BAB II
KAJIAN LITERATUR

2.1. Kajian Pustaka

Bab ini akan membahas beberapa definisi yang berkaitan dalam penelitian ini seperti *framework* aplikasi website, laravel, ISSAF dan definisi lainnya serta bab ini akan menampilkan beberapa penelitian yang sejenis yang digunakan penulis sebagai perbandingan dan referensi dalam penelitian. Berikut ini pembahasan lengkapnya:

2.1.1. Aplikasi Manajemen

Pada penelitian ini penulis memiliki suatu projek aplikasi manajemen asset, dimana aplikasi tersebut dalam pengembangan dengan berbasis web yang menggunakan *framework* laravel. Aplikasi manajemen ini bertujuan agar setiap aset yang dimiliki dapat terdata dan tersusun secara teratur sehingga aset yang dimiliki tidak mengalami ketidakjelasan yang kemudian aplikasi dapat diakses dan dipantau selama berjalan secara daring atau *online* [11].

2.1.2. Framework Laravel

Pada saat ini website memiliki beberapa *framework* dalam pembuatannya, seperti codeigniter, laravel, bootstrap dan lainnya. Dalam penelitian ini aplikasi manajemen asset berbasis web dalam pengembangan dengan menggunakan *framework* laravel versi 8, yang merupakan sebuah kerangka kerja *open-source* yang diciptakan oleh Taylor Otwell berbentuk *framework bundle* berbasis PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya [12] [13]. Berdasarkan faktor tersebut maka Laravel memiliki keunggulan yang akan meningkatkan kecepatan pengembangan web, performance lebih cepat, reload data lebih stabil, memiliki keamanan data, menggunakan fitur canggih seperti blade menggunakan konsep HMVC (Hierarchical Model View Controller), yang rersedia pada *library* yang telah siap untuk digunakan serta adanya fitur pengelolaan migrations untuk pembuatan skema table pada database [13].

2.1.3. Kerentanan Keamanan

Hal yang tidak lepas dari suatu aplikasi yang telah dibuat yaitu bagaimana mengatasi kerentanan keamanan dari suatu aplikasi tersebut, banyak cara bagi

pihak yang tidak bertanggung jawab (*attacker*) untuk mencari kelemahan dalam aplikasi sehingga dapat menimbulkan berbagai kerugian dan kerusakan. Dalam melakukan keseluruhan tahapan untuk menguji keamanan sistem mengacu pada uji penetrasi (*penetration testing flow);* yang dalam mengimplementasikannya dapat dikelompokkan seperti *planning* (perencanaan), *attack* (penyerangan), *discover* (menemukan),dan *report* (laporan) [14]. Secara umum, uji penetrasi (pentest) dapat dikategorikan menjadi 3 (tiga) tahapan utama [15]:

1. **Sebelum Penyerangan (*pre attack*)**

Tahapan awal yaitu menyelidiki dan mencari berbagai informasi yang berkaitan dengan target sebagai persiapan dalam penyerangan seperti pengintaian yang dapat disekitar ping bahkan sampai menemukan IP alamat di jaringan, memperoleh informasi berguna dari karyawan perusahaan, mencari informasi di tempat sampah perusahaan untuk menemukan tanda terima telekomunikasi layanan, pencurian. Namun mencari informasi dibatasi oleh kemauan dan perilaku etis yang telah disepakati antara klien dan tim uji penetrasi.

2. **Penyerangan (*attack*)**

Setelah melakukan persiapan, penyerang akan langsung menggunakan informasi yang mereka dapat untuk memulai serangannya, seperti SQL Injection, Denial Of Service (DoS), Cross-site Scripting (XSS), dan berbagai cara lainnya.

