BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digitalisasi, aplikasi *web* merupakan alat utama yang digunakan pemerintah Indonesia untuk mengelola informasi secara digital yang tujuannya untuk memberikan layanan kepada masyarakat. Dengan adanya *website* masyarakat dapat mengakses dan menerima informasi dengan cepat dan efisien, seperti berita, layanan pengaduan, dan informasi publik lainnya.

Dinas Kearsipan dan Perpustakaan Daerah Kabupatan Semarang melalui laman *web* nya mempunyai peran penting dalam menyampaikan informasi publik seperti kegiatan dan program literasi, arsip dan perpustakaan, layanan digital dan, berita terkini, sehingga masyarakat dapat mengakses informasi yang dikeluarkan dengan cepat dan efisien.

Namun, seiring dengan berkembangnya teknologi informasi terdapat juga ancaman siber yang makin berkembang dari masa ke masa. Berdasarkan data milik Badan Siber dan Sandi Negara (BSSN) hasil penelusuran pada *darknet*, ditemukan adanya 56.128.160 temuan *data exposure* yang berdampak pada 461 *stakeholder* di Indonesia. Pada kasus *web defacement* ditemukan sebanyak 5.780 yang menargetkan beberapa *domain* dan sebanyak 4.071 *web defacement* terkait judi *online* yang menargetkan situs pemerintah(BSSN, 2024). Temuan ini menunjukkan bahwasannya situs *web* pemerintah pun sangat rentan terhadap segala bentuk serangan siber, termasuk 2 diantara serangan yang paling umum yaitu *SQL Injection* dan *Cross Site Scripting (XSS)*.

Salah satu teknik umum di dunia *cyber crime* yaitu *SQL Injection*. Sebagian besar aplikasi *modern* menggunakan basis data yang persisten untuk meneruskan informasi. Serangan injeksi terjadi ketika seseorang dengan sengaja mengirimkan perintah *SQL* berbahaya ke *server database* melalui saluran yang ilegal. Saluran yang paling umum digunakan untuk serangan ini adalah *input* data yang tidak divalidasi. *SQL Injection* adalah kerentanan yang terjadi karena *input* pengguna tidak divalidasi dengan baik, dan ini adalah salah satu metode serangan yang umumnya digunakan dalam aplikasi *web* (Riyanti dkk., 2024)

Penetration testing adalah sub kategori dari *ethical hacking* yaitu sebuah metode dan prosedur yang bertujuan untuk menguji dan melindungi keamanan informasi. *Penetration testing* merupakan aktifitas mengevaluasi sistem keamanan yang sudah dibuat dengan cara melakukan simulasi serangan menggunakan metode yang biasa digunakan oleh peretas. Kegiatan ini perlu mendapat persetujuan legal dari pemilik sistem tersebut. Seperti halnya sekarang sedang terjadi pembobolan maupun *hacking* pada perusahaan maupun *website* tertentu yang mengakibatkan kerugian pada pihak yang bersangkutan. Data tersebut dijual dan disalahgunakan oleh pihak yang tidak berwajib dan diperjualbelikan untuk kepentingan sendiri. (Widi Linggih Jaelani dkk., 2023).

Information System Security Assessment Framework (*ISSAF*) merupakan salah satu metode pemindai kerentanan atau *penetration testing* yang umum digunakan peniliti keamanan siber untuk menguji, serta menilai keamanan dari sebuah teknologi sistem informasi. Dengan menggunakan metode ini peneliti dapat menilai suatu keamanan sistem informasi secara terstruktur dan sistematis.

Dalam skripsi ini, penulis menggunakan *Information System Security Assessment Framework (ISSAF)* sebagai pendekatan sistematis untuk mendeteksi kerentanan serta melakukan pengujian keamanan terhadap aplikasi *web* Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Semarang. Fokus utama dalam pengujian ini adalah mendeteksi adanya kemungkinan kerentanan terhadap *SQL Injection* dan *XSS*, di mana serangan *XSS* dan *SQL Injection* termasuk diantara serangan yang paling berbahaya (Anugrah, 2024).

