1. PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi merupakan bagian terpenting dari perkembangan kehidupan manusia pada abad dua puluh (20). Hal ini didukung oleh pendapat dari Guru Besar Filsafat Sekolah Tinggi (STF) Diriyakara, Franz Magnis-Suseno bahwa ada 2 (dua) hal yang menjadi alasan peradaban manusia tidak bisa dipisahkan dengan teknologi yakni pertama, manusia modern tidak memiliki pilihan lainnya untuk menjamin pemenuhan kebutuhan dasarnya selain pemanfaatan teknologi dan kedua, kemenangan (dominasi) budaya teknologi sudah tidak dapat dihindarkan (digagalkan lagi) (Riyanto, 2016). Manusia perlu memahami fakta bahwa kemajuan teknologi bukan untuk dihindarkan tetapi dimanfaatkan sebagai sarana penyebaran informasi dan data secara lebih efektif dan efisien. Teknologi informasi memiliki peran besar dalam membantu pekerjaan manusia dari yang awalnya berbentuk konvensional (tradisional) menjadi modern (digitalisasi), salah satunya pemanfaatan aplikasi sistem elektronik berbasis website yang semakin menjamur.

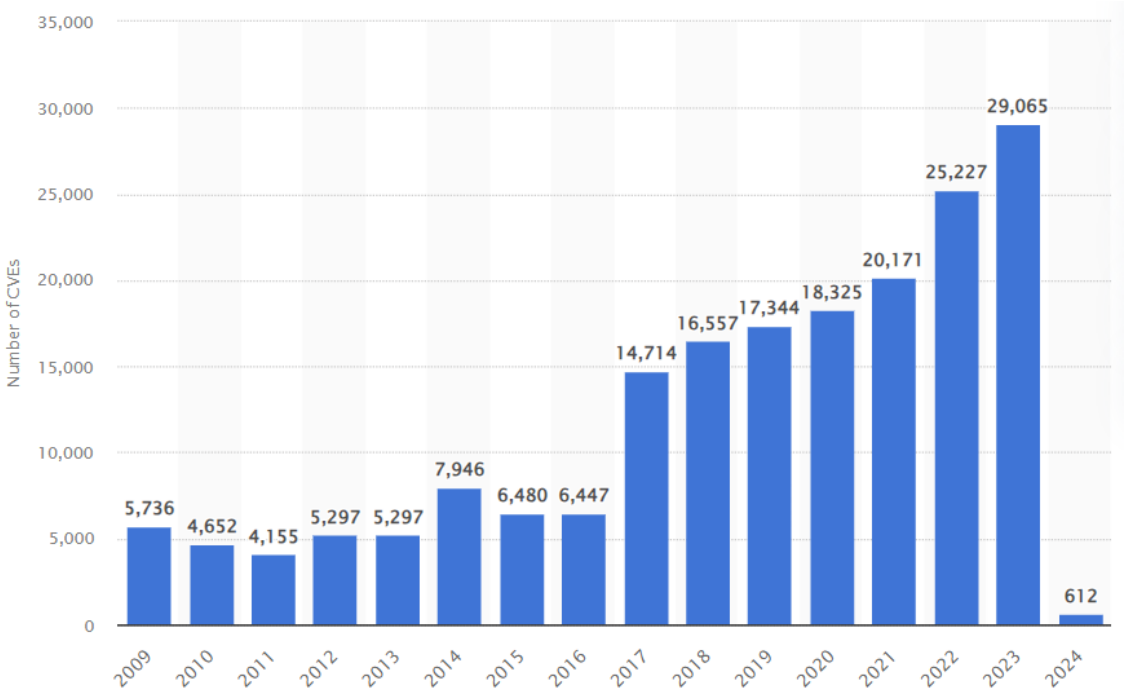
Perlu diketahui sisi lain dari implementasi teknologi informasi yang bisa bersifat destruktif. Dengan adanya penggunaan aplikasi berbasis website yang semakin marak sehingga berdampak positif terhadap pekerjaan manusia, namun dampak negatif yang bisa terjadi dalam hal keamanan informasi/data (*digital*). Dikutip dari artikel ilmiah yang ditulis oleh Jack Hance dan rekan-rekan mengenai *Distributed Attack Deployment Capability for Modern Automated Penetration Testing* bahwa dibutuhkan peningkatan teknik keamanan (pertahanan) terhadap serangan siber dan teknik peretasan yang semakin berkembang (canggih) (Hance, 2022). Aplikasi Website yang digunakan dalam hal-hal yang beresiko tinggi, berkaitan erat dengan data informasi pribadi *user/client* seperti data aktivitas bisnis, keuangan, perbankan, kesehatan dan sebagainya. Hal ini menunjukkan pentingnya keamanan aplikasi website sehingga menjamin keamanan data.