3. **Setelah Penyerangan (*post attack*)**

Pada tahapan ini, tim pentest berbeda dengan penyerang, mereka kembali ke setiap tahapan sebelum terjadi modifikasi ketika ingin melakukan pengujian atau pentest. Dari tahapan tersebut, uji penetrasi memiliki metode dasar sebagai berikut:

Gambar 1. Metode Dasar Penetrasi Testing

2.1.4. Jenis Penetrasi

Ada beberapa jenis penetrasi yang dapat dilakukan oleh seorang penguji sesuai apa yang dapat dilakukannya, berikut jenis-jenis penetrasi [16]:

2.1.4.1.Black Box Testing

Pada pendekatan ini penguji tidak memiliki pengetahuan tentang target yang akan diuji. Penguji hanya mencari tahu semua celah keamanan sistem berdasarkan dengan pengalaman maupun keahlian. Tujuan penguji pada dasarnya bertujuan untuk mengaudit keamanan dari eksternal dengan cara mensimulasikan sebagai *attacker*.

2.1.4.2.White Box Testing

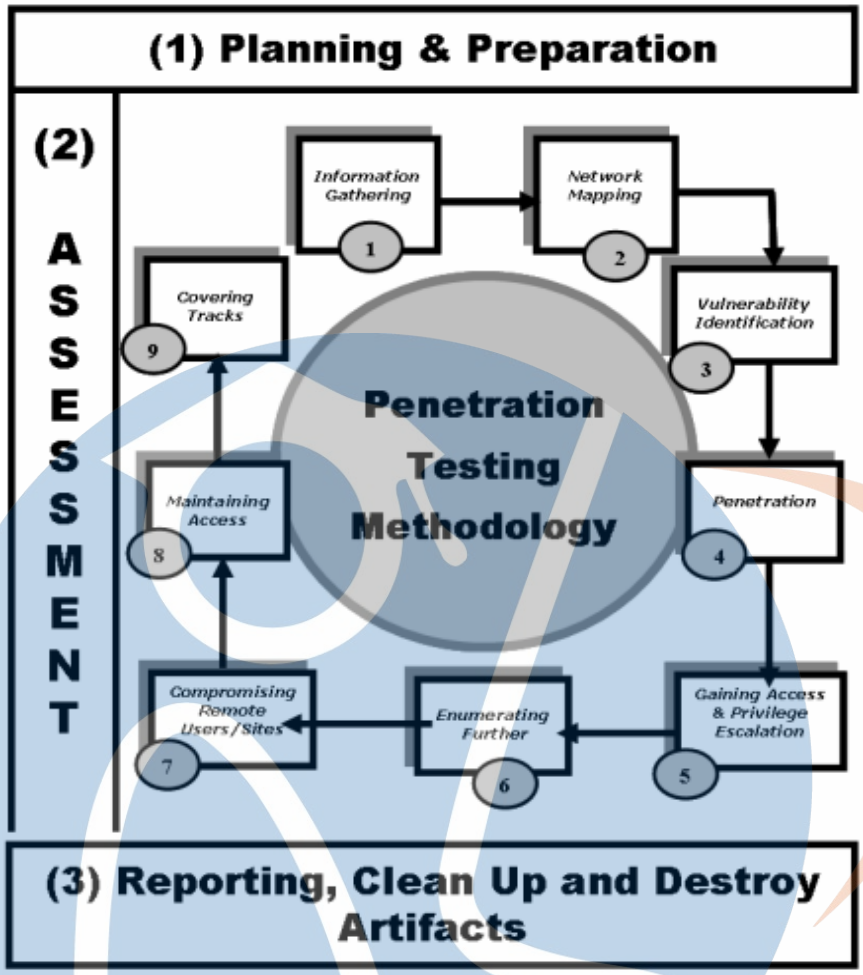
Pada pendekatan ini, seorang penguji diberikan semua informasi secara lengkap seperti konfigurasi jaringan, konfigurasi sistem dan penguji melakukan audit terhadap sistem keamanan dari internal. Penguji mensimulasikan tindakan tersebut seperti bahaya karyawan yang hadir dalam batas target maupun kebijakan. Pengujian ini memerlukan keahlian yang mendalam agar mendapatkan hasil yang lebih baik.

2.1.4.3.Grey Box Testing

Pada pendekatan ini adalah menggabungkan antara dua pendekatan Black Box Testing dengan White Box Testing. Pada pendekatan ini, seorang penguji harus memiliki pengetahuan tentang pengujian sebuah jaringan atau sistem.

