ANALISIS DAN UJI COBA TINGKAT KEAMANAN *WEBSITE* UIN AR-RANIRY MENGGUNAKAN *ACUNETIX WEB VULNERABILITY SCANNER*

TUGAS AKHIR

Diajukan Oleh:
MULYA AKMAL
NIM. 180705004
Mahasiswa Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2023 M / 1445 H

ANALISIS DAN UJI COBA TINGKAT KEAMANAN *WEBSITE* UIN AR-RANIRY MENGGUNAKAN *ACUNETIX WEB VULNERABILITY SCANNER*

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh
Sebagai Beban Studi Memperoleh Gelas Sarjana dalam Ilmu Teknologi Informasi

Diajukan Oleh:
MULYA AKMAL
NIM. 180705004
Mahasiswa Program Studi Teknologi Informasi
Program Studi Teknologi Informasi

Disetujui Oleh:

Pembimbing I	pembimbing II
Malahayati, M.T	Rahmat Musfikar, M.Kom
NIP.198301272015032003	NIP.198909132020121015

Mengetahui,
Ketua Program Studi Teknologi Informasi

Ima Dwitawati, MBA
NIP.198210132014032002

ANALISIS DAN UJI COBA TINGKAT KEAMANAN *WEBSITE* UIN AR-RANIRY MENGGUNAKAN *ACUNETIX WEB VULNERABILITY SCANNER*

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Skripsi
Fakultas Sains dan Teknologi UIN Ar-Raniry dan Dinyatakan Lulus
Serta Diterima sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu Teknologi Informasi

Pada Hari/Tanggal: 27 Juli 2023
9 Muharram 1445 H

Panitia Ujian Munaqasyah Skripsi

Ketua,	Sekretaris,
Malahayati, M.T	**Rahmat Musfikar, M.Kom**
NIP. 198301272015032003	**NIP.198909132020121015**
Penguji I,	Penguji II,
Mulkan Fadhli, S.T., M.T	**Aulia Syarif Aziz, S.Kom., M.Sc**
NIP.198811282020121006	**NIP.199305212022031001**

Mengetahui,
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Ar-Raniry Banda Aceh

Dr. Ir. M. Dirhamsyah, M.T.,
NIP.196210021988111001

LEMBARAN PERNYATAAN KEASLIAN KARYA ILMIAH/SKRIPSI

Nama	:	Mulya Akmal
NIM	:	180705004
Program Studi	:	Teknologi Informasi
Fakultas	:	Sains Dan Teknologi UIN Ar-Raniry Banda Aceh
Judul Skripsi	:	Analisis dan Uji Coba Tingkat Keamanan *Website* UIN Ar-Raniry Menggunakan *Acunetix Web Vulnerability Scanner*

Dengan ini saya menyatakan bahwa dalam penulisan skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggung jawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya; dan
4. Tidak memanipulasi dan memalsukan kata.
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawan atas karya ini;

Bila di kemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains Dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 20 Juli 2023
Yang membuat pernyataan,

Mulya Akmal
NIM: 180705004

ABSTRAK

Nama	:	Mulya Akmal
NIM	:	180507004
Fakultas/Prodi	:	Sains Dan Teknologi/Teknologi Informasi
Judul	:	Analisis Tingkat Keamanan *Website* UIN Ar-Raniry Menggunakan *Acunetix Web Vulnerability Scanner*
Tanggal Sidang	:	27 Juli 2023
Jumlah Halaman	:	68 Halaman
Pembimbing I	:	Malahayati, M.T
Pembimbing II	:	Rahmat Musfikar, M.Kom
Kata Kunci	:	Tingkat Keamanan, *Website, Vulnerability Scanner*, *Acunetix web vulnerability scanner*

Website Universitas Islam Negeri Ar-Raniry merupakan media akses yang dilakukan seluruh mahasiswa, namun *website* siakad UIN Ar-Raniry belum pernah dilakukan pengujian vulnerability, sehingga tidak menutup kemungkinan keamanan *website* siakad.ar-raniry.ac.id di-*hack* oleh sekelompok tertentu untuk mencuri, menghapus atau merusak data. Dengan melakukan pengujian bertujuan mencari tau level tingkatan keamanan *website* yang hasilnya tidak menjamin sistem bebas dari resiko serangan, tetapi dapat meminimalisir serangan yang dapat disalahgunakan, karena untuk menjelajahi semua aspek diperlukan pengujian tingkat lanjut. Jenis penelitian yang digunakan dalam penyususnan Tugas Akhir ini yaitu jenis penelitian eksperimen. Namun, dalam menganalisis hasil *scanning website* menggunakan penelitian kualitatif bersifat deskriptif. Hasil yang ditemukan Acunetix pada *website* siakad UIN Ar-Raniry diperoleh 4 keretanan pada level *low*, 2 pada level *medium*, 0 pada level *high* dan 4 pada level *information* yang berarti kerentanan terjadi karena kesalahan konfigurasi , *site coding* yang lemah dan *website* belum bisa dikategorikan aman karena masih terdapat celah keamanan dan berpotensi diakases tanpa izin yang bisa merusak sistem. *Website* siakad UIN Ar-Raniry perlu dilakukan penelitian mendalam mengenai kerentanan dan meningkatkan keamanan *website*, hal ini berupaya terhindar dari kerentanan serta diperlukan penyajian hasil audit keamanan dalam bentuk lain yang memiliki standar keamanan informasi lain agar hasilnya lebih optimal.

KATA PENGANTAR

Puji syukur kepada Allah swt yang maha pengasih lagi maha penyayang yang telah memberikan kasih sayang, kesempatan dan kesabaran kepada penulis sehingga skripsi ini dapat disusun dengan baik. Salawat dan Salam penulis sampaikan kepada Nabi Muhammad SAW dan keluarga beserta sahabat beliau yang telah membimbing kita ke alam yang penuh dengan ilmu pengetahuan. Atas rahmat Allah swt yang maha kuasa, penulis dapat menyusun tugas akhir dengan judul "Analisis dan Uji Coba Tingkat Keamanan *Website* UIN Ar-Raniry Menggunakan *Acunetix Web Vulnerability Scanner*".

Dalam penyusunan penelitian ini, tidak lepas dari arahan dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mngucapkan rasa hormat dan terima kasih kepada semua pihak yang telah membantu. Pihak-pihak terkait di antaranya ialah:

1. Ayahanda M Ali dan Ibunda Nurasiah selaku orang tua yang selalu mendukung dengan doa-doanya. Serta kakak dan adik Lisya Wirdah dan Ajral Muhsinin yang telah menyemangati.
2. Bapak Dr. Ir. M. Dirhamsyah, M.T selaku Dekan Fakultas dan Teknologi beserta jajarannya.
3. Ibu Ima Dwitawati, MBA selaku ketua program studi Teknologi Informasi.
4. Ibu Malahayati, M.T selaku Dosen Pembimbing I dan Bapak Rahmat Musfikar, M.Kom selaku Dosen Pembimbing II yang telah banyak membantu dan memberikan bimbingan hingga Tugas Akhir ini selesai.

5. Bapak Nazaruddin Ahmat, M.T selaku penasehat akademik.
6. Seluruh Dosen Teknologi Informasi yang telah berbagi ilmu masing-masing bidang.
7. Safira Putri Riskhi yang selalu men-*support* dalam pembuatan Tugas Akhir ini.
8. Sahabat dan teman-teman mahasiswa program studi Teknologi Informasi angkatan 2018 serta seluruh keluarga Teknologi Informasi yang telah memberikan dukungan dan semangat dalam penyelesaian Tugas Akhir ini.
9. Dan semua pihak yang bersangkutan turut ikut mendukung yang tidak bisa disebut namanya satu persatu.

Penulis menyadari bahwa masih banyak terdapat kekurangan. Maka dari itu, kritik serta saran sangat diharapkan untuk menyempurnakan Tugas Akhir ini sehingga dapat berguna bagi penulis dan pembaca. Aamiin.

Banda Aceh, 20 Juli 2023
Penulis,

Mulya Akmal

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Pesatnya perkembangan ilmu Teknologi Informasi yang merupakan ilmu pengetahuan dan teknologi sangat mempengaruhi dalam berbagai aspek kehidupan bisnis sehari-hari. Dalam perkembangannya, internet menjadi salah satu elemen utama dalam mendapatkan informasi yang dibutuhkan. Untuk mengakses informasi melalui internet terdapat banyak cara, salah satunya yaitu dengan mengunjungi *website* yang mana terdapat kumpulan halaman dan hyperlink yang berhubungan digunakan untuk menampilkan informasi tekstual, gambar, suara statis atau dinamis. Namun, bersamaan dengan kemajuan teknologi, risiko keamanan *website* juga semakin meningkat. Ancaman seperti serangan siber, peretasan, dan eksploitasi kerentanan telah menjadi tantangan yang harus diatasi oleh para pemilik *website*.

Aspek keamanan *website* suatu hal yang penting dari sistem informasi, akan tetapi, problema keamanan ini kerap tidak diperhatikan oleh pemilik administrator sistem informasi. Adanya pensyaratan keamanan sistem aplikasi untuk melindungi data yang dapat memenuhi aspek keamanan yaitu; aspek kerahasian *(confidemtiality)*, aspek integritas *(integrity)* dan aspek ketersediaan *(avaibility)* dari keamanan sistem aplikasi (Khaldun, 2020). Untuk meminimalisir pemasalahan tersebut penting bagi administrator sistem untuk secara proaktif mengidentifikasi dan mengatasi potensi kerentanan pada *website* mereka. Salah

satu cara yang efektif untuk melakukan ini adalah dengan menggunakan alat bantu keamanan seperti *Acunetix Web Vulnerability Scanner*.

Acunetix Web Vulnerability Scanner merupakan alat pengujian otomatis pada layanan aplikasi *web* yang memindai aplikasi *web* dengan menyelidiki, mengidentifikasi, dan memulihkan kerentanan seperti SQL *(Structured Query Language) Injection, Cross Site Scripting,* dan kerentanan *web* yang dieksploitasi lainnya. *Acunetix Web Vulnerability Scanner* menggunakan perangkat lunak untuk mengidentifikasi kerentanan yang ada pada *website* karena kerentanan tersebut yang nantinya diketahui dari hasil penilaian risiko *website*. Pengujian kerentanan dilakukan terhadap pengukuran yang absolut untuk memberikan peningkatan kualitas. Hasil evaluasi dipertimbangkan sehingga dapat mengambil tindakan peningkatan pencegahan dan mempelajari prosedur kerja penyerang (Information & Zirwan, 2022).

Universitas Islam Negeri Ar-Raniry merupakan salah satu Perguruan Tinggi yang menyediakan tampilan *website* baik informasi pengenalan Universitas atau informasi yang berkaitan dengan perkuliahan. Alamat *website* tersebut yaitu https://uin.ar-raniry.ac.id. Selain itu juga mempunyai beberapa *website* seperti pendaftaran.ar-raniry.ac.id, siakad.ar-raniry.ac.id. Salah satu *website* yang sering di kunjungi oleh mahasiswa yaitu *website* siakad.ar-raniry.ac.id yang merupakan sebuah *website* portal mahasiswa yang berisikan data seluruh mahasiswa UIN Ar-Raniry Banda Aceh, baik itu berupa biodata mahasiswa, KRS (Kartu Rencana Studi), maupun nilai IP (Indeks Prestasi) atau IPK (Indeks Prestasi Kumulatif) mahasiswa.

Website siakad.ar-raniry.ac.id tidak dapat terhindar dari ancaman keamanan *website* yang berpotensi di-*hack* oleh sekelompok tertentu untuk mencuri data, menghapus data atau merusak data, maka perlu untuk dilakukan pengujian. Pengujian tersebut mengacu pada tingkatan keamanan *website* siakad.ar-raniry.ac.id yang hasilnya bukan menggaransikan sistem bebas dari resiko serangan, tetapi dapat meminimalisir serangan yang dapat disalahgunakan, karena untuk menjelajahi semua aspek diperlukan pengujian tingkat lanjut.

Fokus penelitian ini yaitu melakukan *scanning* pada *website* siakad.ar-raniry.ac.id untuk menguji dan mencari celah keamanan serta kelemahan *website* dengan menggunakan *Acunetix Web Vulnerability Scanner*. Penggunaan *Acunetix Web Vulnerability Scanner* berfungsi memberikan solusi terhadap kelemahan yang diamati dan mengelola lanjut terhadap *website* yang diuji dan meningkatkan keamanan sistem.

Oleh karena itu penulis tertarik untuk meneliti dengan judul **"Analisis Dan Uji Coba Tingkat Keamanan *Website* UIN Ar-Raniry Menggunakan *Acunetix Web Vulnerability Scanner*".** Dengan harapan pada penelitian ini menghasilkan laporan analisis yang dapat dijadikan informasi dalam upaya meminimalisir keretanan yang terdapat pada *website* siakad.ar-raniry.ac.id. dan dapat mengambil tindakan untuk meningkatkan sistem keamanan *website* siakad.ar-raniry.ac.id.

1.2 Rumusan Masalah

Berdasarkan konteks permasalahan di atas, maka dapat dirumuskan yaitu bagaimana menguji dan menganalisis tingkat keamanan *website* UIN Ar-Raniry menggunakan *Acunetix Web Vulnerability Scanner*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini diperoleh tingkat keamanan yang terdapat pada *website* UIN Ar-Raniry Banda Aceh menggunakan perangkat lunak (*software*) *Acunetix Web Vulnerability Scanner* serta diberikan solusi dari permasalahan yang ditemukan.

1.4 Batasan Masalah

Untuk mencapai tujuan dan sasaran dari penelitian ini, masalah-masalah yang dibatasi sebagai berikut:

1. Perangkat lunak (*software*) yang digunakan untuk melakukan pengujian tingkat keamanan dalam penelitian ini adalah *Acunetix Web Vulnerability Scanner*.
2. Pengujian dan analisis tingkat keamanan hanya relevan pada situs *web* siakad.ar-raniry.ac.id.
3. Penulis hanya menguji dan menemukan titik lemah *website* serta memberikan usulan atau saran berdasarkan analisis yang dilakukan dalam penelitian ini.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh pada penelitian ini adalah sebagai berikut:

1. Menjadikan acuan untuk lebih meningkatkan aspek keamanan *website* UIN Ar-Raniry Banda Aceh.
2. Untuk membantu mengurangi kemungkinan kerugian akibat lemahnya keamanan *website*, dengan harapan adanya penelitian ini, para pembaca atau mahasiswa mendapatkan gambaran bagaimana mekanisme kerjanya.

1.6 Sistematika Penulisan

Sistem penulisan dalam penelitian ini sebagai berikut:

Bab I Pendahuluan, bab ini meliputi latar belakang masalah, rumusan masalah, tujuan masalah, batasan masalah dan manfaat penulisan, serta sistematisasi penulisan.

Bab II Latar belakang Teori, bab ini memuat penelitian terdahulu dan penjelasan tentang masalah yang dibahas.

Bab III Metode penelitian, yang menguraikan lokasi, objek penelitian, bahan, alat, sistem kerja, teknik pengumpulan data, analisis data dan kerangka berpikir.

Bab IV Hasil dan pembahasan, yang meliputi analisis hasil pengolahan data dan membahas tingkat keamanan *website* UIN Ar-Raniry menggunakan *Acunetix Web Vulnerability Scanner*.

Bab V kesimpulan dan saran, bab ini meluputi beberapa kesimpulan yang diambil berdasarkan hasil dari penelitian dan saran.

