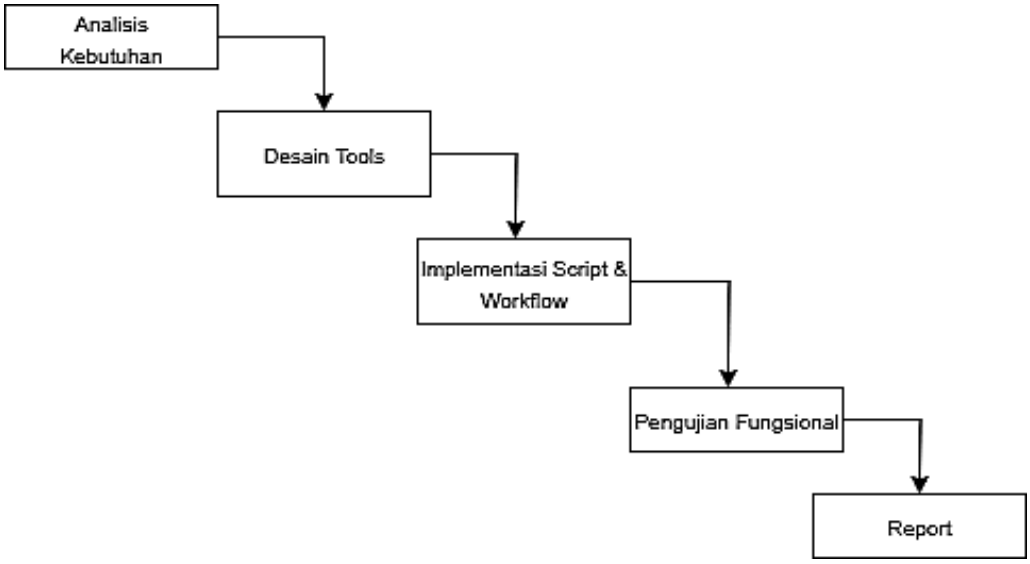
BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Model pengembangan perangkat lunak yang digunakan adalah *Waterfall*, karena struktur tahapannya yang sekuensial dinilai cocok untuk membangun sistem *CLI* yang sistematis, dapat diuji, serta terdokumentasi dengan baik. Model *Waterfall* terdiri dari beberapa tahapan yang dilalui secara berurutan, menyerupai aliran air terjun, sehingga memudahkan dalam memastikan bahwa setiap komponen *tools* dan *workflow* keamanan dikembangkan sesuai kebutuhan analisis.

Tahapan pengembangan menggunakan model *Waterfall* dijabarkan pada Gambar 3.1.



Gambar 3. 1 Alur Model *Waterfall*

1. Kebutuhan Sistem

Tahapan ini dilakukan dengan mengidentifikasi kebutuhan fungsional dan teknis dalam pembangunan *tools* enumerasi modular bernama Cipherion. Informasi dikumpulkan melalui studi terhadap metode *passive reconnaissance*, eksplorasi fitur *tools open-source* seperti *Subfinder, DNSx, Dirsearch, HTTPX*, serta kebutuhan dokumentasi hasil pengujian yang komunikatif dan proporsional. Fokus

utama adalah merancang *workflow CLI* yang efisien, legal, dan mendukung analisis keamanan *website* Ninja SHL secara sistematis.

2. Perancangan Sistem

Tahap desain berfokus pada perancangan struktur modular *tools* Cipherion, alur integrasi antar submodul, format *output*, dan konfigurasi eksekusi melalui terminal. Setiap *submodule* (misalnya s*ubdomain scanner, port enumerator, directory brute-forcer*) dirancang agar dapat beroperasi secara terpisah maupun terintegrasi dalam satu *workflow*. Desain modular ini bertujuan memudahkan *debugging*, dokumentasi, dan ekspansi fitur di masa depan. Format *output* juga dirancang agar siap digunakan dalam laporan akademik dan analisis kerentanan.

3. Implementasi dan Pengujian Modul

Tahapan ini merupakan proses realisasi desain dengan menggunakan bahasa pemrograman *Python* dan metode *CLI* di sistem operasi *Kali Linux*. Implementasi meliputi penyusunan skrip eksekusi, integrasi antarmuka terminal dengan berbagai t*ools* enumerasi, serta penyesuaian parameter agar *output* konsisten dan mudah dianalisis. Pengujian modul dilakukan secara parsial terhadap fungsi utama.

