Analisis Kerentanan Keamanan *Website* Dengan Menggunakan Metode *Penetration Testing* dan *Framework* OWASP (Studi Kasus : *Website* Instansi XYZ)

Ghaly Pramudya Adhi Mukti
Fakultas Rekayasa Industri
Sistem Informasi
ghalypramudya@student.telkomuniversity.ac.id

Muhammad Nasrullah
Fakultas Rekayasa Industri
Sistem Informasi
emnasrul@telkomuniversity.ac.id

Purnama Anaking
Fakultas Rekayasa Industri
Sistem Informasi
purnamaanaking@telkomuniversity.ac.id

Abstrak **—**Keamanan sistem informasi menjadi salah satu masalah utama dalam perkembangan teknologi. Sistem informasi harus memastikan kerahasiaan, ketersediaan, dan integritas pada semua sumber daya informasi. Instansi XYZ Surabaya adalah salah satu instansi kesehatan yang menggunakan sistem informasi berbasis *website* untuk media informasi kesehatan. Namun bedasarkan data telah terjadi peretasan terhadap layanan kesehatan, tercatat selama periode bulan januari sampai dengan bulan september tabun 2020 telah terjadi sebanyak 123 kejadian serangan *cyber* yang menargetkan layanan kesehatan dan pada 2021 terjadi kasus peretasan data yang terjadi pada Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan. Berdasarkan kasus tersebut sistem informasi pada instansi juga beresiko menjadi sasaran dari serangan *cyber*, maka perlu dilakukan pengujian keamanan untuk meminimalisir celah kerentanan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab. Tujuan dilakukan penelitian ini adalah melakukan analisis kerentanan pada website Instansi XYZ untuk mengetahui apa saja kerentanan pada website dan memberikan rekomendasi kepada pihak Instansi. Metode yang digunakan pada penelitian ini adalah penetration testing dengan pendekatan framework OWASP. Hasil penelitian ini diharapkan dapat memberikan rekomendasi yang dapat digunakan oleh pengembang *website* Instansi XYZ dalam meningkatkan keamanan bagi *website* mereka.

***Kata kunci*— Keamanan Sistem, OWASP, *Penetration Testing*, *Website*.**

I. PENDAHULUAN

Seiring dengan perkembangan teknologi, informasi sangat mudah untuk diperoleh dan disebarluaskan serta menjadi salah satu aset penting bagi perusahaan [1]. Perusahaan harus memiliki kemampuan untuk menyediakan informasi yang akurat dan cepat, sehingga sistem informasi menjadi bagian penting bagi perusahaan karena dapat mendukung berbagai kegiatan di perusahaan. Tetapi, sistem informasi dapat menyebabkan munculnya ancaman dan risiko dari penggunaan teknologi karena dimanfaatkan oleh pihak yang tidak bertanggung jawab[2]**.** Berdasarkan data dari laporan Positive Technologies yang telah dirangkum oleh Kompas.id, tercatat selama periode bulan januari sampai dengan bulan september tabun 2020 telah terjadi sebanyak 123 kejadian serangan cyber yang menargetkan layanan kesehatan. Berdasarkan data yang telah dirangkum sebagian besar penyerangan dilakukan dengan malware dengan presentase sebesar 61%, kemudian serangan berupa rekayasa sosial sebesar 46%, serangan berusa peretasan sebesar 30%, serangan credential compromise sebesar 17% dan serangan pada situs kesehatan sebesar 7%[3]. Permasalahan keamanan sistem informasi dapat menimbulkan kerugian bagi perusahaan, sehingga perlu dilakukan tindakan untuk mencegah dan memperbaiki keamanan sistem informasi[2].

Berdasarkan data pada tekno.kompas.com terjadi kasus peretasan data yang terjadi pada Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan tahun 2021. Data yang berhasil dicuri berupa Nomor Kartu, NIK, Nomor Ponsel, E-mail, alamat dan gaji dari 279 juta penduduk indonesia yang terdaftar pada BPJS Kesehatan. Data tersebut dijual oleh akun peretas yang bernama “kotz” pada forum online[3]. Karena permasalahan tersebut mendorong organisasi untuk membangun keamanan sistem, tidak terkecuali sistem keamanan pada *website*. Website adalah sebuah halaman yang menyediakan informasi atau data yang dapat diakses saat terkoneksi dengan internet. Website sangat dibutuhkan untuk menyebarkan data dan informasi secara lebih luas[4]. Seperti pada instansi kesehatan contohnya juga menerapkan sistem informasi berbasis website.