Pada beberapa tahun terakhir, ketika hampir seluruh data masyarakat Indonesia didigitalisasi maka banyak sekali serangan siber yang terjadi. Hingga bagian poin ini ditulis (Minggu, 5 November 2023) telah terjadi peretasan terhadap situs laman website Kementerian Pertahanan (KemHan) Republik Indonesia yakni kemhan.go.id. Diketahui bahwa pelaku peretasan yang menamai dirinya sebagai “Two2” telah berhasil mendapat akses terhadap data yang memang tidak bersifat rahasia sebesar 1,64TB dan dijual pada situs dark-web (Yuslianson,

2023). Menurut Ketua Lembaga Riset Keamanan Siber yaitu *Communication and Information System Security Research Center* (CISSReC), Pratama Persadha menyatakan bahwa kasus kebocoran data pada situs pemerintahan tersebut, menunjukkan adanya celah keamanan pada Kementerian Pertahanan RI, sehingga apabila tidak ditindaklanjuti dapat membahayakan keamanan serta kedaulatan negara (Dewi, 2022). Selain kasus yang terjadi pada website resmi pemerintah, juga terjadi pada website non-pemerintah baik itu bisnis maupun *non-profit company.*

Beragam kasus serangan siber dikelompokkan menjadi beberapa jenis teknik *exploitation*. Beberapa contoh diantaranya *Denial of Service* yakni penyerang melakukan *request* yang berlebihan (masif) terhadap situs website sehingga server website kesulitan menanggapi request tersebut. Hal ini menyebabkan proses kinerja server website menjadi lambat bahkan bisa terhenti. Pada hari Rabu, 24 Juni 2020, Badan Siber dan Sandi Negara (BSSN) Republik Indonesia (RI) mengkonfirmasi serangan terhadap situs website Dewan Perwakilan Rakyat (DPR) RI (www.dpr.go.id) telah mendapat serangan *Distributed Denial of Service* (DDOS) sehingga dapat diakses kembali malamnya pukul 22.08 WIB (Hafis, 2020). Selain itu, metode atau jenis *defacing* website juga merupakan hal yang paling sering dilakukan *hacker* yakni dengan mengubah tampilan website pemilik, sehingga menyebabkan turunnya kepercayaan *client* terhadap keamanan data serta reputasi pemilik website tersebut. Hingga kini salah satu website pemerintah daerah yakni Dinas Kesehatan Kabupaten Agam (https://dinkes.agamkab.go.id/readme.php) telah di *hack* oleh "Anon7" sehingga mengubah tampilan website menjadi tidak normal dan bersifat memberikan rasa takut (tampilan background dinamis skeleton). Hal yang sama (defacing website) juga terjadi pada situs website pemerintah daerah lainnya yakni kabupaten Luwu Timur (https://sirasul.luwutimurkab.go.id/readme.html07). Dari 2 kasus yang paling sering terjadi pada website pemerintahan, terdapat juga kasus peretasan yang membuat kegaduhan dan berdampak secara nasional bagi Negara Indonesia. Pada tahun 2022, seorang peretas yang menamakan dirinya "Bjorka" melakukan pencurian data peduli lindungi yakni salah satu aplikasi meta data kesehatan Indonesia, diluncurkan pada masa Covid-19 (*Coronavirus Disease*) dan menjual pada *blackmarket dark-web*. Hal ini kemudian dipamerkan berupa *sample* data dari pemimpin-pemimpin pemerintahan Indonesia (Yuslianson, 2023). Kasus-kasus peretasan di atas menunjukkan bahwa tingkat kejahatan siber (*cyber crime*) terus meningkat setiap tahunnya dan perlu adanya tindakan pencegahan (Analisa dan *Offensive Testing System*), *defensive* terhadap serangan dan *recovery* dari dampak serangan.