BAB II

LANDASAN TEORITIS

2.1 Penelitian Terdahulu

Penelitian terdahulu penting dibuat dalam sebuah karya ilmiah termasuk Tugas Akhir untuk pemetaan terhadap temuan dari riset-riset yang dilakukan sebelumya, sehingga dapat menghindari dari duplikasi dan plagiasi dan otentitas penelitian ini dapat dipertanggungjawabkan secara ilmiah. Dalam tulisan ini, penulis meriset penelitian yang telah dilakukan berhubungan dengan objek kajian tentang analisis tingkat keamanan *website* UIN Ar-Raniry menggunakan *Acunetix Web Vulnerability Scanner*. Adapun penelitian terdahulu yang terkait dengan penelitian ini dapat dilihat pada tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Judul	Hasil Penelitian	Perbedaan
1	Hasbullah Jamaluddin dan Nurul Fitriani Suaeb 2018	Analisis keamanan *website* terhadap *sniffing* process pada jaringan nirkabel menggunakan aplikasi *wireshark* (studi kasus: simak UNISMUH)	*Website* Simak Unismuh belum menggunakan sertifikat SSL, maka sangat rentang terhadap serangan MITM (man in the middle) meski *browser* yang digunakan merupakan versi terbaru.	Uji coba keamanan *website* pada jaringan nirkabel aplikasi *wireshark* terhadap *sniffing process.*

2	Bagus Wicaksono 2018	Pengujian Celah Keamanan Aplikasi Berbasis Web Menggunakan Teknik Penetration Testing Dan DAST (*Dynamic Application Security Testing*)	Pada pengujian celah keamanan dengan metode DAST pada *website* bagusw.win yang meliputi serangan *Cross Side scripting, broken access control*, serta SQL injection pada *website* dapat dibuktikan, sehingga dapat diakukan dengan proses evaluasi untuk memperbaiki ketiga celah keamanan.	Menggunakan teknik *penetration testing* dan DAST (*Dynamic Application Security Testing*)
3	Azzah Hania Dalila 2022	Analisis Keamanan Pada *Website* PT. Aneka Tirta Talenta Menggunakan Metode Penetration Testing	Pengujian kerentanan SQL Injection dengan tools exploit yaitu Burp Suite, SQL Map dan JSQL bahwa tidak ada kerentanan SQL Injection pada *website* anekatirta-security.my.id. Perbaikan kerentanan tingkat High berhasil dengan melakukan debugging dan perbaikan pada source code. Terdapat beberapa solusi pada	Melakukan dengan metode *penetration testing* dan hanya fokus pada SQL Injection dan

			tingkat Medium dan Low agar dapat diperbaiki oleh pengelola *website*.	
4	Shafira Khairunnisa 2022	*Analisis Keamanan Website Repository Institut Teknologi Telkom Purwokerto Menggunakan Metode Vulnerability Assessment*	*Scanning* Port pemantauan keamanan ada beberapa port yang terbuka dan tertutup di *server* repository IT Telkom Purwokerto. Disarankan menggunakan *firewall* untuk mengizinkan port tertentu sambil memblokir yang lain berupaya mempertahankan jaringan dari serangan *cyber*	Menguji tingkat keamanan *website* menggunakan pendekteksi celah yang berbeda dan objek uji coba yang berbeda.

Berdasarkan beberapa penelitian terdahulu di atas dapat disimpulkan bahwa tingkat keamanan *website* sangat diperlukan bertujuan untuk melindungi data dan informasi *website*. Oleh karena itu pengujian ini merupakan salah satu cara untuk mengetahui tingkatan keamanan *website*. Metode yang diterapkan pada penelitian-penelitian diatas seperti; Penetration Testing, DAST, dan *vulnerability assesment* dengan menggunakan bantuan *software*. *Software* atau tools tersebut seperti; Acunetix, Nmap, Nessus, Vega, OWASP ZAP. Namun, tidak semua tools tersebut memberikan solusi terhadap masalah pada penelitiannya, hanya

sebagiannya dan fokus penelitian yang khusus pada injeksi tertentu. Dengan adanya penelitian penulis, melakukan pengujian *website* menggunakan metode *vulnerability assessment* berupa *Acunetix Web Vulnerability Scanner*, selain menjadi alat *scanning website* juga memiliki kelebihan yaitu dapat memberikan solusi atau rekomendasi dalam meningkatkan tingkat keamanan *website*.

2.2 Informasi

Banyak definisi tentang informasi yang dikemukakan oleh para ahli. Berikut ini akan disampaikan pengertian informasi dari berbagai sumber (Teguh Wahyono: 2004):

1. Gordon B. Davis dalam bukunya *Management Informations System: Conceptual Foundations, Structure, and Development* menyebutkan informasi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.
2. Berry E. Cushing dalam buku *Accounting Information System and Business Organization*, menyebutkan bahwa informasi merupakan sesuatu yang menunjukkan hasil pengolahan data yang diorganisasi dan berguna kepada orang yang menerimanya.
3. Robert N. Anthony dan John Dearden dalam buku *Management Control Systems*: *Concept and Practise* mengatakan informasi sebagai suatu kenyataan, data, *item* yang menambah pengetahuan bagi penggunanya.

4. Stephen A. Moscove dan Mark G. Simkin dalam buku *Accounting Information System*: *Concepts and Practise*, menyebutkan bahwa informasi sebagai kenyataan atau bentuk-bentuk yang berguna yang dapat digunakan untuk pengambilan keputusan.

Berdasarkan keempat pengertian seperti tersebut di atas dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat untuk pengambilan suatu keputusan.

2.3 Keamanan Informasi

3.3.1 Pengertian Keamanan Informasi

Keamanan informasi merupakan sebuah penjagaan atau perlindungan asset perusahaan terhadap gangguan bisnis, penyingkapan informasi rahasia dan mutasi data rahasia. Hal tersebut biasanya dipaparkan sebagai menjaga keutuhan, ketersediaan, dan kerahasiaan asset organisasi, informasi dan metode (Vacca, 2009). Manajemen sistem keamanan informasi yaitu rangkaian kegiatan berkelanjutan yang ditinjau berkala dalam memastikan asset yang berkaitan berfungsi dengan aman (Gondodiyiti, 2007). Sehingga disimpulkan keamanan informasi adalah serangkaain kegiatan bertujuan melindungi aset-aset suatu perusahaan yang terkait dengan sistem informasi maupun informasi dalam memastikan ketersediaan, kerahasiaan, dan integritasnya meningkat.

2.3.2 Prinsip-Prinsip Keamanan Informasi

Ada tiga tujuan paling utama dalam mencapai keamanan informasi (Vacca, 2009), sebagai berikut:

1. Kerahasiaan berarti laporan atau informasi yang hanya disediakan pada orang yang benar-benar membutuhkan akses. Kerahasiaan dilakukan dengan mengenkripsi informasi sehingga orang-orang tertentu saja yang menguraikannya dan atau menolak aksesnya. Semua aspek sistem harus dijaga kerahasiaannya. Ini berarti bahwa akses ke seluruh lokasi pencadangan serta file log dicegah apabila mengandung informasi rahasia.
2. Integritas berarti hanya orang yang memiliki wewenang yang dapat menambah atau memperbarui informasi. Modifikasi data yang tidak sah dapat menyebabkan hilangnya integritas data. Dalam hal ini, akses ke informasi mesti diberhentikan hingga keutuhan informasi dipulihkan.
3. Ketersediaan atau availability yaitu laporan atau sebuah informasi yang terdapat pada saat dibutuhkan. Jika informasi terkait prosedur tersedia, maka proses tidak dapat berjalan.

2.4 Serangan Aplikasi *Web*

Bisnis, sekolah, pemerintah dan lainnya bergantung pada teknologi dan aplikasi *web*. Oleh karena itu, perlindungan aplikasi siteus *web* mengkaitkan pendekatan yang menggunakan fitur keamanan umum, sebagai berikut:

2.4.1 Peningkatan *Server Web* (*Hardening the Web server*)

Dengan meningkatkan *server web* bertujuan untuk mengamankan sistem operasi dan layanan sistem *server web* penting untuk melindungi dari jenis serangan lain, tetapi tidak mencegah serangan pada aplikasi *web*. Ini dikarenakan input pengguna dari *web browser* melalui HTTP harus dilakukan oleh aplikasi web ditingkat aplikasi tersebut.

2.4.2 Perlindungan Jaringan (*Protecting the Network*)

Alat keamanan siber tidak selalu keamanan siber dapat memblokir serangan siber, tetapi tidak selalu dapat memblokir serangan pada aplikasi situs *web*. Ini karena banyak alat keamanan siber mengabaikan lalu lintas konten HTTP yang merupakan cara menyerang aplikasi situs *web*. Penyerang menggunakan protokol ini untuk menargetkan kerentanan dalam perangkat lunak aplikasi situs *web* karena konten transmisi HTTP tidak divalidasi. Serangan yang paling umum terhadap aplikasi *web* adalah *Cross-Site Coding*, SQL *injection*, XML *injection*, dan *Command Injection/Directory Traversal* (Ciampa, 2012). Di bawah ini adalah jenis serangan *cyber* yang paling umum:

1. XSS (*Cross Site Scripting*)

   XSS atau *Cross Site Scripting* yaitu sebuah jenis serangan injeksi kode. XSS dijalankan oleh penyerang yang mengimpor HTML atau kode skrip klien lainnya ke halaman *web*. Serangan ini tampaknya berasal dari situs ini, memungkinkan penyerang untuk melewati keamanan sisi klien, meng-*host* aplikasi yang berbahaya dan memperoleh informasi sensitif (Ciampa, 2012).

2. SQL *Injection*

SQL *Injection* merupakan sebuah teknik yang mengekploitasi kerentanan yang muncul di lapisan basis data aplikasi. Kerentanan itu ditimbulkan ketika input pengguna tidak difilter secara baik dari karakter *escape string* yang ditambahkan pada pernyataan SQL dan ketika input pengguna bukan tipe yang kuat yang mengakibatkan eksekusi SQL yang di harapkan. Ini adalah contoh nyata dari jenis kerentanan terjadi umumnya ketika Bahasa pemograman atau skrip ditambahkan ke yang lain (Ciampa, 2012).

3. XML *Injection*

Cara kerja serangan XML *Inection* menyamai dengan serangan injeksi SQL. Penyerang yang mendapatkan *website* yang tidak menyaring input pengguna dapat menyuntikkan tag dan data XML ke dalam *database*. Jenis yang spesifik pada serangan injeksi XML yaitu injeksi XPath yang mencoba untuk mengeksploitasi kueri XML *Path Language* (XPath) yang dihasilkan dari input pengguna (Ciampa, 2012).

4. *Command Injection/ Directory Traversal*

Serangan traversal direktori yang mengeksploitasi suatu kerentanan dalam program aplikasi *website* atau *software server web* untuk memungkinkan bagi pengguna menavigasi dari direktori root ke direktori lain yang dibatasi. Kemampuan untuk berpindah ke diktori lain mengakibatkan kemungkinan bagi pengguna yang tidak

berwenang untuk melihat file yang sensitif atau rahasia, memasukkan perintah untuk dieksekusi pada *server* dikenal yang disebut *command injection* (ciampa, 2012). Hal itu misalnya, *browser* menimta halaman dinamis dari *server web* (www.server.net) untuk mendapatkan tampilan. Cara melihatnya dengan membuat sebuah URL http://www.server.net/dynamic.asp?view=display.html. Akan tetapi, kerentanan dalam kode aplikasi mengakibatkan penyerang melakukan serangan traversal direktori (Ciampa, 2012).

2.5 *Vulnerability*

2.5.1 Pengertian *Vulnerability*

Vulnerability yaitu titik dimana sistem menjadi rawan akan serangan (Lethinen, Russell, dan Gangemi, 2006). Kerentanan atau yang di sebut *vulnerability* merupakan suatu kelemahan yang terdapat dalam sistem menguatkan penyerang untuk menganggu keutuhan sistem. *Vulnerability* ini disebabkan oleh kata sandi yang mudah di tebak atau lemah, virus komputer, *malware*, SQL *injection*, *bug* perangkat lunak dan sebagainya. Oleh karena itu dapat disimpulkan bahwa *vulnerability* ini adalah suatu kelemahan terjadi pada sistem, yang berarti penyerang dapat membahayakan fungsionalitas dan keutuhan sistem menjadi rentan terhadap serangan (Vacca, 2009).

2.5.2 Jenis - Jenis *Vulnerability*

Komputer dan jaringan sangat rentan terhadap serangan, dan sistem komputer memiliki beberapa jenis kerentanan (Lethinen, Russell, dan Gangemi, 2006), yaitu:

1. *Physical Vulnerabilities* (kerentanan fisik) adalah memungkinkan orang yang tidak berwenang membobol *server* jaringan, menghancurkan perangkat jaringan, mencuri data cadangan atau mencetak.
2. Kerentanan pada *network* atau komputer yang muncul diakibatkan adanya bencana alam atau ancaman dari lingkungan. Seperti kebakaran, gempa bumi, pemadaman listrik, debu, suhu yang tidak stabil, dan sebagainya. Hal ini di sebut dengan *natural vulnerabilities*.
3. Kerentanan jaringan komputer disebabkan oleh kegagalan *hardware failure* yang memungkinkan orang yang tidak berwenang dengan mudah memberikan *security hole* dan menyebabkan kegagalan sistem disebut dengan *hardware and software vulnerabilities*.
4. *Media vulnerabilities* adalah kerentanan yang mungkin muncul di media cadangan paket dikarenakan dapat dirusak atau di curi. Media cadangan paket seperti *disk, cartridge, tape*, dan lainnya.
5. *Media Vulnerabilities* (kerentanan media) adalah kerentanan yang mungkin muncul di media cadangan paket *disk, cartridge, tape, chip memori printout* bahwa dapat di rusak atau di curi yang dikarenakan debu.
6. *Emanation Vulnerabilities* (kerentanan radiasi) adalah kerentanan pada perangkat elektronik yang mempunyai radiasi elekromagnetik

dengan mencegat sinyal yang dipancarkan komputer, *network*, sistem nirkabel dan dapat menyimpan serta mengirimkan informasi yang rentan

7. *Communication Vulnerabilities* (kerentanan komunikasi) adalah kerentanan yang memungkinkan orang yang tidak berwenang membobol sistem ketika komputer terhubung ke jaringan atau diakses melalui modem atau internet.
8. *Human Vulnerabilities* (kerentanan manusia) adalah kerentanan terbesar yang diciptakan oleh sejumlah orang yang menggunakan sistem dan mengelolanya.
9. *Exploiting vulnerabilities* adalah kerentanan yang dilakukan menyebabkan dieksploitasi berbagai macam cara. *Logging* merupakan salah satunya dikarenakan tidak adanya perlindungan pada kata sandi serta sistem dengan kontrol yang paling kecil.

Teknik atau taktik dilakukan bagi penyerang dalam menyerang *website* via internet dan *network* (O'Brien, 2005) sebagai berikut:

1. *Denial of service* yaitu taktik yang paling biasa di lakukan dalam jaringan dengan metode membombardir alat pada *website*. Hal ini seperti menyumbat sistem, memperlambat kinerja, merusak situs atau dengan memperbanyak permintaan.
2. Penyelidikan internet untuk mengidentifikasi macam layanan, komputer serta koneksinya. Oleh karena itu memungkinkan bagi

penyerang untuk mengeksploitasi kerentanan pada program dan perangkat lunak komputer tertentu. Hal ini disebut *scans*.