2.1.5. ISSAF

Untuk memudahkan dalam menganalisa suatu kerentanan keamanan maka penelitian ini menggunakan kerangka pengujian penetrasi yaitu ISSAF (Information System Security Assessment Framework) yang dirancang untuk menganalisa sistem dengan 3 (tiga) fase umum, yaitu [17]:



Gambar 2. Rincian Information System Security Assessment Framework

2.1.5.1. PHASE 1 - PLANNING AND PREPARATION

Fase ini terdiri dari langkah-langkah untuk bertukar informasi awal, merencanakan dan mempersiapkan pengujian. Sebelum melakukan pengujian, akan melakukan yang akan ditandatangani dari kedua belah pihak. Hal ini akan memberikan dasar untuk penugasan dan perlindungan hukum timbal balik serta akan menentukan tim perikatan tertentu, tanggal pasti, waktu pengujian, jalur eskalasi dan pengaturan lainnya. Beberapa contoh yang dipertimbangkan dalam hal ini diantaranya:

1. Identifikasi kontak individu dari kedua sisi
2. Pertemuan pembukaan untuk mengkonfirmasi ruang lingkup, pendekatan dan metodologi.
3. Setuju dengan kasus uji dan jalur eskalasi tertentu.

2.1.5.2. PHASE 2 - ASSESSMENT

Dalam fase ini dimana akan benar-benar melakukan tes penetrasi yang terdiri dari 9 pengujian (*assessment*). Setiap langkah penilaian akan mewakili tingkat akses yang lebih besar terhadap informasi aset yang dimiliki, berikut 9 (sembilan) tahap *assessment:*

1. Information Gathering

Pengumpulan informasi pada dasarnya membutuhkan internet untuk menemukan semua informasi mengenai target (perusahaan dan/atau individual) dengan menggunakan cara teknis (DNS/WHOIS) dan cara non-teknis (mesin pencari, portal berita, mail, dll.). Hal ini merupakan tahap awal dari setiap audit keamanan informasi, yang cenderung diabaikan oleh banyak orang ketika melakukan segala jenis pengujian pada sistem informasi, pengumpulan informasi dan penambangan data yang sangat penting yang akan memberikan semua informasi yang mungkin membantu dalam melakukan tes. Saat melakukan pengumpulan informasi, penting untuk berimajinasi, seperti menjelajahi setiap jalan yang mungkin mendapatkan lebih banyak informasi tentang target dan sumber dayanya. Apa pun yang bisa didapatkan selama tahap pengujian ini berguna, seperti brosur perusahaan, kartu nama, selebaran, iklan surat kabar, dokumen internal, dan sebagainya. Pengumpulan informasi tidak mengharuskan penilai (assessor) untuk menjalin kontak dengan sistem sasaran. Informasi dikumpulkan (terutama) dari sumber publik di internet dan organisasi yang menyimpan informasi publik seperti agen pajak, perpustakaan, dll. Penilaian (assessment) umumnya terbatas dengan waktu dan sumber daya. Oleh karena itu, sangat penting untuk mengidentifikasi poin yang kemungkinan besar akan rentan, dan menjadi fokus penilaian mereka. Bahkan alat terbaik tidak berguna jika tidak digunakan secara tepat dalam waktu dan tempat yang tepat. Itu sebabnya penilai yang berpengalaman menginvestasikan sejumlah besar waktu dalam pengumpulan informasi.

2. Network Mapping

Sama halnya dengan proses sebelumnya, ketika semua informasi yang mungkin tentang target telah diperoleh, pendekatan yang lebih teknis diambil untuk menjejakkan (*footprint*) jaringan dan sumber daya yang menjadi pertanyaan bagi

penilai. Informasi spesifik jaringan dari informasi yang didapat sebelumnya dan diperluas untuk menghasilkan topologi jaringan yang mungkin bagi target. Banyak alat (*tools*) dan aplikasi yang dapat digunakan dalam tahap ini untuk membantu menemukan teknis informasi tentang jaringan yang terlibat dalam pengujian seperti host yang sedang berjalan, mengidentifikasi port atau layanan yang krusial, pemetaan jaringan (router, *firewall*), mengidentifikasi rute menggunakan *Manajement Information Base* (MIB).

