2. LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada tinjauan pustaka akan dibahas teori-teori mengenai metode yang digunakan beserta penjelasan yang sesuai dengan penelitian.

2.1.1 Kejahatan Siber (*Cyber Crime*) dan Peretas (*Hacker*)

Kejahatan siber merupakan tindakan atau kegiatan dengan maksud jahat terhadap ketahanan dan keamanan sebuah aset digital atau teknologi informasi (Pertiwi, 2022). Berdasarkan Undang-Undang (UU) Nomor 11 tahun 2008 Pasal 30 tentang Informasi dan Transaksi Elektronik (ITE), bahwa:

- Ayat (1) “Setiap orang dengan sengaja dan tanpa hak atau melawan hukum mengakses komputer dan/atau Sistem Elektronik milik orang lain dengan cara apa pun”.
- Ayat (2) “Setiap orang dengan sengaja dan tanpa hak atau melawan hukum mengakses Komputer dan/atau Sistem Elektronik dengan cara apa pun dengan tujuan untuk memperoleh Informasi Elektronik dan/atau Dokumen Elektronik”.
- Ayat (3) “Setiap orang dengan sengaja dan tanpa hak atau melawan hukum mengakses Komputer dan/atau Sistem Elektronik dengan cara apa pun dengan melanggar, menerobos, melampaui, atau menjebol sistem pengamanan”.

Hal selanjutnya dipertegas pada Pasal 32 ayat 1 UU No. 11/2008 bahwa “Setiap orang dengan sengaja dan tanpa hak atau melawan hukum dengan cara apa pun mengubah, menambah, mengurangi, melakukan transmisi, merusak, menghilangkan, memindahkan, menyembunyikan suatu Informasi Elektronik dan/atau Dokumen Elektronik milik orang lain atau milik publik”. Berdasarkan 2 pasal UU ITE yang diatur oleh pemerintah Indonesia tersebut, telah jelas secara eksplisit mengenai tindakan-tindakan yang melawan hukum dalam bidang (Informasi dan Transaksi Elektronik) ITE (KOMINFO, 2023). Pada lanjutan pasal 30 juga akan menegaskan mengenai sanksi yang diberikan kepada seorang penjahat siber terbukti bersalah yakni hukuman maksimal 8 tahun penjara dan/atau denda maksimal Rp 800.000.000,- (Wijoseno & Widhiyaastuti, 2023). UU ITE tersebut menunjukkan kepedulian dan atensi Pemerintah Republik Indonesia mengenai keamanan sistem informasi, jaringan komputer dan data dari masyarakat pengguna layanan informasi.

C*yber crime* menargetkan objek serangan terhadap komputer, jaringan komputer dan atau perangkat yang terhubung ke internet. Pelaku kejahatan siber disebut peretas (*hacker*). *Hacker* dikategorikan menjadi 2 berdasarkan jumlah pihak yang terlibat yakni secara perseorangan/individu dan kelompok atau komunitas/organisasi. Sedangkan berdasarkan tujuan (*main goal*) dari *hacker* dibagi menjadi 3 (Buxton, 2022), yakni:

a. *Black Hat. Black hat hacker* merupakan tipe seorang peretas yang secara illegal melakukan peretasan untuk niat atau motivasi bersifat kriminal. Berupaya untuk mengakses secara *unauthorized* (tidak sah) kepada objek yang ditarget. Setelah akses berhasil didapat maka tipe *black hat hacker* akan melakukan pencarian kerentanan keamanan dan berusaha untuk mengeksploitasinya. Cara eksploitasi yang dilakukan membahayakan keamanan suatu jaringan komputer atau komputer itu sendiri, seperti dengan menanamkan virus, trojan dan atau mencuri, menghancurkan dan menyalahgunakan data pribadi orang lain. Motivasi atau niat finansial yakni keuntungan ekonomi menjadi alasan utama dari *black hat hacker.*

b. *White Hat. White hat hacker* merupakan tindakan yang dilakukan *hacker* dalam mengidentifikasi dan melakukan perbaikan terhadap kerentanan keamanan. Sering disebut sebagai *ethical security hacker,* karena memiliki izin atau secara legal (sah) melakukan tindakan peretasan dalam mencari tahu kerentanan keamanan sistem informasi sebuah organisasi. Secara teknis, *white hat* akan melakukan langkah-langkah seperti halnya *black hat* dalam mengakses bahkan melakukan eksploitasi serangan pada objek yang akan dites. Hanya saja karena telah terjadi kesepakatan atau perjanjian kerja dengan objek atau target yang diuji sehingga seorang *white hat hacker* biasanya memiliki *privilege* berupa pengetahuan lebih mengenai sistem informasi yang akan diuji tersebut. Hasil dari *penetration testing* yang dilakukan akan disampaikan dalam bentuk *formal report* kepada organisasi pemilik sistem informasi tersebut. Laporan akhir dari *assessment* dan analisis kerentanan akan disertai dengan langkah-langkah yang bisa dilakukan untuk mengatasi kerentanan tersebut.

c. *Grey Hat. Grey hat hacker* memiliki keterbatasan informasi terhadap target objek yang akan diuji (tanpa persetujuan dengan pemilik sistem informasi). Namun, berbeda dengan *black hat hacker*, *grey hat hacker* tidak memiliki motivasi atau niat untuk melakukan kejahatan siber yang merugikan pihak yang diuji. Sehingga ketika ditemukan adanya kerentanan keamanan akan dilaporkan kepada pemiliki sistem informasi tanpa melakukan eksploitasi

yang bersifat destruktif atau merugikan pihak lain. *Grey hat hacker* biasanya akan tetap mendapat imbalan atau *reward* dari tindakan uji coba *penetration testing* yang telah dibuat dan disampaikan kepada pemilik sistem informasi.

Gambar 2.1 *Different Motivation When Breaking Into Systems of White, Black and Grey Hat Hacker.*

Sumber: Buxton, O. (2022, October 12). *Hacker Types: Black Hat, White Hat, and Gray Hat Hackers*. Hacker Types: Black Hat, White Hat, and Gray Hat Hackers. https://www.avast.com/c-hacker-types

Secara singkat perbedaan utama dari 3 jenis *hacker* di atas yakni bahwa *white hat hacker* akan membantu organisasi untuk mengembangkan tingkat keamanan informasi yang lebih baik dengan kerentanan yang berhasil di eksploitasi. *Black hat hacker* memiliki motivasi untuk tujuan kriminal atau ancaman yang bersifat destruktif sehingga merugikan pihak yang diuji. Sedangkan *grey hat hacker* bukan merupakan ancaman tetapi juga bukan seorang *hacker* yang memiliki *privilege* sebanyak *white hat hacker*.

