerability Database* (NVD) untuk kerentanan terkait komponen. Gunakan alat analisis komposisi perangkat lunak untuk mengotomatisasi proses ini. Selain itu, berlangganan email pemberitahuan tentang kerentanan keamanan terkait komponen yang Anda gunakan untuk tetap terinformasi dan segera mengambil tindakan perbaikan yang diperlukan.
d. Menggunakan komponen dari sumber resmi melalui tautan aman. Lebih memilih paket yang ditandatangani untuk mengurangi kemungkinan menyertakan komponen berbahaya yang dimodifikasi.