Penetration testing (PT - Pentest) merupakan sebuah metode aktif dalam mengakses dan mengevaluasi keamanan aset-aset *digital* seperti jaringan, web, server, aplikasi dan sebagainya. Sistem *penetration testing* atau percobaan serangan sebagai bagian untuk mengukur keamanan *digital* khususnya sebuah situs website perlu diimplementasikan oleh pemilik website. *Penetration testing* dilakukan dengan mencoba mengidentifikasi kerentanan (*vulnerabilities*) dan mencoba mengeksploitasi sebuah sistem atau aset digital yang menjadi objek *testing* (Ghanem, 2022). Berikut data statistika mengenai jumlah kerentanan dan eksposur keamanan IT secara umum di seluruh dunia sejak tahun 2009 hingga tahun 2024 (*Number of common IT security vulnerabilities and exposures / CVEs*)



Gambar 1.1 *Number of Common IT Security Vulnerabilities and Exposures* (CVEs) *Worldwide from* 2009 *to* 2024

Sumber: Petrosyan, A. (2024, January 9). *Number of common IT security vulnerabilities and exposures (CVEs) worldwide from 2009 to 2024 YTD*. Statista.com. https://www.statista.com/statistics/500755/worldwide-common-vulnerabilities-and-exposures.

Berdasarkan data *graphic bar* di atas, jumlah kerentanan tertinggi terjadi pada tahun 2023 sebanyak 29.065 dan jumlah kerentanan pada minggu pertama bulan Januari 2024 telah mencapai sebanyak 612 (Petrosyan, 2024). Oleh karena pertambahan jumlah tingkat kerentanan pada bidang IT sejak satu dekade terakhir menunjukkan pentingnya dilakukan

penetration testing oleh pemilik aset *digital* secara periodik atau rutin. Hal ini dapat mencegah terjadinya eksploitasi kerentanan oleh pihak–pihak yang tidak bertanggung jawab.

Penetration testing memiliki standar pengujian secara umum, yang sering menjadi *step by step* atau langkah-langkah dalam setiap jenis *penetration testing.* Langkah pertama pengumpulan dan analisis informasi (data dari target yang akan diuji), dilanjutkan perencanaan, pelaksanaan pengujian hingga membuat *report* atau laporan evaluasi dari hasil *penetration testing* yang dilakukan (Angir, Noertjahyana, & Andjarwirawan, 2015). Proses ini biasanya akan dilakukan oleh seorang teknisi siber yang hendak melakukan *penetration testing.* Waktu yang dibutuhkan oleh seorang *tester* akan berbeda-beda tergantung seberapa kompleks suatu sistem atau objek yang hendak diuji dan seberapa banyak akses yang dimiliki oleh *tester* terhadap objek yang diuji. Dalam mempercepat proses dasar seperti pengumpulan informasi dapat dibuat secara automatis sehingga mempermudah pekerjaan seorang *tester.* Namun, terdapat langkah-langkah dalam metode *penetration testing* yang perlu dilakukan oleh setiap penguji salah satunya dari proses pengumpulan informasi dan perencanaan pengujian. Pada perencanaan yang sama kemungkinan besar akan digunakan berulang kali pada tindakan *penetration testing* lainnya (Cooper, 2021). Pemanfaatan *script automation* yang dapat dijalankan secara berulang atau *repeat*, diharapkan mampu mengurangi proses atau tindakan yang berulang secara manual yang memakan waktu lebih lama. Seperti telah dibahas sebelumnya, bahwa tindakan peretasan website yang semakin marak dan terus bertambahnya kerentanan di bidang IT maka perlu dilakukan proses *penetration testing* dengan mudah dan cepat secara automasi, sehingga dapat dilakukan oleh *user/client* pemilik website baik secara mandiri maupun dengan bantuan *tester*.