2.1.2 *Penetration Testing / Vulnerability Testing*

Penetration Testing sering disingkat Pentest merupakan komponen penting dari *Security Audit.* Pentest adalah metode untuk mengevaluasi keamanan sistem komputer atau jaringan dengan simulasi uji coba serangan. Terdapat 2 tipe *penetration testing* berdasarkan asal ruang lingkup serangan pentest antara lain (Andre, 2023):

a. *External Testing.* Uji coba serangan yang dilakukan dari luar jaringan atau sistem dengan melakukan analisa pada informasi jaringan publik yang tersedia, *network enumeration phase* dan keamanan *device* yang digunakan.

b. *Internal Testing.* Uji coba serangan dari dalam jaringan atau sistem yang akan menampilkan dan mengevaluasi beberapa hal, seperti jumlah *network access point* yang mewakili beberapa *logical* dan *physical segment.*

Secara umum, sebuah organisasi akan memiliki *internal testing* yang akan secara berkala melakukan pentest terhadap sistem keamanan organisasi, dilakukan oleh orang atau tim yang sering disebut *pentester*. Berdasarkan tingkat kedalaman tindakan *penetration testing* yang dilakukan dibagi menjadi 3 metode, antara lain (Harjowinoto, Noertjahyana, & Andjarwirawan, 2016):

a. *Passive Penetration Testing.* Melakukan pemetaan dan pengujian terhadap control yang ada di dalam *web application, login* dan konfigurasinya, sehingga memodelkan pemetaan terhadap target/objek sistem.
b. *Active Penetration Testing.* Melakukan kegiatan aktif dalam pengujian seperti manipulasi *input,* pengambilan hak akses dan aktif melakukan eksploitasi terhadap kerentanan yang telah didapat.
c. *Aggressive Penetration Testing.* Melakukan eksploitasi terhadap kerentanan, *reserve engineering* terhadap *software* dan sistem, *backdoor attack*, *download code/file* serta mencoba untuk mengambil akses atas kepemilikan informasi (finansial) yang ada di *server.*

2.1.3 *Penetration Testing Execution Standard* (PTES)

Penetration Testing Execution Standard (PTES) merupakan suatu *framework* (kerangka kerja) atau panduan *standard* yang menggambarkan pendekatan terstruktur untuk melakukan pentest terhadap suatu keamanan sistem sebuah organisasi secara menyeluruh. Tujuan utamanya adalah untuk meningkatkan keamanan jaringan atau sistem dari sebuah organisasi dengan mendeteksi adanya kerentanan atau celah keamanan (Kholiq & Khoirunnisa, 2019). PTES biasanya digunakan untuk melakukan evaluasi dan *assessment* kerentanan pada *high level organization* (skala besar). Terdapat 7 fase dari metode PTES, antara lain ("The Penetration Testing Execution Standard — Pentest-Standard 1.1 Documentation", 2016):

a. *Pre-engagement Interactions.* Melakukan *planning* atau perencanaan secara *detail* mengenai proses *penetration testing* yang akan dilakukan. Proses ini cukup penting ketika hendak melakukan suatu tindakan pentest pada sistem sebuah organisasi. Persiapan yang baik perlu menentukan beberapa aspek atau parameter awal pentest sebagai berikut:
    - Ruang lingkup. Secara umum untuk membatasi *scope* dari aspek pengujian hingga hal teknis, contohnya aspek waktu yang dibutuhkan untuk menyelesaikan satu *task*

pentest yang diberikan dan aspek apa saja yang akan diuji seperti DoS (*Denial of Service*), SQL Injection, *Phising,* XSS dan sebagainya beserta *tool* yang digunakan seperti OWASP ZAP, Kali Linux, Nmap Burp Suite dan sebagainya.

- Matriks estimasi waktu. *Timeline* atau penjadwalan projek pentest yang ditentukan harus jelas dari tanggal mulai hingga tanggal akhir. Hal ini perlu diketahui dengan baik oleh pihak organisasi pemilik sistem informasi dan disepakati bersama *pentester*.

Selain 2 (dua) aspek di atas, terdapat beberapa aspek yang perlu diperhatikan pada fase *pre-engagement interactions* antara lain, *scoping meeting, survey by questionnaries from client* dan sebagainya, yang bersangkutan dengan rencana secara detil projek pentest dari awal hingga akhir antara *client* dan *pentester.*

b. *Intelligence Gathering.* Bertujuan mendapatkan informasi sebanyak-banyaknya dari objek atau target dengan menyediakan standarisasi *reconnaissance* pada tahap awal dari sebuah pentest. Terdapat 3 (tiga) *level intelligence gathering,* antara lain:

   - *Level* 1. Hanya memerlukan usaha yang tidak terlalu banyak dan sulit karena informasi yang hendak diperoleh, sepenuhnya diautomasikan melalui *tool* atau cara lain yang *simple* dan mampu mengefisienkan waktu.
   - *Level* 2. Merupakan kombinasi dari hasil informasi yang didapat dari *automation* pada *level* 1 dan beberapa analisis *manual.* Contoh informasi *level* 2 adalah pengetahuan informasi mengenai bisnis (lokasi fisik bisnis, hubungan bisnis, struktur organisasi dan sebagainya).
   - *Level* 3. Melakukan pentest yang lebih *modern* dengan cakupan yang lebih luas. Membutuhkan semua informasi dari *level* 1 dan *level* 2 serta analisis manual secara mendalam dari informasi yang didapat sebelumnya.

   Tiga Tingkatan ini diperlukan dalam menentukan sejauh mana metode *pentest* yang perlu dilakukan, informasi apa saja yang diperlukan sesuai dengan aspek tujuan, kebutuhan dan tingkat keamanan suatu organisasi.

c. *Threat Gathering. Pentester* akan melakukan pencarian terhadap celah keamanan (*vulnerabilities*) berdasarkan informasi yang berhasil dikumpulkan pada tahap sebelumnya. Berikut proses pemodelan *threat* tingkat tinggi (*high level*) antara lain:

   - Mengumpulkan dokumentasi (seperti *file*) yang relevan. Mengumpulkan informasi dan dokumen yang berkaitan dengan sistem atau aplikasi yang akan diuji.