3. Vulnerability Identification

Untuk memulai bagian ini, penilai akan memilih poin tertentu untuk menguji dan bagaimana mengujinya. Pada identifikasi kerentanan, penilai akan melakukan beberapa aktivitas untuk mendeteksi titik lemah yang dapat dieksploitasi dan kerentanan dapat diklasifikasikan berdasarkan risiko bisnis dan risiko teknis. Kegiatan pencarian kerentanan meliputi:

1. Identifikasi layanan yang rentan menggunakan spanduk layanan
2. Lakukan pemindaian kerentanan untuk mencari kerentanan yang diketahui.
3. Lakukan verifikasi positif palsu dan negatif palsu (contohnya dengan mengkorelasikan kerentanan satu sama lain dan dengan informasi yang diperoleh sebelumnya)
4. Identifikasi jalur serangan dan skenario untuk eksploitasi

Klasifikasi risiko kerentanan dapat dilihat pada table dibawah:

Tabel 1. Klasifikasi Risiko Kerentanan

	Risiko Rendah untuk Bisnis	Risiko Sedang untuk Bisnis	Risiko Tinggi untuk Bisnis
Risiko Teknis Tinggi	**Status: MED (Contohnya kemampuan sistem yang dapat dikompromi pada proses bisnis yang tidak penting)**	**Status: HIGH (Contohnya kemampuan sistem yang tidak dapat dikompromi pada proses bisnis penting)**	**Status: HIGH (Contohnya kemampuan system yang tidak dapat dikompromi pada proses bisnis kritis)**

Risiko Teknis Sedang	**Status: LOW (Contohnya kemampuan sistem yang dapat dikompromi pada proses bisnis yang tidak penting)**	**Status: MED (Contohnya kemampuan sistem yang dapat dikompromi pada proses bisnis yang penting)**	**Status: HIGH (Contohnya kemampuan system yang tidak dapat dikompromi pada proses bisnis kritis**
Risiko Teknis Rendah	**Status: LOW (Contohnya kebocoran non kritis pada sistem yang tidak penting untuk bisnis)**	**Status: LOW (Contohnya kebocoran non kritis pada sistem yang penting dalam mendukung bisnis)**	**Status: MED (Contohnya kebocoran pada sistem yang sangat penting dalam bisnis)**

4. Penetration

Penilai mencoba untuk mendapatkan akses yang tidak sah dengan menghindari keamanan dan mencoba untuk mencapai tingkat akses seluas mungkin. Beberapa kegiatan yang dapat dilakukan sebagai berikut:

1. Mengembangkan alat (*tools*)/skrip pengembangan

Beberapa keadaan, mungkin bagi penilai membuat alat dan skrip mereka sendiri untuk menghemat biaya.

2. Membuat dokumen

Dokumentasi ini akan berisi penjelasan rinci tentang jalur eksploitasi, dampak yang dinilai dan bukti adanya kerentanan.

5. Gaining Access and Privilege Escalation

Kegiatan dibagian ini akan memungkinkan penilai untuk mengkonfirmasi dan mendokumentasikan kemungkinan intrusi (penyusupan) dan/atau penyebaran serangan otomatis. Beberapa contoh cara mendapatkan akses ke akun yang tidak memiliki hak istimewa dengan:

1. Penemuan kombinasi nama pengguna/sandi seperti *dictionary attacks, brute force attacks*.
2. Kata sandi kosong atau kata sandi default dalam akun sistem.
3. Memanfaatkan pengaturan awal vendor seperti parameter konfigurasi jaringan, kata sandi, dan lainnya.
4. Menemukan *public services* yang memungkinkan operasi tertentu kedalam sistem seperti menulis/membuat/membaca file.
5. Bukti eksploitasi konsep yang diperoleh atau dikembangkan, diuji dalam lingkungan yang terisolasi, dan diterapkan pada sistem yang disusupi. Pada tahap ini tujuannya untuk mendapatkan hak administratif, hambatan yang dihadapi tingkat penambalan (*patch*) dan penguatan sistem; dan alat integritas sistem (termasuk antivirus) yang dapat mendeteksi bahkan dalam beberapa kasus dapat memblokir tindakan bukti eksploitasi.