Instansi XYZ Surabaya adalah salah satu instansi kesehatan yang menggunakan sistem informasi berbasis website untuk media informasi kesehatan. Pada sistem informasi xyz.com berisi informasi terkait UGD, Rawat Inap, Rawat jalan, Jadwal Dokter, Pendaftaran Pasien, dan juga informasi lainnya. Berdasarkan kasus serangan *cyber* yang menargetkan instansi kesehatan pada 2020 dan juga kasus peretasan data pada BPJS Kesehatan maka sistem informasi pada instansi juga beresiko menjadi sasaran dari serangan *cyber*, hal ini didukung dengan hasil wawancara kepada koordinator IT pada Instansi XYZ yang menyatakan bahwa *website* Instansi XYZ pernah terjadi tindakan peretasan dengan mengganti tampilan *website*, maka perlu dilakukan

pengujian keamanan untuk meminimalisir celah kerentanan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab.

Cara terbaik untuk mengetahui tingkat kemanan suatu sistem yatu dengan mencoba penetration testing[5]. Penetration testing merupakan pengujian untuk mengidentifikasi kerentanan baru, menguji keamanan dan menetapkan kontrol untuk mengurangi resiko dari sistem informasi[4]. Pada penerepan penetration testing pada *website*, terdapat panduan atau *framework* yang digunakan dalam pengujian dan analisis keamanan pada *website*, panduan yang dimaksud yaitu (*Open Web Application Security Project*) OWASP. OWASP merupakan organisasi yang mempunyai visi untuk mentaga kseamanan *website* dengan menyediakan *resource*[8]. Pada penelitian ini pengujian keamanan *website* dilakukan dengan *penetration testing* dan menggunakan metode OWASP yang bertujuan untuk mencari celah keamanan yang ada pada *website* Instansi XYZ Surabaya. Hasil dari penelitian ini dapat digunakan oleh pengelola atau pengembang *website* untuk memperbaiki celah kerentanan pada *website*.

II. KAJIAN TEORI

A. Keamanan Sistem Informasi

Keamanan sistem informasi adalah bagaimana kita dapat mencegah ancaman atau paling tidak kita dapat mendeteksi adanya ancaman pada sebuah sistem informasi[2]. Keamanan Sistem Informasi juga bisa diartikan sebagai cara tau upaya kita dałam mencegah dari serangan pengguna komputer atau pengakses jaringan yang tidak bertanggung jawab[2].

B. OWASP

(*Open Web Application Security Project*) OWASP merupakan suatu komunitas yang memungkinkan sebuah organisasi dapat mengembangkan dan memelihara website yang bisa dipercaya[14]. Sedangkan OWASP Top 10 adalah sebuah kerangka yang dibentuk oleh komuitas OWASP yang berisikan 10 urutan ancaman-anacaman yang dapat membahayakan dan mengancam bagi suatu website[6].

C. Penetration Testing

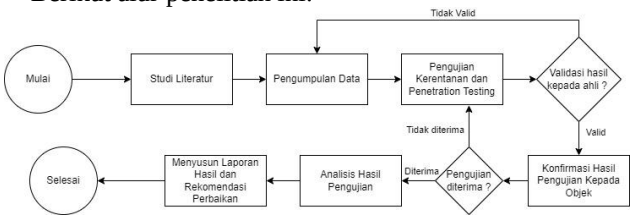
Penetration Testing adalah metode pengujian yang membantu dalam proses pembentukan strategi keamanan sistem informasi berdasarkan hasil pengujian dan identifikasi kerentanan yang akurat. Tujuan dari *Penetration Testing* yang utama adalah mengidentifikasi kerentanan keamanan dalam keadaan yang terkendali, sehingga dapat mengantisispasi sebelum pihak yang tidak bertanggung jawab dapat mengeksploitasi sistem tersebut [7].

III. METODE

Metode yang digunakan pada penelitian ini yaitu metode *black box testing* dengan pendekatan *framework* OWASP karena berfokus pada keamanan sistem pada *website*. OWASP merupakan aplikasi *open source* sehingga siapapun bisa melakukan pengujian keamanan pada *website*. Pada penelitian ini juga menggunakan teknik *penetration testing* dengan menggunakan *Operation System* Kali Linux.