Terdapat banyak metode dan tahapan atau *framework* (sistem kerja) yang digunakan untuk melakukan sebuah tindakan atau aktivitas *penetration testing.* Sistem pengembangan atau kerja mengenai *pentest* terhadap website yang popular sekarang yakni *Open Web/Worldwide Application Security Project* (OWASP). OWASP merupakan suatu komunitas *open source* (sumber daya terbuka) yang dibuat dengan tujuan membantu organisasi dalam mengembangkan, membeli dan memelihara aplikasi website yang dapat dipercaya dengan tingkat keamanan tinggi dan kerentanan rendah ("Tentang OWASP - OWASP Top 10:2021," 2021). OWASP sering digunakan oleh *pentester professional* (pengembang) dan ahli teknologi untuk mengamankan website. Hal ini menunjukkan bahwa OWASP memiliki riwayat penggunaan yang baik, sehingga dipercaya untuk dianalisa dan digunakan. Pengembangan OWASP menjadi lebih cepat karena tersedia sebagai *platform framework non-profit* yang *open-source.* Selain itu, *Penetration Testing Execution Standard* (PTES) juga merupakan salah satu

metode s*tandard* atau *framework* dasar yang masih jarang digunakan oleh instansi atau organisasi. Hal ini disebabkan karena sifat PTES yang *high level management* (memiliki langkah yang umum dan digunakan untuk sistem Perusahaan atau organisasi besar). Pada dokumentasi PTES sendiri, tidak memiliki detail *tool* atau pedoman teknis yang disarankan untuk melaksanakan *pentest* (PTES, 2014). Hal ini menyebabkan seorang *penetration testing* perlu melakukan kombinasi beberapa metode *pentest* lainnya dalam mengikuti langkah-langkah *standard* umum *penetration testing* berdasarkan PTES. Oleh karena itu, ketika metode pengembangan automasi *penetration testing* terhadap keamanan aplikasi website diperbandingkan antara OWASP dan PTES dengan mengkombinasikan berbagai *tools* yang diperlukan, diharapkan menjadi salah satu alternatif dalam melakukan *penetration testing* terhadap aplikasi website (melalui OWASP) serta memberikan langkah-langkah komprehensif beserta *evaluation report* sesuai PTES.

Melihat pentingnya faktor keamanan terhadap jaringan komputer maka selain *tool* yang dikembangkan, banyak sekali penelitian mengenai keamanan jaringan komputer. Penelitian berupa implementasi langkah-langkah hingga *literatur review* dengan berbagai metode telah banyak dilakukan oleh para profesional *cyber security* maupun mahasiswa. Beberapa penelitian sebelumnya telah mengidentifikasi berbagai metode dan alat yang digunakan dalam pengujian keamanan aplikasi website. Eksplorasi implementasi *tools* dan teknik tertentu secara automatis dengan pilihan beberapa *tools penetration testing* jarang dilakukan, dimana kebanyakan dari mereka fokus pada aspek manual dari *penetration testing*. Sebagai contoh, penelitian oleh Kuncoro dan Rahma (2022) menyoroti pentingnya automasi dalam pengujian keamanan dan mengeksplorasi aspek-aspek tertentu dari metode OWASP. Meskipun demikian, belum banyak penelitian yang secara khusus menekankan integrasi antara metode OWASP dan PTES dalam pengembangan *automation script* untuk *penetration testing*. Selain itu, sudah ada berbagai aplikasi dan *platform* yang mendukung pengujian keamanan, namun masih terdapat *research gap* terkait efektivitas dan efisiensi penggunaan automasi, terutama dalam konteks kombinasi metode OWASP dan PTES.