4. Integrasi Tools dan Validasi

Setelah seluruh *submodule* berhasil diimplementasikan, dilakukan proses integrasi ke dalam satu *workflow* enumerasi lengkap. Validasi dilakukan dengan menguji Cipherion pada target *website* Ninja SHL, termasuk proses identifikasi *subdomain* aktif, direktori sensitif, dan potensi *endpoint* yang rentan. Hasil pengujian dianalisis dan didokumentasikan menggunakan format tabel, diagram, dan screenshot untuk mendukung narasi skripsi. Validasi ini bertujuan membuktikan bahwa *workflow* yang dibangun mampu mengidentifikasi kerentanan secara sistematis.

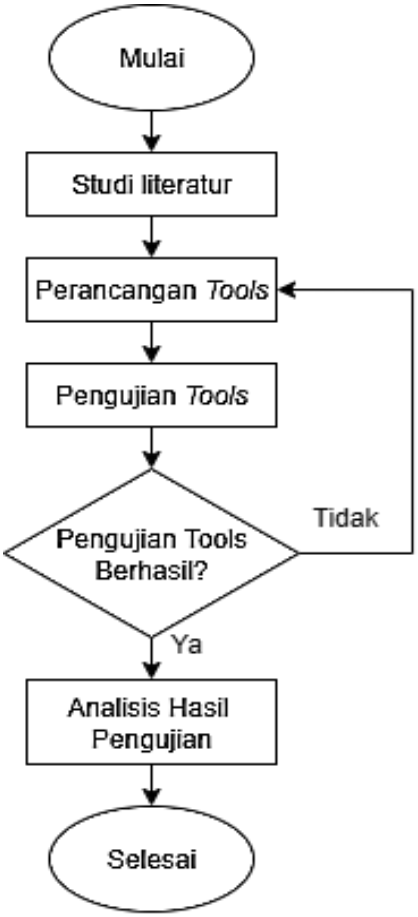
5. Operasional dan Pemeliharaan

Tools yang telah diuji kemudian digunakan untuk keperluan analisis keamanan lanjutan. Pemeliharaan dilakukan melalui perbaikan bug, pembaruan *dependency*, serta penambahan fitur atau *submodule* berdasarkan kebutuhan eksplorasi selanjutnya. Dokumentasi *workflow* dijaga agar tetap rapi, dapat direproduksi, dan

siap dijadikan portofolio akademik maupun profesional di bidang *cybersecurity*. Tahapan ini juga mencakup pengelolaan hasil pengujian agar tetap terstruktur dan tidak menimbulkan risiko penggunaan ilegal.

3.2 Alur Penelitian

Alur penelitian ini bertujuan untuk mengarahkan penelitian dengan struktur yang sistematis, serta mempermudah dalam memahami proses alur penelitian. Berikut merupakan alur penelitian yang dijelaskan pada Gambar 3.2.



Gambar 3. 2 Alur Penelitian

3.2.1 Studi Literatur

Studi literatur dalam penelitian ini berfokus pada teori, konsep, dan penelitian sebelumnya yang relevan dengan teknik enumerasi dalam *penetration testing*

aplikasi *web*, terutama dalam konteks pengumpulan informasi awal secara sistematis dan legal. Kajian ini mencakup eksplorasi teknik *black-box testing*, identifikasi *subdomain* dan direktori tersembunyi, serta pendekatan audit konfigurasi layanan yang terekspos ke publik. Tujuan utama dari kajian ini adalah memahami bagaimana enumerasi tahap awal dapat memberikan gambaran menyeluruh tentang permukaan serangan (*attack surface*) suatu sistem *web*.

Untuk mengatasi kendala keterpisahan dan ketidakefisienan berbagai *tools* enumerasi yang ada, penelitian ini mendalami konsep modular *CLI-based enumeration* sebagai pendekatan inovatif yang memungkinkan integrasi berbagai teknik pengumpulan data ke dalam satu sistem kerja yang utuh. Selain itu, penelitian ini meninjau struktur komputasi *terminal* berbasis *Kali Linux* sebagai lingkungan yang optimal dalam menjalankan proses enumerasi secara mandiri. Studi ini juga mengevaluasi metode dokumentasi otomatis hasil *scan* serta keakuratan informasi yang diperoleh dari sistem publik. Literatur yang digunakan mencakup buku akademik, jurnal ilmiah internasional, dan artikel konferensi yang relevan, sehingga memberikan landasan teoritis yang kuat bagi pengembangan Cipherion sebagai *tools* enumerasi yang efektif dan terstruktur.