6. Enumerating Further

Dalam tahapan ini, pentester akan melakukan pencacahan (*enumerating*) lebih lanjut untuk mencari kerentanan lainnya aseperti:

1. Mendapatkan kata sandi terenkripsi untuk cracking offline (contohnya dengan membuang SAM pada sistem Windows, atau menyalin etc/passwd dan /etc/shadow dari sistem Linux)
2. Mendapatkan kata sandi (atau terenkripsi) dengan sniffing atau teknik lainnya.
3. Melacak (*sniff*) lalu lintas dan menganalisisnya.
4. Mengumpulkan cookie dan gunakan untuk mengeksploitasi session dan untuk serangan kata sandi.
5. Pengumpulan alamat email
6. Mengidentifikasi routes dan jaringan
7. Memetakan jaringan internal.

7. Compromise Remote Users/Site

Satu lubang cukup untuk mengekspos seluruh jaringan, terlepas dari seberapa aman jaringan perimeter itu. Komunikasi antara pengguna/situs jarak jauh dan jaringan perusahaan dapat dilengkapi dengan autentikasi dan dienkripsi dengan menggunakan teknologi seperti VPN, untuk memastikan bahwa data ketika dalam perjalanan melalui jaringan tidak dapat dipalsukan atau disadap. Namun, tak menjamin titik akhir (*endpoints*) komunikasi tidak terganggu. Dalam skenario seperti itu, penilai harus mencoba berkompromi dengan pengguna jarak jauh, telekomunikasi jarak jauh, dan/atau situs jarak jauh suatu perusahaan. Mereka dapat memberikan akses istimewa ke jaringan internal.

8. Maintaining Access

Tahapan ini memberikan akses secara berlanjut setelah dapat mengeksploitasi sistem seperti:

1. Membuat Saluran Terselubung (*covert channels*)

Saluran rahasia juga dapat digunakan untuk menyembunyikan kehadiran pada sistem atau di jaringan. Saluran rahasia dapat berupa terowongan (*tunnels*) protokol (seperti icmp-tunnel, http-tunnel dll...) dan dapat menggunakan VPN tunnels. Lakukan langkah-langkah berikut untuk menggunakan saluran rahasia: Metodologi - siapkan saluran terselubung (*covert channel*) di jaringan target Identifikasi saluran terselubung yang dapat digunakan, Pilih alat terbaik yang tersedia untuk saluran terselubung, Uji saluran terselubung menggunakan teknik *Common Detection*.

2. Pintu Belakang (*backdoors*)

Pintu belakang dimaksudkan untuk selalu dapat kembali ke sistem tertentu, bahkan jika akun yang digunakan untuk meretas sistem tidak lagi tersedia (contohnya, telah dihentikan). *Backdoors* dapat dibuat dengan beberapa cara. Dengan menggunakan rootkit, dengan membuka port untuk ‘mendengarkan’ sistem target.

3. Rootkit

Rootkit akan memungkinkan untuk memiliki lebih banyak kekuatan daripada yang dilakukan oleh administrator sistem dari sebuah system dan dapat mengontrol sistem jarak jauh sepenuhnya. Seringkali rootkit juga memungkinkan penyembunyian file, proses dan/atau soket jaringan, sementara masih memungkinkan individu yang mengendalikan rootkit untuk mendeteksi dan menggunakan sumber daya tersebut.

Catatan: penggunaan saluran penutup (*covert channels*), pemasangan backdoors dan penyebaran rootkit sering tidak dilakukan sebagai uji penetrasi, karena risiko jika salah satu dari saluran tersebut tetap terbuka atau setelah pengujian, dan dapat terdeteksi oleh penyerang yang lain.