Dalam konteks *automation penetration testing* berbasis script, PTES dapat menjadi kerangka kerja yang berguna untuk memastikan bahwa pemeriksaan keamanan dilakukan dengan cara yang terstruktur dan komprehensif. Namun, masih ada pertanyaan tentang seberapa lama waktu yang dibutuhkan untuk menerapkan metode PTES secara efektif dalam automation penetration testing berbasis script. Mengingat bahwa *automation penetration testing* berbasis script sering kali digunakan untuk meningkatkan efisiensi dan konsistensi dalam

proses *penetration testing*, penting untuk mengevaluasi seberapa lama waktu yang diperlukan untuk menerapkan metode PTES dalam konteks ini. Pengerjaan *penetration testing* dari masa ke masa dilakukan secara manual dan lebih kompleks. Namun, pada kenyataannya, jumlah para ahli *penetration testing* tidak banyak tersedia, dan proses yang dilakukan manual memakan waktu yang lebih lama dan biaya yang cukup besar (Abu-Dabaseh & Alshammari, 2018). Hal ini akan membantu dalam memahami sejauh mana PTES dapat diintegrasikan ke dalam proses automation penetration testing secara efisien waktu tanpa mengorbankan keefektifan kualitas hasil.

Tabel 1.1

Perbandingan *Manual* dan *Automation Penetration Testing*

Aspek Perbedaan	***Automated***	***Manual***
Proses Pengujian	Proses cepat, standar dan pengujian yang mudah diulang.	Proses manual, tidak standar (tergantung subjek yang melakukan *pentest*), intensif modal dan biaya yang cukup tinggi.
Kerentanan Pengelolaan *Database* Serangan	*Database* serangan dipelihara dan diperbaharui.	Pemeliharaan *database* serangan secara manual, bergantung pada *database public.*
Pengembangan dan Pengelolaan Eksploitasi	Pengembangan dan pemeliharaan semua eksploitasi menggunakan *software* automasi, efektifitas eksploitasi maksimal dan aman untuk dijalankan dalam berbagai vektor atau aspek serangan.	Pengembangan eksploitasi membutuhkan waktu dan memerlukan keahlian yang signifikan. Eksploitasi publik dapat mencurigakan dan berpotensi tidak aman untuk dijalankan.
Pelaporan (*Reporting*)	Laporan-laporan diotomatiskan dan dapat disesuaikan.	Memerlukan pengumpulan data hasil uji secara manual.
Pembersihan (*Cleanup*)	Produk atau *tools* pengujian otomatis telah menawarkan solusi atau fitur *cleanup.*	Penguji harus secara manual mengurangi perubahan pada sistem setiap kali kerentanan ditemukan.

Modifikasi Jaringan	Sistem pengujian tetap, tidak berubah.	Seringkali menghasilkan banyak perubahan sistem dan alur pengujian.
*Logging/*Audit	Secara otomatis mencatat semua *history* aktivitas pengujian secara rinci.	Proses *audit* yang lama, merepotkan karena *manual* dan sering tidak akurat.
Pelatihan Penguji	Penguji menggunakan alat otomatis lebih mudah dan memiliki penjelasan mengenai *tools* yang digunakan.	Penguji perlu mempelajari cara pengujian yang tidak standar sehingga pelatihan dapat disesuaikan dan membutuhkan waktu.