3. Program yang memindai ulang pada paket data yang dikirim via internet mengambil kata sandi dan seluruh kouta paket disebut dengan *sniffer*.
4. *Spoofing* adalah alamat email yang dipalsukan menipu pengguna agar menyimpan informasi sensitif seperti sandi atau nomor kartu kredit.
5. Suatu strategi yang tidak dikenal terdiri dari perintah yang mengeksploitasi *vulnerability* ditemunkan pada program perangkat lunak di sebut *Trojan horse*.
6. Setelah titik masuk yang asli masuk terdeteksi, penyerang secara terus menerus melakukan *login* ulang sehingga menjadi sulit terdeteksi dan lebih mudah. Ini dinamakan *back door*.
7. Applet jahat atau juga di sebut *malicious applets* yaitu sebagian ditulis dalam bahasa Java berupa bahasa komputer yang populer, mengeksploitasi kemampuan komputer, memodifikasi dokumen pada *hard drive*, membobol *password* atau dengan mengirimkan email yang palsu.
8. Melakukan *war dialing* artinya sebuah program yang otomatis untuk menelpon ribuan nomor telepon lewat koneksi modem.
9. Teknik lainnya yaitu sebuah perintah pada program komputer yang menimbulkan tindakan kejahatan yang di sebut sebagai *logic bomb.*

10. Suatu teknik yang menghancurkan atau membajak komputer dengan mengirimkan sejumlah besar data ke penyimpanan sementara komputer pada memori komputer yang disebut dengan pembebanan penyimpanan sementara komputer (*Buffer Overflow*).
11. *Password Cracker* perangkat lunak yang dapat menebak kata sandi.
12. Suatu metode yang dipakai untuk memperoleh akses pada sistem komputer dengan berbicara dengan karyawan perusahaan yang tidak menaruh curiga untuk mengesksrak informasi berharga seperti *password* yang disebut sebagai rekayasa sosial (*Social Engineering*).
13. *Dumpster diving* yaitu program yang memanggil ribuan nomor telpon melalui koneksi modem secara otomatis.

2.6 *Vulnerability Scanner*

Vulnerability scanner adalah sebuah program komputer yang didesain untuk mencari dan memetakan sistem untuk kelemahan pada aplikasi, komputer atau jaringan. Meningkatnya penggunaan internet membuat semakin banyaknya *website* yang bermunculan, dengan adanya *website* membuat pengembang *website* dapat menyampaikan informasinya kepada pengguna internet dengan mudah, namun sangat disayangkan kejahatan internet terus meningkat seiring bermunculannya ragam artikel yang membahas masalah *hacking* (Perdana, 2010:4).

Berikut beberapa jenis dari *tool vulnerability scanner* yaitu:

1. Acunetix

Acunetix website vulnerability scanner merupakan perangkat lunak yang dikembangkan untuk melakukan *scanning*. Kelebihan dari *tools* ini adalah kemampuannya untuk memberikan solusi dari kelemahan yang ditemukan dan mengelola *traceability* dari setiap *vulnerabilities* tersebut. Selain itu, acunetix menyediakan fungsi-fungsi tambahan yang dapat digunakan untuk melakukan pengujian lebih lanjut terhadap *website* yang diuji.

2. W3af

W3af merupakan *tools open source* berbasis python. Fungsi yang ditawarkan tidak jauh berbeda dengan acunetix berkaitan dengan *scanning vulnerabilities* pada *website*. Perbedaannya w3af menampikan hasil scan yang lebih teknis dibandingkan acunetix dan tidak memberikan solusi secara langsung untuk setiap *vulnerabilities* yang ditemukannya.

3. Wireshark

Wireshark adalah *packet analyzer* dan *open-source*. *Tools* ini seringkali digunakan untuk menemukan masalah pada jaringan, pengembangan perangkat lunak dan protokol komunikasi, dan pendidikan. Wireshark bersifat *cross-platform* dan menggunakan pcap untuk meng-*capture* paket jaringan. Wireshark dapat berjalan pada hampir semua sistem operasi yang tersedia.

4. OWASP ZAP

ZAP (*Zed Attack Proxy*) buatan OWASP (*The Open Web Security Project*) merupakan tools penetration testing yang digunakan untuk menemukan *vulnerabilities* pada aplikasi *web*. ZAP menyediakan pemindaian otomatis dan seperangkat *tools* untuk menemukan *vulnerabilities* secara manual.

5. Subgraph Vega

Vega adalah aplikasi *open-source* yang dapat digunakan untuk menguji keamanan aplikasi *web*. Vega dapat membantu menemukan dan memvalidasi SQL *Injection*, XXS (*Cross-Site Scripting*), dan *vulnerabilities* lainnya.

6. Nmap

Nmap atau disebut juga sebagai *network mapper* merupakan sebuah perangkat lunak yang berfungsi untuk mengeksplorasi serta mengaudit keamanan jaringan. Dibuat dengan tujuan untuk memeriksa cangkupan jaringan besar dengan cepat, tetapi perangkat lunak ini juga dapat berjalan pada satu host.

7. Nessus Scanner

Nessus Scanner merupakan sebuah aplikasi yang dibuat oleh *Tenable Security* untuk menemukan celah maupun *Vulnerability* yang terdapat pada sebuah *network*, Nessus sendiri memiliki beberapa fungsi diantaranya untuk *Vulnerability Scanning*, *Configuration Editing*, *Compliance Checks* dan lainnya.

2.7 *Acunetix Web Vulnerability Scanner*

Pemindai aplikasi situs *web acunetix* yaitu perangkat lunak untuk mendeteksi celah keretanan *website* (Kristison Zakaria, Onno W. Purbo, Dan Elvira Wardah: 2017). Kelebihan dalam menggunakan alat ini adalah menyediakan solusi untuk kerentanan yang diamati dan mengelola keterlacakan setiap kerentanan ini. Selain itu, *acunetix* menawarkan fitur tambahan yang memungkinkan pengujian lebih lanjut dilakukan di situs yang diuji. *Acunetix Web Vulnerability Scanner* adalah alat yang dirancang untuk menemukan kerentanan dalam penggunaan *web* pada penyerangan yang dilakukan oleh orang tidak berwenang, sehingga dapat mengeksploitasi *website* untuk memperoleh akses yang tidak sah pada data serta sistem dengan berbagai kerentanan. Berikut enam kerentanan yang dapat diidentifikasi oleh acunitix:

1. XSS (*Cross-Site Scripting*)

   Kerentanan XSS terjadi ketika aplikasi *web* tidak memvalidasi atau menyaring input pengguna dengan benar, sehingga memungkinkan serangan injeksi skrip berbahaya yang dieksekusi oleh *browser* pengguna.

2. *Injection Attacks*

   Kerentanan serangan injeksi terjadi ketika input yang diterima oleh aplikasi *web* tidak diverifikasi atau diolah dengan benar. Hal ini dapat memungkinkan penyerang menyisipkan perintah atau kode berbahaya ke dalam permintaan atau parameter yang dieksekusi oleh sistem, menyebabkan kerentanan yang dapat disalahgunakan.

3. CSRF (*Cross-Site Request Forgery*)

Kerentanan CSRF terjadi ketika aplikasi *web* tidak melaksanakan mekanisme perlindungan yang memvalidasi sumber permintaan yang sah. Hal ini memungkinkan penyerang untuk memanipulasikan permintaan yang dilakukan oleh pengguna yang sah, melakukan tindakan yang tidak diinginkan tanpa sepengetahuan mereka.

4. SQL *Injection*

Kerentanan SQL *injection* terjadi ketika input pengguna tidak diverifikasi atau diolah dengan benar sebelum digunakan dalam perintah SQL. Hal ini dapat memungkinkan penyerang menyisipkan kode SQL berbahaya, yang sapat mengakibatkan manipulasi basis data, pengungkapan data sensitif atau bahkan penghapusan data.

5. Server Misconfiguration

Kerentanan konfigurasi *server* terjadi ketika *server web* tidak dikonfigurasi dengan benar, meninggalkan celah keamanan yang dapat disalahgunakan. Misalnya, pengaturan akses yang tidak tepat, izin file yang tidak benar, atau pengaturan default yang lemah dapat menyebabkan serangan dan penyalahgunaan.

6. IDOR (*Insecure Direct Object References*)

Kerentanan IDOR terjadi ketika aplikasi *web* tidak memvalidasi atau melindungi dengan benar akses langsung terhadap objek, seperti ID unit atau referensi langsung ke file atau data sensitif. Hal ini

memungkinakan penyerang untuk mendapatkan akses tidak sah ke objek atau data yang seharusnya tidak mereka akses.

Acunetix Web Vulnerability Scanner mempunyai beberapa fitur inovatif menjadikannya sebagai pemimpin pasar global (Febri Al Fajar: 2020), yaitu:

1. Pemeriksaan XXS dan injeksi SQL. XSS adalah suatu penyerangan yang menguatkan peretas dalam menjalankan script yang tidak aman di *browser* pengunjung situs *web*. Sedangkan SQL *Ijection* yaitu suatu metode penyerangan yang melakukan perubahan perintah SQL dalam mengakses data pada *database*.
2. Teknologi Acusensor pemindai kerentanan *web acunetix* memiliki fitur Teknologi AcuSensor. Ini adalah teknologi keamanan terbaru yang menguatkan untuk mengidentifiksdi kerentanan yang muncul dan tidak terdeteksi saat pemindaian aplikasi *web* tradisional. Manfaat memakai Teknologi AcuSensor termasuk mempercepat proses menemukan dan memperbaiki kerentanan yang ada, memberikan informasi lebih rinci tentang setiap kerentanan yang terdeteksi, dan menghilangkan kerentanan injeksi SQL yang mengandalkan deteksi pesan kesalahan. Untuk memeriksa masalah konfigurasi aplikasi *web*, termasuk kemampuan untuk memperbaikinya dari *server web*.
3. Pemindai port dan peringatan jaringan acunetix memindai port yaitu melakukan scan pada port yang terbuka di *server web* dan mencari peringatan jaringan.

4. *Legal and Regulatory Compliance Acunetix* (kepatuhan terhadap hukum dan peraturan acunetix) yaitu dapat membuat laporan yang memberi tahu apakah aplikasi *web* sesuai dengan keamanan VISA PCI.
5. Pemindaian kerentanan teknologi Ajax dan *web* 2.0 yaitu biasanya client *Script Client Engine* (mesin penganalisa skrip klien) yang memungkinkan untuk melakukan scan secara menyeluruh pada aplikasi Ajax dan *web* aplikasi 2.0 terbaru dan yang paling kompleks serta menemukan kerentanannya.
6. Uji area pelindungan kata sandi dan formulir *web* acunetix yaitu memiliki *Macro Recording Tool* (alat perekam makro) yang dapat dijalankan untuk menguji area yang dilindungin kata sandi dan formulir *web*.
7. *Google hacking database* (GHDB) *acunetix* yaitu mempunyai memiliki kueri basis data peretasan google. Ini digunakan untuk memeriksa konten situs *web* dan mengidentifikasi data sensitif sebelum "search engine hacker" melakukannya.

Metode kualitatif menggunakan alat berupa perangkat lunak yang umum digunakan dan metode untuk menguji keamanan aplikasi (Sutanta, 2011). Langkah selanjutnya adalah:

1. Fase inisiasi, pada fase ini pencarian dan peninjauan dokumentasi keamanan aplikasi dilakukan.

2. Fase investigasi, penyelidikan ini dilakukan dalam program aplikasi *server web.*
3. Fase pengujian, pada fase ini pengujian keamanan melalui aplikasi yang dilakukan dengan memakai alat yang disebut dengan *Acunetix Web Vulnerability Scanner*, menggunakan metode biasa digunakan untuk menguji aplikasi dan sistem keamanan rahasia.
4. Fase verifikasi, pada fase ini verifikasi keamanan aplikasi dilakukan dan administrator diberitahu untuk melakukan perbaikan berdasarkan peneliatian dan pengujian berbagai aspek *programming edge.*

Berikut merupakan informasi yang terdapat pada *acunetix web vulnerability scanner* dapat dilihat pada tabel 2.2:

Tabel 2.2. *Vulnerability Information* Pada *Acunetix Web Vulnerability Scanner*

Type	*Vulnerability information*
Ctitical	Mengeksekusi kode pada aplikasi *web* atau *server* aplikasi, atau mengakses data sensitif.
High	Mengakses sumber daya dan data aplikasi. Ini dapat memungkinkan penyerang mencuri informasi sesi atau data sensitif dari aplikasi atau *server*.
Medium	Kesalahan dan kekurangan dalam konfigurasi aplikasi. Dengan mengeksploitasi masalah keamanan dapat mengakses informasi sensitif di aplikasi atau server.
Low	Kebocoran informasi, kesalahan konfigurasi, dan kurangnya beberapa langkah keamanan. Memanipulasi orang untuk mengikuti tindakan tertentu atau mengungkapkan informasi penting.
Info	Tidak diklarifikasikan sebagai kerentanan

Acunetix juga mempunyai beberapa istilah konsep-konsep penting yang dapat dipahami antara lain:

1. Durasi

Durasi yaitu waktu yang diperlukan dalam melakukan pemindaian keamanan pada aplikasi *web*. Durasi pemindaian akan bervariasi tergantung pada ukuran dan kompleksitas aplikasi yang sedang dipindai.

2. Request

Pada Acunetix, *request* merujuk pada permintaan yang dibuat oleh alat ini ke aplikasi *web* yang sedang dipindai. Permintaan ini mencakup berbagai tindakan yang dilakukan untuk menguji keamanan aplikasi, seperti mengirimkan data masukan atau menjalankan serangan tertentu.

3. *Average Response Time* (Waktu Rata-rata Respon)

Average Response Time adalah waktu rata-rata yang dibutuhkan oleh aplikasi *web* yang sedang dipindai untuk merespons permintaan dari Acunetix. Waktu respon yang cepat biasanya diinginkan, karena memungkinkan Acunetix untuk mengidentifikasi potensi kerentanan dengan lebih efisien.

4. *Paths Identified* (Jalur-jalur yang Diidentifikasi):

Paths Identified mengacu pada jalur-jalur atau rute-rute spesifik dalam aplikasi *web* yang telah ditemukan dan diidentifikasi oleh Acunetix selama pemindaian. Jalur-jalur ini merupakan daftar dari URL dan parameter-parameter yang bisa diakses oleh alat tersebut selama pengujian.

BAB III

METODE PENELITIAN

3.1 Tempat dan Objek Penelitian

3.1.1 Tempat Penelitian

Lokasi atau tempat penelitian yaitu di luar jaringan Universitas Islam Negeri Ar-Raniry Banda Aceh.

3.1.2 Objek Penelitian

Objek penelitiaan ini adalah sebuah *website* siakad.ar-raniry.ac.id yang berisikan sistem informasi data portal mahasiswa UIN Ar-Raniry Banda Aceh.

3.2 Perangkat

Perangkat-perangkat yang digunakan pada penelitian ini adalah perangkat keras (*Hardware*), dan perangkat lunak (*Software*).

3.2.1 Perangkat Keras

Perangkat keras yang digunakan pada penelitian ini yaitu laptop. Laptop merupakan alat dipakai dalam mendukung pengujian kerentanan keamanan pada *website*. Untuk spesifikasi laptop yang digunakan pada penelitian ini seperti pada tabel 3.1:

Tabel 3.1 *Device Specifications*

Model	Acer Aspire A514-52K
Device name	LAPTOP-KC2ERA8S
Processor	Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz
Installed	RAM 4,00 GB (3,84 GB usable)
Device ID	1DEE9961-0BFD-4F53-BC08-B86F068F297A
Product ID	00327-35162-28318-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini sebagai berikut:

1. Windows 10 versi 21H2.
2. *Acunetix Web Vulnerability Scanner* versi 14.6.
3. Situs *Website* yang di *scanning* adalah siakad.ar-raniry.ac.id.

3.3 Metode Penelitian

Metode adalah teknik, tatacara atau prosedur. Sedangkan penelitian dasarnya merupakan suatu upaya pencarian atau disebut dengan istilah *research* yang artinya kembali (Bambang Suggono, 2007). Metode yang digunakan pada penelitian ini yaitu *vulnerability assessment*. Metode penelitian *vulnerability assessment* adalah suatu proses penilaian keretanan yang digunakan untuk mengidentifikasi, menilai, dan mengukur kerentanan dalam suatu sistem, infrastruktur, atau organisasi terhadap potensi ancaman dan risiko keamanan (Mona Fronita, 2023). Metode ini bertujuan untuk menemukan potensi celah atau titik lemah dalam keamanan sehingga langkah-langkah mitigasi dan perlindungan

yang tepat dapat diambil untuk mencegah atau mengurangi kemungkinan serangan atau pelanggaran keamanan.