9. Covering Tracks

Catatan: pada umumnya praktik secara normal selama pengujian penetrasi bertindak seterbuka mungkin (kecuali jika diminta oleh klien) dan untuk menghasilkan informasi dan catatan terperinci dari semua aktivitas, jadi bagian di bawah ini sebagian besar untuk tujuan referensi:

1. Sembunyikan File

Menyembunyikan file penting jika penilai keamanan perlu menyembunyikan aktivitas yang telah dilakukan selama ini dan setelah merusak sistem dan untuk mempertahankan sembali saluran. Ini juga penting untuk menyembunyikan alat sehingga ini tidak perlu diunggah ke server target setiap kali.

2. Hapus Log

Pentingnya tahap ini mudah dipahami tetapi biasanya diremehkan, setelah penyerang berhasil mengkompromikan sistem, dia akan menyimpannya tanpa memberi tahu administrator, untuk alasan yang jelas. Semakin lama penyerang tetap berada di sistem yang disusupi, semakin baik peluang dia untuk dapat mencapai tujuannya lebih jauh di jaringan. Selama proses kompromi sistem, beberapa aktivitas yang mencurigakan dan/atau salah dicatat. Seorang penyerang yang terampil tahu bahwa log perlu diolah. Dia memodifikasinya untuk menutupi jejaknya dan menipu kehadirannya.

Catatan: ini hanya efektif jika tidak ada server *Syslog* jarak jauh yang digunakan. Jika ya, server *Syslog* jarak jauh ini harus diretas dan dibersihkan juga.

3. Menonaktifkan pemeriksaan integritas

Dalam kasus di mana pemeriksaan integritas statis oleh sistem seperti *Tripwire* telah diterapkan, sangat sulit untuk membuat perubahan apa pun pada sistem tanpa terdeteksi dan dilaporkan. Namun, jika penyebaran alat integritas sistem tidak dilakukan dengan benar, contohnya dengan meninggalkan file dengan tanda tangan file dan program yang valid diserver yang sama, dimungkinkan untuk memodifikasi sistem dan membuat ulang tanda tangan.

4. Menonaktifkan Anti-Virus (AV)

Saat ini, di sebagian besar stasiun kerja dan server, terdapat perangkat lunak Anti-Virus yang melindungi sistem dari perangkat lunak berbahaya yang terkenal (seperti exploit, viri, worm, dll), fokus dari langkah ini dalam pengujian penetrasi adalah untuk dapat menonaktifkan atau mengalahkan perangkat lunak AV sehingga penguji dapat melakukan aktivitas tanpa hambatan, dan kemungkinan untuk mengaktifkan kembali AV nanti. Di sebagian besar solusi AV yang dikelola secara terpusat, perangkat lunak AV dimulai ulang setelah jumlah waktu ketika dihentikan oleh penguji. “Masa tenggang” memungkinkan penguji untuk melakukan beberapa tugas agar perangkat lunak AV tetap dinonaktifkan untuk jangka waktu yang lebih lama. Kemungkinan hal yang dapat dilakukan oleh penguji (sebagian besar memerlukan akses tingkat Administrator) seperti buat file batch sehingga layanan AV dihentikan setiap 30 detik, nonaktifkan layanan AV, Blokir port manajemen pusat, Menerapkan Rootkit seperti eksploitasi POC, harus disesuaikan agar dapat sepenuhnya mencakup aktivitas penguji. Dalam kebanyakan kasus jika ada patroli AV, rootkit (biasanya pada win32) akan terdeteksi sebelum instalasi. Jadi, memodifikasi rootkit diperlukan di sebagian besar situasi. Penting juga untuk memperhatikan bahwa beberapa rootkit tidak akan berfungsi pengaturan sistem yang berbeda. Contohnya root-kit yang

digunakan dapat bekerja pada win2k-SP3 tetapi tidak dapat mencakup apa pun pada SP4.