Sumber: Abu-Dabaseh, F., & Alshammari, E. (2018). Automated Penetration Testing: An Overview. *Computer Science & Information Technology*, *8*(6), pp. 125–126. https://doi.org/10.5121/csit.2018.80610

Oleh karena itu, melatarbelakangi pemikiran diatas maka, seberapa lama waktu yang dibutuhkan untuk menerapkan metode *Penetration Testing Execution Standard* (PTES) berbasis pengembangan *automation script penetration testing* secara menyeluruh akan diketahui melalui penelitian ini. Seperti yang diketahui bahwa metode PTES memiliki *detail* aspek atau komponen *penetration testing* yang terstruktur dan menyeluruh. Selain itu, perbandingan keefektifan hasil dari *automation script* yang diterapkan dengan metode PTES dan penggunaan *tools* pada metode OWASP yang telah banyak dikembangkan, perlu dianalisa untuk melihat seberapa tepat hasil *penetration testing* yang diuji coba.

1.2 Rumusan Masalah

Mencermati pembedahan ketajaman latar belakang pemikiran yang telah diuraikan di atas, maka rumusan masalahnya adalah sebagai berikut:

1. Seberapa lama waktu penerapan metode *Penetration Testing Execution Standard* (PTES) dalam *automation penetration testing* berbasis *script* untuk memastikan pendekatan yang terstruktur dan menyeluruh?
2. Bagaimana perbandingan keefektifan hasil automasi *script penetration testing* dengan metode PTES dan metode OWASP?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah di atas maka tujuan dari penelitian ini adalah:

1. Menentukan durasi waktu yang diperlukan untuk menerapkan metode PTES dalam *automation penetration testing* berbasis *script*, guna menyediakan pemahaman yang jelas terkait efisiensi waktu pelaksanaan.
2. Membandingkan keefektifan hasil automasi *script penetration testing* dengan metode PTES dan menggunakan *tool* dari metode OWASP.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah menerapkan prosedur standar dari *Penetration Testing execution Standard* (PTES) secara automasi, sehingga mudah digunakan oleh *user* dengan latar belakang *pentester* maupun orang awam.

1.5 Ruang Lingkup

Ruang lingkup penelitian akan dibatasi pada beberapa hal sebagai berikut:

1. Automasi hanya akan berfokus pada 3 tahapan PTES yakni *intelligence* (*information*) *gathering, vulnerability analysis* dan *exploitation.*
2. *Penetration testing* akan berfokus pada *vulnerability* atau kerentanan tertentu. Berikut daftar kerentanan yang akan diuji atau dideteksi*,* antara lain:
   a. SQL *Injection.*
   b. *Brute Force.*
   c. *Cross Site Request Forgery (CSRF).*
   d. *File Inclusion.*
   e. *Cross-Site Scripting (DOM XSS, Reflected XSS dan Stored XSS).*
   f. *Authorisation Bypass.*
   g. Javascript*.*
   h. *Open Redirect URL HTML.*

   Berikut pembagian *tools* yang akan di-*import* beserta kerentanan *website* yang akan diuji:

Tabel 1.2

Kerentanan *Website* dan Penggunaan *Tools*

Kerentanan	***Tool***					
	OWASP ZAP	Kali Linux (*Meta-sploit)*	Burp Suite	SQLMap	Acunetix	Nesus VS
SQL *Injection*		✓		✓		
Brute Force		✓				
CSRF	✓	✓	✓			
File Inclusion	✓		✓	✓		
XSS	✓		✓		✓	
Authorisation Bypass	✓	✓	✓	✓		
Javascript	✓	✓	✓			✓
Open Redirect URL	✓		✓			

3. *Tools* (*software* dan *operating system*) yang akan digunakan untuk menyediakan *environment penetration testing,* sebagai berikut:
   - *Operating system* yang digunakan adalah Kali Linux yang telah dikembangkan sebagai *open source operating system* khusus untuk tujuan *hacking* dan *penetration testing.*
   - Kali Linux *Virtual Machine* (VMware) digunakan dalam membuat dan menjalankan *automation penetration script* serta sebagai wadah pemasangan *website target.*
   - *Damn Vulnerable Web App* (DVWA) merupakan *web application* (WA) dan *web server* (WS) PHP/MySQL yang akan digunakan sebagai objek simulasi *penetration testing*, sehingga berdasarkan aspek *legality*, proses *pentest* terhadap *DVWA* merupakan jenis

white box penetration testing karena sebagai *pentester* memiliki akses terhadap objek (DVWA) secara penuh.