3.3.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penyususnan Tugas Akhir ini yaitu jenis penelitian eksperimen. Penelitian eksperimen adalah penelitian yang dilakukan terhadap variabel yang data-datanya belum ada sehingga perlu dilakukan proses manipulasi melalui pemberian *treatment*/perlakuan tertentu terhadap subjek penelitian yang kemudian diamati/diukur dampaknya (data yang akan datang) (Jaedun: 2011). Namun, dalam menganalisis hasil *scanning website* menggunakan penelitian kualitatif. Penelitian kualitatif adalah jenis penelitian yang menghasilkan penemuan-penemuan yang tidak dapat di peroleh dengan menggunakan prosedur-prosedur statistik atau cara-cara lain dari kuantifikasi (Nur Sayidah, 2018).

3.3.2 Pendekatan Penelitian

Adapun yang menjadi pendekatan penelitian pada Tugas Akhir ini adalah pendekatan deskriptif analisis, yaitu dengan suatu metode yang membuat deskripsi, gambaran secara sistematis, faktual, akurat dengan fakta-fakta, sifat-sifat serta hubungan antara fenomena yang di pelajari dan dianalisis sesuai dengan data yang diperoleh (Muhammad Nazir, 1998). Pada penelitian ini, deskriptif analisis mengacu pada hasil *scanning* tingkat keamanan *website* dengan menggunakan *acunetix vulnerability web scanner*.

3.4 Teknik Pengumpulan Data

Teknik yang digunakan pada pengumpulan data dalam penelitian ini adalah:

3.4.1 Teknik Perpustakaan

Teknik perpustakaan adalah mengumpulkan dokumen-dokumen yang berhubungan dengan keamanan *web* yang terdapat pada buku, majalah dan hasil pencarian internet. Teknik ini mengumpulkan literatur dan data yang bersumber dari buku serta jurnal yang berhubungan dengan penelitian ini (Rahmadi: 2011). Kemudian, apabila telah menemukan celah keamanan yang dilakukan pada pengujian mengacu pada langkah-langkah pemecahan masalah yang diperoleh dari jurnal atau buku referensi terkait dengan masalah yang dipecahkan

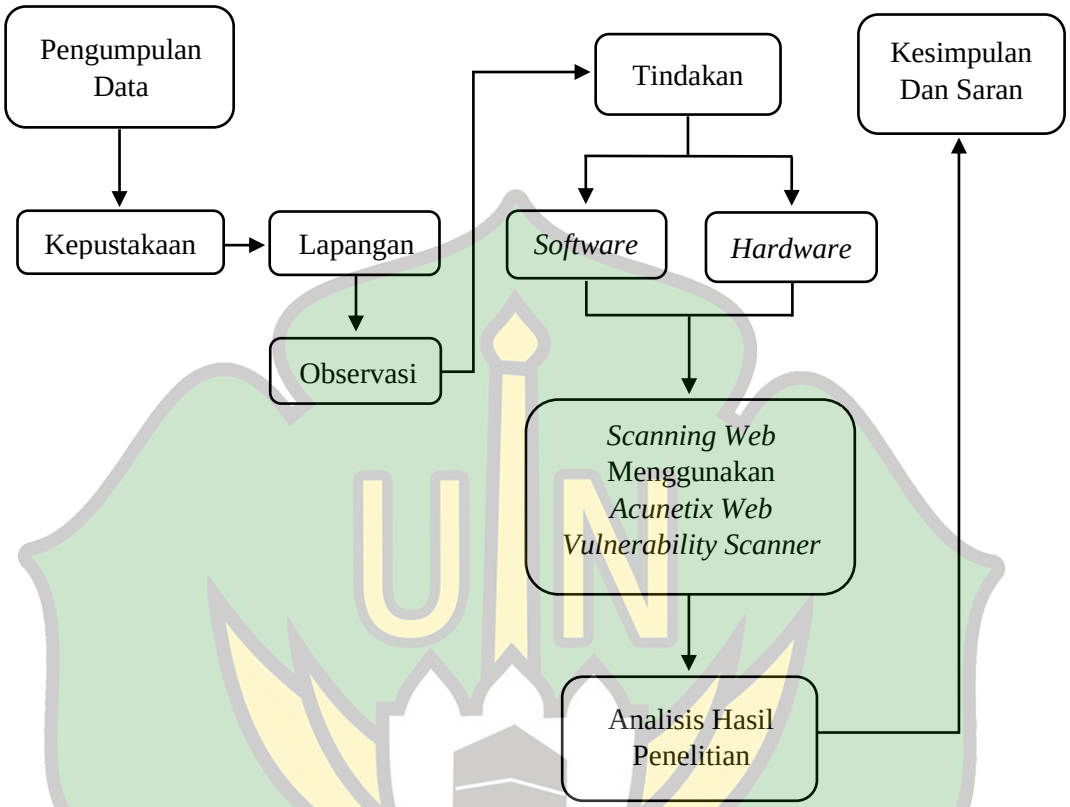
3.4.2 Teknik Observasi

Teknik observasi yaitu dengan mengobservasi atau mengamati perangkat lunak dan jaringan yang digunakan. Teknik observasi dilaksanakan apabila sebuah situs *web* dapat dilakukan serta dicoba sifat atau fitur yang memiliki kemungkinan adanya celah. Teknik yang dilakukan dalam penelitian ini yaitu terdapat pada *website* siakad.ar-raniry.ac.id.

3.5 Analisis Data

Data yang diperoleh dari pengamatan tingkat keamanan situs *web* yang menggunakan pemindaian *acunetix web vulnerability scanner* yang dilakukan pada situs *web* siakad.ar-raniry.ac.id. Kemudian data yang diperoleh dianalisis untuk mengetahui hasil tingkat keamanan yang terdapat pada halaman *website*.

3.6 Alur Berpikir



Gambar 3.1 Alur Berpikir

Sesuai dengan diagram alir penelitian yang terdapat pada Gambar 3.1, penelitian ini berlangsung dalam beberapa tahap:

1. Menghimpun data-data secara sistematis dan terarah untuk memperoleh pemahaman secara mendalam.
2. Mengumpulkan dan meneliti literatur, buku, e-book, artikel untuk mendukung penelitian
3. Mengamati dan mengambil suatu data penelitian secara langsung dari lapangan yang menjadi target penelitian sehingga data yang diperoleh bersifat konkret.

4. Melakukan pengamatan langsung pada terhadap objek yang diteliti dengan mengidentifikasi sumber data yang relevan.
5. Melakukan tindakan penelitian dengan mengumpulkan perangkat yang di perlukan dalam melakukan *scanning web*.
6. Menyiapkan dan mengkonfigurasi perangkat keras (*hardware*) serta perangkat lunak (*software*) yang diperlukan untuk mendukung pelaksanaan penelitian.
7. Mengintervensi untuk melakukan *scanning* pada situs *web* siakad.ar-raniry.ac.id menggunakan pemindai kerentanan *acunetix web vulnerability scanner* untuk mendapatkan informasi tentang keamanan situs.
8. Menganalisis data yang diperoleh dari pengujian dan menganalisis tingkat keamanan *website* siakad.ar-raniry.ac.id serta solusi untuk meminimalisir keretanan.
9. Membuat kesimpulan dari data yang sudah dianalisis pada pengujian *web*, celah keamanan dan rekomendasi yang dapat digunakan untuk mengamankan situs *web*.

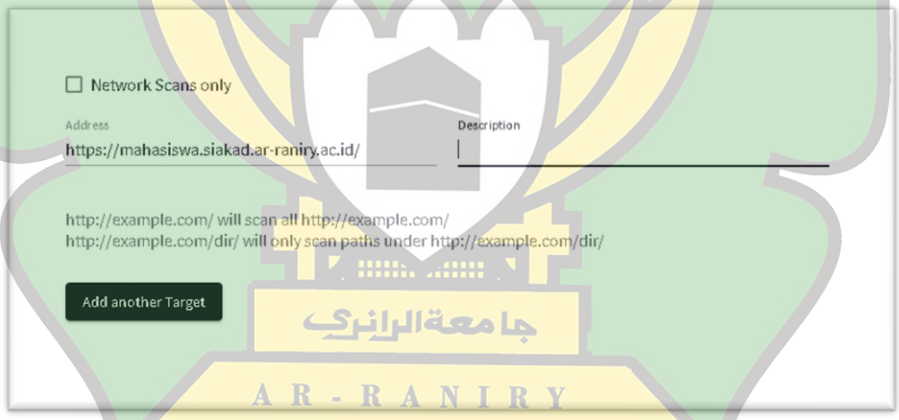
BAB IV

HASIL DAN PEMBAHASAN

Keamanan *website* dianalisis menggunakan *Acunetix Web Vulnerability Scanner* dengan tahapan sebagai berikut:

4.1 *Add Target*

Add target merupakan meng-input dari URL target *address* dan *description* dengan memasukkan *address* yaitu http://mahasiswa.siakad.ar-raniry.ac.id/ seperti yang terlihat pada gambar 4.1:



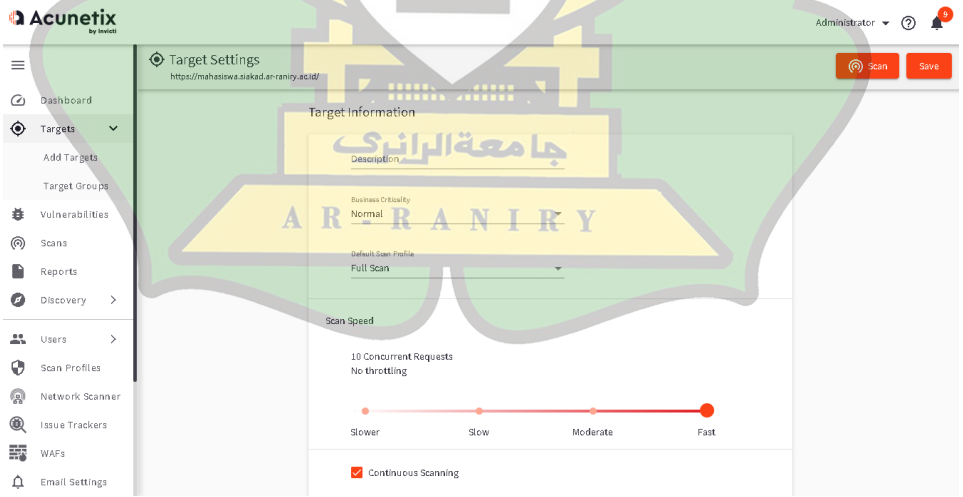
Gambar 4.1 *Add Target*

4.2 *Vulnerability Scanner*

Pada proses pemindaian *web* terdapat *target information* yaitu informasi tentang situs *web* atau aplikasi *web* yang akan dipindai. Hal itu *mencakup URL, parameter, halaman, dan fitur lainnya yang dianalisis keamanannya. Target*

information berfungsi untuk memahami apa yang dipindai dan mengidentifikasi potensi keretanan keamanan yang mungkin ada dalam situs *web* yang dituju.

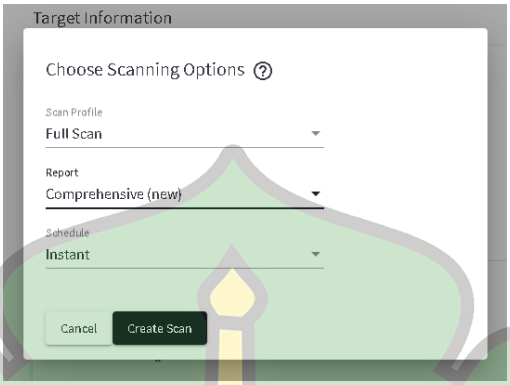
Pada *target information* terdapat *description* yang mengacu pada penjelasan singkat yang diberikan tentang setiap target yang akan diidentifikasi atau dipindai dalam konteks keamanan. *Business criticality* yang merujuk pada sejauh mana informasi yang dicari dianggap kritis. Default scan profile yang mengacu pada pengaturan standar yang digunakan otomatis, pada penelitian ini menggunakan *business criticality* normal. Dan *scan speed* yang mengacu pada tingkat kecepatan saat melakukan proses pemindaian terhadap target informasi. Tingkat kecepatan scan speed yaitu *slower*, *slow*, *modetare*, dan *fast*. Pada penelitian ini menggunakan kecepatan *fast*. Setelah mengatur tingkat kecepatan saat melakukan pemindaian, tahapan selanjutnya dengan mencentang *continuos scanning* dan *save*. Hal ini dilihat pada gambar 4.2:



Gambar 4.2 *Target Settings*

Terdapat 6 tipe scan pada Acunitix yaitu *full scan*, *high risk vulnerability*, *cross-site scripting vulnerability*, *sql injection vulnerability*, *weak password*, dan

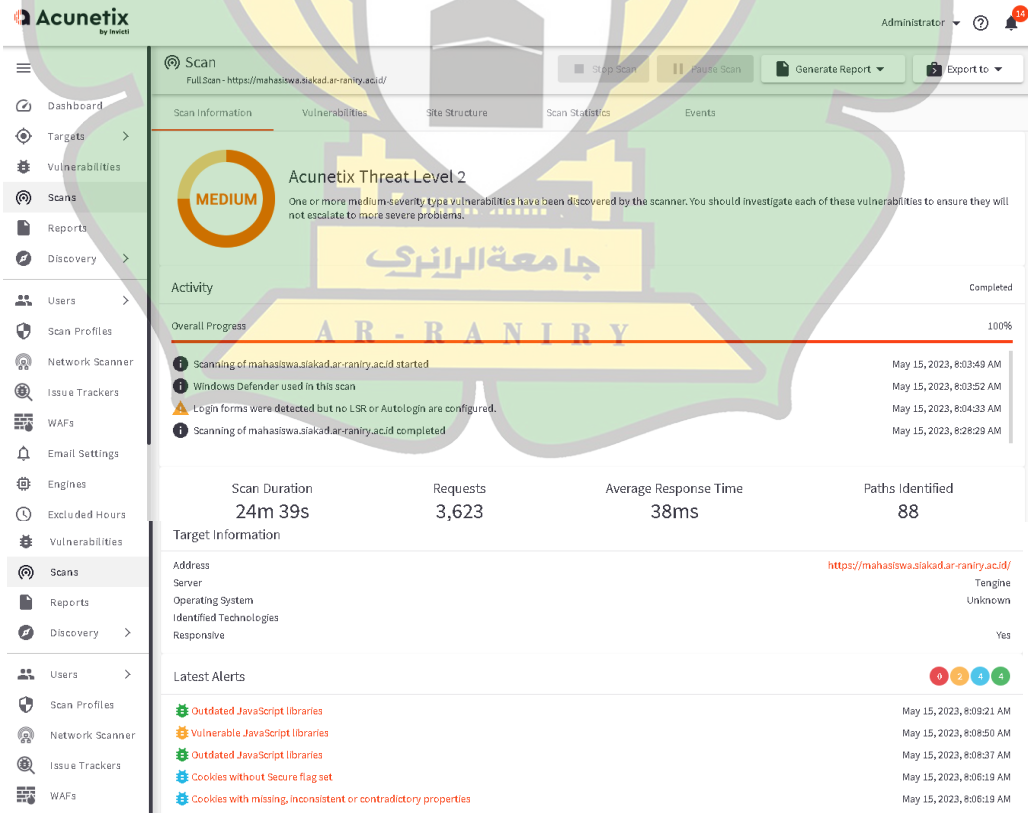
crawl only. Penulis menggunakan tipe acunetix *full scan* untuk mendapatkan hasil kerentanan secara keseluruhan. Kemudian *create scan* seperti pada gambar 4.3:



Gambar 4.3 *Create Scan*

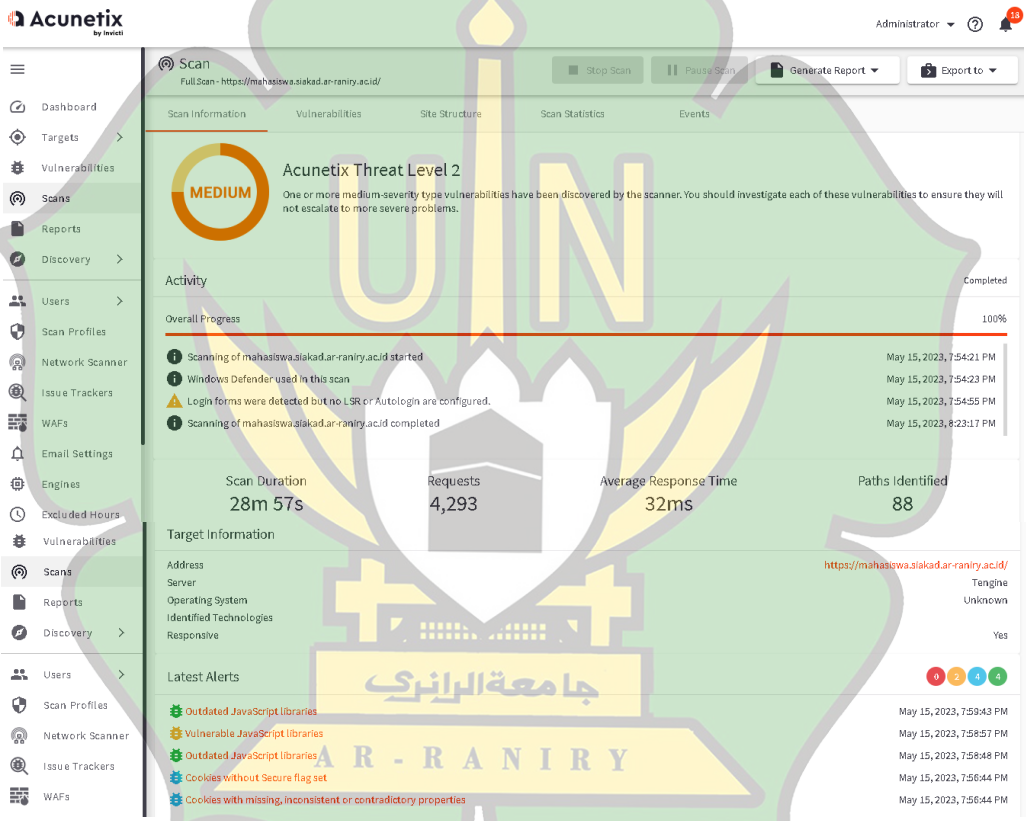
4.3 Hasil *Scanning*

Dari hasil *scanning* atau pengujian yang dilakukan sebanyak 10 kali, 5 kali pada jam kerja dan 5 kali pada luar jam kerja dapat dilihat pada gambar 4.3-4.12:



Gambar 4.4 Hasil *Scanning* 1

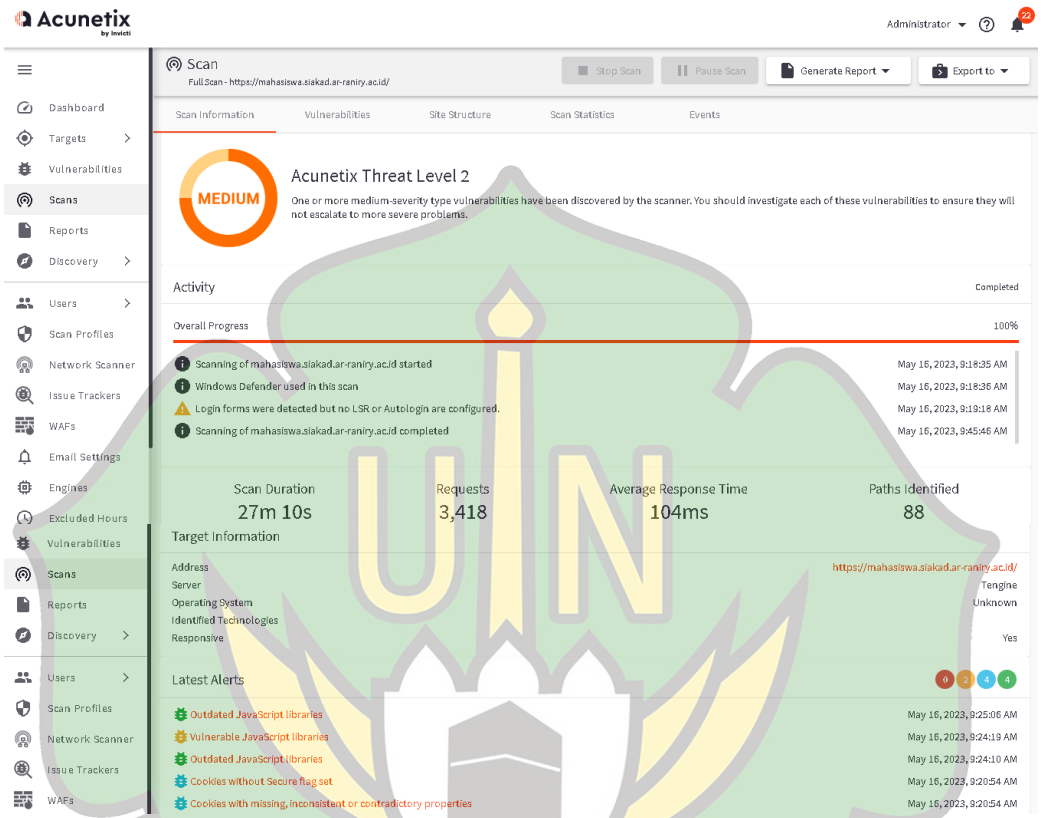
Pada gambar 4.4 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan pada jam kerja tanggal 15 Mei 2023 jam 08:03 WIB dengan durasi *scanning* 24 menit 39 detik, 3.623 permintaan pada *web*, rata-rata waktu respon 38 menit dan identifikasi jalur berjumlah 88.



Gambar 4.5 Hasil *Scanning* 2

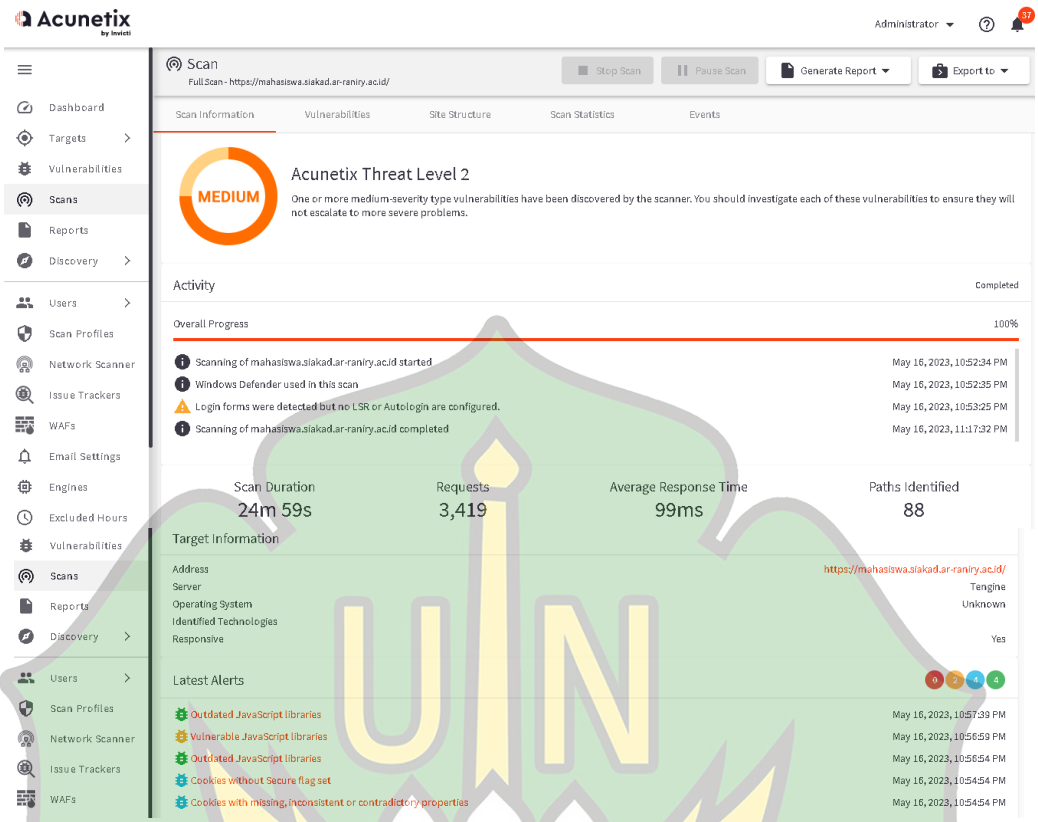
Pada gambar 4.5 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alert*s pada level Low dan 4 *alerts* informational. *Scanning* dilakukan di luar jam kerja pada tanggal 15 Mei 2023 jam 19:58 WIB dengan durasi *scanning* 28 menit

57 detik, 4.293 permintaan pada *web*, rata-rata waktu respon 32 menit dan identifikasi jalur berjumlah 88.



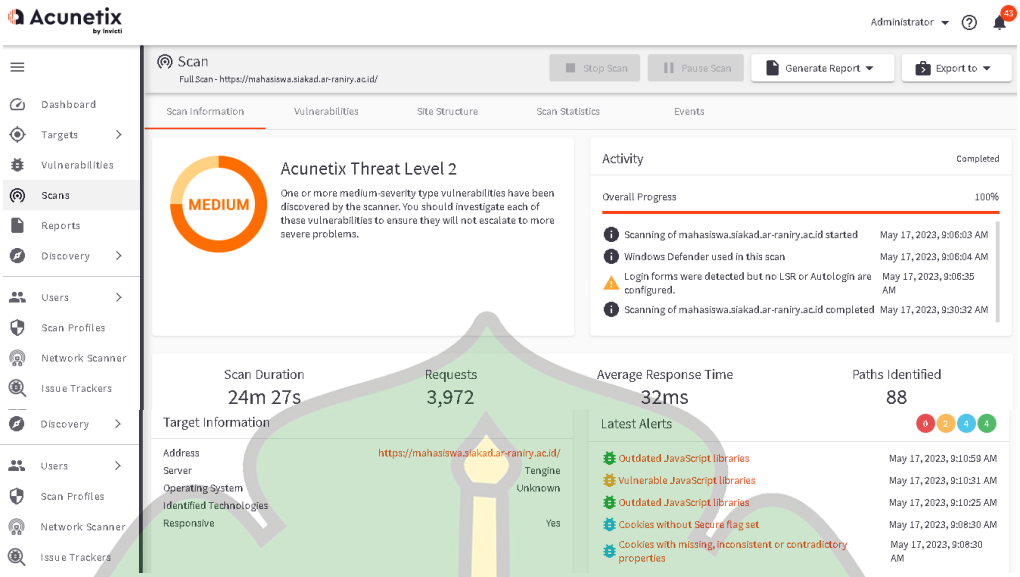
Gambar 4.6 Hasil *Scanning* 3

Pada gambar 4.6 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan pada jam kerja tanggal 16 Mei 2023 jam 09:25 WIB dengan durasi *scanning* 27 menit 10 detik, 3.418 permintaan pada *web*, rata-rata waktu respon 104 menit dan identifikasi jalur berjumlah 88.



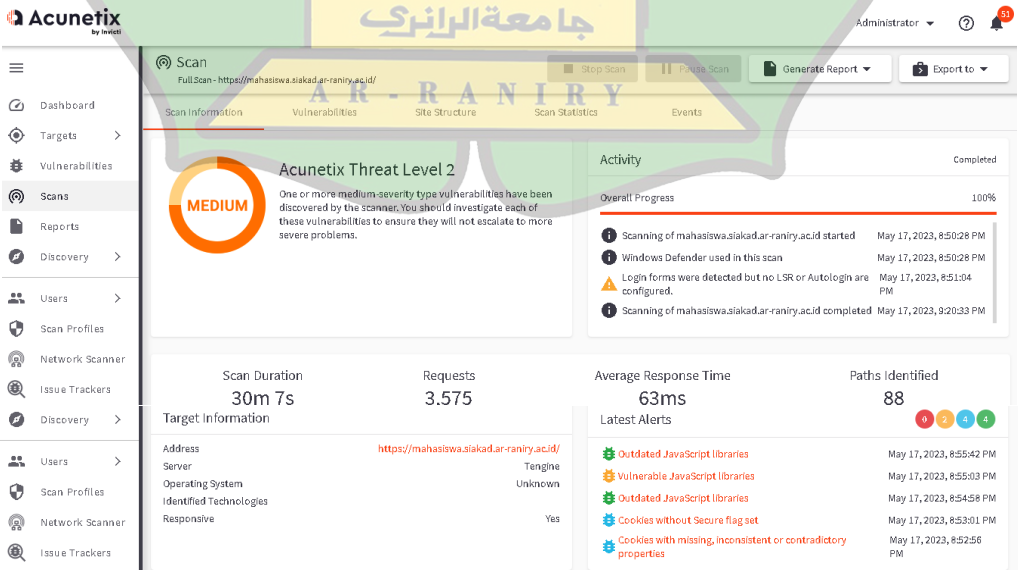
Gambar 4.7 Hasil *Scanning* 4

Pada gambar 4.7 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan di luar jam kerja pada tanggal 16 Mei 2023 jam 10:57 WIB dengan durasi *scanning* 24 menit 59 detik, 3.419 permintaan pada *web*, rata-rata waktu respon 99 menit dan identifikasi jalur berjumlah 88.



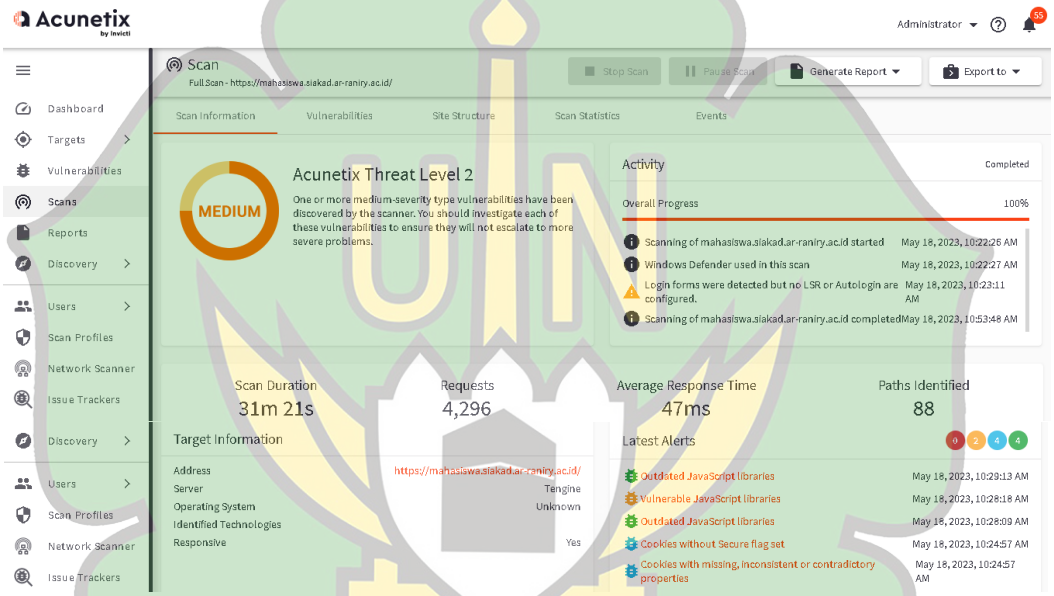
Gambar 4.8 Hasil *Scanning* 5

Pada gambar 4.8 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan pada jam kerja tanggal 17 Mei 2023 jam 09:10 WIB dengan durasi *scanning* 24 menit 27 detik, 3.972 permintaan pada *web*, rata-rata waktu respon 32 menit dan identifikasi jalur berjumlah 88.



