DAFTAR ISI

HALAMAN SAMPUL i
HALAMAN JUDUL ii
HALAMAN PERSETUJUAN iii
HALAMAN PERNYATAAN KEASLIAN v
HALAMAN PERSEMBAHAN vi
HALAMAN MOTTO vii
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii
DAFTAR ISTILAH xiv
DAFTAR LAMPIRAN xix
INTISARI xx
ABSTRACT xxi
BAB I PENDAHULUAN 1
A. Latar Belakang Masalah 1
B. Rumusan Masalah 6
C. Batasan Masalah 6
D. Tujuan Penelitian 6
E. Manfaat Penelitian 7
BAB II TINJAUAN PUSTAKA 8
A. Landasan Teori 8
B. Penelitian Sebelumnya 22
BAB III METODE PENELITIAN 28
A. Tempat dan Waktu Penelitian 28
B. Metode Pengumpulan Data 28
C. Alat dan Bahan Penelitian 30
BAB IV HASIL DAN PEMBAHASAN 35
A. *Intelligence Gathering* 35

B. *Threat Modeling* 39
C. *Vulnerability Analysis* 39
D. *Exploitation* 48
E. Reporting 61
BAB V PENUTUP 66
A. Kesimpulan 66
B. Saran 66
DAFTAR PUSTAKA
LAMPIRAN

DAFTAR TABEL

Tabel 2. 1 Penelitian Sebelumnya 27
Tabel 4. 1 Hasil Pemindaian *Port* 38
Tabel 4. 2 *OWASP TOP 10* 39
Tabel 4. 3 Hasil *Vulnerability OWASP ZAP* 40
Tabel 4. 4 Hasil *Vulnerability OWASP ZAP* (Lanjutan) 41
Tabel 4. 5 *Payload SQLI* 49
Tabel 4. 6 Hasil Pengujian *Scanning Vulnerability* 61
Tabel 4. 7 Hasil Pengujian *Scanning Vulnerability* (Lanjutan) 61
Tabel 4. 8 Kerentanan *OWASP ZAP* dan Hasil *Exploitation* 62

DAFTAR GAMBAR

Gambar 2.1 *CIA Triad* 8
Gambar 2.2 *Penetration Execution Standart (PTES)* 11
Gambar 2.3 *OWASP Zed Attack Proxy* 13
Gambar 2.4 *Proses Kerja OWASP ZAP* 15
Gambar 2.5 *Network Mapper (NMAP)* 17
Gambar 2.6 Cara Kerja *NMAP* 18
Gambar 2.7 *Whois* 19
Gambar 2. 8 *Visual Studio Code* 21
Gambar 3. 1 Konsep Penelitian *PTES* 32
Gambar 4. 1 Pengujian *OWASP ZAP* 40
Gambar 4. 2 Hasil Uji Injeksi Comments 50
Gambar 4. 3 *Url Login kalipelus-banjarnegara.desa.id* 50
Gambar 4. 4 *Url Login kalipelus-banjarnegara.desa.id or 1 = 1--* 51
Gambar 4. 5 Hasil Setelah ditambahkan karakter khusus pada *URL* 51
Gambar 4. 6 Serangan *Cross Site Scripting* Gagal 54
Gambar 4. 7 Hasil serangan script *HTML* Tidak Berhasil 56
Gambar 4. 8 Hasil Pengujian *Exploitasi CSRF* 59
Gambar 4. 9 Hasil Pengujian *Exploitasi Cross Site Scripting* 60

DAFTAR ISTILAH

Absence of Anti-CSRF Tokens	:	Kondisi di mana formulir atau permintaan pada aplikasi *web* tidak dilengkapi dengan token keamanan unik untuk mencegah serangan *Cross-Site Request Forgery (CSRF).* Tanpa token ini, aplikasi rentan terhadap pemalsuan permintaan dari pihak tidak berwenang.
Brute Force	:	Teknik serangan yang dilakukan dengan mencoba berbagai kombinasi kata sandi atau kredensial secara sistematis hingga menemukan kombinasi yang valid untuk mengakses sistem secara tidak sah.
Clickjacking	:	Serangan keamanan yang memanfaatkan lapisan transparan atau elemen *UI* tersembunyi untuk menipu pengguna agar mengklik elemen yang tidak disadari, sehingga mengeksekusi aksi yang tidak diinginkan pada halaman *web* target.
Content Security Policy (CSP)	:	Kebijakan keamanan berbasis header *HTTP* yang membatasi sumber eksternal yang diizinkan untuk memuat konten (seperti *skrip*, gambar, atau *stylesheet*) pada halaman *web* guna mencegah serangan *XSS* dan injeksi konten berbahaya.
Cross Site Request Forgery (CSRF)	:	Serangan keamanan pada aplikasi *web* yang memaksa pengguna yang telah terautentikasi untuk mengirimkan permintaan tanpa disadari ke server target. Serangan ini terjadi karena aplikasi mempercayai permintaan yang datang dari *browser* pengguna tanpa melakukan validasi tambahan. Penyerang memanfaatkan sesi *login* yang masih aktif dengan menyisipkan permintaan berbahaya melalui tautan, gambar, atau form tersembunyi. Akibatnya, server mengeksekusi aksi seperti pengiriman