- Mengidentifikasi dan mengategorikan aset utama dan sekunder. Mengidentifikasi dan mengkategorikan aset utama yang paling penting dalam konteks pengujian serta aset sekunder yang mungkin terpengaruh atau digunakan oleh serangan.
- Mengidentifikasi dan mengkategorikan ancaman dan komunitas ancaman. Mengidentifikasi dan mengkategorikan berbagai ancaman yang mungkin ada terhadap aset-aset tersebut, seperti peretas, malware, atau pengguna yang tidak sah.
- Memetakan komunitas ancaman ke aset utama dan sekunder. Menghubungkan atau memetakan komunitas ancaman ke aset utama dan sekunder yang potensial menjadi target serangan.

Terdapat beberapa akses yang perlu diperhatikan saat pemodelan kerentanan (*threat*), antara lain:

- *Business Asset Analysis* dengan parameter yang perlu dimodelkan yakni data organisasi seperti kebijakan, rencana dan prosedur organisasi, informasi setiap divisi (produksi, marketing, keuangan, operasional), data pegawai, data pelanggan dan sebagainya.
- *Business Process Analysis* dengan parameter yang perlu dimodelkan yakni proses pendukung infrastruktur teknis, proses pendukung informasi aset dan proses pendukung *human assets.*
- *Threat Agent Analysis.* Berikut contoh tabel klasifikasi *threat agent/community:*

Tabel 2.1

Contoh Klasifikasi *Threat Agent/Community*

Internal	**Eksternal**
Pegawai	Rekan Bisnis
Manajemen (eksekutif)	Pesaing
Administrators (*network, server*)	Kontraktor
Developers	*Suppliers*
Teknisi	Negara (Pemerintah)

Sumber: *The Penetration Testing Execution Standard — pentest-standard 1.1 documentation*. (n.d.). Pentest-Standard.readthedocs.io. Retrieved November 10, 2023, from https://pentest-standard.readthedocs.io/en/latest/index.html

Motivation Modelling (tujuan atau motivasi pemodelan kerentanan target *website*) terdiri dari aspek *profit* (langsung atau tidak langsung), tindakan haktivisme, kesenangan (reputasi semata) dan balas dendam dari individu atau organisasi lainnya.

d. *Vulnerability Analysis.* Proses untuk menemukan kerentanan dalam sistem yang dapat dimanfaatkan oleh pihak tak bertanggung jawab. Kerentanan tersebut dapat bervariasi, mulai dari konfigurasi *host* dan layanan yang salah, hingga desain aplikasi yang tidak aman. Meskipun proses yang digunakan untuk mencari kelemahan berbeda-beda dan sangat tergantung pada komponen tertentu yang diuji, ada beberapa prinsip utama yang berlaku dalam proses tersebut. Proses ini dimulai dari *testing* baik secara aktif maupun pasif, setelah itu melakukan validasi terhadap korelasi hasil dari berbagai *tool* yang telah digunakan. Langkah terakhir dari *vulnerability analysis* adalah *research* dengan membaca (studi literatur) mengenai temuan yang diperoleh.

e. *Exploitation.* Proses ini berfokus pada memperoleh akses ke sistem yang dituju dengan melewati (*bypass*) pembatasan keamanan. Fokus utamanya adalah pada proses identifikasi celah masuk ke dalam organisasi melalui jaringan dan mengidentifikasi dokumen atau aset yang ditarget. Perlu diingat ketika melakukan eksploitasi harus dengan hati-hati dan perlu memastikan bahwa identitas penyerang tersembunyi. Beberapa prinsip yang perlu diketahui dalam eksploitasi yakni *Precision Strike,* penyesuaian akses eksploitasi (*customization*), *Zero-Day Angle* dan *Fuzzing*.

f. Proses untuk menentukan nilai dari sistem yang berhasil disusupi dan untuk mempertahankan kendali atas sistem di masa mendatang. Setelah melakukan eksploitasi, *pentester* perlu melakukan identifikasi pada data-data sensitif, seperti pengaturan konfigurasi, saluran komunikasi.

g. *Reporting.* Membuat laporan yang sesuai dengan ketentuan dasar pentest *report.* Struktur laporan dibagi menjadi 2 (dua) bagian utama yakni yang pertama menjelaskan mengenai tujuan dan metode serta yang kedua menjelaskan hasil (*result*) yang dilakukan. Berikut poin-poin yang perlu ada pada laporan akhir secara lengkap, antara lain:

    1. Tujuan Pengujian.
    2. Lingkup pengujian.

3. Tanggal Pengujian.
4. Metodologi pengujian.
5. Hasil pengujian.
6. Laporan akhir.

2.1.4 Kali Linux

Kali Linux merupakan sistem operasi *open-source* yang dikhususkan untuk tujuan *hacking* dan *penetration testing* pada jaringan komputer. Dirilis oleh Offensive Security pada tahun 2013, Kali Linux merupakan salah satu generasi *operating system linux,* turunan dari Debian Linux dan telah menjadi standar industri serta pengembangan keamanan jaringan (Meilinaeka, 2023). Berikut kelebihan dan kekurangan dari Kali Linux, antara lain:

a. Kelebihan:
   - Terintegrasi dengan lebih dari 600 security *tools.*
   - Gratis dan *open source* (fleksibilitas pengembangan).
   - Kompatibilitas *hardware* yang baik.
   - Populer dan memiliki komunitas pengguna yang besar.

b. Kekurangan:
   - Tidak cocok untuk pengguna pemula.
   - Kurangnya dukungan resmi dari lembaga pemerintah atau negara mengenai standarisasi penggunaannya.
   - Kurang *stabil* karena sering mengalami *update* (*open-source risk*).

Berikut jenis-jenis Kali Linux, antara lain:

a. Kali Linux Full dapat *support* terhadap lebih dari 600 *hack tools.*
b. Kali Linux Light merupakan versi mini, untuk *hardware* yang lebih tua dan memerlukan sistem dasar.
c. Kali Linux ARM mendukung *ponsel*, *tablet*, Raspberry Pi, Odroid, dan Chromebook.
d. Kali Linux Virtual merupakan versi khusus dijalankan pada *virtual machine* di atas sistem operasi utama (misalnya Windows).
e. Kali Linux NetHunter berfokus pada *penetration testing* perangkat seluler.
f. Kali Linux Docker merupakan versi yang berjalan dalam Docker.
g. Kali Linux AWS dapat diinstal pada layanan *cloud Amazon Web Services* (AWS).

2.1.5 *Open Web/Worldwide Application Security Project* (OWASP)

OWASP merupakan suatu yayasan atau komunitas non-profit untuk meningkatkan keamanan perangkat lunak secara *global.* Berdiri pada tahun 2001, OWASP beroperasi awal sebagai organisasi yang bertujuan untuk meningkatkan keamanan aplikasi melalui proyek-proyek *open-source, local chapters,* konferensi akademik dan menjadi komunitas yang besar. Tiga poin penting mengenai OWASP yakni:

- OWASP menggerakkan proyek-proyek pengembangan *open source* perangkat lunak untuk meningkatkan keamanan aplikasi.
- Memiliki 250 lebih *local chapters* dan puluhan ribu anggota di seluruh dunia.
- Mengadakan konferensi terkemuka dalam industri pengembangan *software security* dan *training session* bagi para profesional di bidang keamanan aplikasi.