Untuk mendukung penelitian ini, penulis juga melakukan kajian terhadap beberapa penelitian terdahulu yang berkaitan dengan uji penetrasi *website* diantara lain penelitian oleh (Mu'min dkk., 2024) yaitu menguji sebuah *website* yang diklaim tidak memiliki kerentanan *SQL* menggunakan metode injeksi *SQL*. Penelitian oleh (Kushardianto dkk., 2024) yang melakukan uji penetrasi terhadap *website* udacoding pada kegiatan pengabdian masyarakat. Penelitian (Faizi dkk., 2023) melakukan pengujian kerentanan pada *website* Universitas Singaperbangsa Karawang menggukan metode *Penetration Execution And Standart* (*PTES*).

Berdasarkan latar belakang masalah yang sudah diuraikan penulis akan melakukan pengujian keamanan aplikasi berbasis *web* milik Dinas Perpustakaan dan Kearsipan Daerah Kabupaten Semarang dan menentukan judul skripsi sebagai berikut "**PEMINDAI KERENTANAN APLIKASI *WEB* DINAS KEARSIPAN DAN PERPUSTAKAAN DAERAH KABUPATEN SEMARANG MENGGUNAKAN *INFORMATION SYSTEM SECURITY ASSESSMENT FRAMEWORK* (*ISSAF*)**"

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini sebagai berikut :

1. Bagaimana penerapan metode *information system security assessment framework* (*ISSAF*) dengan *penetration testing* terhadap *website* Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Semarang?
2. Apa saja jenis celah keamanan yang ditemukan?
3. Apa rekomendasi mitigasi yang diperlukan untuk meningkatkan keamanan sistem?

1.3 Batasan Masalah

Agar pembahasan lebih terarah sesuai dengan judul yang telah ditentukan, penulis hanya membahas pokok-pokok pembahasan sebagai berikut:

1. Pengujian hanya berfokus pada *website* Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Semarang dengan *url* : https://arpusda.semarangkab.go.id/
2. Fokus pengujian hanya pada kerentanan *SQL Injection* dan *Cross-site scripting (XSS)*.
3. Tahapan pengujian hanya menggunakan 4 dari 8 tahapan metode *Information System Security Assessment Framework (ISSAF)*.

1.4 Tujuan

Tujuan yang ingin dicapai melalui penelitian di dalam skripsi sebagai berikut:

1. Menerapkan metode *Information System Security Assessment Framerok (ISSAF)* dalam melakukan *penetration testing* terhadap *website* Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Semarang.
2. Mengidentifikasi serta melakukan penilaian terhadap celah keamanan *SQL Injection dan Cross Site Scripting (XSS)*
3. Memberi rekomendasi mitigasi untuk memperkuat sistem.

1.5 Manfaat

Manfaat dari penelitian ini sebagai berikut:

1. Bagi Penulis

a. Sebagai pemenuhan syarat kelulusan Strata Satu (S1), Teknik Informatika Fakultas Teknik Universitas Islam Sumatera Utara

b. Menambah wawasan serta keterampilan dalam melakukan evaluasi keamanan sistem informasi disektor pemerintahan.

2. Bagi Instansi Terkait

a. Meningkatkan keamanan sistem informasi yang dikelola

b. Sebagai bahan evaluasi

1.6 Metodologi Penelitian

Metode yang penulis gunakan dalam penulisan skripsi ini sebagai berikut:

1. Studi Literatur

   Proses pengumpulan data melalui studi literatur yaitu dengan cara mengumpulkan refrensi berupa karya tulis ilmiah seperti jurnal yang erat kaitannya dengan tema penulisan skripsi ini.

2. Pengujian dan Ananlisis

   Menerapkan metode *Information System Security Assessment Framework (ISSAF)* dalam melakukan pengujian dan analisis. Menurut(Prasetyo & Alimyaningtias, 2024) terdapat 3 langkah untuk menggunakan metode *ISSAF* yaitu :

   a. *Planning and Preparation*
   b. *Assessment*
   c. *Reporting*

1.7 Sistematika Penulisan

Sistematika penulisan Tugas Skripsi ini dibagi atas beberapa bab, di mana masing-masing bab dibagi atas beberapa sub agar mempermudah penjelasan 6 mengenai penelitian yang dilakukan dan mempermudah pembaca dalam memahami isi penelitian. Adapun sistematika penulisan Tugas Skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan berisi tentang Latar Belakang Masalah, Rumusan Masalah, Tujuan, Manfaat, Batasan Masalah, dan Sistematika Penulisan dalam pembuatan Tugas Skripsi.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang digunakan sebagai pendukung untuk membahas tentang masalah apa yang dibahas pada penelitian ini. Teori pendukung diperoleh dengan studi literatur dan dokumentasi *internet.*