4. Jenis *Automation Script* yang akan dibuat adalah *Bourne Again Shell* (*Bash*). *Bash* merupakan *Shell* atau *Script* paling umum yang digunakan pada *Operating System* Linux. Oleh karena itu, *Bash Scripting* sangat familiar dan mudah diimplementasi pada Linux.
5. Tahapan PTES dan *import tools* yang digunakan pada *Bash automation script*, antara lain:

Tabel 1.3

Segmentasi *Tools* pada Tahapan *Penetration Testing Execution Standard* (PTES)

Tools	**Tahapan PTES**						
	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**
Whois		✓					
Nmap		✓					
Acunetix				✓			
Kali Linux (Metasploit)					✓		
theHarvester		✓					
Nesus VS				✓			
OWASP ZAP				✓	✓		
Burp Suite		✓		✓			
Wireshark		✓		✓			
SQLMap					✓		
Tanpa *tools*	✓		✓			✓	✓

Sumber: Dewi, B. T. K., & Setiawan, M. A. (2022). *Kajian literatur: Metode dan tools pengujian celah keamanan aplikasi berbasis web* (p.5). Universitas Islam Indonesia.

Keterangan:

F1: *Pre-Engagement Interaction*

F2: *Intelligence Gathering*

F3: *Threat Modelling*

F4: *Vulnerability Analysis*

F5: *Exploitation*

F6: *Post Exploitation*

F7: *Reporting*

Catatan: Penggunaan *tools* akan disesuaikan dengan bagian *exploitation* dari objek *website* yang akan diimplementasi *penetration testing.*

6. Aspek *Penetration Testing Execution Standard* (PTES) akan divalidasi melalui perbandingan studi literasi untuk menilai apakah langkah yang dilakukan telah sesuai dengan standar keamanan atau dasar dari *penetration testing* yang terstruktur menyeluruh.
7. *Output* dari proyek ini adalah membuat *automation script* yang dapat digunakan untuk memudahkan proses *penetration testing,* serta menentukan berapa lama waktu yang dibutuhkan dalam implementasi PTES menggunakan bantuan *automation script.*
8. Kontribusi penelitian pada bidang akademik yakni mengevaluasi tingkat keamanan website dengan mengidentifikasi dan membandingkan keefektifan hasil *penetration testing* yang telah dilakukan menggunakan OWASP ZAP dan *automation script*.

1.6 Metodologi Penelitian

Langkah-langkah yang akan dilakukan mencakup metodologi penelitian dalam pengerjaan proyek, antara lain:

1. Studi Literatur. Tahap awal akan melakukan studi literatur yang mendalam tentang beberapa sebagai berikut:
   - Keamanan aplikasi web.
   - Daftar kerentanan dan cara kerja exploitasi pentest terhadap metode SQL *Injection, Brute Force, Cross Site Request Forgery (CSRF), File Inclusion, Cross-Site Scripting (DOM XSS, Reflected XSS dan Stored XSS), Authorisation Bypass,* Javascript dan *Open Redirect URL HTML.*
   - Metode *penetration testing* berbasis *script automation* dan *Bash Linux.*
   - *Penetration Testing Execution Standard* (PTES).
   - *Open Web/Worldwide Application Security Project* (OWASP).
   - *Open Web/Worldwide Application Security Project Zed Attack Proxy* (OWASP ZAP).
2. Melakukan *setting up tools* OWASP ZAP, Kali Linux *Virtual Machine* (VMware) dan *Damn Vulnerable Web App* (DVWA).

- *Download* dan *install software Kali Linux Virtual Machine.*
- *Download* dan *install software* OWASP ZAP.
- *Download* dan *install Web Application* dan *Web Server* DVWA.