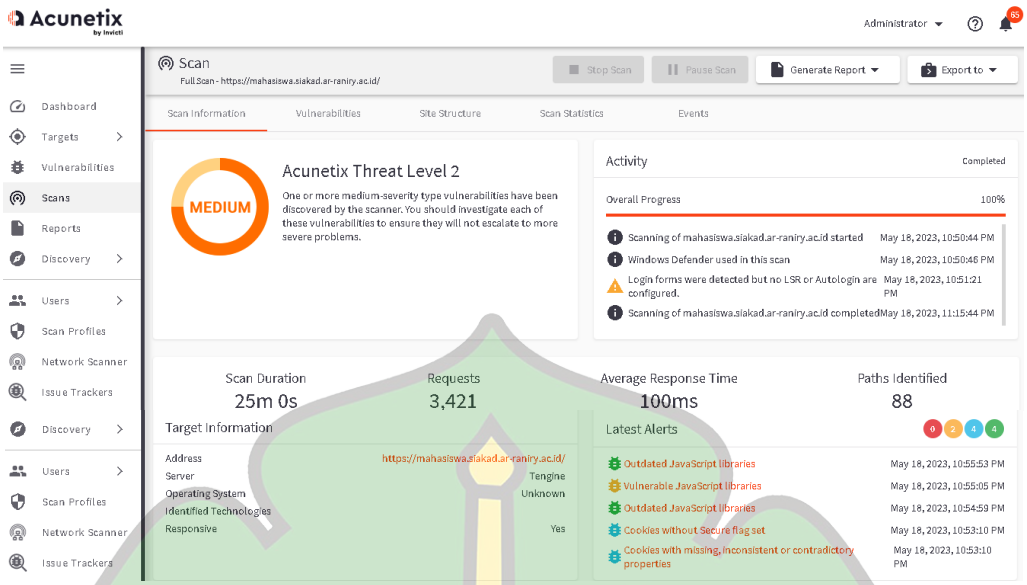
Gambar 4.9 Hasil *Scanning* 6

Pada gambar 4.9 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan di luar jam kerja pada tanggal 17 Mei 2023 jam 20:55 WIB dengan durasi *scanning* 30 menit 07 detik, 3.575 permintaan pada *web*, rata-rata waktu respon 63 menit dan identifikasi jalur berjumlah 88.



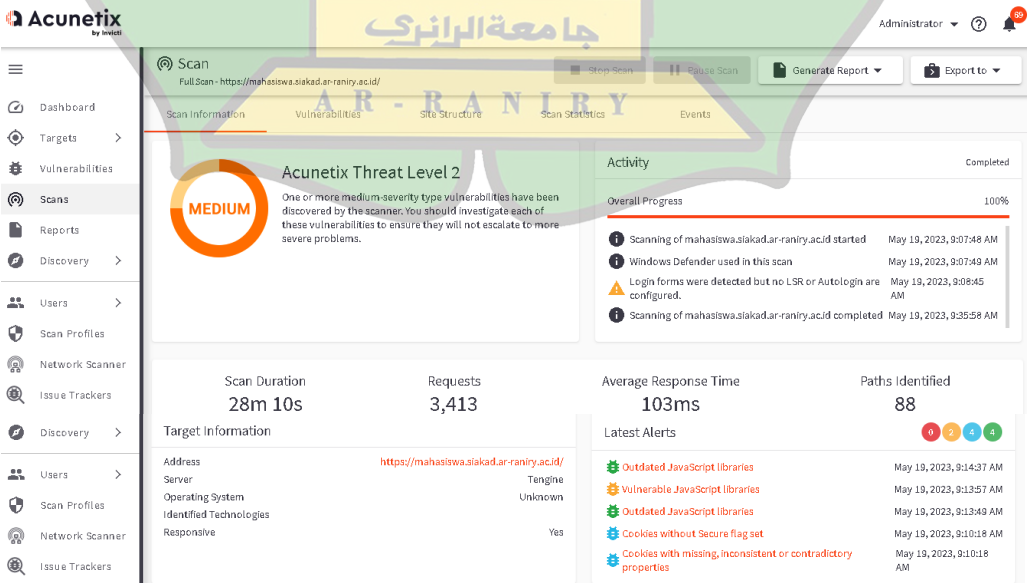
Gambar 4.10 Hasil *Scanning* 7

Pada gambar 4.10 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan pada jam kerja tanggal 18 Mei 2023 jam 10:24 WIB dengan durasi *scanning* 31 menit 21 detik, 4.296 permintaan pada *web*, rata-rata waktu respon 47 menit dan identifikasi jalur berjumlah 88.



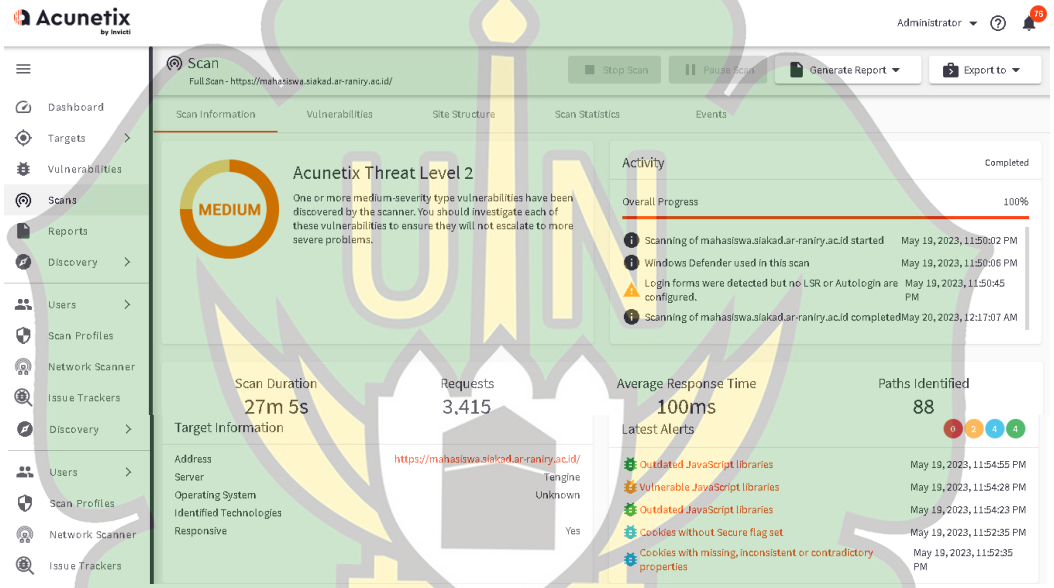
Gambar 4.11 Hasil *Scanning* 8

Pada gambar 4.11 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan di luar jam kerja pada tanggal 18 Mei 2023 jam 22:53 WIB dengan durasi *scanning* 25 menit 0 detik, 3.421 permintaan pada *web*, rata-rata waktu respon 100 menit dan identifikasi jalur berjumlah 88.



Gambar 4.12 Hasil *Scanning* 9

Pada gambar 4.12 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan pada jam kerja tanggal 19 Mei 2023 jam 09:13 WIB dengan durasi *scanning* 28 menit 10 detik, 3.413 permintaan pada *web*, rata-rata waktu respon 103 menit dan identifikasi jalur berjumlah 88.



Gambar 4.13 Hasil *Scanning* 10

Pada gambar 4.13 menunjukkan hasil *scanning website* berada pada level medium dengan rincian 0 *web alerts* level high, 2 *web alerts* level medium, 4 *web alerts* pada level Low dan 4 *alerts* informational. *Scanning* dilakukan di luar jam kerja pada tanggal 19 Mei 2023 jam 23:54 WIB dengan durasi *scanning* 27 menit 05 detik, 3.415 permintaan pada *web*, rata-rata waktu respon 100 menit dan identifikasi jalur berjumlah 88.

4.4 *Result Analysis*

Hasil yang ditemukan merupakan kerentanan yang dapat diukur dari jumlah kerentanan pada *website* siakad UIN Ar-Raniry dan level yang ditemukan oleh Acunetix. Pada Tugas Akhir ini penulis melakukan 10 kali *Scanning* yang dilakukan pada jam kerja dan luar jam kerja selama 5 hari, seperti tabel 4.1 dan 4.2 di bawah ini:

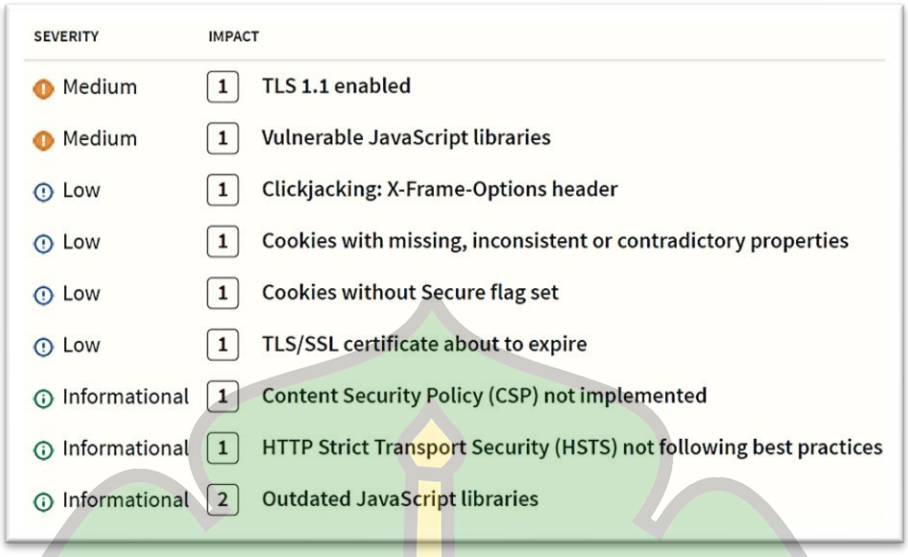
Tabel 4.1 *Scanning* Jam Kerja

Scan	Tanggal	Jam	Durasi	Requests	Average Response Time	Paths Identified	Celah			
							High	Medium	Low	Info
1	15/05/23	08:03	24 m 39s	3,623	38ms	88	0	2	4	4
3	16/05/23	09:18	27m 10s	3,418	104ms	88	0	2	4	4
5	17/05/23	09:06	24m 27s	3,972	32ms	88	0	2	4	4
7	18/05/23	10:22	31m 21s	4,296	47ms	88	0	2	4	4
9	19/05/23	09:07	28m 10s	3,413	103ms	88	0	2	4	4

Tabel 4.2 *Scanning* Luar Jam Kerja

Scan	Tanggal	Jam	Durasi	Requests	Average Response Time	Paths Identified	Celah			
							High	Medium	Low	Info
2	15/05/23	19:54	28m 57s	4,293	32ms	88	0	2	4	4
4	16/05/23	22:52	24m 59s	3,419	99ms	88	0	2	4	4
6	17/05/23	20:50	30m 7s	3,575	63ms	88	0	2	4	4
8	18/05/23	22:50	25m 0s	3,421	100ms	88	0	2	4	4
10	19/05/23	23:50	27m 5s	3,415	100ms	88	0	2	4	4

Berdasarkan tabel 4.1 dan 4.2 dari penelitian yang telah penulis lakukan, terdapat beberapa keretanan seperti terlihat pada gambar 4.14.



SEVERITY	IMPACT	
Medium	1	TLS 1.1 enabled
Medium	1	Vulnerable JavaScript libraries
Low	1	Clickjacking: X-Frame-Options header
Low	1	Cookies with missing, inconsistent or contradictory properties
Low	1	Cookies without Secure flag set
Low	1	TLS/SSL certificate about to expire
Informational	1	Content Security Policy (CSP) not implemented
Informational	1	HTTP Strict Transport Security (HSTS) not following best practices
Informational	2	Outdated JavaScript libraries

Gambar 4.14 *Vulnerabiliy* Hasil *Scanning*

Pada gambar 4.14 menunjukkan bahwa terdapat keretanan 2 medium, 4 low dan 4 informational. Maka berikut penjelasan kerentanan-kerentanan tersebut:

a. TLS 1.1 *Enabled*

Server *website* mendukung enkripsi melalui TLS 1.1, yang secara resmi tidak digunakan lagi pada Maret 2021 sebagai hasil dari masalah keamanan bawaan. Saat mengincar Standar Keamanan Data PCI (*Payment Card Industry*). DSS (*Decision Support System*) adalah sistem pendukung keputusan dalam berbagai situasi dan konteks bisnis, disarankan untuk menggunakan TLS 1.2 atau lebih tinggi. Menurut PCI, "30 Juni 2018 adalah batas waktu untuk menonaktifkan SSL/TLS awal dan menerapkan protokol enkripsi yang lebih aman untuk memenuhi Standar Keamanan Data PCI (PCI DSS) untuk pengamanan data pembayaran.

Penyerang mungkin dapat mengeksploitasi masalah ini untuk melakukan serangan *man-in-the-middle* dan mendekripsi komunikasi antara layanan yang terpengaruh. Disarankan untuk menonaktifkan TLS 1.1 dan menggantinya dengan TLS 1.2 atau lebih tinggi. Menonaktifkan TLS 1.1 dan menggantinya dengan TLS 1.2 bertujuan untuk meningkatkan keamanan komunikasi data. TLS (*Transport Layer Security*) adalah protokol keamanan yang digunakan untuk melindungi informasi yang dikirim melalui jaringan, seperti data pribadi, login, atau transaksi online.

TLS 1.1 memiliki beberapa kerentanan keamanan yang ditemukan dalam implementasinya, dan TLS 1.2 adalah versi yang lebih kuat dan aman. Dengan menggunakan TLS 1.2, ada peningkatan keamanan yang signifikan, termasuk algoritma enkripsi yang lebih kuat dan perlindungan terhadap serangan keamanan yang lebih baru. Oleh karena itu, dengan beralih dari TLS 1.1 ke TLS 1.2, organisasi dan penyedia layanan dapat meningkatkan perlindungan data mereka terhadap serangan dan menjaga komunikasi mereka lebih aman dan terjamin.

b. *Vulnerable Javascript Libraries*

Website siakad UIN Ar-Raniry menggunakan satu atau beberapa pustaka *JavaScript* yang rentan. Satu atau lebih kerentanan dilaporkan versi perpustakaan. Oleh karena itu, *website* siakad UIN Ar-Raniry harus mengkonsultasikan detail Serangan dan referensi *website* untuk informasi

lebih lanjut tentang perpustakaan yang terpengaruh dan kerentanan yang dilaporkan serta disarankan untuk meningkatkan ke versi terbaru.

Meningkatkan versi rentan dari pustaka *JavaScript* ke versi terbaru bertujuan untuk meningkatkan keamanan dan stabilitas aplikasi atau situs *web*. Pustaka *JavaScript* yang rentan sering kali memiliki kerentanan keamanan yang telah ditemukan dan diperbaiki dalam versi terbaru. Dengan memperbaruinya dapat mengatasi kerentanan tersebut dan mengurangi risiko serangan yang mungkin dieksploitasi oleh penyerang. Selain itu, versi terbaru pustaka *JavaScript* juga mengatasi perbaikan bug, peningkatan kinerja, fitur baru, dan pembaruan lainnya.

c. Clickjacking: X-Frame-Options Header

Clickjacking adalah teknik berbahaya menipu pengguna *website* agar mengklik sesuatu yang berbeda dari yang dirasakan pengguna yang mereka klik, sehingga berpotensi mengungkap informasi rahasia atau mengendalikan komputer mereka saat mengklik halaman *website* yang tampaknya tidak berbahaya. *Server* tidak mengembalikan header X-Frame-Options dengan nilai DENY atau SAMEORIGIN, yang artinya bahwa situs *website* ini dapat berisiko terkena serangan clickjacking. Header respons HTTP X-Frame-Options dapat digunakan untuk menunjukkan apakah *browser* harus diizinkan atau tidak untuk merender halaman di dalam bingkai atau *iframe*. Situs dapat menggunakan ini untuk menghindari serangan clickjacking, dengan memastikan bahwa kontennya

tidak disematkan ke dalamnya situs tidak terpercaya. Dampaknya tergantung pada aplikasi *website* yang terpengaruh.