		data, perubahan konfigurasi, atau input komentar seolah berasal dari pengguna yang sah. *CSRF* melanggar prinsip integrity dan confidentiality karena tindakan dilakukan tanpa persetujuan pengguna. Pencegahan *CSRF* dilakukan dengan penerapan token unik pada setiap permintaan, validasi origin dan referer, serta pembatasan metode *HTTP* yang sensitif.
Cross Site Scripting (XSS)	:	Kerentanan keamanan pada aplikasi *web* yang memungkinkan penyerang menyisipkan dan menjalankan skrip berbahaya di browser pengguna lain. Serangan ini terjadi ketika aplikasi gagal memvalidasi dan memfilter input pengguna dengan benar. Skrip yang disisipkan biasanya berupa *JavaScript* dan dijalankan dalam konteks domain yang sah. Dampaknya meliputi pencurian *cookie* sesi, pengambilalihan akun, manipulasi tampilan halaman, dan pengalihan pengguna ke situs berbahaya. *XSS* melanggar prinsip confidentiality dan integrity karena data pengguna dapat diakses dan diubah tanpa izin. Pencegahan *XSS* dilakukan dengan validasi input yang ketat, encoding output, penerapan *Content Security Policy (CSP),* serta pembatasan eksekusi skrip dari sumber tidak tepercaya.
Exploitation	:	Tahapan dalam *penetration testing* yang bertujuan memanfaatkan kerentanan yang telah diidentifikasi untuk mendapatkan akses tidak sah, mengekstrak data, atau menguji dampak serangan terhadap sistem.
Denial of Service (DoS)	:	Serangan yang bertujuan mengganggu ketersediaan layanan sistem dengan membanjiri target menggunakan lalu lintas jaringan berlebihan atau mengeksploitasi

		kelemahan sistem sehingga pengguna sah tidak dapat mengakses layanan tersebut.
Header HTTP	:	Bagian dari protokol *HTTP* yang berisi metadata mengenai permintaan atau respons, seperti jenis konten, kebijakan keamanan, informasi server, dan cookie. Header keamanan seperti *CSP, HSTS*, dan *X-Frame-Options* berperan penting dalam melindungi aplikasi *web*.
Injection	:	Kerentanan yang terjadi ketika data yang tidak tervalidasi dimasukkan ke dalam perintah atau query sistem (seperti *SQL*, *OS command*, atau *LDAP*), memungkinkan penyerang mengeksekusi perintah berbahaya. *SQL Injection* merupakan bentuk paling umum dari serangan *injection*.
Intelligence Gathering	:	Tahap awal *penetration testing* yang bertujuan mengumpulkan informasi tentang target (seperti alamat IP, domain, layanan yang berjalan, versi perangkat lunak) untuk mendukung perencanaan pengujian keamanan selanjutnya.
NMAP (Network Mapper)	:	Perangkat lunak open source untuk pemindaian jaringan yang digunakan mengidentifikasi *host* aktif, *port* terbuka, layanan yang berjalan, dan sistem operasi target melalui analisis paket jaringan.
OWASP (Open Web Application Security Project)	:	Organisasi nirlaba yang menyediakan standar, metodologi, dan alat untuk meningkatkan keamanan aplikasi *web*. *OWASP Top 10* merupakan daftar kerentanan keamanan aplikasi *web* paling kritis yang diperbarui secara berkala.
Penetration Testing	:	Proses pengujian keamanan sistem secara legal dan terukur dengan mensimulasikan serangan nyata untuk

		mengidentifikasi kerentanan, mengevaluasi dampaknya, dan memberikan rekomendasi perbaikan.
Penetration Testing Execution Standard (PTES)	:	Kerangka kerja standar untuk *penetration testing* yang terdiri dari tujuh fase: *pre-engagement interaction, intelligence gathering, threat modeling, Vulnerability analysis, exploitation, post-exploitation,* dan *reporting*.
Port	:	Nomor logika pada protokol jaringan yang digunakan untuk mengidentifikasi layanan atau aplikasi tertentu yang berjalan pada sebuah host (misalnya *port 80* untuk *HTTP, port 443* untuk *HTTPS*).
SQL Injection	:	Teknik serangan yang memanfaatkan input yang tidak divalidasi pada aplikasi *web* untuk menyisipkan perintah *SQL* berbahaya ke dalam *query database*, memungkinkan penyerang mengakses, memodifikasi, atau menghapus data secara tidak sah.
SSL/TLS (Secure Socket Layer / Transport Layer Security)	:	Protokol kriptografi yang menyediakan komunikasi aman melalui jaringan dengan mengenkripsi data yang ditransmisikan antara klien dan server, ditandai dengan penggunaan protokol *HTTPS* pada *URL website*.
Vulnerability	:	Kelemahan pada sistem, aplikasi, atau konfigurasi yang dapat dimanfaatkan oleh pihak tidak berwenang untuk melanggar keamanan. Kelemahan ini muncul akibat kesalahan desain, implementasi kode, konfigurasi yang tidak tepat, atau pengelolaan sistem yang buruk. *Vulnerability* memungkinkan terjadinya serangan yang berdampak pada kebocoran data, perubahan data, atau gangguan layanan. Tingkat risiko *Vulnerability* ditentukan oleh tingkat kemudahan eksploitasi dan besarnya dampak yang ditimbulkan. Identifikasi

		Vulnerability dilakukan melalui pengujian keamanan seperti *Vulnerability scanning* dan *penetration testing*.
Web Application Firewall (WAF)	:	Sistem keamanan yang berfungsi sebagai filter antara pengguna dan aplikasi *web* untuk mendeteksi serta memblokir lalu lintas berbahaya seperti *SQL Injection*, *XSS*, dan serangan *DDoS* sebelum mencapai aplikasi.

DAFTAR LAMPIRAN

Lampiran 1. Surat Izin Penelitian
Lampiran 2 Hasil Wawancara
Lampiran 3 Dokumentasi Wawancara
Lampiran 4 Dokumentasi Observasi
Lampiran 5 Kartu Bimbingan Dosen Pembimbing 1
Lampiran 6 Kartu Bimbingan Dosen Pembimbing 2