BAB III METODOLOGI PENELITIAN

Bab ini menguraikan tahapan-tahapan sistematis yang digunakan untuk melakukan kajian penelitian. Tahapan-tahapan tersebut merupakan kerangka yang dijadikan pedoman penelitian untuk mencapai tujuan yang telah ditetapkan

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil dan pembahasan dari *penetration testing* terhadap *website* Dinas Kearsipan dan Perpustakaan Daerah Kabupaten Semarang.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan uraian bab – bab penulisan skripsi dan saran yang diajukan untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 *Website*

2.1.1 Pengertian *Website*

Website merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh *browser* seperti *Mozila Firefox, Google Chrome* atau yang lainnya. Dokumen pada *website* disebut juga *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman yang lain, baik diantara halaman yang disimpan dalam *server* maupun *server* diseluruh dunia.(Manuhutu, 2021). *Website* juga dapat diartikan sebagai sarana penyampaian informasi dalam bentuk digital yang dapat diakses melalui *gadget* dengan mudah, cepat, dan efisien.

2.1.2 Jenis-Jenis *Website*

Menurut Saputra & Fahrizal, (2021) jenis-jenis *web* terbagi atas:

Jenis *Web* berdasarkan sifatnya:

1. *Website* dinamis, merupakan salah satu *website* yang menyediakan konten atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrogaman yang digunakan antara lain *PHP*, *ASP*, *.NET* dan memanfaatkan *database MySQL* atau *MS SQL*.
2. *Webstie* statis, merupakan *website* yang kontennya sangat jarang berubah. Bahasa pemrogaman yang digunakan adalah *HTML* dan belum memanfaatkan *database.*

Jenis *Web* berdasarkan fungsinya:

1. *Personal Website, website* yang berisi informasi pribadi seseorang.
2. *Commercial website, website* yang dimiliki oleh perusahaan yang bersifat bisnis.
3. *Government website, website* yang dimiliki oleh instansi pemerintah, pendidikan, yang bertujuan memberikan pelayanan kepada pengguna.
4. *Non-Profit Organization Website,* dimiliki oleh organisasi yang bersifat *non-profit* atau tidak bersifat bisnis.

2.1.3 Keamanan *Website*

Keamanan *website* adalah suatu aktivitas untuk melindungi *website* dan jaringannya dari berbagai ancaman. Mulai dari pencurian data, hingga perusakan *software* dan *hardware.* Umumnya, keamanan *website* disebut juga dengan *cyber security.* Keamanan *website* adalah salah satu prioritas utama pengembang *web.* Jika seseorang mengabaikan perlindungan ini, seorang peretas dapat memperoleh informasi penting dan dapat mengubah format *website* tersebut.(Azwan dkk., 2022)

2.1.4 Aspek Keamanan *Website*

Aspek keamanan *website* merupakan elemen-elemen yang sangat penting dan harus diperhatikan untuk melindungi *website* dari ancaman serangan siber, menjaga integritas data, dan memastikan setiap layanannya berfungsi sebagaimana mestinya.

Menurut (Handayani dkk., 2023) konsep keamanan harus memenuhi minimalnya 3 (tiga) aspek yaitu:

1. Kerahasiaan (*confidentiality*). Dapat menjamin bahwa data bersifat rahasia, maksudnya hanya dapat diakses oleh pihak yang berhak.
2. Keutuhan (*integrity*). Dapat menjamin bahwa data tetap utuh dan lengkap, dan dapat menjaga dari kerusakan atau ancaman lain yang mengakibatkan berubah informasi dari aslinya.
3. Ketersediaan (*availability*). Dapat menjamin pengguna dapat mengakses informasi kapanpun tanpa adanya gangguan dan tidak dalam format yang tidak bisa digunakan.