A. Alur Penelitian

Berikut alur penelitian ini:



IV. HASIL DAN PEMBAHASAN

A. Information Gathering

Pada tahapan ini dilakukan pengumpulan informasi-informasi yang dibutuhkan seputar *webiste* xyz.com dan antrianxyz.com. Informasi yang dibutuhkan merupakan informasi yang mendalam pada *website*. Hasil *information gathering* bisa dilihat pada tabel dibawah ini.

Perintah	Hasil
whois	Diketahui informasi penting terkait website, seperti nama *domain*, *server domain*, tanggal *update*, status *domain*, *email registrasi*, dll.
nslookup	Diketahui informasi seputar nama server secara mendetail.

B. Network Mapping

Network Mapping merupakan tahapan pemetaan dan pemindaian yang berguna untuk mengumpulkan informasi terkait perangkat yang tersedia atau port yang terbuka pada infrastruktur jaringan *website* dengan menggunakan perintah *nmap* pada *tools kali linux*. Hasil *Network Mapping* bisa dilihat pada tabel dibawah ini.

Perintah	Hasil
nmap -v -sn	Pada kedua *website* terdapat 1 *host* yang aktif atau terhubung.
nmap -vv -sS	Ditemukan dua port yang bersifat open pada kedua website, yaitu port 80/tcp dan 443/tcp.
nmap -O	Sistem operasi yang digunakan oleh kedua *website* adalah linux 2.6.

C. Exploitation

Pada tahap ini peneliti melakukan pengujian celah kerentanan pada *website*. Informasi yang sebelumnya sudah terkumpul dari tahapan sebelumnya dapat digunakan sebagai acuan untuk melakukan pengujian ini. Pengujian kerentanan dilakukan dengan dua cara, yaitu : *Scanning* menggunakan *tools* OWASPZap dan Penetration testing. Hasil dari eksploitasi pada kedua *website* bisa dilihat pada tabel dibawah ini.

Daftar Kerentanan	Kerentanan yang ditemukan
A01 : Broken Access Control.	Kerentanan pada bagian akses data pendaftaran *website* antrianxyz.com
A02 : Cryptographic Failures.	Kerentanan *Hidden file Found*, *Suspicious*

	Comment, Re-examine Cache, pada kedua *website*.
A03 : Injection.	*User Controllable HTML* pada xyz.com.
A04 : Insecure Design.	*Big Redirect Detected* pada kedua *website*.
A05 : Security Misconfiguration.	Kerentanan *Cloud Metadata, CSP Wildcard, CSP Script-unsafe, CSP Style-unsafe, CSP Header not set, Missing Anti-clickjacking, Server Leak via X-Powerred-By, Strict Transport Header not set, X Content-type Header missing, User Agent Fuzzer, Cookie without samesite,* pada kedua *website*.
A06 : Vulnerable and Outdated Component.	Kerentanan *Vulnerable JS Library* pada kedua *website.*
A07 : Identification and Authentication Failures.	Tidak ditemukan kerentanan.
A08 : Software and Data Integrity Failures.	Kerentanan *Cross-Domain JavaScript* pada xyz,com.
A09 : Security Logging and Monitoring.	Kerentanan pada bagian Login admin xyz.com.
A10 : Server-Side Request Forgery (SSRF),	Kerentanan pada *Request HTTP* pada antrianxyz.com

D. Reporting

Nama Kerentanan	Solusi
A01	Gunakan teknologi Blockchain yang berisi patch atau blok yang berguna untuk mengamankan data pengguna saat dikirim ke server. Untuk menerapkan blockchain[8]
A02	*Hidden File Found* : Periksa apakah pada *website* terdapat *file* yang tersembunyi dan dapat diakses. Lalu pertimbangkan apakah *file* tersebut merupakan *file* yang penting atau digunakan, jika *file* tersebut tidak penting sebaiknya dihilangkan atau dihapus[9].
	Suspicious Comment : Hapus komentar yang berisi informasi terkait kekurangan atau kelemahan dari *website*. Karena komentar tersebut bisa dimanfaatkan oleh penyerang untuk mencari tahu kelemahan pada *website*[9].
	Re-examine Cache : Gunakan konfigurasi *header cache* yang tepat untuk mengantisipasi kebocoran data sensitif yang dapat diakses melalui cache pengguna. Terapkan pengaturan *cache* “no-cache” agar tidak ada konten yaang disimpan dalam cache atau terapkan durasi waktu pada *cache* untuk membatasi waktu pengguna[10].
A03	Validasi inputan dapat berupa pemeriksaan karakter yang diperbolehkan atau seperti batasan panjang inputan yang bisa diterima. Kemudian lakukan pembersihan *output* sebelum menulisnya kedalam atribut *HTML*. Pembersihan dapat berupa *encoding* yang dapat mengamankan inputan pengguna yang mengandung isi sensitif tidak dapat dieksploitasi[11].
A04	*Big Redirect Detected* : Implementasikan pola kerja development yang aman dengan menggunakan pola *AppSec*.[9].
A05	*Cloud Metadata* : Validasi kembali data apapun yang dikirimkan dari *request HTTP* kepada *server*