3.3 Perancangan Tools

Tahap persiapan tools dilakukan untuk memastikan seluruh komponen yang dibutuhkan oleh *Tools* Cipherion dapat berfungsi secara optimal dalam proses enumerasi terhadap *website* Ninja SHL. Lingkungan pengujian dikonfigurasi menggunakan sistem operasi *Kali Linux* yang mendukung eksekusi modul terminal secara efisien.

Langkah awal meliputi instalasi serta konfigurasi sejumlah modul yang mencakup identifikasi *subdomain*, pemindaian *port* layanan, deteksi direktori tersembunyi, serta analisis teknologi situs termasuk sistem manajemen konten dan *endpoint login*. Validasi terhadap respons *HTTP* juga dilakukan untuk menjamin integritas data yang diperoleh dari sistem publik.

Struktur folder modular disiapkan untuk menampung *output* tiap tahap pengujian. Pengaturan ini mempermudah dokumentasi *log*, penempatan bukti visual, dan penyusunan hasil analisis yang sistematis. Verifikasi koneksi jaringan juga dilakukan agar *tools* dapat berinteraksi optimal dengan *website* Ninja SHL sebagai target legal dalam skenario simulasi *black-box*.

Integrasi internal antar modul dirancang melalui skrip *Python* agar eksekusi *workflow* berlangsung secara berurutan dan hasil tiap tahap dapat digabungkan ke dalam satu laporan akhir yang mudah dianalisis.

3.3.1 Kebutuhan Tools

Kebutuhan tools dalam penelitian ini mencakup seluruh komponen perangkat lunak dan konfigurasi sistem yang diperlukan untuk membangun dan menjalankan *Tools* Cipherion secara efektif dalam proses enumerasi terhadap *website* Ninja SHL. *Tools* Cipherion dirancang berbasis *Command-Line Interface (CLI)* dan berjalan di atas sistem operasi Kali Linux, sehingga kebutuhan tools dikelompokkan menjadi beberapa bagian sebagai berikut.

1. Perangkat Keras

Berikut ini adalah sejumlah komponen perangkat keras yang diperlukan dalam pelaksanaan penelitian dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Fungsi
1.	HP 245 G7 Notebook PC, AMD Ryzen 5 3500U (8 CPU), RAM 8 GB, SSD 512 GB, DirectX 12	Sebagai pusat pemrosesan sistem.
2.	Server Target (Ninja.shl)	Server ninja.shl, IP 110.93.14.30

2.Perangkat Lunak

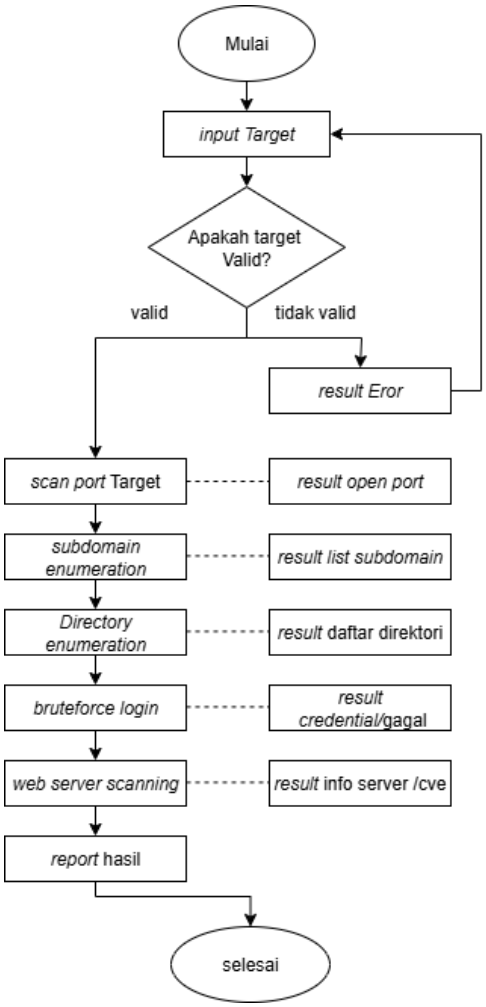
Di samping kebutuhan akan perangkat keras, sistem yang dikembangkan dalam penelitian ini juga memerlukan berbagai perangkat lunak pendukung. Rincian lengkap perangkat lunak yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Kebutuhan Perangkat Lunak