2.2 Pemindai Kerentanan (*Penetration Testing*)

2.2.1 Pengertian *Penetration Testing*

Penetration Testing adalah metode pengujian penetrasi atau *pentesting* (jangan dikelirukan dengan pengujian *ballpoint* atau pulpen), melibatkan simulasi serangan nyata untuk menilai risiko yang terkait dengan potensi pelanggaran keamanan. Pada *pentest* (sebagai lawan dari penilaian kerentanan), penguji tidak hanya menemukan kerentanan yang dapat digunakan oleh penyerang tetapi juga mengeksploitasi kerentanan, jika memungkinkan, untuk menilai apa yang mungkin diperoleh penyerang setelah eksploitasi berhasil.(Mulyanto dkk., 2022)

Menurut Hidayatulloh & Saptadiaji (2021) *Penetration Testing* adalah upaya yang dilakukan secara sah untuk mengeksploitasi sistem komputer dengan tujuan membuat sistem tersebut menjadi aman. *Penetration Testing* yang dilakukan dengan baik dapat menghasilkan rekomendasi untuk mengatasi dan memperbaiki masalah yang ditemukan selama pengujian.

2.2.2 Jenis-Jenis *Penetration Testing*

Menurut Hidayatulloh & Saptadiaji (2021) *penetration testing* terbagi menjadi 2, yaitu :

1. *Black Box Testing* : pengujian ini dilakukan oleh pihak yang tidak memiliki informasi sama sekali tentang sistem operasi, versi *server*, atau jaringan yang terdapat di dalam sistem, sehingga pelaku *penetration testing* harus mencari segala informasi yang dibutuhkan untuk melakukan pengujian terhadap sistem tersebut.
2. *White Box Testing* : Pengujian ini dilakukan oleh pihak yang telah diberikan seluruh informasi tentang sistem operasi, versi *server*, atau jaringan yang digunakan dengan tujuan untuk menemukan kerentanan yang ada sehingga dapat langsung diperbaiki oleh pihak pengelola suatu organisasi.

2.3 *OWASP*

Open Web Application Security Project (*OWASP*) merupakan organisasi *non-profit* yang didirikan pada tahun 2001 untuk membantu pemilik situs *web* dan pakar keamanan melindungi aplikasi *web* dari serangan dunia maya. *OWASP* memiliki 32.000 sukarelawan di seluruh dunia yang melakukan penilaian dan penelitian keamanan.(Alanda dkk., 2021)

Berikut 10 teratas kerentanan dari *OWASP* versi 2021(*Beranda - OWASP Top 10:2021*, n.d.):

1. *Broken Access Control :* Akses Kontrol menetapkan sebuah peraturan yang dimana *user* tidak dapat melakukan sebuah aksi diluar *permission* yang diberikan. Kegagalan atas hal ini dapat mengakibatkan pengeluaran

informasi yang tidak diizinkan, modifikasi, atau penghancuran dari semua data atau pemberlakuan sebuah fungsi bisnis di luar *limit* sebuah *user*.

2. *Cryptographic Failures* : sebelumnya dikenal sebagai *Sensitive Data Exposure*, yang lebih merupakan gejala yang luas daripada akar penyebab, fokusnya adalah kegagalan yang terkait dengan kriptografi (atau ketiadaannya), yang sering menyebabkan paparan data sensitif.
3. *Injection* : Beberapa injeksi yang biasa terjadi adalah *SQL*, *NoSQL*, perintah *OS*, pemetaan relasi objek(*ORM*), *LDAP*, dan bahasa ekspresi(*EL*) atau injeksi perpustakaan navigasi grafik objek. Konsepnya adalah identik di antara semua mesin penerjemah. Penelaahan kode sumber adalah metode terbaik dalam mendeteksi apakah aplikasi tersebut beresiko untuk diinjeksi.
4. *Insecure Design* : kondisi di mana aplikasi atau sistem memiliki kelemahan pada desainnya yang dapat dieksploitasi oleh pihak yang tidak berwenang, sehingga menimbulkan risiko keamanan. Kerentanan ini biasanya terjadi karena kurangnya perencanaan atau penerapan praktik keamanan selama tahap desain sistem atau aplikasi.
5. *Security Misconfiguration* : Tidak memiliki pertahanan yang sesuai atau *security hardening* yang diperlukan diseluruh bagian dari *stack* aplikasi atau tidak benar dalam melakukan konfigurasi untuk *permission* pada *cloud services.*
6. *Vulnerable and Outdated Components* : komponen perangkat lunak, *library*, *framework*, atau modul yang memiliki kerentanan keamanan karena menggunakan versi lama yang sudah tidak didukung (*outdated*) atau memiliki celah keamanan yang belum diperbaiki (*vulnerable*). Komponen

