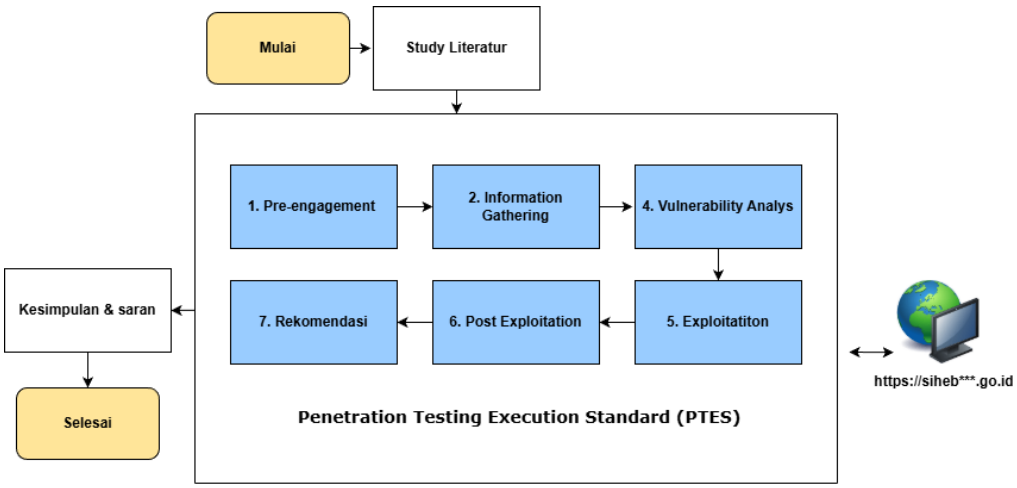
BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang disusun secara sistematis untuk menncapai tujuan penelitian yang diinginkan. pemilihan metologi penelitian merujuk kepada framework *Penetration Testing Execution Standard* (PTES), karena framework ini menyediakan struktur yang lebih operasional dan teknis pada pengujian penetrasi terutama pada fase eksploitasi yang menjadi focus pada penelitian yang di lakukan (Fikri dkk., 2023). Dibandingkan dengan framework lain seperti OWASP Testing Guide dan NIST SP 800-115, *framework* PTES dipilih karena menawarkan tingkat kedalaman teknis yang lebih memadai untuk penelitian eksploitasi khususnya *SQL Injection* (Fadhli, 2024). *OWASP Testing Guide* yang lebih menjurus pada proses pengujian aplikasi web dan pemindian kerentanan secara umum (Adrian Audie Setiawan dkk., 2025), sedangkan NIST SP 800-115 lebih berfokus kepada aspek prosedural dan tata Kelola pengujian keamanan, bukan aplikasi web seperti, jaringan proses audit dan manajemen resiko (Aziz dkk., 2025). Penelitian (Safitra dkk., 2023) menekankan bahwa framework PTES memudahkan proses pengujian karena menyediakan tahapan eksploitasi yang terstruktur dan mudah direplikasi. *Penetration Testing Execution Standard* (PTES) dinilai paling sesuai untuk penelitian yang dilakukan karena memberikan pedoman yang bisa di adaptasikan langsung pada konteks penelitian yang dilakukan. Tahapan penelitian selengkapnya ditunjukkan pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian (Dasmen dkk., 2023)

3.1.1 Studi Literatur

Proses Studi literratur, diperlukan untuk mempelajari dan memahami komponen komponen yang relevan terkait dengan *SQL Injection*, Algoritma Pencarian, Aplikasi web, agar dapat berjalan dengan lancar dan sesuai dengan kaidah penulisan ilmiah. Penelitian ini mengacu pada paper dan dokumen tertulis yang berkaitan dengan *penetration testing*, *Time Based blind SQL injection,* Algoritma Pencarian Khususnya *(Linear search ,Binary search dan Ternary search)* lalu *framework* PTES (*Penetration Testing Execution Standard)* yang di gunakan sebagai dasar rujukan dari tahapan penelitian. Referensi ini menjadi dasar dalam membangun pemahaman mendalam untuk merancang eksperimen penelitian dan pengujian algoritma secara sistematis.