No	Nama Software	Fungsi
1	*Kali Linux*	Sistem operasi utama untuk pengujian dan eksekusi *workflow* terminal
2	*Python 3.10*	Bahasa pemrograman utama untuk pengembangan skrip modular *Tools* Cipherion
3	*HTTPX*	Validasi koneksi dan status *HTTP* dari *subdomain*
4	*VSCode / Nano Editor*	Editor teks untuk penulisan dan modifikasi skrip enumerasi

Dalam perancangan sistem ini, dilakukan integrasi antara beberapa modul enumerasi ke dalam satu *workflow* modular pada *Tools* Cipherion untuk melakukan analisis sistem *website* Ninja SHL secara otomatis. Integrasi ini dirancang melalui skrip *Python* utama yang mengatur urutan eksekusi masing-masing modul, seperti enumerasi *subdomain*, validasi *HTTP*, pemindaian *port*, pencarian direktori tersembunyi sehingga tiap tahap pengujian berjalan konsisten dan terdokumentasi.

3.3.2 Perancangan Kerja Sistem



Gambar 3. 3 Perancangan Kerja *Tools* Cipherion

Dalam perancangan sistem ini, dilakukan integrasi antara beberapa modul enumerasi ke dalam satu *workflow* modular pada *Tools* Cipherion untuk melakukan analisis sistem *website* Ninja SHL secara otomatis. Integrasi ini dirancang melalui skrip *Python* utama yang mengatur urutan eksekusi masing-masing modul, sehingga tiap tahap pengujian berjalan secara konsisten dan terdokumentasi.

Setelah integrasi selesai, dilakukan pengujian terhadap fungsi tiap modul dengan menjalankan skenario simulasi pada target *website*. Jika modul tidak merespon dengan benar, maka hasil *output* akan kosong atau menampilkan *error handler* yang telah disiapkan. Namun, jika pengujian berhasil, maka hasil tiap modul seperti daftar *subdomain*, *port* terbuka, direktori tersembunyi, hingga

endpoint login WordPress akan tersimpan dalam direktori output yang telah ditentukan.

Output tersebut kemudian digunakan sebagai landasan untuk tahapan analisis selanjutnya, seperti pengujian *login brute-force* dan validasi *CMS*. Jika sistem mendeteksi *CMS* yang *valid* dan *endpoint login* aktif, maka *workflow* akan melanjutkan proses *brute-force login* menggunakan *WPScan*. Sebaliknya, jika *CMS* tidak terdeteksi, maka tahapan *login* akan dilewati dan sistem mencatat status sebagai "*skipped*". Semua hasil akhir akan direkam dalam *log* modul yang terstruktur untuk keperluan dokumentasi dan visualisasi hasil uji.

3.4 Pengujian Tools

3.4.1 *Black-Box Testing*

Tabel 3. 3 *Instrumen Black-Box Testing* Skenario Pengujian

No.	Skenario/fitur	*Output* yang Diharapkan	Hasil		Catatan
			Berhasil	Tidak Berhasil	
1.	*Port Scanning*	Menampilkan *port* aktif pada target *host* beserta jenis layanan (misal: *HTTP/SSH*).			
2.	*Subdomain Enumeration*	Menampilkan list *subdomain valid* pada *domain* target.			
3.	*Directory Enumeration*	Menampilkan path direktori yang dapat diakses dengan status kode (200 OK) atau kode lainnya.			
4.	*Bruteforce Login*	Menampilkan kredensial valid jika proses *brute-force* berhasil.			
5.	*Web server Scanning*	menampilkan informasi dasar server, seperti *banner HTTP*, versi perangkat lunak yang digunakan, serta teknologi yang mendukung layanan *web* pada target.			

Metode *black box testing* terhadap *website* Ninja SHL diujicobakan oleh tiga profesional yang ahli di bidang keamanan *cyber* dan *penetration testing*. Proses evaluasi dilakukan oleh tiga penguji teknis, yaitu Zoya Anzasmara (*Engineer*), Irwin Mandela Damanik *(Senior Analyst*), dan Maulana Ilyas Pratama *(Engineer*).Pengujian dilakukan tanpa akses terhadap konfigurasi internal sistem, melainkan melalui pendekatan eksternal yang bertujuan untuk mengidentifikasi *subdomain*, layanan aktif, serta potensi kerentanan awal dari sisi publik. Instrumen Black-Box Testing Skenario Pengujian ditampilkan pada Tabel 3.3.