Konfigurasikan server *website* siakad UIN Ar-Raniry untuk menyertakan header X-Frame-Options dan header CSP dengan frame-ancestors pengarahan. Disarankan untuk mengkonsultasikan referensi *website* untuk informasi lebih lanjut tentang kemungkinan nilai untuk header ini.

Berikut langkah-langkah untuk terhindar dari clickjacking X-Frame-Options header:

1) Memeriksa apakah *server web* mendukung pengaturan header X-Frame-Options. Memastikan bahwa *server web* dapat mengirimkan header tersebut. Jika menggunakan platform atau framework tertentu, carilah dokumentasi resmi yang menjelaskan cara mengonfigurasi header ini.
2) Tentukan nilai yang sesuai untuk header X-Frame-Options. Pilihan umum adalah "DENY" atau "SAMEORIGIN". "DENY" akan mencegah *website* dimuat dalam frame apa pun, sedangkan "SAMEORIGIN" membatasi pembebanan frame hanya dari domain yang sama.
3) Konfigurasikan *server web* untuk mengirimkan header X-Frame-Options dengan nilai yang telah tentukan. Ini bisa dilakukan melalui pengaturan *server* atau melalui konfigurasi

di dalam kode aplikasi, tergantung pada platform yang gunakan.

4) Uji dan verifikasi pengaturan header. Setelah mengatur header X-Frame-Options, pastikan untuk menguji aplikasi *web* dan memastikan bahwa pengaturan tersebut berfungsi seperti yang diharapkan. Selain itu juga dapat menggunakan alat pengujian clickjacking online yang tersedia untuk memverifikasi keamanan situs *web*.

Selain X-Frame-Options, ada juga beberapa langkah lain yang dapat dilakukan untuk meningkatkan keamanan dan melindungi dari clickjacking, seperti:

1) Menggunakan CSP (*Content Security Policy*) untuk mengontrol sumber daya yang diizinkan dalam halaman.
2) Menggunakan frame-busting *JavaScript* untuk memastikan bahwa situs tidak dimuat dalam frame yang tidak diinginkan.
3) Memantau secara rutin kerentanan keamanan dan memperbarui perangkat lunak serta pustaka yang digunakan dalam situs *web*.

d. *Cookies with Missing, Inconsistent or Contradictory Properties*

Setidaknya salah satu properti *cookie* berikut menyebabkan *cookie* menjadi tidak valid atau tidak kompatibel dengan keduanya properti berbeda dari *cookie* yang sama, dengan lingkungan tempat *cookie* digunakan. Meskipun ini bukan kerentanan itu sendiri,

kemungkinan akan menyebabkan perilaku tak terduga oleh aplikasi, yang pada gilirannya dapat menyebabkan masalah keamanan sekunder. *Cookie* tidak akan disimpan, atau dikirimkan, oleh *browser website*. Pastikan konfigurasi *cookie* sesuai dengan standar yang berlaku.

Langkah-langkah untuk mengkonfigurasi *cookie* sesuai dengan standar yang berlaku sebagai berikut:

1) Setel atribut *Secure*. Jika situs *web* menggunakan protokol HTTPS, pastikan untuk mengatur atribut *Secure* pada *cookie*. Dengan melakukan ini, *cookie* hanya akan dikirimkan melalui koneksi aman, sehingga melindungi data pengguna dari serangan pemantauan atau pencurian melalui jaringan.
2) Setel atribut HttpOnly. Mengatur atribut HttpOnly pada *cookie* akan mencegah akses *JavaScript* ke *cookie* tersebut. Hal ini membantu melindungi *cookie* dari serangan CSS (*Cross-Site Scripting*) di mana penyerang mencoba mencuri informasi pengguna melalui kode *JavaScript* yang tidak sah.
3) Atur masa berlaku (*expiration*) yang sesuai. Tentukan waktu berakhirnya *cookie* dengan benar. Jika ingin *cookie* berlaku hanya selama sesi browsing, atur masa berlaku menjadi nol atau biarkan kosong. Jika ingin *cookie* tetap bertahan untuk jangka waktu tertentu, tetapkan nilai waktu yang sesuai dalam *cookie*.

4) Gunakan pengaturan SameSite. Pengaturan SameSite pada *cookie* membantu mencegah serangan CSRF (*Cross-Site Request Forgery*) dengan membatasi pengiriman *cookie* pada permintaan yang berasal dari domain yang sama. Selain itu juga dapat mengatur nilainya menjadi "Strict" untuk mencegah pengiriman *cookie* pada permintaan dari luar domain, atau "Lax" untuk membatasi pengiriman pada permintaan melalui tautan eksternal.
5) Gunakan kebijakan *cookie* yang sesuai. Beberapa negara atau wilayah memiliki persyaratan hukum tertentu terkait penggunaan *cookie*. Pastikan untuk mematuhi kebijakan privasi dan peraturan yang berlaku di yurisdiksi saat mengonfigurasi *cookie*.
6) Lakukan tes dan validasi. Setelah mengatur *cookie*, lakukan pengujian dan validasi untuk memastikan bahwa konfigurasi tersebut berfungsi seperti yang diharapkan. Periksa header *cookie* yang dikirim oleh *server web*. Pastikan bahwa atribut dan pengaturan yang ditentukan diterapkan dengan benar.

e. *Cookies Without Secure Flag Set*

Satu atau lebih *cookie* tidak memiliki *flag* Aman. Saat *cookie* disetel dengan bendera aman, itu menginstruksikan *browser* bahwa *cookie* hanya dapat diakses melalui saluran SSL/TLS yang aman. Ini adalah sebuah perlindungan keamanan penting untuk *cookie* sesi. *Cookie* dapat

dikirim melalui saluran yang tidak terenkripsi. Jika memungkinkan, siakad UIN Ar-Raniry harus menyetel tanda aman untuk *cookie* ini.

Langkah-langkah menyetel tanda aman (*Secure flag*) untuk *cookie* sebagai berikut:

1) Identifikasi *cookie* yang tidak memiliki tanda aman, periksa kode sumber situs *web* dan temukan semua pengaturan *cookie* yang tidak memiliki tanda aman (*Secure flag*) yang diinginkan. Dan perlu menentukan *cookie* mana untuk diberi tanda aman.
2) Atur tanda aman pada *cookie* untuk setiap *cookie* yang ingin diberi tanda aman, memastikan bahwa *cookie* tersebut diatur hanya saat menggunakan protokol HTTPS. Misalnya, jika menggunakan *JavaScript* untuk mengatur *cookie*, memastikan untuk menambahkan pengaturan atribut `{secure: true}` saat membuat atau mengubah *cookie*. Ini akan memastikan bahwa *cookie* hanya akan dikirim melalui koneksi aman.
3) Tes dan validasi, setelah mengatur tanda aman pada *cookie*, lakukan pengujian untuk memastikan bahwa pengaturan ini berfungsi dengan baik. Pastikan bahwa *cookie* yang diberi tanda aman hanya dikirimkan melalui koneksi HTTPS dan tidak dikirimkan melalui koneksi HTTP.

Namun, pengaturan tanda aman hanya efektif saat situs *web* diakses melalui protokol HTTPS. Pastikan bahwa situs *web* diakses melalui koneksi aman dengan sertifikat SSL yang valid agar *cookie* yang diberi tanda aman dapat berfungsi sebagaimana mestinya. Selain itu, melakukan tes dan validasi secara menyeluruh untuk memastikan bahwa pengaturan *cookie* dan tanda aman telah diterapkan dengan benar, dan penggunaan *cookie* sesuai dengan persyaratan keamanan dan privasi yang berlaku.

f. TLS/SSL *Certificate About To Expire*

Salah satu sertifikat TLS/SSL yang digunakan oleh *server* akan kedaluwarsa. Setelah sertifikat kedaluwarsa, sebagian besar *browser website* akan memberikan peringatan keamanan kepada pengguna akhir, bertanya mereka untuk mengonfirmasi keaslian rantai sertifikat secara manual. Perangkat lunak atau sistem otomatis mungkin diam-diam menolak untuk terhubung ke *server*. Lansiran ini tidak selalu disebabkan oleh sertifikat *server* (daun), tetapi mungkin dipicu oleh sertifikat perantara. Silakan merujuk ke nomor seri sertifikat di detail peringatan untuk mengidentifikasi sertifikat yang terpengaruh. Jika *server* aplikasi mendeteksi sertifikat kedaluwarsa dengan sistem yang berkomunikasi dengannya, maka *server* aplikasi dapat terus memproses data seolah-olah tidak terjadi apa-apa, atau sambungan mungkin dihentikan secara tiba-tiba. Disarankan untuk menghubungi Otoritas Sertifikat untuk memperbarui

sertifikat SSL. Untuk menghindari sertifikat TLS/SSL yang kadarluarsa, berikut adalah beberapa langkah yang dapat diambil:

1) Perencanaan pembaruan penting untuk memiliki perencanaan yang baik untuk memperbarui sertifikat TLS/SSL sebelum tanggal kedaluwarsa. Jangan menunggu hingga sertifikat benar-benar kedaluwarsa sebelum memperbarui, karena ini dapat menyebabkan gangguan layanan. Tandai tanggal kedaluwarsa sertifikat di kalender atau gunakan pengingat untuk memastikan bahwa mengambil tindakan tepat waktu.
2) Pemantauan otomatis pertimbangkan untuk menggunakan alat pemantauan otomatis yang dapat memberi tahu ketika sertifikat mendekati tanggal kedaluwarsa. Terdapat berbagai layanan pemantauan sertifikat yang dapat memberikan peringatan melalui email atau pemberitahuan lainnya ketika sertifikat hampir kadarluarsa.
3) Perpanjangan otomatis untuk menghindari kekhawatiran tentang sertifikat yang kadarluarsa, pertimbangkan menggunakan fitur perpanjangan otomatis yang ditawarkan oleh penyedia sertifikat atau platform hosting. Ini memungkinkan sertifikat diperbarui secara otomatis sebelum tanggal kedaluwarsa tanpa intervensi manual.
4) Kalender pembaruan sertifikat. Membuat kalender atau daftar pembaruan sertifikat sebagai bagian dari proses pengelolaan

dan pemeliharaan situs *web* atau sistem. Dalam daftar ini, catat tanggal kedaluwarsa sertifikat dan batalkan pengingat sebelum tanggal tersebut untuk memastikan pembaruan tepat waktu.

5) Kerjasama dengan penyedia sertifikat. Jika mengalami kesulitan dalam memperbarui sertifikat TLS/SSL, jangan ragu untuk menghubungi penyedia sertifikat. Mereka dapat memberikan panduan dan bantuan yang diperlukan dalam proses perpanjangan atau pembaruan sertifikat.

g. CSP (*Content Security Policy*) *Not Implemented*

CSP adalah lapisan keamanan tambahan yang membantu mendeteksi dan memitigasi jenis tertentu serangan, termasuk XXS (*Cross Site Scripting*) dan serangan injeksi data. CSP dapat diimplementasikan dengan menambahkan header Content-Security-Policy. Nilai header ini adalah string yang berisi arahan kebijakan yang menjelaskan kebijakan keamanan konten *website* siakad UIN Ar-Raniry. Menerapkan CSP harus menentukan daftar asal yang diizinkan untuk semua jenis sumber daya yang dimiliki situs siakad UIN Ar-Raniry memanfaatkan. Misalnya, jika memiliki situs sederhana yang perlu memuat *skrip*, *stylesheet*, dan gambar yang dihosting secara lokal, serta dari *library* jQuery dari CDN mereka, header CSP dapat terlihat seperti berikut:

Content - Security - Policy :
Default - src ' self ' ;
Script - src ' self ' https : // code . jquery . com ;

Terdeteksi bahwa aplikasi *website* siakad UIN Ar-Raniry tidak menerapkan CSP sebagai CSP tajuk hilang dari respon. Direkomendasikan untuk mengimplementasikan CSP (Content Security Policy) ke dalam aplikasi *website* siakad UIN Ar-Raniry. CSP dapat digunakan untuk mencegah dan/atau mengurangi serangan yang melibatkan injeksi konten/kode, seperti lintas situs *scripting*/serangan XSS, serangan yang memerlukan penyematan sumber daya berbahaya, serangan yang melibatkan berbahaya penggunaan *iframe*, seperti serangan clickjacking, dan lain-lain.

Sebaiknya terapkan CSP ke dalam aplikasi *website* siakad UIN Ar-Raniry. Mengkonfigurasi kebijakan keamanan konten melibatkan penambahan header HTTP Kebijakan Keamanan Konten ke halaman *website* dan memberinya nilai untuk mengontrol sumber daya yang diizinkan untuk dimuat oleh agen pengguna untuk halaman itu. Untuk mengkonfigurasi kebijakan konten CSP pada halaman *website*, berikut adalah beberapa langkah yang dapat ikuti:

1) Tentukan kebijakan konten yang diinginkan. Identifikasi kebijakan konten yang sesuai dengan kebutuhan dan tujuan situs *web*. Kebijakan konten dapat digunakan untuk mengontrol sumber daya yang diizinkan untuk dimuat di halaman *web*, seperti skrip *JavaScript*, CSS, gambar, font, dan lainnya. menentukan sumber daya yang diizinkan berdasarkan domain, tipe, atau sumber asal.

2) Tambahkan kebijakan konten ke header HTTP. Membuka file konfigurasi *server* atau berkas sumber halaman *web*, dan cari bagian yang berkaitan dengan header HTTP. perlu menambahkan kebijakan konten sebagai header ke respons HTTP. Header yang relevan untuk kebijakan konten adalah `Content-Security-Policy` atau `Content-Security-Policy-Report-Only`. Pertimbangkan untuk menggunakan mode "Report-Only" terlebih dahulu untuk memantau pelanggaran kebijakan sebelum menerapkan mode yang ketat.
3) Atur nilai kebijakan konten. Menentukan nilai kebijakan konten sesuai kebutuhan situs *web*. Contoh pengaturan kebijakan konten yang umum adalah:
   - `default-src: Menentukan sumber daya yang diizinkan secara umum.
   - `script-src`: Menentukan sumber daya skrip *JavaScript* yang diizinkan.
   - `style-src`: Menentukan sumber daya CSS yang diizinkan.
   - `img-src`: Menentukan sumber daya gambar yang diizinkan.
   - `font-src`: Menentukan sumber daya font yang diizinkan.
   - `frame-src`: Menentukan sumber daya *frame* atau *iframe* yang diizinkan.

- `connect-src`: Menentukan sumber daya yang diizinkan untuk koneksi jaringan.

4) Uji dan validasi. Setelah mengatur kebijakan konten, pastikan untuk menguji situs *web* secara menyeluruh untuk memastikan bahwa kebijakan konten berfungsi sebagaimana mestinya. Periksa konsol pengembang di-*browser* untuk memantau adanya pelanggaran kebijakan yang perlu diperbaiki.

Namun, konfigurasi kebijakan konten dapat berbeda-beda tergantung pada platform atau framework yang digunakan. Pastikan untuk merujuk ke dokumentasi resmi platform atau framework yang gunakan untuk petunjuk yang lebih spesifik.

h. HSTS (HTTP *Strict Transport Security*) *Not Following Best Practices*

HSTS menginstruksikan browser *website* untuk hanya terhubung ke situs *website* menggunakan HTTPS. Terdeteksi bahwa implementasi HSTS aplikasi *website* siakad UIN Ar-Raniry tidak seketat biasanya disarankan. HSTS dapat digunakan untuk mencegah dan/atau mengurangi beberapa jenis serangan MitM (*man-in-the-middle*). Sebaiknya terapkan praktik terbaik HSTS di *website* siakad UIN Ar-Raniry. Disarankan untuk konsultasikan referensi web untuk informasi lebih lanjut. Berikut adalah beberapa manfaat dan tujuan yang dapat dicapai dengan meningkatkan HSTS:

1) Penegasan penggunaan HTTPS. HSTS membantu memastikan bahwa semua koneksi ke situs *web* dilakukan melalui protokol

HTTPS yang aman. Dengan meningkatkan HSTS dapat memperkuat penegasan ini dan menghindari penggunaan HTTP yang tidak aman.