	NGINX. Hal ini bisa terjadi jika pada konfigurasi *server* terdapat *header* "host"[12].
	CSP Wildcard : Tetapkan sumber daya yang lebih sepesifik seperti contoh media hanya diperbolehkan dari example.org dan example.net [13].
	CSP Script-unsafe : Hapus skrip yang mengandung "**unsafeinline**" dari kebijakan CSP dan atur kebijakan dengan lebih berhati-hati untuk memperbolehkan skrip hanya dari sumber yang valid dan bisa dipercaya[13].
	CSP Style-unsafe : Hapus *style* unsafeinline dari kebijakan CSP dan atur kebijakan dengan hati-hati untuk memperbolehkan *style* hanya dari sumber yang dapat dipercaya[13]. Ubah *style* yang mengandung "**unsafeinline**" menjadi "**nonce**"
	CSP Header not set : Pastikan bahwa *application server, web server, load balancer*, dan lainnya terkonfigurasi menggunakan *Content-Security Policy header* [9].
	Missing Anti-clickjacking : Aktifkan fitur *ContentSecurity-Policy* dan *X-FrameOptions HTTP header* pada halaman *website*. Fitur tersebut bisa diterapkan pada konfigurasi *APACHE* seperti "**set X-Frame-Options "SAMEORIGIN"**"[9].
	Server Leak via X-Powerred-By : Matikan atau hapus informasi *server* pada *header* "HTTP X-Powered-By" untuk mengurangi informasi terkait teknologi atau kerangka kerja pada *server* yang dapat dimanfaatkan oleh penyerang[9].
	Strict Transport Header not set : Pastikan bahwa semua konfigurasi pada *server* seperti *application server, web server* terkonfigurasi menggunakan "Strict-TransportSecurity"[9].
	X Content-type Header missing : Pastikan bahwa bagian pada konfigurasi *server* yaitu "Content-Type header" telah terkonfigurasi menggunakan fitur "X-Content TypeOptions" dan telah dikonfigurasi menjadi "nosniff"[9].
	User Agent Fuzzer : -
	Cookie without samesite : Memastikan setelan *SameSite attribute* disetel secara ketat seperti "**SameSite=Strict**" untuk memperketat semua *cookie*[9].
A06	*Vulnerable JS Library* : Lakukan pembaruan pada Jqueri atau library dengan versi yang terbaru yang sudah diperbaiki oleh pengembang, agar mengurangi resiko menjadi celah kerentanan[12].
A08	*Cross-Domain JavaScript* : Pilih dan Pastikan bahwa sumber *file* hanya dari pihak ketiga yang bisa dipercaya. Kemudian sumber harus tidak bisa dikontrol oleh *end-user aplication*. Contoh pihak ketiga yang sering digunakan seperti *javasript, framework laravel, bootstrap*, dll[14].
A09	Pastikan semua aktivitas *login* dicatat dan melakukan *monitoring log* untuk mendeteksi tindakan yang mencurigakan. Lakukan

	pemblokiran sementara pada akun yang melakukan login gagal. Misalnya, jika ada 5 kali upaya login yang gagal dalam waktu singkat, blokir akun selama 10 menit.
A10	-

V. KESIMPULAN

Berdasarkan temuan kerentanan diatas dapat disimpulkan bahwa kedua *website* rentan terhadap ancaman serangan, walaupun sebagian besar dari kerentanan memiliki tingkat resiko yang sedang. Penelitian ini telah memberikan informasi terkait kondisi keamanan dan kerentanan pada *website* Instansi XYZ. Dengan menggunakan kedua metode yaitu *penetration testing* dan OWASP, peneliti mampu menemukan celah kerentanan dan melakukan *penetration attack* pada *website*. Dari temuan kerentanan membuktikan bahwa, dengan menggunakan kedua metode tersebut mampu untuk mengetahui kerentanan yang ada pada *website* dan memberikan rekomendasi perbaikan yang bertujuan untuk meningkatkan keamanan pada website Instansi XYZ Surabaya.