OWASP memiliki visi utama yakni "*No More Insecure Software*" dengan misi "*To be the global open community powering secure software through education, tools, and collaboration*". *Main value* atau prinsip dari OWASP yakni terbuka (transparansi), inovatif, global dan integritas. Pada intinya, OWASP berfungsi sebagai pusat global untuk kolaborasi, pendidikan dan inovasi untuk menghasilkan *software* yang lebih aman dengan komitmen transparansi, integritas dan keterlibatan anggota komunitas (OWASP, 2020). Berdasarkan klasifikasi kerentanan, terdapat 2 (dua) metode OWASP yang paling sering digunakan dalam sebuah *penetration* testing (Dewi dan Setiawan, 2022), diantaranya:

- OWASP Top 10. Metode penerapan dengan melakukan segmentasi terhadap 10 jenis kerentanan, yakni:
    a. *Injection*: Kerentanan injeksi terjadi ketika input dari pengguna tidak diatur dengan benar dan dapat dieksekusi sebagai perintah atau query pada sistem. Contoh yang umum adalah SQL *injection*, NoSQL *injection*, dan Command *injection*.
    b. *Broken Authentication*: Kerentanan ini terjadi ketika mekanisme otentikasi dan sesi tidak diatur dengan benar, yang dapat memungkinkan penyerang untuk mengambil alih akun pengguna atau menjalankan tindakan atas nama pengguna yang sah.
    c. *Sensitive Data Exposure*: Ketika aplikasi tidak memadai melindungi data sensitif, seperti informasi pengguna atau informasi keuangan, dari akses yang tidak sah.
    d. XML *External Entities* (XXE): Serangan XXE terjadi ketika aplikasi memproses XML yang berisi referensi ke entitas eksternal yang tidak aman. Hal ini dapat memungkinkan penyerang untuk membaca atau mengunggah *file* lokal, serta melakukan serangan *denial-of-service* (DoS).

e. *Broken Access Control*: Kerentanan ini terjadi ketika tidak ada atau kurangnya kontrol akses yang memadai, yang dapat memungkinkan penyerang untuk mengakses fungsi atau data yang tidak seharusnya mereka akses.
f. *Security Misconfiguration*: Ketika konfigurasi keamanan tidak benar, termasuk pengaturan standar atau *default* yang tidak aman, penggunaan kredensial *default*, atau pengaturan yang tidak perlu, yang dapat memberikan peluang bagi penyerang untuk mengeksploitasi celah keamanan.
g. *Cross-Site Scripting* (XSS): XSS terjadi ketika aplikasi web tidak memvalidasi atau membersihkan *input* dari pengguna, yang memungkinkan penyerang untuk menyisipkan skrip berbahaya dalam halaman web yang dilihat oleh pengguna lain.
h. *Insecure Deserialization*: Kerentanan ini terjadi ketika aplikasi tidak memvalidasi *input* yang diterima dalam proses deserialisasi, yang dapat memungkinkan penyerang untuk melakukan serangan injeksi, mengeksekusi kode berbahaya, atau bahkan menjalankan serangan *denial-of-service* (DoS).
i. *Using Components with Known Vulnerabilities*: Ketika aplikasi menggunakan komponen atau perangkat lunak pihak ketiga yang rentan terhadap kerentanan yang diketahui, yang dapat dimanfaatkan oleh penyerang untuk menjalankan serangan.
j. *Insufficient Logging* dan *Monitoring*: Kerentanan ini terjadi ketika aplikasi tidak memiliki atau memiliki *log* yang tidak memadai serta mekanisme pemantauan, yang dapat menghambat deteksi dan respons terhadap serangan.

Menurut A. Elanda dan R. L. Buana (2021) dalam penelitiannya menggunakan metode OWASP Top 10 dengan tahapan pengujian meliputi: identifikasi kerentanan, penetrasi OWASP ZAP, penetrasi OWASP ZAP berdasarkan OWASP Top 10 dan rekomendasi. Kesimpulan yang didapat bahwa target memiliki tingkat kerentanan sedang berdasarkan pengujian yang dilakukan menggunakan OWASP ZAP mendeteksi 13 kerentanan dan 4 kerentanan berdasarkan OWASP Top 10 yang meliputi: Sensitive Data Exposure, Security Misconfiguration, Cross Site Scripting, dan Insecure Deserialization. Penelitian ini menggunakan *tools* OWASP ZAP dan Acunetix (dalam Dewi dan Setiawan, 2022, p.3).

- OWASP Versi 4 atau *Web Security Testing Guide* (WSTG). Metode ini terdiri dari 11 (sebelas) tahapan pengujian yakni *Information Gathering*, *Configuration and Deployment Management*, *Identity Management*, *Authentication*, *Authorization*, *Session Management*, *Input Validation*, *Testing for Error Handling*, *Testing for weak Cryptography*, *Business Logic*, dan *Client Side Testing* (Dewi dan Setiawan, 2022).

Tabel 2.2

Tahapan dan *Tools* Pengujian Menggunakan Metode OWASP Versi 4

Tools	Tahapan				
	A1	A2	A3	A4	A5
OWASP ZAP	✓	✓	✓	✓	✓
Mozilla Firefox	✓	✓	✓	✓	✓
Google Chrome	✓	✓	✓	✓	
Netsparker	✓	✓	✓	✓	✓
HAVIJ 1.15				✓	
Brutus	✓				

Sumber: Dewi, B. T. K., & Setiawan, M. A. (2022). *Kajian Literatur: Metode dan Tools Pengujian Celah Keamanan Aplikasi Berbasis Web* (pp. 3-4). Universitas Islam Indonesia.