Untuk memperoleh skor akhir Black-Box, digunakan metode perhitungan berikut ini:

$$\%\ keberhasilan\ = \frac{Jumlah\ fitur\ berhasil}{jumlah\ fitur} \times 100\% \qquad (3.1)$$

3.4.2 *System Usability Scale*

Evaluasi terhadap aspek usabilitas dari sistem dilakukan sebagai tahap lanjutan setelah seluruh proses pengujian fungsional dinyatakan selesai dan sistem telah memenuhi kriteria operasional dasar. Metode yang digunakan dalam evaluasi ini adalah System Usability Scale (SUS), yang merupakan pendekatan standar untuk mengukur tingkat kemudahan penggunaan dan kepuasan pengguna terhadap sistem yang dikembangkan. Untuk memberikan gambaran yang lebih jelas mengenai skenario pengujian yang diterapkan dalam metode SUS, representasi rinci dari tahapan dan komponen evaluasi disajikan pada Tabel 3.4.

Tabel 3. 4 Kuisioner *SUS*

No.	Pernyataan
P1	Saya Berfikir Untuk menggunakan *TOOLS ENUMERATION* CIPHERION DALAM ANALISIS KEAMANAN *WEBSITE* Pada waktu berikutnya
P2	Saya Merasa *TOOLS ENUMERATION* CIPHERION DALAM ANALISIS KEAMANAN *WEBSITE* ini Rumit untuk Digunakan
P3	Saya Merasa *TOOLS ENUMERATION* CIPHERION Mudah digunakan secara mandiri
P4	Saya Memerlukan bantuan dari teknisi lain un tuk menggunakan *TOOLS ENUMERATION* CIPHERION
P5	*TOOLS ENUMERATION* CIPHERION Telah berjalan dengan baik dan mampu melakukan *Information Gathering*.
P6	Saya menemukan beberapa hal yang tidak konsisten atau membingungkan saaat menggunakan *TOOLS ENUMERATION* CIPHERION
P7	Saya Percaya Pengguna lain Dapat dengan mudah memahami cara menggunakan *TOOLS ENUMERATION* CIPHERION Tanpa bantuan Teknis yang berlebihan
P8	Saya *merasa TOOLS ENUMERATION* CIPHERION Membingungkan Pada Awal Penggunaan
P9	Saya merasa percaya diri saat menggunakan *TOOLS ENUMERATION* CIPHERION dalam melakukan *Penetration Testing*
P10	Saya Perlu membiasakan diri terlebih dahulu sebelum dapat menggunakan *TOOLS ENUMERATION* CIPHERION dengan lancar dan konsisten

Penilaian terhadap setiap butir pertanyaan dilakukan menggunakan Skala *Likert*, sebagaimana tercantum dalam Tabel 3.5.

Tabel 3. 5 Skala *Likert*

Skala Penilaian	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Penghitungan skor kontribusi adalah sebagai berikut:

a) Pernyataan ganjil (S1, S3, S5, S7, S9) → Skor kontribusi = Skor - 1
b) Pernyataan genap (S2, S4, S6, S8, S10) → Skor kontribusi = 5 - Skor
c) Total skor kontribusi dari 10 pernyataan → Dikalikan dengan 2.5 untuk mendapatkan Total Skor SUS per responden

Dalam *System Usability Scale (SUS),* setiap pernyataan ganjil merupakan pernyataan positif, sehingga skor kontribusinya dihitung dengan mengurangi 1 dari jawaban responden untuk menyesuaikan skala interpretasi. Sebaliknya, pernyataan genap bersifat negatif, sehingga skor kontribusinya dihitung dengan mengurangkan jawaban responden dari 5 agar hasilnya tetap mencerminkan tingkat kepuasan dan kemudahan penggunaan secara konsisten.

Untuk memperoleh skor akhir *SUS*, digunakan metode perhitungan berikut ini:

$$s = \frac{\sum x}{n} \times 2.5 \qquad (3.2)$$

Keterangan:

s = Total Skor *SUS*

$\sum x$ = Jumlah Skor Kontribusi

n = Jumlah Responden

Berdasarkan skor akhir yang telah dihitung, sistem ini selanjutnya dinilai menggunakan skema kategori *SUS* sebagaimana dijelaskan dalam Tabel 3.6.

Tabel 3. 6 Bobot Skor *SUS*

Rentang Skor SUS	Nilai	Kategori
< 51	E	*Awful*
51 – 67	D	*Poor*
68	C	*OK*
68 – 80.3	B	*Good*
> 80.3	A	*Excellent*