3.2 *Penetration Testing Execution Standard* (PTES)

3.2.1 *Pre-engangement interaction*

Tahapan awal *framework Penetration testing execution standard* (PTES) ini dilakukan komunikasi antara peneliti/penguji dengan pihak yang bertanggung

jawab dan mengelola sistem atau aplikasi web yang akan diuji yang dijadikan target. Proses ini mencakup perizinan untuk melakukan pengujian keamanan serta penentuan ruang lingkup dan apa tujuan yang dilakukan dalam pengujian ini, lingkup pengujian disepakati hanya mencakup satu celah kerentanan *critical* yang di temukan dengan tujuan akademis, bukan untuk merusak atau mengubah data sistem, Tahapan awal *framework Penetration testing execution standard* (PTES) ini dilakukan komunikasi antara tim penguji dengan pihak yang bertanggung jawab dan mengelola sistem atau aplikasi web yang akan diuji yang dijadikan target. Proses ini mencakup perizinan untuk melakukan pengujian keamanan serta penentuan ruang lingkup dan apa tujuan yang dilakukan dalam pengujian ini seperti menentukan perangkat keras dan lunak yang di lakukan selama pengujian serta menentukan etika dan Batasan teknis (Madani dkk., 2024). Selain itu perlunya kesepakatan dengan memberi batasan serta aturan etika yang harus dipatuhi agar proses pengujian berjalan dengan aman, legal, dan sesuai dengan standar yang berlaku dengan menentukan target aplikasi web yang akan diuji dan metode pentest apa yang di gunakan ,Komunikasi yang transparan dan kesepakatan yang matang dalam tahap *Pre-engagement Interactions* berperan penting dalam memastikan bahwa proses pengujian dilakukan secara legal, efisien, dan tidak menimbulkan risiko bagi seluruh pihak yang terlibat (Aliero dkk., 2020).

3.2.2 *Information Gathering*

Tahapan *Information gathering* merupakan tahapan mengumpulkan informasi terkait aplikasi web target yang akan di uji sebanyak mungkin yang akan berguna nantinya pada proses *penetration testing*. Informasi yang diperoleh bisa

berupa hasil scanning IP address web target yang berupa alamat IP publik yang dimiliki aplikasi web target tersebut lalu hasil scanning informasi menggunakan *who is* tentang informasi penting terkait domain dan alamat IP milik target pengujian maupun instansi pengelola dapat di ketahui melalui proses ini, dan scanning port menggunakan tools NMAP untuk mengidentifikasi *port-port* terbuka dan pelayanan yang berjalan pada server target untuk mengetahui hipotesis awal terkait celah keamanan yang terbuka yang berpotensi pada aplikasi web target, seluruh temuan itu di kumpulkan karena bisa jadi berguna untuk di tahapan tahapan selanjutnya(De Paoli & Johnstone, 2023).

3.2.3 *Vulnerability Analysis*

Tahapan *Vulnerability Analysis* atau analisis kerentanan, meruapakan tahapan yang krusial karena celah kerentanan yang akan dijadikan objek atau teknik pengujian ditemukan di sini sebagai validasi dari tahapan sebelumnya, pada tahapan ini aplikasi web target akan dipindai (*scanning*) untuk dianalisis dan identifikasi guna mengungkap potensi kerentanan yang ada pada sistem target yang akan diuji, Proses pemindaian kerentanan ini menggunakan alat pemindaian otomatis seperti *Acunetix* untuk menentukan celah kerentanan mana yang termasuk ke dalam kategori celah yang *critical* yang mana pada celah kritikal ini ditemukan detail dari kerentanan tersebut yang berupa parameter atau *endpoint* dan *payload* yang rentan untuk nantinya temuan tersebut sangat berguna untuk dilakukannya proses selanjutnya yaitu tahapan eksploitasi yang lebih mendalam terkait celah keamanan yang ditemukan (P. Kumar dkk., 2023).