Keterangan:

A1: *Authentication*

A2: *Authorization*

A3: *Session Management*

A4: *Input Validation*

A5: *Testing for Error Handling*

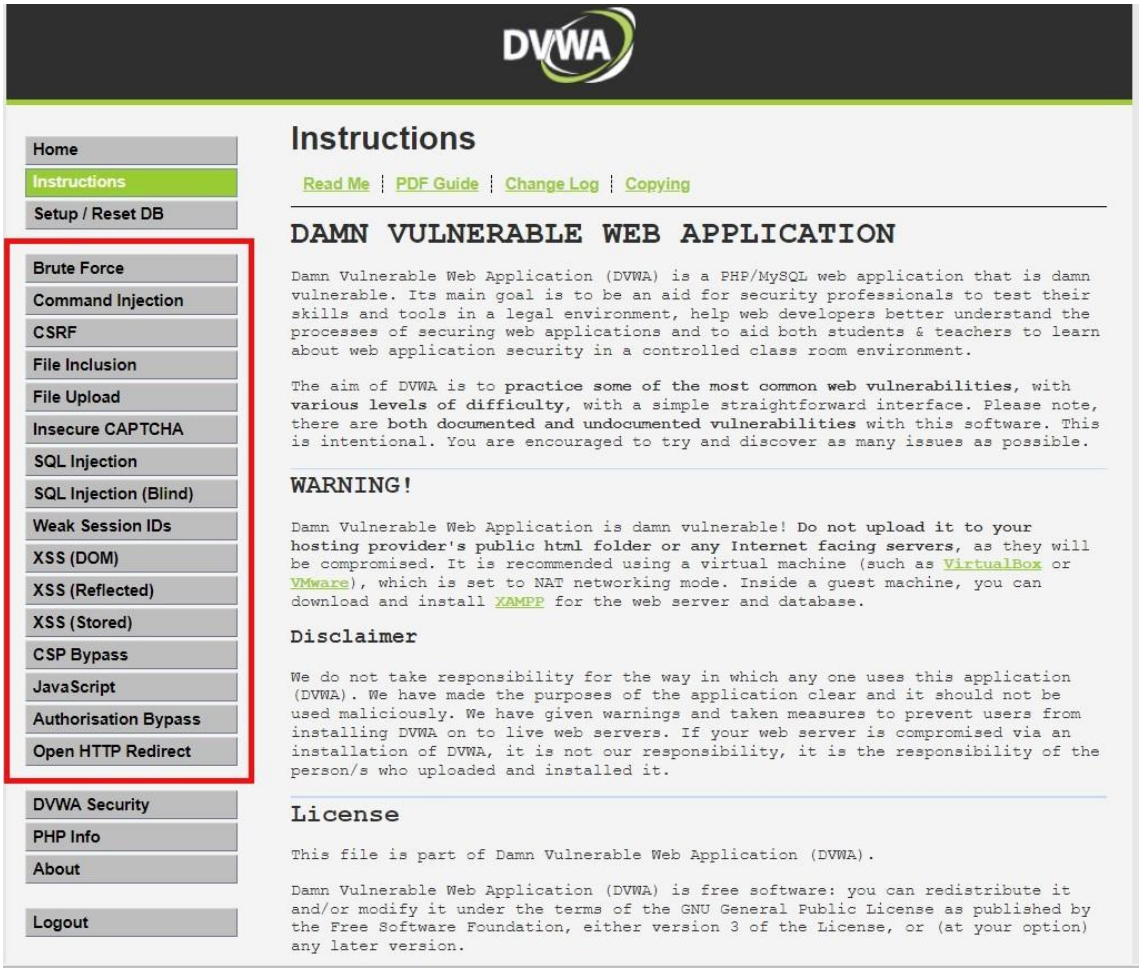
2.1.6 *Damn Vulnerable Web App* (DVWA)

DVWA merupakan *web application* berbasis PHP/MySQL dengan berbagai kerentanan yang dengan sengaja dirancang dan di-*develop.* Tujuan utama DVWA adalah sebagai alat bantu bagi profesional keamanan untuk menguji keterampilan dan alat-alat mereka dalam lingkungan yang legal. Hal ini memudahkan tindakan *ethical hacking* melalui *penetration testing* yang legal atau terautorisasi. Terdapat *level* kerentanan yakni level *low, medium* dan *high.* Namun, perlu diingat bahwa tidak semua kerentanan pada DVWA terdokumentasi sehingga sebagai seorang *pentester* perlu melakukan *testing* yang lebih menyeluruh menggunakan alat-alat atau *tools* yang dimiliki. Untuk *web server testing* menggunakan *web server localhost* XAMPP yang perlu dikonfigurasi. DVWA direkomendasikan untuk dijalankan pada *virtual machine* seperti

VirtualBox atau *VMWare* (“DVWA,” n.d.). Adapun beberapa jenis kerentanan yang dapat ditemukan dan dieksploitasi pada DVWA (Firman, 2022), antara lain:

a. *Brute Force.*
b. *Command Injection.*
c. *Cross-Site Request Surgery* (CSRF).
d. *File Inclusion.*
e. *File Upload.*
f. *Insecure CAPTCHA.*
g. *SQL Injection.*
h. *SQL Injection* (*Blind*).
i. *Weak Session IDs.*
j. *Cross-Site Scripting* (XSS) *DOM.*
k. *Cross-Site Scripting* (XSS) *Reflected.*
l. *Cross-Site Scripting* (XSS) *Stored.*
m. *CSP Bypass.*
n. *JavaScript.*
o. *Authoritation Bypass.*
p. *Open HTTP Redirect.*

Berikut gambar tampilan dari *website* DVWA:



Gambar 2.2 Tampilan *Menu* Utama *Damn Vulnerable Web Application* (DVWA)

Sumber: Firman, M. N. (2022). *Apa Itu DVWA Dalam Cyber Security? – Widya Security*. Widya Security; PT Widya Adijaya Nusantara. https://widyasecurity.com/2024/01/17/apa-itu-dvwa-dalam-cyber-security/

2.1.7 Burp Suite Community Edition

Burp Suite merupakan *tools* yang sering digunakan dalam melakukan *penetration testing* pada *website* dan *mobile app.* Terdapat berbagai fitur Burp Suite yang biasa digunakan yakni sebagai *scanner* untuk melakukan *automation scan vulnerability* pada *website* target, proxy untuk mengatur ip dan *port* yang digunakan sebagai proxy, intruder sebagai alat untuk melakukan brute force, repeater digunakan untuk mengirim *request* berkali-kali kepada *website target* tanpa harus mengirim *request* melalui *browser* serta sebagai alat decoder untuk melakukan proses *encode* dan *decode* (Gunawan, 2020).

2.1.8 OWASP Zed Attack Proxy (ZAP)

OWASP ZAP merupakan salah satu *tool open-source* yang dikembangkan oleh berbagai relawan *developer* OWASP (*Open Web Application Security Project*) untuk keperluan *web security.* ZAP dikembangkan agar *developer* dan *pentester* dapat melakukan *penetration testing* secara manual dan otomatis. Terdapat beberapa fungsi atau fitur utama dari OWASP ZAP yakni *penetration testing* dalam mengidentifikasi kerentanan keamanan seperti SQL *injection, Cross-Site Scripting* (XSS), dan sebagainya. Selain itu, juga terdapat *proxy functionality, automation scanning, spider* (*crawling*) *functionality* dan berbagai fitur *reporting* lainnya (Cara Menggunakan OWASP ZAP, 2024).