3. Melakukan *import tools* yang akan digunakan ketika membuat *automation script* yakni:
   a. *Whois.*
   b. *Nmap.*
   c. *Acunetix.*
   d. *Kali Linux (Metasploit).*
   e. *theHarvester.*
   f. *Nesus VS.*
   g. *OWASP ZAP.*
   h. *Burp Suite.*
   i. *Wireshark.*
   j. *SQLMap.*
4. Pembuatan *automation script* menggunakan *script Bash*. Penelitian ini akan melibatkan pembuatan *automation script* untuk 3 tahapan PTES, antara lain:
   a. *Intelligence gathering.*
   b. *Vulnerability analysis*
   c. *Exploitation*.
5. Simulasi *Penetration Testing*. *Automation script* yang dikembangkan akan digunakan untuk melakukan simulasi pengujian penetrasi terhadap *web server* dan aplikasi web.
   - Lakukan pengumpulan data pada fase *Pre-engagement Interactions* dan validasi ruang lingkup *pentest* untuk keperluan perencanaan.
   - Melakukan fase *Intelligence Gathering.*
   - Melakukan fase *Threat Modeling* terhadap data sensitif.
   - Melakukan fase *Vulnerability Analysis.*
   - Melakukan fase *Exploitation* menggunakan *script* yang dibuat melalui Kali Linux.
   - Melakukan fase *Post Exploitation.*
   - Membuat laporan akhir berisi hasil *penetration testing.*
6. Perbandingan dan Analisis Hasil. Hasil dari pentest metode PTES menggunakan *automation script* akan dibandingkan dengan penggunaan metode OWASP menggunakan beberapa beberapa *tools* secara manual seperti OWASP ZAP.

7. Evaluasi dan Kesimpulan. Penelitian ini akan mengevaluasi hasil pengujian dan memberikan kesimpulan tentang efektivitas serta waktu yang diperlukan dalam implementasi metode *automation script* dibandingkan metode penggunaan *tools* secara manual (OWASP), kontribusi terhadap keamanan dan relevansi dengan kerangka kerja PTES secara terstruktur menyeluruh.
8. Membuat laporan akhir proyek penelitian dalam menjawab rumusan masalah sebelumnya.

1.7 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan untuk menyusun skripsi ini adalah sebagai berikut:

a. Bab 1 Pendahuluan. Membahas latar belakang permasalahan, perumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup, metode penelitian yang digunakan dan relevan terhadap penelitian ini.
b. Bab 2 Landasan Teori. Membahas teori-teori penunjang yang digunakan dalam membuat skripsi ini, serta relevan terhadap topik penelitian, antara lain adalah teori mengenai kejahatan siber (*cyber crime*), peretasan, *penetration testing / vulnerability testing, Penetration Testing Execution Standard* (PTES), *Open Web Application Security Project* (OWASP), Kali Linux, dan *Damn Vulnerable Web App* (DVWA).
c. Bab 3 Analisa Masalah dan Desain Sistem. Membahas metodologi yang digunakan dengan memberikan penjelasan mengenai teknik yang digunakan dalam penelitian ini, analisa permasalahan serta desain sistem/penetration testing yang digunakan dalam menyelesaikan masalah yang ada.
d. Bab 4 Implementasi Sistem. Membahas implementasi *penetration testing* dan penggunaan program *automation script* (PTES) serta penggunaan beberapa *tools* secara manual (OWASP) dalam penelitian berdasarkan desain sistem penetration testing yang ada pada bab sebelumnya.
e. Bab 5 Pengujian Sistem. Membahas pengujian dari program *automation script* yang telah ditentukan dan dibahas pada bab sebelumnya.
f. Bab 6 Kesimpulan dan Saran. Membahas kesimpulan dan saran yang didapat dari hasil pengujian yang telah dilakukan guna menjawab rumusan masalah pada bab sebelumnya.