2) Menghindari serangan *downgrade*. Meningkatkan HSTS ke tingkat yang lebih baru membantu mencegah serangan *downgrade*, di mana penyerang mencoba memaksa koneksi ke versi yang tidak aman atau menggunakan protokol HTTP yang rentan. Dengan HSTS yang diperbarui, *browser* akan memaksa koneksi melalui HTTPS dan tidak memperbolehkan *downgrade* yang tidak sah.
3) Peningkatan kepatuhan. Mengikuti praktik terbaik dan menggunakan versi HSTS yang lebih baru membantu memenuhi standar kepatuhan yang berlaku, seperti kepatuhan PCI DSS (*Payment Card Industry Data Security Standard*) atau kepatuhan industri lainnya yang mengharuskan penggunaan HSTS yang aman.
4) Pencegahan serangan MITM (*man-in-the-middle*). Meningkatkan HSTS ke tingkat yang lebih baru dapat membantu mencegah serangan MITM (*man-in-the-middle*), di mana penyerang mencoba memperoleh atau memodifikasi data sensitif dalam komunikasi. HSTS yang diperbarui dapat memastikan bahwa koneksi hanya dilakukan dengan server yang memiliki sertifikat SSL yang valid.

5) Peningkatan kepercayaan pengguna. Dengan memastikan bahwa koneksi ke situs *web* selalu dilakukan melalui HTTPS yang aman, dapat meningkatkan kepercayaan pengguna. Pengguna akan melihat tanda penguncian atau indikator keamanan di *browser*, memberikan keyakinan bahwa informasi terlindungi saat berinteraksi dengan situs.

6) Pemantauan dan pembaruan. Dengan meningkatkan HSTS ke tingkat yang lebih baru dapat mengikuti perkembangan terbaru dalam keamanan *web* dan memanfaatkan fitur-fitur baru yang ditawarkan. memastikan untuk memantau pembaruan dan revisi yang dikeluarkan oleh standar HSTS untuk menjaga situs *web* tetap aman dan sejalan dengan praktik terbaik.

i. *Outdated* Javascript *Libraries*

Siakad UIN Ar-Raniry menggunakan versi lama dari satu atau beberapa pustaka JavaScript. Tersedia versi yang lebih baru. Meskipun versi Anda tidak ditemukan terpengaruh oleh kerentanan keamanan apapun, disarankan untuk melakukannya menjaga perpustakaan tetap *up to date*. Dan juga disarankan *website* siakad UIN Ar-Raniry untuk meningkatkan ke versi terbaru. Berikut adalah langkah-langkah umum untuk meningkatkan *library JavaScript* yang sudah lama ke versi terbaru:

1) Identifikasi *library* yang perlu ditingkatkan. Meninjau dependensi *JavaScript* yang digunakan dalam aplikasi *web* dan tentukan *library* mana yang sudah lama dan perlu ditingkatkan

ke versi terbaru. Periksa dokumentasi *library* dan sumber daya pengembang untuk mengetahui versi terbaru yang tersedia.

2) Pelajari perubahan dan pembaruan. Mekakukan riset tentang perubahan dan pembaruan yang ada di versi terbaru *library JavaScript*. Periksa catatan rilis (*changelog*) untuk mengetahui perbaikan bug, peningkatan keamanan, peningkatan kinerja, dan fitur baru yang ditawarkan oleh versi terbaru.
3) Buat rencana pembaruan. Membuat rencana untuk meningkatkan *library JavaScript* secara bertahap. Tentukan urutan pembaruan berdasarkan ketergantungan dan kemungkinan dampak pada kode yang ada. Juga, pastikan untuk menguji pembaruan di lingkungan pengembangan sebelum mengimplementasikannya secara langsung di lingkungan produksi.
4) Periksa kompatibilitas. Memeriksa apakah versi terbaru *library JavaScript* yang akan tingkatkan kompatibel dengan versi framework atau *library* lain yang digunakan dalam aplikasi *web*. Pastikan tidak ada konflik atau masalah kompatibilitas yang mungkin muncul.
5) Buat cadangan dan uji. Sebelum melakukan pembaruan, pastikan untuk membuat cadangan (*backup*) kode dan data aplikasi *web*. Selain itu, lakukan pengujian menyeluruh setelah meningkatkan *library JavaScript* untuk memastikan bahwa

aplikasi berjalan dengan baik dan tidak ada masalah atau bug yang muncul.

6) Terapkan pembaruan secara bertahap. Tingkatkan *library JavaScript* satu per satu atau dalam kelompok kecil untuk menghindari risiko perubahan yang terlalu drastis pada aplikasi *web*. Lakukan uji coba setelah setiap pembaruan untuk memverifikasi bahwa aplikasi masih berfungsi seperti seharusnya.
7) Perbarui kode aplikasi. Setelah memperbarui *library JavaScript,* periksa dan perbarui bagian kode yang menggunakan fungsi, metode, atau API yang berubah atau dihapus dalam versi terbaru. Pastikan memahami perubahan tersebut dan mengadopsi sintaks atau penggunaan yang baru sesuai dengan dokumentasi *library* yang ditingkatkan.
8) Uji kembali dan rilis. Setelah selesai meningkatkan semua *library JavaScript* yang sudah lama, lakukan pengujian menyeluruh untuk memastikan bahwa aplikasi *web* berjalan dengan baik dan tidak ada masalah yang muncul. Jika semua berfungsi dengan baik, Anda dapat merilis pembaruan ke lingkungan produksi.

Meningkatkan *library JavaScript* ke versi terbaru untuk menjaga keamanan, performa, dan fungsionalitas aplikasi *web*. Pastikan untuk

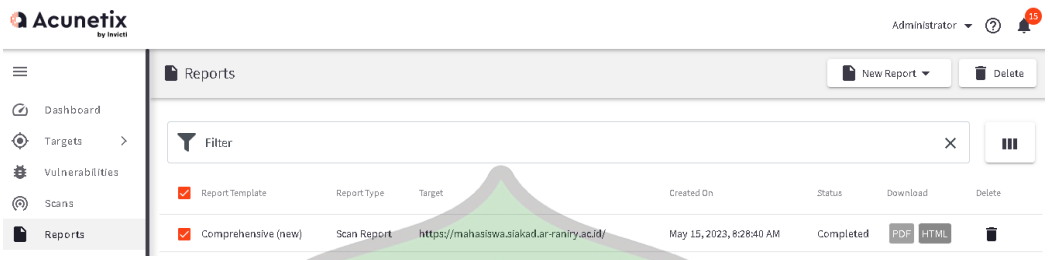
memeriksa dokumentasi resmi *library* JavaScript dan mengikuti pedoman yang diberikan pengembang untuk melakukan pembaruan secara tepat.

Hasil yang terdapat dari *scanning website* yang menggunakan *Acunetix Web Vulnerability Scaner* mencapai pada level dua (medium), sehingga harus adanya evaluasi terhadap celah pada kerentanan yang di temukan. Dari keseluruhan hasil *scanning* yang dilakukan dalam 5 hari dengan 10 kali pengujian menunjukkan bahwa kerentanan yang didapatkan pada *website* siakad UIN Ar-Raniry tidak ditemukan perbedaan, yang membedakan hanya pada *scan duration*, *request*, dan *average response time*. Tingkat keamanan *website* siakad UIN Ar-Raniry berada pada level 2 (medium) dengan jumlah 0 *high*, 2 *medium*, 4 *low* dan 4 pada info. Hasil ini menunjukkan bahwa *website* siakad UIN Ar-Raniry belum dikategorikan aman. Hal ini dikarenakan *website* masih memilki celah keamanan yang harus diperhatikan oleh pengelola *website*, sehingga pengelola *website* bisa meminialisir jika adanya ancaman dan pengakses informasi yang dilakukan tanpa izin berpotensi merusak sistem.

4.4 Report

Tahap terakhir yaitu melakukan pelaporan hasil menggunakan fitur *reports* pada Acunetix. *Report* adalah hasil laporan yang dihasilkan setelah proses pemindaian keamanan selesai dilakukan pada suatu *target information*. Laporan ini berisi informasi tentang potensi keretanan keamanan yang ditemukan saat pemindaian, analisis resiko, rekomendasi perbaikan, dan informasi lainnya. *Report* bertujuan untuk pengelola *website* dapat melihat hasil *scan* langsung dari

Acunetix. Hasil *report* dapat disimpan dalam bentuk PDF dan HTML seperti pada gambar 4.15:



Gambar 4.15 *Report*

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan terhadap *website* siakad UIN Ar-Raniry. Maka terdapat kerentanan-kerentanan seperti clikjacking, HTTP Strict Transport Security (HSTS), TLS 1.1 *Enabled*, beberapa *web alert informational*. Hasil yang ditemukan *Acunetix Web Vulneraility Scanner* pada *website* siakad UIN Ar-Raniry berada pada level *medium* yaitu level 2 termasuk kategori *high* 0 pada *scanning*, 2 *medium*, 4 *low* dan 4 pada info yang berarti kerentanan terjadi karena kesalahan konfigurasi, *site coding* yang lemah dan *website* belum bisa dikategorikan aman karena masih terdapat celah keamanan dan berpotensi diakases tanpa izin yang bisa merusak sistem. Maka solusi yang ditawarkan terhadap kerentanan yaitu:

1. Meningkatkan TLS.1 menjadi TLS.2 untuk meningkatkan keamanan komunikasi data.
2. Meningkatkan *javascript libraries* ke versi terbaru bertujuan mengurangi kerentanan dan mengurai resiko serangan.

3. Mengkofigurasi *server website*, menyertakan header x-frame-options dan *header* CSP dengan frame-anccestors pengarahan.
4. Mengkofigurasi *cookie* yang sesuai dengan standar seperti; setel atribut *secure*, setel atribut HttpOnly dan lainnya.
5. Menginstruksi *browser* bahwa *cookie* hanya dapat diakses melalui saluran SSL/TLS yang aman.
6. Menghubungi otoritas sertifikat bertujuan memperbarui sertifikat SSL.
7. Menerapkan CSP ke dalam aplikasi *website*, seperti; uji validasi.
8. Mengkonsultasikan referensi *website*.

5.2 Saran

Dari penelitian yang telah dilakukan, perlu dilakukan penelitian mendalam mengenai kerentanan *website* siakad UIN Ar-Raniry, agar penilaian dari kerentanan-kerentanan yang dialami memiliki penilaian yang lebih akurat. Serta diperlukan penyajian hasil audit keamanan ini dalam bentuk kerangka audit yang lain yang memiliki standar keamanan informasi lain agar hasilnya lebih optimal.

DAFTAR KEPUSTAKAAN

Basri. Pendekatan Kriptografi Hybrid pada Keamanan Dokumen Elektronik dan HypertextTransfer Protocol Secure (HTTPS) (Analisis Potensi Implementasi Pada Sistem Keamanan*), Jurnal Ilmu Komupter*. 1(2). 2015.

Ciampa. *Security & Guide to Network Security Fundamentals.* Boston: Course Technology. 2012.

Sutanta. *Sistem Informasi Manajemen*. Yogyakarta: Andi. 2022.

Febri Al Fajar. Analisis Keamanan Aplikasi *Web* Prodi Teknik Informatika UIKA Menggunakan *Acunetix Web Vulnerability. Jurnal Inova-Tif*. Vol. 3 No. 2. 2020.

Frenly Kristianto, Syaiful Rahman, dan Syamsul Bahri. Analisis Keretanan Pada *Website* Servio Menggunakan *Acuenitix Web Vulnerability*. *Journal of Technology Research in Information System and Engineerin*. Vol 9 no. 1. 2022.

Gondodiyoto. *Audit Sistem Informasi + Pendekatan CobIT*. Jakarta: Mitra Wacana Media. 2007.

Informasi, J. & Zirwan, A. *Pengujian dan Analisis Keamanan Website Menggunakan Acunetix Vulnerability Scanner*. *4*(1), 1–3. 2022.

Jaedun, A. *Metodologi Penelitian Eksperimen*. Fakultas Teknik UNY. 2011.

Khaldun, U. I. *Analisis keamanan aplikasi web prodi teknik informatika uika menggunakan acunetix web*. 2, 110–120. 2020.

Kristison Zakaria, Onno W. Purbo, Dan Elvira Wardah. Mendeteksi backdoor dengan aplikasi shell detector: Kristison Zakaria. 2017.

Lehtinen, R., Russell, D., Gangemi, G.T. *Dasar-Dasar Keamanan Komputer*. (Edisi ke-2), Washington D.C: O'Reilly Media. 2006.

Mona Fronita. Analisis Celah Keamanan *Website* Sitasi Menggunakan *Vulnerability Assessment*. J*urnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*. Vol. 9 no. 1. 2023.

Nazwita, & Ramadhani. *Analisis Sistem Keamanan Web Server dan Database Server Menggunakan Suricata*, Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI). 9, 308-317. 2017.

O'brien, J. A. *Pengantar Sistem Informasi: Perspektif Bisnis dan Manajeral*. (Edisi 12). Terjemahan Dewi Fitriasari dan Deny Amos, Jakarta: Salemba Empat. 2005.

Rahmadi. *Pengantar Metodelogi Penelitian*. Banjarmasin: Antasari Press. 2011.

Rifky Lana Rahardian. Analisisi Keamanan *Web* New Kuta Golf Menggunakan Metode *Vulnerability Assessment* Dan Perhitungan Metrics. *Jurnal Informatika Dan Teknologi Komputer*. Vol 2 no. 3. 2022.

Rerung, R.R. *Program Web Dasar*. Yogyakarta: Deepublish. 2018.

Shelly, G.B., Woods, D.M, & Dorin, W.J. HTML: Konsep dan Teknik Komprehensif (edisi ke-5.), Boston: Cengage Learning. 2009.

Teguh Wahyono. *Sistem Informasi, Konsep Dasar, Analisa Desain Dan Implementasi.* Yogyakarta: Graha Ilmu. 2004.

Turban, Rainer, & Potter. *Introduction to Information Technology*. (2 edition), John Wiley & Sons, Inc, New York. 2003.

https://www.acunetix.com/support/docs/a360/issues/vulnerability-severity-levels/

DAFTAR RIWAYAT HIDUP

Nama/NIM	:	Mulya Akmal/180705004
Tempat/Tanggal Lahir	:	Aceh Besar/08 November 2000
Jenis Kelamin	:	Laki-Laki
Pekerjaan	:	Mahasiswa
Agama	:	Islam
Kebangsaan/Suku	:	Indonesia/Aceh
Status	:	Belum Menikah
Alamat	:	Gp Lambaro Kueh, Kec. Lhoknga Kab. Aceh Besar

Orang Tua

Nama Ayah	:	M Ali
Pekerjaan	:	PNS/Guru
Nama Ibu	:	Nurasiah
Pekerjaan	:	Ibu Rumah Tangga
Alamat	:	Gp Lambaro Kueh, Kec. Lhoknga Kab. Aceh Besar

pendidikan

SD/MI	:	SD Kulam Data
SMP/MTs	:	MTsN 1 Lhoknga
SMA/MA	:	SMK Negeri 2 Banda Aceh
PT	:	Mahasiswa S1 Teknologi Informasi UIN Ar-Raniry Banda Aceh (2018-Sekarang)

Demikan riwayat hidup ini saya buat sebenarnya agar dapat dipergunakan sebagaimana mestinya.

Banda Aceh, 20 Juli 2023

Penulis,

Mulya Akmal