2.1.9 SQLMap

SQLMap merupakan salah satu *tool penetration testing* yang bersifat *open-source* dengan beberapa fungsi otomasi seperti mendeteksi, mengeksploitasi kelemahan SQL *Injection* dan mengambil alih *server* dari *database.* SQLMap *support* terhadap sistem *management database* seperti, MySQL, Oracle PostgreSQL, MariaDB dan sebagainya (Irawan, 2021).

2.1.10 Whois

Whois merupakan layanan *scanning* yang digunakan untuk mencari informasi terhadap sebuah *website.* Informasi yang bisa didapat antara lain, nama pemilik *domain,* IP *address, email* bahkan kontak pribadi pemilik *website* tersebut. Informasi pada sebuah *domain* tersebut didapat dari perusahaan *registrar* dan *registries.* Setiap kali sebuah *website* didaftarkan nama *domain* melalui perusahaan *registrar* dan *registries.* Kedua perusahaan tersebut saling berkolaborasi secara langsung dengan *Internet Corporations for Assigned Names and Number* (ICANN). ICANN akan mengatur *database* pada internet dan memastikan tingkat keamanan dari setiap data pemilik *website.* Terdapat dua jenis informasi Whois yakni *Thin Model* (Informasi bersifat umum seperti registrar, nama server dan tanggal pendaftaran) dan *Thick Model* (Informasi tambahan seperti kontak pendaftar, administratif dan juga teknisi) (Syifaudin, 2024).

2.1.11 Nmap

Nmap (*Network Mapper*) merupakan suatu *open-source tools* yang berfungsi untuk melakukan eksplorasi mendapatkan informasi dan *vulnerability scanning* sebuah jaringan. Kali Linux menjadi salah satu *operating system* yang menyediakan Nmap secara langsung atau *default* terinstal. Terdapat beberapa fungsi Nmap yang diperlukan untuk proses *penetration*

testing yakni *port discovery* (mendeteksi *port-port* yang terbuka pada sebuah jaringan), *network mapping* dan *vulnerability scanning* (Siahaan & Lie, 2021).

2.1.12 theHarvester

theHarvester merupakan sebuah *tools* yang memiliki kemampuan *information gathering* seperti *email, subdomain, hosts, open port* dan informasi mengenai *database server.* theHarvester dibuat menggunakan bahasa pemrograman Python (Ardiansyah, 2024).

2.1.13 Wireshark

Wireshark merupakan sebuah *tool* atau aplikasi *capture* paket data bersifat *open-source* yang digunakan untuk melakukan pemindaian serta menangkap *traffic* data pada jaringan internet. Wireshark secara umum digunakan sebagai *troubleshoot tool* pada jaringan yang bermasalah serta untuk menguji sebuah *software* (termasuk *website*). Wireshark menggunakan *library* atau *tool* bantuan untuk melakukan *capture internet network traffic* yaitu Pcap (Saputro & Zakaria, n.d.).

2.1.14 Metasploit

Metasploit adalah *tools* keamanan komputer yang memberikan informasi tentang kerentanan keamanan dan mendukung pengujian penetrasi. Dimiliki oleh Rapid7, perusahaan keamanan siber berbasis di Amerika Serikat, Metasploit terkenal dengan Metasploit *Framework*, *tools* sumber terbuka yang digunakan untuk mengembangkan dan menjalankan eksploitasi kode pada sistem target jarak jauh. Metasploit juga mencakup *tools* anti-forensik dan remediasi, beberapa di antaranya terintegrasi dalam Metasploit *Framework*. Sistem operasi Kali Linux sudah menyertakan Metasploit sebagai instalasi bawaan (Fahmi, 2022).

2.1.15 Nslookup

Name server lookup atau yang sering disebut nslookup merupakan *command line*, perintah atau *tool* sederhana untuk mengambil informasi DNS (*Domain Name System*) dari sebuah *domain* atau *host* guna keperluan *troubleshoot* jaringan. Nslookup tersedia di sebagian besar sistem operasi seperti, Linux, Windows dan macOS. Dengan melakukan *query* DNS, pengguna atau seorang *pentester* dapat memperoleh informasi *detail* mengenai *target* seperti alamat IP yang terkait dengan *domain* tersebut, *server* yang mengelola *domain* ataupun informasi lainnya mengenai sebuah *domain target*. Bahasa sederhana dari sebuah *tool* nslookup

yakni sebagai sebuah *search engine* untuk mengkonversi nama *domain* atau *host* menjadi sebuah alamat IP (Apa Itu Nslookup? Andalan Untuk Melakukan Troubleshoot, 2023).

2.1.16 Wget

World Wide Web Get (Wget) merupakan *tool download manager* sederhana yang diperuntukan untuk *operating system* Linux. Wget berfungsi untuk mengunduh berbagai jenis *file* yang ada di internet. Selain itu, wget juga dapat digunakan pada *local network* guna kepentingan *transfer* data antar perangkat komputer dalam jaringan yang sama (K, 2019).

2.1.17 Netcat

Netcat adalah sebuah *tool* jaringan untuk membaca dan menulis data melalui koneksi jaringan menggunakan koneksi atau protokol TCP dan UDP. Hal ini dapat digunakan untuk menyerang keamanan sebuah jaringan internet dari *website* yang terpublikasi di internet. Netcat dapat digunakan dalam pembuatan koneksi jaringan untuk *debugging* dan penyelidikan jaringan yang dapat mempengaruhi aplikasi *web* (Geeksforgeeks, 2020).

2.1.18 WhatWeb

WhatWeb merupakan *tool* yang digunakan untuk mendeteksi teknologi *web* seperti *server, Content Management System* (CMS), *framework,* JavaScript *libraries,* dan *statistic/analytics packages.* WhatWeb juga dapat mengidentifikasi *version numbers, email address, account* Ids, SQL *errors* dan berbagai informasi dari sebuah *website* (WhatWeb - next Generation Web Scanner., n.d.).