ini sering menjadi target serangan karena dapat dieksploitasi untuk mendapatkan akses tidak sah atau merusak sistem.

7. *Identification and Authentication Failures* : kegagalan sistem untuk secara efektif mengidentifikasi dan mengautentikasi pengguna atau entitas, sehingga memungkinkan akses tidak sah atau pelanggaran terhadap kontrol keamanan. Hal ini terjadi ketika mekanisme identifikasi (siapa pengguna) dan autentikasi (verifikasi identitas pengguna) tidak diterapkan dengan benar atau memiliki kelemahan yang dapat dieksploitasi.
8. *Software and Data Integrity Failures* : kerentanan keamanan yang terjadi ketika perangkat lunak, proses pembaruan, atau data yang digunakan oleh aplikasi tidak memiliki mekanisme untuk memastikan integritasnya. Hal ini memungkinkan penyerang untuk memodifikasi, memalsukan, atau menyisipkan kode atau data berbahaya, sehingga membahayakan sistem atau penggunanya.
9. *Security Logging and Monitoring Failures* : kegagalan dalam merekam (*logging*) atau memantau (*monitoring*) kejadian-kejadian yang berkaitan dengan keamanan pada sistem atau aplikasi. Kegagalan ini dapat menyebabkan organisasi tidak mengetahui atau terlambat mengetahui adanya serangan, pelanggaran, atau aktivitas yang mencurigakan, yang seharusnya bisa segera ditangani untuk mencegah kerusakan lebih lanjut.
10. *Server-Side Request Forgery* : jenis kerentanan keamanan yang terjadi ketika aplikasi *web* memungkinkan penyerang untuk mengirimkan permintaan *HTTP* dari *server* yang rentan ke tujuan lain, yang mungkin di luar kendali aplikasi. Dalam serangan *SSRF*, penyerang dapat

memanfaatkan *server* aplikasi untuk melakukan permintaan ke sumber daya internal atau eksternal, yang biasanya tidak dapat diakses langsung oleh penyerang.

2.4 *ISO/IEC* 27001

ISO/IEC 27001 merupakan standar internasional yang membantu organisasi mengelola keamanan informasi. Standar ini penting karena memberikan panduan untuk melindungi data dari ancaman seperti pencurian, kebocoran, atau akses yang tidak sah. Dengan menerapkan *ISO* 27001, perusahaan dapat memastikan bahwa mereka mematuhi hukum, meningkatkan kepercayaan pelanggan, dan mengelola risiko keamanan secara efektif (Nurbojatmiko dkk., 2025). Menurut Vakhula (2024) terdapat 11 versi standar *ISO/IEC* 27001 yaitu :

1. Intelijen ancaman
2. Keamanan informasi untuk penggunaan layanan *cloud*
3. Kesiapan TIK untuk kelangsungan bisnis
4. Pemantauan keamanan fisik
5. Manajemen konfigurasi
6. Penghapusan informasi
7. Penyembunyiaan data
8. Pencegahan kebocoran data
9. Kegiatan pemantauan
10. Pemfilteran *web*
11. Pengkodean yang aman

2.5 *SQL Injection*

SQL Injection merupakan suatu teknik eksploitasi dengan cara melakukan modifikasi perintah *SQL* pada *form input* suatu aplikasi yang nantinya akan memungkinkan penyerang untuk mengirimkan sintaks atau perintah kepada *database* suatu aplikasi(Putranto dkk., 2022)