3.2.4 *Exploitation*

Tahapan Eksploitasi merupakan proses pemanfaatan kerentanan *Time-Based blind SQL injection* untuk mengekstraksi data sensitif dari *database* dengan merancang sistem eksploitasi otomatis berupa *script pyton* yang mengirimkan *payload* SQL berbasis *delay* waktu menggunakan perintah SLEEP(n) sebagai pertanda berhasil tertebaknya karakter. Sistem otomatis ini mengelola pengiriman request, pengukuran waktu respon pemilihan logika algoritma pencarian serta check point. Tiga algoritma pencarian yaitu *Linear search,Binary search* dan *Ternary search* diimplementasikan untuk mengekstraksi nama *database* jumlah tabel daftar tabel hingga data sensitif seperti *username* dan juga *password* seluruhnya dapat dijalankan melalui *command line argument parser* dengan mode otomatis tertentu (Sommervoll dkk., 2024). Selain memperoleh hasil ekstraksi, sistem juga mencatat total *request* dan total waktu eksekusi sebagai parameter evaluasi sehingga tahap eksploitasi tidak hanya membuktikan bahwa keturunan dapat dimanfaatkan tetapi juga melakukan evaluasi efektifitas masing masing algoritma pada berbagai skenario delay.

3.2.5 *Post Exploitation*

Tahapan *post exploitation* merupakan langkah lanjutan setelah tahapan eksploitasi berhasil dilakukan dengan mengevaluasi hasil dari eksploitasi dan ekstraksi *database* yang sebelumnya sudah diperoleh, seperti nama *database,* jumlah dan nama tabel jumlah kolom, nama akun *user* hingga *username* dan juga *password.* (Tahir & Risky, 2024) evaluasi ini tidak hanya menekankan pada keberhasilan penebakan data, tetapi juga pada aspek parameter yang menyertainya,

yaitu total waktu eksekusi (*runtime*) dan jumlah permintaan (*request*) yang terkirim ke server selama proses ekstraksi berlangsung. Kedua parameter ini dipilih karena secara langsung mencerminkan pengaruh pada sistem target sekaligus efektivitas metode atau algoritma ekstraksi yang digunakan. (Dasmen dkk., 2023).

Analisis pada tahapan ini membandingkan kinerja tiga algoritma pencarian (*linear search*, *binary search* dan *ternary search*) pada skenario serangan *Time based Blind SQL injection* dengan variasi *delay sleep* antara 1 hingga 6 detik. Analisis ditujukan untuk menilai bagaimana perubahan nilai *sleep* mempengaruhi *runtime* dan juga *request* pada setiap algoritma, serta mengidentifikasi pola performa yang muncul. Hasil dari tahapan *post eksploitation* ini digunakan sebagai dasar dalam menentukan algoritma mana yang paling optimal efisien dan efektif dalam ekstraksi data pada *database* melalui kerentanan *Time base blind SQL injection.*

3.3 Penyusunan Kesimpulan & Saran

Tahap akhir dalam penelitian ini adalah menyusun kesimpulan dan rekomendasi berdasarkan hasil eksploitasi yang telah diperoleh. Kesimpulan mencakup temuan utama dari eksploitasi yang dilakukan serta implikasi dari hasil pengujian terhadap keamanan sistem target.. Penelitian ini juga memberikan rekomendasi mitigasi perbaikan untuk mencegah serangan dan eksploitasi *Blind SQL Injection* serupa pada aplikasi web (Bimandaru dkk., 2023). Seluruh hasil penelitian ini kemudian disusun dalam laporan akhir yang mencakup dokumentasi langkah-langkah eksploitasi, hasil analisis, serta kesimpulan yang diperoleh.