2.1.19 Sublist3r

Sublist3r merupakan paket *tool* yang dibuat menggunakan Python untuk mengenumerasi atau menghitung *subdomain* dari sebuah *website* menggunakan OSINT. Hal ini dapat membantu dalam proses *debug* jaringan atau *pentest* untuk mengumpulkan *subdomain* dari *domain website target*. Sublist3r akan mengenumerasi *subdomain* menggunakan berbagai mesin pencari seperti Google, Yahoo, Bing, Baidu dan sebagainya. Selain itu, Sublist3r juga menghitung *subdomain* menggunakan *tool* lainnya seperti Netcraft, Virustotal, ThreatCrowd, DNSdumpster dan ReverseDNS (Sublist3r | Kali Linux Tools, 2024).

2.1.20 Metagoofil

Metagoofil merupakan salah satu *tool information gathering* yang gratis dan bersifat *open-source.* Metagoofil dirancang untuk mengekstrak semua informasi metadata dari dokumen yang ter-*publish* di *website.* Dengan menggunakan dua *libraries* utama yakni Hachoir dan PdfMiner untuk mengekstrak data. Setelah mengekstrak semua data yang tersedia pada sebuat *website*, metagoofil akan memberikan laporan mengenai nama pengguna, *versi* perangkat lunak/*server* serta nama mesin pencari yang digunakan untuk *penetration testing* dalam fase pengumpulan informasi atau *information gathering* (Metagoofil - Tool to Extract Information from Docs, Images in Kali Linux, 2021).

2.1.21 Wfuzz

Wfuzz merupakan *tool* yang menyediakan *framework* untuk membantu *pentester* dalam mengotomasikan proses *penetration testing* dengan memberikan *analysis security assessments* sehingga dapat memberikan informasi serta melakukan eksploitasi terhadap aplikasi *web* (Mendez, n.d.). Secara khusus Wfuzz dirancang untuk melakukan *bruteforce* terhadap *website* dan dapat digunakan untuk menemukan *resources* yang tidak terkait dengan *directory,* menemukan *scripts,* melakukan *bruteforce* GET dan POST *parameter* sehingga dapat memungkinkan pengecekan kerentanan berbagai jenis injeksi (SQL *Injection,* XSS, LDAP dan sebagainya), serta melakukan *bruteforce* langsung terhaadp *form login* menggunakan *username* dan *password* dengan teknik *fuzzing* (Wfuzz | Kali Linux Tools, 2024).

2.1.22 Hydra

Hydra adalah *framework* Python yang bersifat *open-source* untuk melakukan *bruteforce* pada layanan *login website* sehingga memberikan akses *pentester* mendapatkan kredensial atau akses masuk (Getting Started | Hydra, n.d.).

2.1.23 Nikto

Nikto merupakan *web server scanner* bersifat *open-source* yang digunakan untuk memindai kerentanan terhadap *server web* termasuk beberapa *file* dan program berbahaya menggunakan LibWhisker rfp untuk melakukan pemeriksaan keamanan atau informasi yang cepat. Nikto memiliki beberapa fitur seperti pemeriksaan format *database, reporting via* teks atau HTML*, support cookies,* SSL *support, proxy support* dan sebagainya (Nikto | Kali Linux Tools, 2024).

2.2 Tinjauan Studi

Berikut merupakan penelitian-penelitian terkait yang telah dilakukan sebelumnya sebagai bahan pertimbangan dan perbandingan studi literatur pada penelitian skripsi ini.

2.2.1 Analisis Metode *Open Web Application Security Project* (OWASP) pada Pengujian Keamanan Website: *Literature Review* (Kuncoro dan Rahma, 2022)

a. Masalah yang diangkat dari penelitian ini adalah celah kerentanan dan hak akses ilegal yang dimiliki oleh pihak tidak berwenang. Penelitian ini bertujuan untuk mengeksplorasi dan menganalisis penggunaan metode Open Web Application Security Project (OWASP) dalam pengujian keamanan sistem komputer, terutama fokus pada website. Dalam konteks ini, masalah yang muncul adalah meningkatnya kebutuhan akan keamanan sistem komputer, terutama dengan maraknya penggunaan koneksi internet yang dapat meningkatkan risiko terjadinya tindak kejahatan komputer.
b. Penelitian ini menggunakan metode *Scoping Review* untuk melakukan seleksi literatur terkait pengujian keamanan sistem komputer dengan metode OWASP. Metode ini memungkinkan peneliti untuk menyusun pemahaman menyeluruh terhadap literatur yang relevan dengan fokus dan tujuan penelitian.
c. Hasil penelitian yang dapat disimpulkan penulis bahwa penelitian ini memberikan gambaran tentang pentingnya pengujian keamanan sistem informasi, khususnya yang berbasis aplikasi website. Pengujian keamanan dianggap krusial untuk memberikan perlindungan dan kenyamanan kepada pengguna sistem. Melalui pencarian celah keamanan dan pengujian, penelitian ini berhasil mengidentifikasi beberapa kelemahan dan kerentanan pada sistem. Kelemahan tersebut, jika dieksploitasi, dapat dimanfaatkan oleh pihak yang tidak berwenang dan tanpa akses yang sah. Dengan maraknya insiden kebocoran data dan penyusupan oleh pihak yang tidak sah, penelitian ini menyoroti perlunya melakukan pengujian keamanan secara berkala dan bertahap. Hal ini bertujuan untuk menyediakan sistem yang kuat dan aman bagi pengguna. Hasil analisis menunjukkan bahwa metode OWASP Top Ten sering digunakan, dan alat terbuka seperti ZAP menjadi pilihan utama dalam pengujian keamanan sistem berbasis website. Penelitian ini mencatat bahwa penelitian yang fokus pada pengujian keamanan menggunakan metode OWASP versi 4.2 masih kurang ditemukan. Sementara metode OWASP Top Ten masih dominan, penelitian lebih lanjut pada versi terbaru dapat memberikan wawasan tambahan. Penelitian menunjukkan bahwa dengan berkembangnya celah keamanan dan variasi

serangan, metode yang telah digunakan mungkin kurang optimal. Kondisi ini menunjukkan adanya kebutuhan untuk terus beradaptasi dengan ancaman yang semakin beragam.

d. Meskipun literatur yang telah diidentifikasi memberikan gambaran umum tentang pengujian keamanan sistem dengan fokus pada kerangka kerja OWASP, masih terdapat kebutuhan yang signifikan untuk mengeksplorasi lebih lanjut aspek automasi, khususnya dalam konteks penggunaan skrip pentest. *Research gap* yang dapat diidentifikasi terletak pada kurangnya informasi yang mendalam mengenai efisiensi, efektivitas, dan potensi tantangan yang mungkin dihadapi dalam menerapkan automasi skrip pentest, terutama dalam kerangka kerja OWASP.