Ada beberapa jenis *SQL Injection* diantaranya *Tautologies, Logically Incorrect Query, Union Query, Piggy Backed Query, Store Procedures*. *Tautologi* adalah serangan untuk menghasilkan kondisi *TRUE* dengan menggunakna '=' (sama) dengan *query*. Jenis *SQL Injection* ini dapat melewati halaman otentikasi dan mendapatkan data yang diekstraksi.(Yulia Andarini dkk., 2023)

2.6 *Cross Site Scripting (XSS)*

Cross Site Scripting atau serangan *XSS* merupakan salah satu jenis serangan *cyber* berbahaya dan pernah menyerang beberapa *platform* popular seperi *Facebook*, *Google*, dan *Paypal*. Serangan ini mengeksploitasi kerentanan *XSS* untuk mencuri data, mengendalikan sesi pengguna, menjalankan kode jahat, atau digunakan sebagai bagian dari serangan *phising*.(Charly dkk., 2022)

Menurut Ade Gustiyonoo dkk. (2024) Ada beberapa serangan *XSS* antara lain:

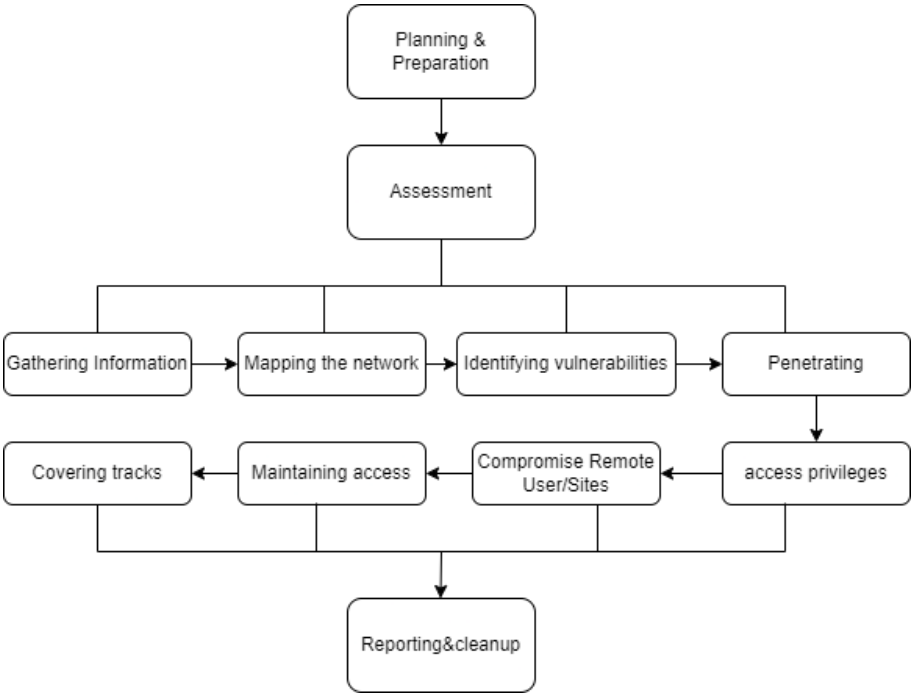
1. *Reflected XSS* adalah serangan di mana pengguna mengirimkan *input* ke aplikasi yang kemudian dipantulkan kembali ke pengguna yang sama. Serangan ini memanfaatkan permintaan yang tidak cukup disaring oleh sistem, sehingga skrip berbahaya dapat dieksekusi saat mengirimkan permintaan ke situs *web* yang rentan.

2. *Stored XSS* adalah serangan di mana kode berbahaya disuntikkan secara permanen ke *server* target, misalnya dalam *database*. Serangan ini terjadi ketika korban meminta informasi yang disimpan dari *server* dan menerima skrip berbahaya dari *server*, misalnya dari kolom pesan atau komentar.
3. *DOM-bassed XSS* (*XSS* berbasis) adalah serangan di mana *Payload* serangan mengubah "linkungan" *DOM* di *browser* korban yang digunakan oleh skrip sisi klien asli. Akibatnya, kode sisi klien berjalan secara tak terduga tanpa mengubah halaman itu sendiri, karena modifikasi berbahaya telah terjadi di lingkungan *DOM*