REFERENSI

[1] S. Nurul, S. Anggrainy, and S. Aprelyani, "Faktor-Faktor Yang Mempengaruhi Keamanan Sistem Informasi: Keamanan Informasi, Teknologi Informasi DAN Network (Literature Review SIM)," *Jurnal Ekonomi Manajemen Sistem Informasi*, vol. 3, no. 5, 2022, doi: 10.31933/jemsi.v3i5.

[2] C. Chazar, "Standar Manajemen Keamanan Sistem Informasi Berbasis ISO IEC 27001 2005," *Jurnal Informasi*, vol. 7, no. 2, 2015.

[3] W. K. Pertiwi and O. Yusuf, "BPJS Kesehatan Akui Ada Kemungkinan Peretasan Data 279 Juta Warga RI," KOMPAS.COM. Accessed: Jan. 18, 2024. [Online]. Available: https://tekno.kompas.com/read/2021/05/25/13304797/bpjs-kesehatan-akui-ada-kemungkinan-peretasan-data-279-juta-warga-ri

[4] T. Ariyadi, T. Langgeng Widodo, N. Apriyanti, and F. Sasti Kirana, "Analisis Kerentanan Keamanan Sistem Informasi Akademik Universitas Bina Darma Menggunakan OWASP," *TECHNO.COM*, vol. 22, no. 2, pp. 418–429, 2023.

[5] I. O. Riandhanu, "Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi," *Jurnal Informasi dan Teknologi*, vol. 4, no. 3, Oct. 2022, doi: 10.37034/jidt.v4i3.236.

[6] S. Hidayatulloh and D. Saptadiaji, "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)," *Jurnal Algoritma*, vol. 19, no. 1, pp. 77–86, 2021, [Online]. Available: http://jurnal.itg.ac.id/

[7] M. Hasibuan and A. M. Elhanafi, "Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box," *sudo Jurnal Teknik Informatika*, vol. 1, no. 4, pp. 171–177, Dec. 2022, doi: 10.56211/sudo.v1i4.160.

[8] I. Riadi, R. Umar, and I. Busthomi, "Optimasi Keamanan Autentikasi dari Man in the Middle Attack (MiTM) Menggunakan Teknologi Blockchain," *JIEET : (Journal Information Engineering and Educational Technology)*, vol. 04, no. 01, pp. 15–19, 2020.

[9] D. Lee, "Analisis Kerentanan Aplikasi Akademik Berbasis Website XYZ Menggunakan OWASP," *Jurnal Khatulistiwa Informatika*, vol. 11, no. 2, pp. 92–102, 2023.

[10] A. Rohim and L. Setiyani, "JIPAKIF NUSANTARA Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara Analisis Celah Keamanan E-Learning Perguruan Tinggi Menggunakan Vulnerability Assessment," *JIPAKIF : Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara*, vol. 1, no. 1, pp. 1–10, 2023, [Online]. Available: http://jurnal.edunovationresearch.org/

[11] N. Christina Sari *et al.*, "Deteksi Kerentanan SQL Injection pada Website Menggunakan Vulnerability Assessment Info Artikel," *Journal of Data Insights*, vol. 2, no. 1, pp. 9–17, 2024, doi: 10.26714/jodi.

[12] J. Khatib Sulaiman and U. Pakuan, "Analisis Keamanan Website Menggunakan Open Web Application Security Web (OWASP) I Wayan Sriyasa, Victor Ilyas Sugara," *Indonesian Journal of Computer Science*, vol. 13, no. 2, pp. 3315–3327, 2024.

[13] R. Muhammad Fauzi, R. Hermawan, D. Rosian Adhy, and S. Maesaroh, "Analisis Kerentanan Keamanan Web Menggunakan Metode OWASP Dan PTES di Web Pemerintahan Desa XYZ," *Jurnal Orang Elektro*, vol. 13, no. 2, pp. 225–231, 2024, [Online]. Available: https://XYZ.g-desa.id/,

[14] S. Hidayatulloh and D. Saptadiaji, "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)," *Jurnal Algoritma*, vol. 19, no. 1, pp. 77–86, 2021, [Online]. Available: http://jurnal.itg.ac.id/