2.1.5.3. PHASE 3 - REPORTING, CLEANING UP AND DESTROY ARTIFACTS

Dalam fase ini akan melampirkan laporan apa saja yang telah dilalui kemudian membersihkan dan menghancurkan artifak yang berkaitan dengan pengujian sebelumnya sehingga sistem berjalan normal.

2.1.5.3.1. REPORTING

Laporan (*reporting*) dalam metode ISSAF sebagai berikut:

1. **Laporan Verbal**

Ketika dalam pengujian penetrasi jika masalah kritis telah teridentifikasi harus segera dilaporkan untuk memastikan bahwa organisasi mengetahuinya. Pada titik ini, kekritisan masalah harus didiskusikan dan melakukan tindakan pencegahan untuk melindungi dari masalah.

2. **Laporan Akhir**

Setelah selesainya semua kasus uji yang telah ditentukan dalam ruang lingkup pekerjaan/pentest, laporan tertulis yang menjelaskan hasil rinci dari pengujian dan tinjauan harus disiapkan dengan rekomendasi untuk melakukan perbaikan. Laporan harus didokumentasikan dengan baik dan terstruktur. Hal-hal yang harus dipastikan dalam laporan adalah bagian-bagian:

1. Ringkasan Manajemen
2. Lingkup proyek (dan bagian Di Luar Lingkup)
3. Alat yang telah digunakan (termasuk eksploit)
4. Tanggal & waktu pengujian aktual pada sistem
5. Setiap keluaran dalam pengujian yang dilakukan (tidak termasuk laporan pemindaian kerentanan; dapat disertakan sebagai lampiran)

2.1.5.3.2. CLEANING UP AND DESTROY ARTIFACTS

Semua informasi yang dibuat dan/atau disimpan pada sistem yang diuji harus dihapus dari sistem. Jika karena alasan tertentu tidak memungkinkan dihapus dari

sistem jarak jauh, maka semua file pengujian (dengan lokasinya) harus disebutkan dalam laporan teknis sehingga staf teknis klien dapat menghapusnya setelah laporan diterima.

2.2. Penelitian Terkait

Dalam penelitian ini, ada beberapa referensi yang digunakan sebagai acuan serta membandingkan dengan beberapa penelitian terkait dengan topik yang dibawa dalam penelitian ini. Beberapa penelitian memiliki kesamaan seperti tujuan yang ingin dicapai dalam penelitiannya, yaitu memantau suatu sistem keamanan namun, dengan pendekatan dan studi kasus yang berbeda- beda. Berikut tabel perbandingan dari penelitian terkait yang digunakan Semua informasi yang dibuat dan/atau disimpan pada sistem yang diuji harus dihapus dari sistem. Jika karena alasan tertentu tidak memungkinkan dihapus dari sistem jarak jauh, maka semua file pengujian (dengan lokasinya) harus disebutkan dalam laporan teknis sehingga staf teknis klien dapat menghapusnya setelah laporan diterima:

Tabel 2. Penelitian Terkait

No	Penelitian	Tools	Kesimpulan
1.	ANALISIS PERBANDINGAN METODE WEB SECURITY PTES, ISSAF DAN OWASP DI DINAS KOMUNIKASI DAN INFORMASI KOTA BANDUNG Tio Revolino Syarif, Didit Andri Jatmiko,	OWASP, Accunetix	Penelitian ini membandingkan berbagai metode web security dimana penelitian ini menggunakan diskominfo.bandung.go.id sebagai penetrasi testing untuk mencari kerentanan keamanan website. Dari hasil penelitian, bahwa website tersebut lebih cocok dengan metode PTES dan OWASP dikarenakan lebih mudah dipahami dibandingkan