2.7 *Information System Security Assessment Framework (ISSAF)*

Framework ISSAF dikembangkan oleh *OISSG* (*Open Information System Security Group*). Metodologi *penetration testing* melalui *framework ISSAF* dibuat untuk mengevaluasi jaringan, sistem dan kontrol aplikasi. *Framework ISSAF* memberikan tahapan proses pengujian penetrasi secara optimal yang bertujuan memberikan arahan kepada auditor melakukan pengujian secara lengkap dan benar, serta menghindari kesalahan dalam melakukan pengujian serangan yang bersifat acak. Ada tiga langkah dalam kerangka kerja *ISSAF* yang meliputi persiapan dan perancangan, pengujian, serta pelaporan dan pembersihan jejak serangan.(Sanjaya dkk., 2020)

Tahapan menggunakan *framework ISSAF* dapat dilihat pada gambar 2.1



Gambar 2.1 *ISSAF Framework*

1. *Planning & Preparation*

   Tahap perencanaan serta persiapan di mana tim uji penetrasi dan pihak terkait menentukan tujuan, ruang lingkup, batasan, metode pengujian, serta mendapatkan persetujuan legal.

2. *Assessment*

   Pada tahap ini dilakukan analisis awal terhadap target untuk memahami struktur dan arsitektur sistem. *Assessment* adalah bagian utama yang dibagi lagi menjadi beberapa proses berikut:

   a. *Gathering Information* (pengumpulan Informasi)

      Mengumpulkan sebanyak mungkin informasi tentang target, baik dengan menggunakan teknik *passive reconnaissence* (tanpa interaksi langsung dengan target) maupun *active reconnaissance*

(melibatkan interaksi langsung) seperti : menggumpulkan *IP Address*, *DNS Record, e-mail* pegawai, informasi *server*.

b. *Mapping the Network* (pemetaan jaringan)

Membuat peta struktur jaringan yang bertujuan untuk memahami bagaimana perangkat-perangkat terhubung satu sama lain. Ini termasuk mendeteksi perangkat aktif, *service* yang berjalan , *port* jaringan yang terbuka, serta *firewall* atau *IDS* (*intrusion detection system*) yang digunakan

c. *Identifying Vulnerabilities* (Identifikasi Kerentanan)

Mencari celah keamanan dalam sistem atau aplikasi seperti : sistem operasi yang tidak diperbaharui, aplikasi yang rentan terhadap *SQL Injection, XSS, buffer overflow,* atau konfigurasi *server* yang lemah

d. *Penetration* (penetrasi)

Melakukan eksploitasi terhadap kerentanan yang ditemukan untuk mendapatkan akses ke sistem.

e. *Covering Tracks* (Menghilangkan Jejak)

Setelah berhasil masuk ke sistem, tim uji penetrasi perlu menghapus atau memodifikasi *log* untuk menghindari deteksi. Dalam praktik *ethical hacking* langkah ini hanya disimulasikan dan didokumentasikan, bukan untuk merusak sistem.

f. *Mantaining Access* (Mempertahankan akses)

Mencoba mempertahakan akses ke sistem untuk jangka panjang seperti penanaman *backdoor* atau menambahkan akun pengguna

baru. Tujuan untuk menguji seberapa besar risiko jika serangan nyata terjadi.

g. *Compromise Remote User/Site*

Jika memungkinkan, serangan diperluas ke *user* atau sistem lain yang terhubung dengan target utama. Seperti dari *server* satu meretas ke *server* lainnya di dalam jaringan internal.

h. *Access Privileges* (meningkatkan hak akses)

Setelah masuk ke sistem tim uji penetrasi mencoba meningkatkan hak akses, seperti dari *user* biasa menjadi *administrator* atau *root*.

3. *Reporting & Cleanup*

Tahap akhir berupa pembuatan laporan terkait semua temuan, metode yang digunakan, tingkat risiko, dan rekomendasi perbaikan. *Cleanup* artinya mengembalikan semua sistem ke kondisi semula dan memastikan tidak ada *backdoor, malware,* atau modifikasi lain yang tertinggal.

Tabel 2.1 *Roadmap ISSAF*

NO	METODOLOGI	TUJUAN
1	*Information Gathering*	Mengumpulkan informasi umum tentang target.
2	*Mapping Network*	Mengumpulkan informasi layanan pada sisi jaringan.
3	*Vulnerability Identifying*	Memindai Kerentanan
4	*Penetration*	Sebagai alat eksploitasi kerentanan *SQL injection.*
5	*Privilege escalation*	Memperoleh informasi akun.