2.2.2 Analisis Keamanan Webserver Menggunakan *Penetration Test* (Fachri, Fadlil dan Riadi, 2021)

a. Penelitian ini bertujuan untuk mengevaluasi keamanan sistem informasi, khususnya web server yang digunakan sebagai Sistem Informasi Akademik (SIA) di lingkungan perguruan tinggi. Fokusnya adalah mengidentifikasi kelemahan dan kerentanan pada web server serta meningkatkan tingkat keamanan sistem.

b. Metode yang digunakan dalam penelitian ini mencakup beberapa tahap, antara lain *Information Gathering, Vulnerability Assessment, Gaining Access, Maintaining Access,* dan *Clearing Track*. Penelitian menggunakan pendekatan *penetration testing* yang bersifat legal dan resmi.

c. Hasil penelitian menunjukkan bahwa terdapat empat kelemahan tingkat tinggi, empat kelemahan tingkat menengah, dan dua kerentanan tingkat rendah pada *web server*. Selain itu, ditemukan beberapa *port* yang masih terbuka, memungkinkan peretas untuk dengan mudah masuk ke dalam sistem, mengeksploitasi informasi yang ada dalam Sistem Informasi Akademik tersebut.

d. Pada penelitian ini hanya berfokus pada keamanan *website* Sistem Informasi Akademik secara khusus, sedangkan pada penelitian skripsi ini mengenai *automation script* akan berfokus pada keamanan *web server* secara lebih *general. Tool* yang digunakan hampir sama namun cara kerja yang akan berbeda yakni pada penelitian yang akan datang bersifat automasi.

2.2.3 Improved Deep Recurrent Q-Network of POMDPs for Automated Penetration Testing (Zhang et al., 2022)

a. Masalah yang menjadi topik atau fokus tujuan penelitian ini adalah keterbatasan pada *penetration testing methods*. Meskipun sudah banyak metode pengujian penetrasi yang ada, kebanyakan bersifat ideal dan kurang mencerminkan situasi nyata serangan. Selain itu, pentingnya keamanan jaringan bagi masyarakat awam. Dengan perkembangan teknologi, kehidupan sehari-hari masyarakat semakin erat kaitannya dengan jaringan, dan perlindungan keamanan siber menjadi semakin penting.

b. Metode yang digunakan yakni *Modeling Black-Box Penetration Testing* sebagai POMDP. Penelitian ini memodelkan proses uji penetrasi *black-box* sebagai *Partially Observed Markov Decision Process* (POMDP), menggambarkan interaksi agen dengan lingkungan jaringan. Juga menggunakan algoritma ND3RQN, dengan diperkenalkannya algoritma ND3RQN, yang diaplikasikan dalam uji penetrasi otomatis *black-box.*

c. Hasil yang diperoleh yakni algoritma ND3RQN berhasil menemukan strategi *path* serangan yang lebih baik untuk semua *host* rentan dalam *penetration testing* otomatis *black-box*. Algoritma ini menunjukkan generalisasi dan ketangguhan yang unggul dibandingkan dengan algoritma *state-of-the-art* (SOTA) dalam skenario simulasi berbagai ukuran.

d. Studi-studi sebelumnya cenderung berfokus pada skenario pentest yang ideal, sementara penelitian skripsi ini mengeksplorasi perspektif nyata serangan di dunia nyata. Namun, pelaksanaan pentest yang mudah, cepat dan ideal perlu dikembangkan sehingga mempermudah pemilik sistem atau *website* sebagai seorang awam juga mampus melakukan pentest ke *websitenya* secara automasi melalui *automation script.*

2.2.4 Analisis Keamanan *Web Server Open Journal System* (OJS) Menggunakan Metode ISSAF dan OWASP (Studi Kasus OJS Universitas Lancang Kuning) (Guntoro, Costaner dan Musfawati, 2020)

a. Topik permasalahan yang disoroti pada penelitian ini yakni pentingnya *penetration testing* dalam menjaga keamanan sistem informasi, terutama dalam konteks Universitas Lancang Kuning yang mengalami kerusakan pada sistem *Open Journal System* (OJS) karena serangan. Penelitian ini juga menekankan bahwa meskipun ada banyak metode pengujian penetrasi yang tersedia, kebanyakan dari mereka kurang mencerminkan situasi nyata serangan. Ini menggarisbawahi kebutuhan akan metode pengujian penetrasi yang lebih realistis dan efektif dalam menghadapi ancaman keamanan siber.

b. Metode yang digunakan dalam penelitian ini, yaitu *Modeling Black-Box Penetration Testing* sebagai POMDP, menunjukkan pendekatan inovatif untuk memodelkan proses uji penetrasi *black-box* sebagai *Partially Observed Markov Decision Process* (POMDP). Penggunaan algoritma ND3RQN juga memberikan kontribusi signifikan dalam pengujian penetrasi otomatis *black-box*, menghasilkan strategi jalur serangan yang lebih baik untuk *host* yang rentan. Sedangkan dalam penelitian skripsi ini menggunakan jenis atau metode *White-Box Penetration Testing* dimana telah diketahui penjelasan atau *documentation* bahkan *tutorial* mengenai target *website* yakni DVWA.

c. Hasil penelitian menunjukkan bahwa algoritma ND3RQN berhasil meningkatkan efektivitas *penetration testing* otomatis metode *black-box* dengan menemukan strategi serangan yang lebih baik untuk semua *host* yang rentan. Keunggulan algoritma ini dalam menangani skenario simulasi berbagai ukuran menunjukkan potensi untuk diterapkan dalam *penetration testing* nyata di dunia nyata atau *real*. Sedangkan hasil dari penelitian skripsi ini untuk mengetahui seberapa efektif metode PTES jika dibandingkan dengan OWASP serta waktu yang dibutuhkan apabila menggunakan *tools automation script* dalam suatu rangkaian *penetration testing.*

d. Studi ini mengisi kesenjangan dalam literatur dengan mengeksplorasi pendekatan yang lebih realistis dalam *penetration testing*, dimana penelitian sebelumnya cenderung berfokus pada skenario *pentest* yang ideal. Namun, penelitian ini juga menyoroti pentingnya mengembangkan alat-alat *penetration testing* yang mudah digunakan dan dapat diakses oleh pemilik sistem atau *website* yang profesional maupun awam secara teknis, untuk memungkinkan mereka melakukan pengujian secara otomatis dan efisien. Hal ini selaras dengan manfaat ke depan atau hasil akhir dari penelitian skripsi yang dibuat yakni dengan pengembangan *tools automation script* diharapkan lebih inklusif terhadap pemilik *website* dalam melakukan *penetration testing* secara mandiri, cepat dan tepat (tanpa mengurangi kualitas atau hasil akhir *penetration testing*).