	Universitas Komputer Indonesia		metode ISSAF dikarenakan hasil pengujian kerentenan berbentuk tingkatan level.
2.	Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF I Gede Ary Suta Sanjaya, Gusti Made Arya Sasmita, Dewa Made Sri Arsa, Agusutus 2020	Who is Domain, IP lookup Scanner, NMAP, Owaspzap, Web Site Analysis, Vega	Penelitian ini melaporkan bahwa kerentanan pada website lembaga tersebut berbahaya yang dapat dieksekusi dengan SQL Injection dan XSS, serta celah lainnya yaitu port yang terbuka dan bug.
3.	ANALISIS KEAMANAN WEB SERVER OPEN JOURNAL SYSTEM (OJS) MENGGUNAKAN METODE ISSAF DAN OWASP (STUDI KASUS OJS UNIVERSITAS LANCANG KUNING) Guntoro, Loneli Costaner, dan Musfawati, Juni 2020	Whois Domain, SSL Scan, SQL Map, Zenmap, Acunetix,	Penelitian ini mencari kerentanan terhadap web server yang menjalankan sistem Open Journal System (OJS) pada website journal.unilak.ac.id namun ISSAF framework assestmentnya tidak dijalankan secara menyeluruh dan hasil penelitian ini mengatakan bahwa website tersebut masih tergolong aman namun masih rentan akan serangan seperti DoS (Denial of Service)

4.	SISTEM MONITORING WEBSITE DENGAN METODE ISSAF DI DINAS KOMUNIKASI dan INFORMATIKA KABUPATEN TANGERANG Jajang Ruhiyat Angga Setiyadi, 2017	Whois, Nikto, Hackertarget api	Penelitian ini bahwa keamanan web aplikasi pada DISKOMINFO Kabupaten Tangerang memiliki kerentanan SQL injection dan Cross Site Scripting (XSS), yang memungkinkan bagi seorang attacker mengambil alih system yang dikelola pihak DISKOMINFO

Dari tabel yang diatas, menjelaskan bahwa penelitian yang diambil oleh penulis tidak meniru atau plagiarisme namun, penulis juga tidak melakukan penelitian dari awal melainkan mengembangkan dari penelitian – penelitian yang telah ada sebelumnya. Penelitian ini paling mendekati penelitian dari I Gede Ary Suta Sanjaya, penelitian Jajang Ruhiyat serta penelitian Guntoro yaitu menganalisa kerentanan keamanan web dengan menggunakan metode ISSAF. Perbedaan penelitian ini yaitu pada web yang digunakan dimana web yang dianalisa merupakan web projek yang dikembangkan bersama dengan teman kemudian dianalisa kerentanan keamanannya menggunakan ISSAF. Diharapkan dari penelitian ini dapat memberikan manfaat dalam menganalisa web yang mungkin pembaca akan mengembangkan aplikasi berbasis web secara mandiri sehingga perlu melakukan uji penetrasi kerentanan keamanan yang ada pada aplikasi tersebut sehingga dapat memudahkan ketika melakukan perawatan (*maintenance)* website.

2.3. Posisi Penelitian

Untuk memudahkan dalam memahami posisi penelitian ini dengan penelitian terkait maka ditampilkan dengan menggunakan tabel.

Tabel 3. Posisi Penelitian

No	OWASP	ISSAF	BLACKBOX	WHITEBOX	WSL2
1	Guntoro, Loneli Costaner, dan Musfawati (Teknik Informatika, Fakultas Ilmu Komputer, Universitas Lancang Kuning) ANALISIS KEAMANAN WEB SERVER OPEN JOURNAL SYSTEM (OJS) MENGGUNAKAN METODE ISSAF DAN OWASP (STUDI KASUS OJS UNIVERSITAS LANCANG KUNING)				
2		I Gede Ary Suta Sanjaya, Gusti Made Arya Sasmita, Dewa Made Sri Arsa (Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana) Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF			
3		Jajang Ruhiyat, Angga Setiyadi (Universitas Komputer Indonesia) SISTEM MONITORING WEBSITE DENGAN METODE ISSAF DI DINAS KOMUNIKASI dan INFORMATIKA KABUPATEN TANGERANG			
4		Raihan Sabiq Rabbani (Teknik Informatika, STT Nurul Fikri) ANALISIS KERENTANAN KEAMANAN APLIKASI MANAJEMEN ASSET BERBASIS WEB MENGGUNAKAN METODE ISSAF (INFORMATION SYSTEMS SECURITY ASSESSMENT FRAMEWORK) STUDI KASUS PT.XYZ			