NO	METODOLOGI	TUJUAN
6	*Compromise remote*	Memperoleh akses *remote* ke dalam sistem operasi *server*.
7	*Maintaining access*	Melakukan penanaman *backdoor*
8	*Covering Track*	Menghapus *log* serangan.

2.8 Penelitian Terdahulu

Berikut penulis mencantumkan beberapa hasil penelitian terdahulu sebagai bahan acuan untuk melakukan penelitian ini :

Tabel 2.2 Penelitian Terdahulu

NO	PENELITI	JUDUL PENELITIAN	HASIL PENELITIAN
1.	Muhammad Amirul Mu'min, Zumhur Alamin, Fathir, Sahrul Ramadhan	Uji Penetrasi Injeksi *SQL* terhadap Celah Keamanan *Website* XYZ menggunakan *Tools SQLMap*	*Tools SQLMap* berhasil menemukan data sensitif seperti *username* dan *password*, meskipun *web* tidak menunjukkan kerentanan *SQLi* secara eksplisit. Ini membuktikan efektivitas *SQLMap* dalam mendeteksi celah keamanan dan menegaskan pentingnya pengujian rutin, validasi *input, parameterized query*,

NO	PENELITI	JUDUL PENELITIAN	HASIL PENELITIAN
			WAF, dan pengelolaan hak akses untuk mencegah *SQL Injection.*
2.	Zidan Faizi, Puwantoro, Azhari Ali Ridha	Analisis *Web Security Hole* menggunakan Metode *Penetration Testing Execution and Standard (PTES)*	*Website* unsika.ac.id memiliki 1 kerentanan risiko tinggi, 5 sedang, 5 rendah, dan 6 informasional. Ditemukan 3 celah nyata: *X-Frame-Options Header Not Set, Application Error Disclosure, dan Broken Access Control.* Metode *PTES* terbukti efektif dalam mengidentifikasi dan memvalidasi celah keamanan secara sistematis.
3.	Nur Cahyono Kushardianto, Festy Winda Sari, Antoni Haikal, Muhammad Idris	Penetrasi *Testing* Aplikasi *Website* Udacoding	Ditemukan 10 kerentanan utama dengan level *high, medium*, dan informatif. Celah yang teridentifikasi termasuk *Broken Access Control,*

NO	PENELITI	JUDUL PENELITIAN	HASIL PENELITIAN
			Injection (XSS), Security Misconfiguration, dan *Unrestricted File Upload.* Kerentanan menyebabkan terbukanya data sensitif seperti *password* dalam *plain-text, token session* di *URL*, dan kelemahan pada *login endpoint*. Pengujian menggunakan pendekatan manual & otomatis.

2.9 *Flowchart*

Menurut Zalukhu dkk., (2023)*flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* sistem merupakan suatu urutan proses dalam sistem dengan menunjukkan alat dari media *input*, *output* serta jenis media yang digunakan untuk menyimpan dalam proses pengolahan data, sedangkan *flowchart* program merupakan suatu bagan dengan simbol-simbol tertentu yang menggambarkan suatu urutan dari proses secara detail dan berhubungan antara suatu proses (intruksi) dengan proses lainnya dalam suatu program.

Tabel 2.3 Simbol *Flowchart* (Zalukhu dkk., 2023)

SIMBOL	NAMA	FUNGSI
	*TERMINATOR*	Permulaan/akhir program
	GARIS ALIR (*Flow Line*)	Arah aliran program.
	*PREPARATION*	Proses inisiasi/pemberian harga awal.
	PROSES	Proses perhitungan/proses pengolahan data.
	*INPUT/OUTPUT DATA*	Proses *input/output data*, parameter, informasi.
	*PREDEFINED PROCESS* (Sub Program)	Permulaan sub program/ proses menjalankan sub program.
	*DECISION*	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	*ON PAGE CONNECTOR*	Penghubung bagian-bagian *flowchart* yang berada pada satu halaman.
	*OFF-PAGE CONNECTOR*	Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda.