DETEKSI KERENTANAN JARINGAN PADA SERVER APACHE2 PORTAL AKADEMIK SMART STMIK PALANGKARAYA MENGGUNAKAN BUGHUNTER

PROPOSAL TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Penulisan Tugas Akhir
pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH

NAMA
Faryan Sandhika
C2255201097
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2026

DETEKSI KERENTANAN JARINGAN PADA SERVER APACHE2 PORTAL AKADEMIK SMART STMIK PALANGKARAYA MENGGUNAKAN BUGHUNTER

PROPOSAL TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Penulisan Tugas Akhir
pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH

NAMA
Faryan Sandhika
C2255201097
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2026

PERSETUJUAN

DETEKSI KERENTANAN JARINGAN PADA SERVER APACHE2 PORTAL AKADEMIK SMART STMIK PALANGKARAYA MENGGUNAKAN BUGHUNTER

Proposal tugas akhir ini telah disetujui dan disahkan

Pembimbing I,

Elia Zakharia, M.T
NIK. 19920526201604

Pembimbing II,

Rommi Kaestria, M.Kom
NIK. 198605242011103

Mengetahui
Ketua STMIK Palangkaraya,

Ir. Hj. Siti Maryamah, M.M
NIK. 196902021995004

Mengesahkan
Ketua Prodi Teknik Informatika,

Ir. Abdul Hadi, S.T., M.Kom
NIK. 198505102021101

PENGESAHAN

DETEKSI KERENTANAN JARINGAN PADA SERVER APACHE2 PORTAL AKADEMIK SMART STMIK PALANGKARAYA MENGGUNAKAN BUGHUNTER

Proposal tugas akhir ini telah diseminarkan, dinilai, dan disahkan
Oleh Tim Seminar pada Tanggal 13 Januari 2026

Tim Seminar Proposal :

1. Sam'ani, S.T., M.Kom ..... 
   Ketua

2. Elia Zakharia, M.T .....
   Sekretaris

3. Rommi Kaestria, M.Kom .....
   Anggota

DAFTAR ISI

PERSETUJUAN ..... **iii**
PENGESAHAN ..... **iv**
DAFTAR ISI ..... **v**
DAFTAR TABEL ..... **vi**
DAFTAR GAMBAR ..... **vii**
DAFTAR LAMPIRAN ..... **viii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 4
1.5 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA **7**
2.1 Kajian Teori 7
2.1.1 HTTP Security Headers 7
2.1.2 Server Apache2 8
2.1.3 Kerentanan Sistem (*System Vulnerability*) 9
2.1.4 *TLS/SSL* (*Transport Layer Security / Secure Sockets Layer*) 10
2.1.5 *Vulnerability Assessment* (VA) 10
2.1.6 *BugHunter* 11
2.1.7 VMware Workstation 12
2.1.8 Kali Linux 12
2.1.9 CVSS (*Common Vulnerability Scoring System*) 12
2.1.10 Prinsip *Security Misconfiguration* 13
2.2 Penelitian Terdahulu Yang Relevan 14
BAB III METODOLOGI PENELITIAN **22**
3.1 Tinjauan Umum 22
3.2 Jenis penelitian 24
3.3 Desain Penelitian 24
3.4 Instrumen Penelitian 28
3.5 Analisis Kebutuhan 31
3.6 Desain 37
3.7 Jadwal Penelitian 40
DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2.1. Penelitian yang relevan 15
Tabel 3.1. Jadwal Penelitian Tugas Akhir 41

DAFTAR GAMBAR

Gambar 3.1 Desain Penelitian Audit Keamanan Server Apache2 25
Gambar 3.2 Analisis Proses Penelitian 35
Gambar 3.3 Desain Proses Penelitian 38

DAFTAR LAMPIRAN

Lampiran 1. Surat tugas pembimbing tugas akhir
Lampiran 2. Lembar konsultasi bimbingan Tugas Akhir
Lampiran 3. Surat izin penelitian
Lampiran 4. Surat tugas penguji seminar

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Digitalisasi layanan pendidikan telah menjadikan portal akademik sebagai bagian penting dari infrastruktur kampus yang menyimpan data sensitif seperti identitas mahasiswa, nilai, dan riwayat akademik. Tingginya ketergantungan institusi pendidikan terhadap layanan digital membuat portal akademik wajib memiliki konfigurasi keamanan yang kuat. Namun, tren global menunjukkan bahwa sektor pendidikan merupakan salah satu target serangan siber terbesar. (Qadir *et al.*, 2025) mengungkapkan bahwa aplikasi web pendidikan menjadi sasaran utama berbagai serangan otomatis karena banyak server menggunakan konfigurasi *default* yang tidak diperkuat *(unhardened).*

Pada level infrastruktur, portal akademik umumnya dijalankan menggunakan *Apache HTTP Server* atau *Nginx*. Namun, penelitian terbaru menunjukkan bahwa banyak institusi pendidikan masih menggunakan konfigurasi standar yang tidak aman. (Wijaya, Sasmita and Pratama, 2024) menemukan bahwa *server* akademik di perguruan tinggi masih belum menerapkan header keamanan penting seperti *Content-Security-Policy, HSTS, X-Frame-Options,* dan *X-Content-Type-Options.* Kondisi ini menyebabkan layanan akademik rentan terhadap serangan seperti *Cross-Site Scripting (XSS), clickjacking,* hingga *Man-in-the-Middle (MITM).*

Kerentanan server pendidikan semakin diperparah oleh implementasi *TLS/SSL* yang tidak optimal. (Altulaihan, Alismail and Frikha, 2023) menunjukkan

bahwa banyak institusi pendidikan masih menggunakan cipher lemah dan protokol *TLS* lama yang rentan terhadap serangan pemutusan enkripsi *(downgrade attack)* dan penyadapan transmisi data. Situasi ini diperburuk oleh fakta bahwa sebagian besar institusi pendidikan tidak melakukan audit keamanan secara berkala.

Bukti empiris dari penelitian lain memperkuat risiko ini. (Cruz, Almeida and Oliveira, 2023) menunjukkan bahwa konfigurasi *server* yang tidak di-*hardening* menyebabkan meningkatnya jumlah serangan otomatis yang berhasil masuk ke permukaan aplikasi. Mereka juga menekankan bahwa alat *scanning open-source* dapat mengidentifikasi kelemahan konfigurasi sebelum dieksploitasi oleh penyerang. Hal tersebut didukung oleh penelitian (Syarifudin and Setiyani, 2023), yang menemukan 11 kerentanan pada sistem akademik perguruan tinggi, termasuk header keamanan yang hilang dan konfigurasi *TLS* yang lemah.

Meskipun banyak penelitian membahas keamanan *web*, sebagian besar dilakukan dalam ruang laboratorium atau simulasi, bukan pada *server* akademik yang benar-benar beroperasi. Selain itu, belum ada penelitian yang secara langsung menggunakan *BugHunter* sebagai alat utama untuk mendeteksi *misconfiguration server Apache2* pada portal akademik dan memetakan tingkat risiko secara kuantitatif melalui model *before–after.* Dengan minimnya penelitian empiris di lingkungan kampus, institusi pendidikan tidak memiliki acuan yang jelas untuk menerapkan *hardening server* berbasis data nyata.

Berdasarkan celah tersebut, penelitian ini dilakukan untuk menganalisis konfigurasi keamanan *server Apache2* pada Portal Akademik SMART STMIK Palangkaraya menggunakan *BugHunter*. Fokus penelitian mencakup deteksi

kerentanan, klasifikasi tingkat risiko menggunakan *CVSS v3.1,* serta evaluasi efektivitas mitigasi seperti penambahan *security headers,* optimasi *TLS,* dan penguatan kontrol akses. Penelitian ini dilakukan secara terotorisasi, berbasis *real case,* dan diharapkan menghasilkan temuan empiris yang dapat digunakan langsung oleh tim IT kampus akademik.

1.2 Perumusan Masalah

"Bagaimana melakukan deteksi kerentanan jaringan pada server Apache2 yang digunakan oleh Portal Akademik SMART STMIK Palangkaraya dengan memanfaatkan alat *BugHunter*, sehingga hasil pemindaian tersebut dapat memberikan gambaran menyeluruh mengenai kondisi keamanan server serta mengidentifikasi konfigurasi yang berpotensi menimbulkan risiko keamanan?"

1.3 Batasan Masalah

Agar penelitian ini terarah dan fokus pada tujuan yang ingin dicapai, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Penelitian ini hanya difokuskan pada proses deteksi dan analisis kerentanan jaringan pada *server web* yang digunakan oleh Portal Akademik SMART STMIK Palangkaraya sebagai studi kasus nyata *(real case),* dengan menggunakan alat *BugHunter* sebagai sistem pemindaian *(vulnerability scanner)* berbasis *open-source*
b. Seluruh proses pengujian dan analisis dilakukan pada lingkungan virtual menggunakan *VMware* dengan sistem operasi *Kali Linux,* di mana aktivitas pemindaian dilakukan dari sisi klien *(external scanning)* tanpa mengubah konfigurasi *internal server* secara langsung.

c. Penelitian ini berfokus pada identifikasi konfigurasi keamanan *server* (seperti *security headers,* konfigurasi *TLS/SSL,* dan pengaturan akses) tanpa melakukan serangan eksploitasi aktif *(penetration attack)* maupun modifikasi terhadap sistem asli.

1.4 Tujuan dan Manfaat

a. Tujuan

Untuk mengidentifikasi dan menganalisis tingkat kerentanan jaringan pada *server Apache2* Portal Akademik SMART STMIK Palangkaraya menggunakan alat *BugHunter* berbasis *open-source* dalam lingkungan *virtual VMware (Kali Linux),* guna memperoleh gambaran empiris mengenai kondisi keamanan *server* serta menghasilkan rekomendasi konfigurasi yang dapat meningkatkan perlindungan terhadap ancaman siber.

b. Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi Penulis

Memperoleh pemahaman mendalam mengenai mekanisme deteksi kerentanan jaringan, konfigurasi keamanan pada *server Apache2*, serta penerapan alat *BugHunter* dalam melakukan *vulnerability assessment* pada sistem berbasis web yang dijalankan di lingkungan virtual menggunakan *VMware (Kali Linux).*

2. Bagi Kampus (STMIK Palangkaraya)

Hasil penelitian ini dapat dijadikan referensi dalam meningkatkan keamanan portal akademik SMART, khususnya dalam mengidentifikasi serta

memperbaiki potensi celah keamanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab.

3. Bagi Pengelola TI atau Administrator Server

   Memberikan panduan empiris dalam penerapan analisis keamanan *server* berbasis *BugHunter,* serta membantu menentukan langkah mitigasi yang tepat terhadap kerentanan yang terdeteksi pada *server Apache2*, tanpa mengganggu stabilitas maupun kinerja layanan sistem akademik.

4. Bagi Peneliti Selanjutnya

   Menyediakan dasar ilmiah dan acuan teknis untuk penelitian lanjutan di bidang keamanan jaringan, khususnya yang berkaitan dengan pengujian kerentanan *(vulnerability testing),* penerapan *web security scanning*, dan integrasi alat analisis keamanan *open-source* seperti *BugHunter* pada infrastruktur jaringan pendidikan tinggi.

1.5 Sistematika Penulisan

Agar penulisan skripsi ini terarah, sistematis, dan mudah dipahami, maka disusun berdasarkan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang tinjauan pustaka yang diambil dari penelitian yang relevan beserta susunan kajian teori yang disesuaikan dengan tema Tugas Akhir.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang tahapan yang dilakukan peneliti dalam mengumpulkan informasi atau data yang dibutuhkan.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Teori

2.1.1 HTTP Security Headers

Keamanan jaringan merupakan serangkaian mekanisme yang dirancang untuk melindungi aset digital, data, dan komunikasi dari ancaman siber maupun akses tidak sah. Pada lingkungan perguruan tinggi, portal akademik merupakan layanan inti yang menangani data sensitif seperti nilai mahasiswa, identitas, serta riwayat kegiatan perkuliahan. Penelitian (Qadir *et al.*, 2025) menunjukkan bahwa sektor pendidikan menjadi target serangan siber terbesar karena sebagian besar institusi masih mengandalkan konfigurasi server *default*, tidak menerapkan pembaruan keamanan, serta jarang melakukan audit kerentanan secara berkala.

Jenis ancaman yang paling dominan pada layanan akademik berbasis web meliputi:

- *Cross-Site Scripting (XSS)*
- *Clickjacking*
- *Security Misconfiguration*
- *Man-in-the-Middle (MITM)*
- *Session Hijacking*

Kelemahan ini muncul ketika *server* tidak dikonfigurasi dengan baik, tidak menerapkan *header* keamanan, atau menggunakan protokol komunikasi yang tidak terenkripsi.

2.1.2 Server Apache2

Apache HTTP Server (Apache2) merupakan *web server* yang paling banyak digunakan secara global. Apache Software Foundation (2021) menyebutkan bahwa *Apache* populer karena fleksibilitas modul, stabilitas, serta kompatibilitas lintas platform.

(Mubaiwa and Mukosera, 2022) menjelaskan bahwa konfigurasi *server* yang tidak tepat, misalnya direktori yang tidak dilindungi, informasi *server* yang terekspos (*server signature*), atau *SSL* yang tidak dikonfigurasi dengan benar, termasuk dalam kategori *Security Misconfiguration*, salah satu penyebab utama kebocoran data dan titik masuk serangan.

Apache memiliki beberapa modul keamanan penting, seperti:

- *Mod_security*

  Berfungsi sebagai *Web Application Firewall (WAF)* untuk mendeteksi dan memblokir serangan *web*.

- *Mod_ssl*

  Mengelola enkripsi *TLS/SSL* untuk komunikasi aman.

- *.htaccess Access Control*

  Mengatur hak akses berdasarkan IP, host, atau metode tertentu.

- *Header Module*

  Mengaktifkan *HTTP Security Headers* untuk melindungi browser dari serangan berbasis web.

Penelitian (Wijaya, Sasmita and Pratama, 2024) menemukan bahwa *server Apache* di banyak institusi pendidikan masih menggunakan konfigurasi *default*,

menyebabkan munculnya celah seperti *TLS* lemah, *header* keamanan tidak aktif, dan kontrol akses tidak ketat.

2.1.3 Kerentanan Sistem (*System Vulnerability*)

NIST (2022) mendefinisikan kerentanan *(vulnerability)* sebagai kelemahan pada sistem, konfigurasi, atau implementasi yang dapat dieksploitasi oleh penyerang. Kerentanan terbagi menjadi:

1. Kesalahan konfigurasi *(misconfiguration)*
2. *Protocol weakness (TLS/SSL* lemah*)*
3. *Outdated software*
4. *Vulnerable modules/dependencies*

Penelitian (Syarifudin and Setiyani, 2023) menemukan lebih dari 11 kerentanan pada sistem akademik perguruan tinggi akibat konfigurasi server yang tidak diperkuat *(non-hardened).* Kesalahan ini mencakup header keamanan hilang, port tidak terlindungi, serta sertifikat *TLS* tidak *valid.*

Kerentanan yang menjadi fokus penelitian ini meliputi:

- *HTTP Security Headers*
- *Konfigurasi TLS/SSL*
- *Access Control*

Tingkat risiko dihitung menggunakan *CVSS v3.1,* yang menyediakan metode kuantitatif untuk menilai tingkat keparahan kerentanan.

2.1.4 *TLS/SSL* (*Transport Layer Security / Secure Sockets Layer*)

TLS/SSL merupakan protokol enkripsi untuk menjaga kerahasiaan dan integritas data yang dikirim melalui jaringan. Standar modern *TLS 1.3 (RFC 8446)* memberikan keamanan lebih kuat dengan menghapus cipher yang sudah usang. Kesalahan konfigurasi *TLS* menyebabkan:

- *Downgrade Attack*
- *Man-in-the-Middle*
- *penggunaan cipher lemah*
- *sertifikat digital tidak valid*

Dalam penelitian ini, modul *TLS Scanner* pada *BugHunter* digunakan untuk menilai apakah *server* Portal SMART sudah mengikuti standar keamanan yang direkomendasikan.

2.1.5 *Vulnerability Assessment* (VA)

Menurut NIST SP 800-115 (2022), *Vulnerability Assessment* adalah proses identifikasi, pengukuran, dan analisis kerentanan sistem. Tahapannya meliputi:

1. *Planning*
2. *Scanning*
3. *Analysis*
4. *Reporting*

Penelitian ini menitikberatkan pada tahap Scanning dan Analysis menggunakan *BugHunter,* yang bertujuan mendeteksi kesalahan konfigurasi dan kelemahan yang berpengaruh pada keamanan *server.*

2.1.6 *BugHunter*

BugHunter adalah alat *vulnerability scanning* berbasis Python yang berfungsi mendeteksi berbagai jenis kerentanan pada aplikasi web. Penelitian (Qadir *et al.*, 2025) menegaskan bahwa *bug hunting* melalui pendekatan *Dynamic Application Security Testing* (*DAST)* terbukti lebih efektif mendeteksi kerentanan tingkat tinggi pada kategori *Broken Access Control* dan *Security Misconfiguration* dibandingkan metode statis.

BugHunter bekerja dengan melakukan analisis otomatis terhadap:

- *Misconfiguration* pada server
- *Header* keamanan yang hilang
- Informasi sensitif yang terekspos
- Masalah dan kelemahan *TLS/SSL*
- *Redirect* dan akses file tidak aman

Keunggulan *BugHunter* meliputi:

1. Hasil berupa data kuantitatif *(log, JSON, CSV)*
2. Ringan dan cepat
3. *Non-intrusive*, aman untuk server operasional
4. Mendukung metode *kuantitatif before vs after mitigation*
5. Fokus mendeteksi konfigurasi server, bukan eksploitasi aktif

BugHunter menjadi alat utama dalam penelitian ini karena mampu menghasilkan skor risiko yang objektif dan daftar kerentanan yang dapat dianalisis secara numerik, Penelitian (Abdulghaffar, Elmrabit and Yousefi, 2023) menambahkan

bahwa penggunaan lebih dari satu alat pemindaian dapat menurunkan tingkat *false positive* sehingga meningkatkan kualitas temuan *BugHunter*.

2.1.7 VMware Workstation

VMware Workstation adalah perangkat lunak virtualisasi yang digunakan untuk menjalankan sistem operasi berbasis *virtual*. Dalam penelitian ini, *VMware* berfungsi untuk:

- Menjalankan *Kali Linux*
- Mengisolasi proses pemindaian agar tidak berdampak pada perangkat utama
- Mengendalikan konfigurasi jaringan secara fleksibel
- Melakukan *scanning* dalam lingkungan terisolasi
- Menghindari risiko terhadap sistem operasional kampus

2.1.8 Kali Linux

Kali Linux merupakan distribusi Linux khusus keamanan yang menyediakan berbagai tool untuk analisis kerentanan. Fungsinya dalam penelitian:

- Menjalankan *BugHunter*
- Mengelola *dependency Python*
- Menyediakan lingkungan testing yang aman

2.1.9 CVSS (*Common Vulnerability Scoring System*)

CVSS adalah standar global untuk mengukur tingkat keparahan kerentanan. Faktor yang dianalisis:

- *Base Score*
- *Attack Vector*
- *Impact Metrics*

- *Exploitability Metrics*

CVSS digunakan sebagai alat ukur numerik dalam analisis hasil *scanning BugHunter.*

2.1.10 Prinsip *Security Misconfiguration*

Security Misconfiguration merupakan salah satu ancaman paling umum pada layanan web modern. Kerentanan ini muncul akibat:

- Pengaturan bawaan (*default*) yang tidak aman
- Header keamanan tidak aktif
- Konfigurasi *TLS* lemah
- Direktori atau file sensitif masih dapat diakses

Teori ini menjadi dasar bahwa konfigurasi server merupakan komponen utama yang harus diaudit dalam penelitian ini.

2.2 Penelitian Terdahulu Yang Relevan

Penelitian terdahulu diperlukan sebagai dasar untuk memahami perkembangan riset terkait keamanan server web, deteksi misconfiguration, analisis *header* keamanan, kelemahan *TLS/SSL*, serta efektivitas penggunaan alat pemindaian otomatis dalam mengidentifikasi kerentanan pada layanan berbasis web. Kajian terhadap penelitian sebelumnya membantu menggambarkan bagaimana berbagai metode audit keamanan diterapkan pada lingkungan pendidikan maupun *server* publik, serta menyoroti pola kelemahan konfigurasi yang sering ditemukan. Dengan meninjau penelitian terdahulu, diperoleh pemahaman mengenai sejauh mana kerentanan server terjadi akibat konfigurasi *default*, kurangnya aktivasi *header* keamanan, dan tidak optimalnya penerapan protokol *TLS* modern. Selain itu, kajian pustaka memungkinkan peneliti mengidentifikasi keterbatasan riset-riset sebelumnya, seperti minimnya penelitian yang dilakukan pada server akademik nyata serta kurangnya pendekatan berbasis *before–after* untuk mengevaluasi efektivitas mitigasi. Oleh karena itu, tinjauan terhadap penelitian terdahulu menjadi langkah penting dalam memperkuat landasan ilmiah penelitian ini serta memastikan posisi penelitian dalam konteks pengembangan metode audit keamanan server yang lebih komprehensif dan berbasis kasus nyata.

Tabel 2.1. Penelitian yang relevan

No	Penulis / Tahun	Topik Penelitian	Metode	Pembahasan	Hasil
1	*Wijaya et al (2024)*	*Pengujian Penetrasi Aplikasi Web pada Platform E-learning OASE Universitas Udayana Menggunakan Information System Security Assessment Framework (ISSAF) dan Open Source Security Testing Methodology Manual (OSSTMM)*	*Scanning konfigurasi server (vulnerability assessment)*	*Menilai konfigurasi keamanan server e-learning di beberapa perguruan tinggi, dengan fokus pada penggunaan Apache/Nginx dan penerapan HTTP Security Headers serta TLS.*	*Ditemukan beberapa misconfiguration, seperti header keamanan yang belum aktif dan konfigurasi TLS yang belum mengikuti standar modern, sehingga meningkatkan risiko serangan XSS dan MITM.*
2	*Qadir et al (2025)*	*Evaluasi Komparatif Pendekatan dan Alat untuk Pengujian Keamanan Aplikasi Web yang Efektif*	*Analisis pola serangan & log keamanan*	*Menganalisis pola serangan otomatis terhadap aplikasi web pendidikan pada berbagai institusi, serta memetakan jenis kerentanan yang sering muncul pada sistem akademik daring.*	*Sektor pendidikan teridentifikasi sebagai salah satu target paling rentan; banyak serangan memanfaatkan konfigurasi server yang lemah dan tidak dilakukan hardening secara berkala.*
3	*Altulaihan et al (2023)*	*Survei tentang Pengujian Penetrasi Aplikasi Web*	*Analisis header keamanan (header inspection)*	*Mengevaluasi penggunaan HSTS, CSP, X-Frame-Options, dan header lain pada website institusi pendidikan serta dampaknya terhadap mitigasi serangan berbasis browser.*	*Banyak website pendidikan tidak mengaktifkan CSP dan HSTS secara benar, sehingga masih rentan terhadap XSS, clickjacking, dan serangan berbasis konten lainnya.*

4	*Cruz et al (2023)*	*Solusi Open Source untuk Vulnerability Assessment: Analisis Perbandingan*	*Studi komparatif scanner*	*Membandingkan beberapa alat pemindaian kerentanan open-source pada skenario nyata, serta mengukur kemampuan masing-masing alat dalam mendeteksi misconfiguration dan kelemahan aplikasi web.*	*Scanner yang ringan, terstruktur, dan mudah dikonfigurasi terbukti lebih efektif digunakan pada lingkungan server real-case, terutama untuk kebutuhan audit berkala.*
5.	*Syarifudin & Setiyani (2023)*	*Analisis celah keamanan SIAKAD perguruan tinggi menggunakan Vulnerability Assessment*	Footprinting (Terminal, Zenmap, Whois, Wappalyzer) + scanning OWASP ZAP	*Melakukan pemetaan layanan dan pemindaian kerentanan pada Sistem Informasi Akademik (SIAKAD) di sebuah perguruan tinggi, lalu mengklasifikasikan jenis kerentanan dan tingkat risikonya.*	*Ditemukan 11 kerentanan (5 kategori medium dan 6 low) termasuk header keamanan yang hilang dan konfigurasi TLS yang kurang optimal, serta disusun rekomendasi perbaikan konfigurasi server.*

Berdasarkan hasil telaah terhadap tujuh penelitian terdahulu yang relevan, dapat disimpulkan beberapa poin penting sebagai berikut:

1. server pendidikan memiliki tingkat kerentanan yang tinggi akibat penggunaan konfigurasi *default* dan rendahnya frekuensi audit keamanan berkala. Penelitian (Wijaya, Sasmita and Pratama, 2024). dan (Qadir *et al.*, 2025) menunjukkan bahwa server kampus umumnya masih menggunakan konfigurasi standar bawaan yang tidak aman, seperti header keamanan yang tidak aktif, *TLS* yang belum diperbarui, serta tidak adanya mekanisme kontrol akses yang memadai. Kondisi ini menyebabkan server akademik berada pada tingkat risiko serangan yang tinggi, terutama dari bot otomatis yang aktif menargetkan layanan web institusi pendidikan.
2. *HTTP Security Headers* merupakan aspek pengamanan yang paling sering diabaikan, padahal memiliki peran penting sebagai lapisan mitigasi terhadap serangan berbasis browser. Penelitian (Altulaihan, Alismail and Frikha, 2023) menemukan bahwa mayoritas situs pendidikan tidak menerapkan header penting seperti *Content-Security-Policy, Strict-Transport-Security,* maupun *Referrer-Policy.* Tidak diterapkannya *header* tersebut membuat website rentan terhadap serangan *Cross-Site Scripting (XSS), clickjacking,* dan kebocoran informasi sensitif. Penelitian terdahulu menunjukkan bahwa absennya header keamanan seperti *HSTS, CSP,* dan *X-Frame-Options* merupakan salah satu bentuk misconfiguration yang paling sering ditemukan pada *server* Apache2 di institusi pendidikan (Cruz, Almeida and Oliveira, 2023). Kondisi ini dapat meningkatkan risiko serangan seperti *MITM,*

clickjacking, dan *content-sniffing.* Dengan demikian, pemeriksaan konfigurasi *header* keamanan menjadi bagian penting dalam proses *vulnerability assessment* pada penelitian ini.

3. penelitian mengenai pemanfaatan *scanner open-source* menunjukkan efektivitas tinggi dalam mendeteksi *misconfiguration* secara *non-intrusive.* (Cruz, Almeida and Oliveira, 2023) menegaskan bahwa *scanner open-source* lebih efisien dibandingkan tools enterprise, terutama dalam konteks analisis konfigurasi server pendidikan yang membutuhkan metode pemindaian yang aman dan ringan. *Scanner* seperti *BugHunter* dinilai unggul karena memberikan keluaran kuantitatif yang mudah diinterpretasikan *(JSON, CSV, HTML)* serta dapat digunakan dalam lingkungan nyata tanpa mengganggu layanan produksi. Temuan ini mendukung pemilihan *BugHunter* sebagai alat utama dalam penelitian ini.
4. Penelitian (Syarifudin and Setiyani, 2023) menemukan bahwa sistem akademik di perguruan tinggi Indonesia cenderung memiliki *misconfiguration* pada *level* fundamental. Dalam studinya, ditemukan 11 kerentanan yang didominasi oleh *header* tidak aktif, sertifikat *TLS* tidak valid, direktori sensitif yang dapat diakses, serta pengaturan autentikasi yang lemah. Penelitian tersebut menekankan bahwa sumber kerentanan tidak selalu berasal dari aplikasi, tetapi justru lebih sering berasal dari konfigurasi server yang tidak diperkuat. Hal ini memperkuat urgensi penelitian ini yang berfokus pada identifikasi kondisi konfigurasi *server Apache2* pada portal akademik SMART melalui pendekatan scanning terotorisasi.

5. berdasarkan keseluruhan penelitian terdahulu, belum ada penelitian yang secara khusus menggabungkan tiga aspek penting sekaligus: deteksi *misconfiguration* pada *server* akademik berbasis *Apache2,* penggunaan *scanner open-source* seperti *BugHunter,* serta analisis tingkat risiko menggunakan model kuantitatif *(CVSS v3.1)* dalam skenario *before–after.* Penelitian-penelitian sebelumnya cenderung membahas satu aspek saja secara terpisah, seperti analisis *header* keamanan tanpa *TLS*, atau analisis *TLS* tanpa pemindaian konfigurasi berbasis *tool open-source.* Dengan demikian, penelitian ini mengisi celah yang belum ditangani oleh studi sebelumnya, yaitu melakukan audit keamanan *server* akademik berbasis *Apache2* secara *real-case* dengan metode kuantitatif dan pemindaian terotorisasi menggunakan *BugHunter.*

Berdasarkan hasil telaah terhadap lima penelitian terdahulu yang relevan, dapat disimpulkan bahwa kerentanan pada *server web* pendidikan merupakan permasalahan yang muncul secara konsisten dan berulang dalam berbagai studi. Penelitian (Wijaya, Sasmita and Pratama, 2024) dan (Qadir *et al.*, 2025) menunjukkan bahwa institusi pendidikan berada pada tingkat risiko yang tinggi karena sebagian besar server akademik masih menggunakan konfigurasi *default* tanpa proses *hardening* yang memadai. Kondisi ini menyebabkan *server* kampus menjadi sasaran empuk bagi botnet maupun penyerang manual, terutama pada aspek konfigurasi *HTTP* dan keamanan enkripsi *TLS/SSL.* Temuan tersebut mempertegas bahwa sebelum memperbaiki lapisan aplikasi, institusi pendidikan

harus memastikan bahwa lapisan *server* telah dikonfigurasi sesuai standar keamanan modern.

Selain itu, penelitian mengenai *HTTP Security Headers* memberikan pemahaman yang lebih dalam mengenai akar permasalahan yang sering ditemukan pada layanan pendidikan. Studi (Altulaihan, Alismail and Frikha, 2023) menunjukkan bahwa sebagian besar website pendidikan tidak menerapkan *header* kritikal seperti *HSTS, CSP, X-Frame-Options,* dan *Referrer-Policy.* Tidak diterapkannya *header* tersebut menjadikan sistem sangat rentan terhadap serangan seperti *XSS, clickjacking,* serta kebocoran informasi sesi pengguna. temuan penelitian terdahulu secara langsung mendukung urgensi dilakukannya audit konfigurasi mendalam pada server akademik.

Di sisi lain, penelitian terkait efektivitas *scanner open-source* yang dilakukan (Cruz, Almeida and Oliveira, 2023) memberikan bukti bahwa *tools* ringan dan *non-intrusive* justru lebih tepat digunakan untuk menganalisis konfigurasi server nyata, khususnya pada lingkungan pendidikan yang tidak dapat dihentikan operasionalnya. *Scanner* seperti *BugHunter,* yang dirancang untuk mendeteksi *misconfiguration, header* tidak aktif, serta kelemahan *TLS/SSL,* memiliki efektivitas yang tinggi karena dapat memberikan hasil terstruktur tanpa membebani *server target.* Hal ini menjadi dasar pemilihan *BugHunter* sebagai alat utama dalam penelitian ini, mengingat sifatnya yang aman dan mampu menghasilkan data kuantitatif untuk analisis risiko.

Lebih lanjut, penelitian (Syarifudin and Setiyani, 2023) memperkuat fakta bahwa sistem akademik di Indonesia cenderung memiliki kelemahan pada lapisan

konfigurasi dasar. Dalam kasus perguruan tinggi *XYZ*, ditemukan 11 kerentanan yang mayoritas berada pada kategori *medium* dan *low,* namun merupakan indikator kuat bahwa server pendidikan sering kali dikelola tanpa standar keamanan yang jelas. Kerentanan seperti direktori sensitif terbuka, sertifikat *TLS* tidak *valid,* serta mekanisme autentikasi lemah menjadi pola umum pada server akademik. Temuan tersebut sangat relevan dengan penelitian ini karena fokus utamanya adalah memeriksa konfigurasi *server Apache2* Portal SMART STMIK Palangkaraya yang berpotensi memiliki pola kelemahan serupa.

Meskipun berbagai penelitian terdahulu telah membahas keamanan web server, *misconfiguration,* dan efektivitas *scanner open-source,* sebagian besar penelitian tersebut masih berdiri pada fokus yang terpisah. Tidak ada penelitian yang mengintegrasikan analisis konfigurasi server akademik secara langsung, pemindaian terstruktur menggunakan *BugHunter,* serta evaluasi risiko sebelum dan sesudah penerapan mitigasi. Selain itu, belum ada penelitian yang memfokuskan studinya pada *server* akademik Indonesia dalam konteks *real-case* dengan pendekatan kuantitatif berbasis *CVSS v3.1.*

Oleh karena itu, penelitian berjudul "Deteksi Kerentanan Jaringan pada Server Apache2 Portal Akademik SMART STMIK Palangkaraya Menggunakan *BugHunter*" dirancang untuk mengisi celah penelitian tersebut.

BAB III

METODOLOGI PENELITIAN

3.1 Tinjauan Umum

Penelitian ini dilaksanakan dalam bentuk eksperimen terapan *(applied experimental research)* yang berfokus pada deteksi kerentanan jaringan pada server web yang digunakan oleh Portal Akademik SMART STMIK Palangkaraya. Portal ini merupakan sistem informasi akademik yang digunakan secara rutin oleh mahasiswa dan dosen untuk melakukan login, mengakses informasi perkuliahan, melihat nilai, dan aktivitas akademik lainnya. Dengan posisi tersebut, portal SMART menjadi salah satu komponen infrastruktur kritikal di lingkungan STMIK Palangkaraya yang harus dijaga keamanannya.

Lingkungan penelitian dirancang dalam bentuk laboratorium virtual yang terkontrol menggunakan VMware Workstation. Di atas VMware dijalankan mesin virtual dengan sistem operasi Kali Linux yang berfungsi sebagai *scanner environment.* Pada mesin virtual ini diinstal Python dan alat pemindaian kerentanan *BugHunter* yang akan digunakan untuk menganalisis konfigurasi keamanan server portal SMART, termasuk pengaturan *HTTP Security Headers,* konfigurasi *TLS/SSL,* dan potensi misconfiguration lainnya.

Akses ke server Portal Akademik SMART STMIK Palangkaraya dilakukan melalui jaringan internet dengan pendekatan *black-box* dari sisi klien, sehingga peneliti hanya memanfaatkan respons yang diberikan server tanpa mengubah konfigurasi *internal.* Seluruh aktivitas pemindaian dilakukan secara *non-intrusive*

tidak mengganggu operasional sistem yang sedang berjalan. Hasil pemindaian berupa laporan kerentanan akan dikumpulkan, dikelompokkan berdasarkan tingkat risiko, dan dianalisis secara kuantitatif menggunakan acuan seperti *CVSS v3.1.*

Dengan rancangan lingkungan penelitian tersebut, tahapan metodologi selanjutnya menjelaskan secara rinci proses instalasi dan konfigurasi *BugHunter* pada Kali Linux, tahapan pemindaian terhadap *server Apache2* Portal SMART STMIK Palangkaraya, teknik pengumpulan data, hingga prosedur analisis kerentanan dan penyusunan rekomendasi konfigurasi keamanan yang dapat dimanfaatkan oleh tim Teknologi Informasi STMIK Palangkaraya.

Sumber data yang digunakan dalam penelitian ini meliputi:

Laporan hasil pemindaian *BugHunter,* berupa:

- Daftar kerentanan *(vulnerabilities / misconfigurations),*
- Kategori risiko,
- Skor risiko *(risk score),*
- Informasi *header HTTP* dan konfigurasi *TLS/SSL,*

Log eksekusi dan output terminal *Kali Linux* selama proses pemindaian, Dokumentasi pendukung, seperti :

- Screenshot hasil scan,
- Catatan konfigurasi *VM* dan jaringan,
- Referensi standar keamanan (misalnya *CVSS v3.1 dan NIST SP 800-115)*.

Dengan rancangan lingkungan penelitian tersebut, tahapan metodologi selanjutnya akan menjelaskan secara rinci proses persiapan, instalasi, konfigurasi, pemindaian,

analisis, dan pelaporan hasil yang diperoleh dari penggunaan *BugHunter* terhadap *server Apache2* Portal Akademik SMART STMIK Palangkaraya.

3.2 Jenis penelitian

Jenis penelitian yang digunakan adalah Penelitian Kuantitatif Deskriptif dengan pendekatan Eksperimen Terapan *(Applied Experimental Research).*

Metode ini dipilih karena penelitian berfokus pada:

1. Penerapan langsung *(implementasi)* alat keamanan jaringan *BugHunter* dalam lingkungan nyata yang terkontrol (*VMware* dan *Kali Linux),*
2. Pengukuran terstruktur terhadap tingkat kerentanan *server Apache2* berdasarkan:
   - Tingkat keparahan *(severity),*
   - Kategori risiko,
   - Skor risiko yang kemudian dipetakan ke skala *CVSS v3.1,*
3. Analisis deskriptif kuantitatif, dimana hasil pemindaian *BugHunter* disajikan dalam bentuk tabel, grafik, dan statistik sederhana untuk menggambarkan kondisi keamanan server.

Penelitian ini tidak melakukan penetration testing agresif, tetapi menerapkan *non-intrusive vulnerability assessment,* sehingga fokus pada deteksi dan pengukuran risiko, bukan pada eksploitasi kerentanan.(Mubaiwa and Mukosera, 2022)

3.3 Desain Penelitian

Berdasarkan Gambar 3.1 Desain Penelitian Audit Keamanan *Server Apache2*, berikut merupakan penjelasan untuk setiap proses yang dilakukan dalam penelitian ini. Setiap tahapan disusun secara sistematis agar proses deteksi, analisis, dan

evaluasi kerentanan pada *server* Portal Akademik SMART STMIK Palangkaraya dapat dilakukan secara terarah dan dapat direplikasi.

Berikut flowchart mind map pada Gambar 3.1 Desain Penelitian Audit Keamanan Server Apache2. berikut:



Gambar 3.1 Desain Penelitian Audit Keamanan Server Apache2

1. **Studi Literatur**

   Tahap ini dilakukan dengan mengkaji berbagai referensi ilmiah mengenai keamanan jaringan, keamanan web *server*, konfigurasi Apache2, konsep *security misconfiguration*, serta metode *vulnerability assessment* menggunakan alat seperti *BugHunter*. Studi literatur bertujuan memberikan landasan teori sekaligus menentukan pendekatan penelitian yang tepat.

2. **Penentuan Lingkungan Pengujian**

   Pada tahap ini peneliti menentukan lingkungan eksperimen menggunakan VMware Workstation sebagai platform virtualisasi. Sistem operasi Kali Linux digunakan sebagai perangkat pemindaian, sedangkan target pengujian adalah *server Apache2* Portal Akademik SMART STMIK Palangkaraya. Tahap ini memastikan bahwa penelitian dilakukan pada lingkungan terkontrol dan aman.

3. **Persiapan & konfigurasi Awal**

   Peneliti menyiapkan *BugHunter* di dalam mesin Kali Linux, melakukan konfigurasi direktori output, memeriksa dependensi Python, serta memverifikasi koneksi ke server target. Tahap ini memastikan bahwa seluruh alat dapat berjalan dengan benar sebelum proses pemindaian dilakukan.

4. **Pemindaian Tahap 1 (*Before Mitigation*)**

   BugHunter dijalankan untuk melakukan pemindaian awal terhadap server Apache2 Portal SMART. Proses ini menghasilkan daftar kerentanan yang berkaitan dengan header keamanan, konfigurasi *TLS/SSL*, informasi server yang terekspos, serta pengaturan akses. Seluruh keluaran *BugHunter* disimpan dalam bentuk file laporan dan log untuk dianalisis lebih lanjut.

5. **Analisis Hasil**

   Hasil pemindaian awal dianalisis menggunakan parameter CVSS v3.1 untuk menentukan tingkat keparahan masing-masing temuan. Berdasarkan temuan tersebut, peneliti menyusun rekomendasi perbaikan konfigurasi server yang perlu diterapkan oleh administrator sistem.

6. **Penerapan Mitigasi**

   Berdasarkan rekomendasi yang telah disusun, admin server melakukan perubahan konfigurasi pada Apache2, antara lain penambahan header keamanan, penyesuaian versi dan cipher *TLS*, pengaturan ulang virtual host, serta pembatasan akses terhadap direktori atau *endpoint* tertentu. Peneliti mendokumentasikan perubahan konfigurasi yang dilakukan sebagai bagian dari data penelitian.

7. **Pemindaian Tahap 2 (*After Mitigation*)**

   Setelah mitigasi diterapkan, *BugHunter* dijalankan kembali dengan parameter yang sama seperti pemindaian tahap 1. Tujuannya adalah untuk mengukur perubahan tingkat risiko dan memastikan bahwa kerentanan kritis telah berkurang atau hilang. Hasil pemindaian tahap 2 kembali disimpan dalam bentuk laporan terstruktur.

8. **Penyusunan Dokumentasi Penelitian**

   Pada tahap akhir, peneliti membandingkan hasil pemindaian sebelum dan sesudah mitigasi berdasarkan jumlah kerentanan, kategori *severity*, serta nilai CVSS. Perbandingan ini digunakan untuk menilai efektivitas tindakan mitigasi yang dilakukan pada server Apache2 Portal SMART. Seluruh proses, temuan,

grafik perbandingan, dan interpretasi hasil kemudian disusun ke dalam laporan Tugas Akhir sebagai dasar penarikan kesimpulan dan saran.

3.4 Instrumen Penelitian

Instrumen penelitian merupakan metode dan alat yang digunakan untuk mengumpulkan data selama proses audit keamanan *server* dilakukan. Dalam penelitian ini, data diperoleh melalui observasi teknis pada proses pemindaian menggunakan *BugHunter,* pengumpulan berbagai dokumentasi konfigurasi server maupun hasil pemindaian, serta studi pustaka untuk memperkuat dasar teori. Jenis data yang digunakan terdiri dari data primer dan data sekunder, yang dikumpulkan melalui beberapa teknik berikut.

3.4.1 Observasi Langsung (Data Primer)

Observasi langsung dilakukan pada lingkungan virtual (VMware + Kali Linux) selama proses pemindaian kerentanan *server Apache2* Portal Akademik SMART berlangsung. Selama kegiatan observasi, peneliti memantau respons *server* terhadap *scanning BugHunter,* perubahan status header keamanan, perilaku *protokol TLS/SSL* ketika diuji, serta struktur output yang dihasilkan secara *real-time.*

Data primer yang diperoleh melalui observasi langsung meliputi:

- Respons *HTTP server* selama pemindaian (kode status, *redirect*, *error*).
- Daftar *header* keamanan yang aktif maupun tidak aktif pada Portal SMART.
- Informasi konfigurasi *TLS/SSL,* termasuk versi protokol, *cipher suite,* dan validitas sertifikat.

- Temuan misconfiguration seperti *directory exposure, missing headers,* atau akses file yang tidak aman.
- Deteksi kelemahan lain seperti informasi sensitif yang terekspos pada *endpoint* tertentu.
- Hasil *scoring severity* dan *CVSS* yang dihasilkan otomatis oleh *BugHunter.*
- Catatan kejadian *(timestamp)* dari setiap modul *scanning* yang dijalankan.

Data primer ini menjadi dasar untuk mengevaluasi tingkat kerentanan *server* dan mengukur efektivitas konfigurasi keamanan yang diterapkan pada *server Apache2* Portal Akademik SMART STMIK Palangkaraya.

3.4.2 Studi Dokumentasi (Data Sekunder)

Studi dokumentasi dilakukan dengan mengumpulkan seluruh file, catatan, dan bukti hasil pemindaian yang digunakan atau dihasilkan selama penelitian. Dokumentasi ini memudahkan proses validasi temuan, memperkuat analisis, serta memastikan setiap langkah penelitian dapat direplikasi secara ilmiah.

Dokumen yang dikumpulkan meliputi:

- Daftar File hasil pemindaian *BugHunter*:
  - *Report.html*
  - *Report.json*
  - *Report.csv*
- Screenshot hasil *scanning,* termasuk tampilan *error, warning,* dan daftar kerentanan.
- Catatan proses koneksi ke *server* Portal SMART (alamat *host, port,* status akses).

- *Log terminal* selama menjalankan *BugHunter* pada Kali Linux.
- File pendukung lainnya seperti konfigurasi jaringan VMware (*Bridge, NAT),* serta koneksi internet pengujian.
- Capture konfigurasi server sebelum dan sesudah mitigasi (jika dilakukan).

Data dokumentasi ini memastikan seluruh proses dapat dipertanggungjawabkan serta mendukung keabsahan data primer yang diperoleh melalui observasi langsung.

3.4.3 Studi Pustaka

Studi pustaka dilakukan dengan mengkaji literatur ilmiah yang berkaitan dengan topik keamanan *server, HTTP Security Headers, TLS/SSL,* analisis *misconfiguration,* dan teknik *vulnerability assessment* menggunakan alat *open-source.*

Literatur yang dikaji mencakup:

- Penelitian mengenai kerentanan *server* pendidikan (Wijaya et al., 2024; Qadir et al., 2025).
- Kajian mengenai kelemahan *header* keamanan dan dampaknya (Altulaihan et al., 2023).
- Studi mengenai efektivitas *scanner open-source* (Cruz, Almeida and Oliveira, 2023)
- Penelitian mengenai kerentanan sistem akademik di Indonesia (Syarifudin & Setiyani, 2023).
- Dokumentasi resmi BugHunter, Apache2 *Hardening Guide*, dan OWASP untuk acuan konfigurasi.

Studi pustaka digunakan untuk memperkuat landasan teori, membandingkan hasil penelitian dengan temuan sebelumnya, serta mengidentifikasi celah penelitian yang diisi oleh penelitian ini.

3.5 Analisis Kebutuhan

Bagian ini menjelaskan kebutuhan data, proses yang dianalisis, serta kelemahan yang ditemukan pada *server* Portal Akademik SMART STMIK Palangkaraya. Analisis ini dilakukan untuk memastikan bahwa proses audit keamanan dapat berjalan secara sistematis, terukur, dan sesuai dengan kondisi riil *server* yang digunakan kampus.

3.5.1 Analisis Data

Analisis data bertujuan mengidentifikasi jenis data apa saja yang dibutuhkan untuk melakukan audit keamanan menggunakan *BugHunter.* Pentingnya metadata server dalam analisis telah diperkuat oleh Penelitian (Wartschinski *et al.*, 2022) yang menegaskan bahwa informasi dasar server sangat mempengaruhi deteksi kerentanan. Jenis data ini juga sejalan dengan rekomendasi Penelitian (Ajayi et al., 2022) yang menyatakan bahwa analisis konfigurasi server membutuhkan kombinasi detail header, protokol, dan respons HTTP untuk memetakan risiko keamanan.

a. Data Primer

   Data primer berasal dari hasil pemindaian *(scanning)* langsung menggunakan BugHunter terhadap domain Portal Akademik SMART STMIK Palangkaraya. Data primer ini meliputi:

   1. Daftar kerentanan yang terdeteksi

- *Header* keamanan tidak aktif *(CSP, HSTS, X-Frame-Options, dll.)*
- Sertifikat *TLS/SSL* lemah atau tidak mengikuti standar modern
- *Directory indexing* atau file sensitif yang dapat diakses publik
- *Server* banner terbuka (versi *Apache2, OS, OpenSSL)*
- *Weak Access Control*

2. Detail teknis kerentanan
   - *Severity (low, medium, high)*
   - Deskripsi risiko
   - Parameter bermasalah (misalnya *Strict-Transport-Security = missing*)
   - *Endpoint* yang terdampak
3. Laporan hasil scanning
   - File *JSON* report
   - File *CSV* report
   - Log *scanning BugHunter*
   - Screenshot output untuk dokumentasi
4. Nilai risiko berdasarkan CVSS v3.1
   - *Base score*
   - *Attack vector*
   - *Impact score*
   - *Exploitability score*

   Nilai ini digunakan untuk menilai tingkat bahaya kerentanan secara kuantitatif.

b. Data Sekunder

Data sekunder digunakan sebagai informasi pendukung untuk memastikan hasil analisis sesuai dengan kondisi *server.*

Dokumen Pendukung:

- Jurnal ilmiah terkait *misconfiguration*
- Standar keamanan NIST & OWASP
- Dokumentasi resmi Python, *BugHunter, Apache2, dan TLS*

3.5.2 Analisis Proses

Analisis proses dilakukan untuk menggambarkan alur kerja audit keamanan mulai dari tahap persiapan hingga evaluasi hasil mitigasi. Proses ini bertujuan memastikan bahwa setiap langkah penelitian berjalan secara sistematis dan menghasilkan data yang valid untuk dianalisis. Secara umum, Penelitian (Chaleshtari *et al.*, 2023) menunjukkan bahwa konsistensi hasil dan repeatability sangat penting untuk mencegah false positive. proses penelitian terdiri dari beberapa tahapan sebagai berikut.

1. Persiapan Sistem

   Tahap pertama adalah persiapan sistem, yaitu menentukan target pengujian berupa *server Apache2* Portal Akademik SMART STMIK Palangkaraya, menyiapkan mesin Kali Linux sebagai perangkat pemindaian, serta melakukan instalasi dan konfigurasi *BugHunter.* Tahap ini memastikan lingkungan penelitian siap digunakan untuk melakukan audit keamanan tanpa mengganggu layanan kampus.

2. Pemindaian Awal (*Before*)

   Tahap berikutnya adalah pemindaian awal (*before mitigation*). Pada tahap ini, *BugHunter* dijalankan untuk mengidentifikasi kerentanan yang terdapat pada *server*. Hasil pemindaian berupa file *JSON, CSV,* dan *log* dianalisis untuk mengetahui jenis kesalahan konfigurasi seperti *header* keamanan yang hilang, konfigurasi *TLS* lemah, ataupun informasi server yang terbuka.

3. Analisis Hasil

   Setelah hasil diperoleh, peneliti melakukan analisis kerentanan. Proses ini mencakup klasifikasi tingkat risiko menggunakan *CVSS v3.1*, penentuan prioritas kerentanan yang harus diperbaiki, serta penyusunan rekomendasi mitigasi berdasarkan standar keamanan *server* modern.

4. Mitigasi & Perbaikan

   Selanjutnya dilakukan tahap mitigasi dan perbaikan, di mana rekomendasi seperti aktivasi header keamanan, penguatan *TLS/SSL*, dan *hardening* konfigurasi *server* diterapkan oleh administrator. Tahap ini bertujuan menutup celah yang ditemukan pada pemindaian pertama.

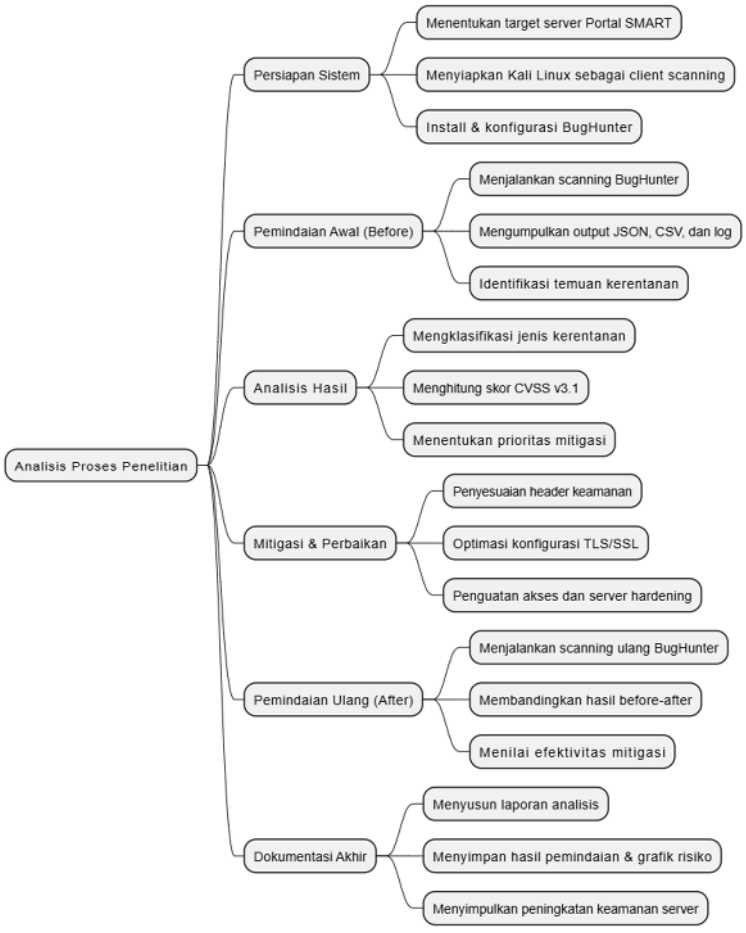
5. Pemindaian Ulang (*After*)

   Untuk mengevaluasi efektivitas mitigasi, peneliti melaksanakan pemindaian ulang (*after mitigation*) menggunakan *BugHunter*. Hasil pemindaian kedua kemudian dibandingkan dengan pemindaian pertama untuk melihat perubahan jumlah kerentanan maupun skor risiko.

6. Dokumentasi Akhir

   Tahap terakhir adalah dokumentasi akhir, yaitu menyusun laporan lengkap yang mencakup hasil pemindaian, analisis *before–after*, grafik perubahan risiko, serta kesimpulan mengenai tingkat peningkatan keamanan server.

Berikut flowchart mind map pada Gambar 3.2 Analisis Proses Penelitian. berikut:



Gambar 3.2 Analisis Proses Penelitian

3.5.3 Analisis Kelemahan

Analisis kelemahan dilakukan untuk mengidentifikasi kekurangan pada konfigurasi *server* Portal Akademik SMART serta keterbatasan metode yang

digunakan sebelumnya. Hasil analisis ini menjadi dasar perlunya dilakukan audit keamanan yang lebih terstruktur dan sesuai kondisi nyata *server* kampus. Kelemahan yang ditemukan adalah sebagai berikut:

1. *Server* kampus tidak memiliki mekanisme pengamanan adaptif, seperti *header* keamanan lengkap, *TLS* modern, maupun firewall aplikasi, sehingga rentan terhadap serangan berbasis *web* seperti *XSS, clickjacking,* dan *MITM.*
2. Konfigurasi *server* masih menggunakan pengaturan *default (unhardened),* termasuk cipher *TLS* lemah, server banner terbuka, dan akses direktori yang belum diamankan.
3. Belum ada penelitian sebelumnya yang menganalisis kondisi nyata Portal SMART, sehingga sebagian besar referensi menggunakan skenario laboratorium yang tidak merepresentasikan kompleksitas *server* kampus.
4. Tidak adanya mekanisme monitoring keamanan *real-time,* sehingga aktivitas berbahaya atau anomali trafik tidak dapat terdeteksi secara cepat.
5. Tidak tersedia audit konfigurasi berkala, mengakibatkan potensi celah keamanan tidak teridentifikasi dalam jangka waktu lama.
6. Kelemahan pada struktur akses dan kontrol, seperti tidak adanya *IP filtering* dan belum dimanfaatkannya .htaccess untuk membatasi akses *endpoint* sensitif.

Kumpulan kelemahan ini menunjukkan bahwa *server* Portal SMART memiliki risiko signifikan terhadap eksploitasi berbasis konfigurasi *(security misconfiguration),* yang merupakan salah satu kategori kerentanan paling sering terjadi menurut standar OWASP. Kondisi ini memperkuat urgensi perlunya

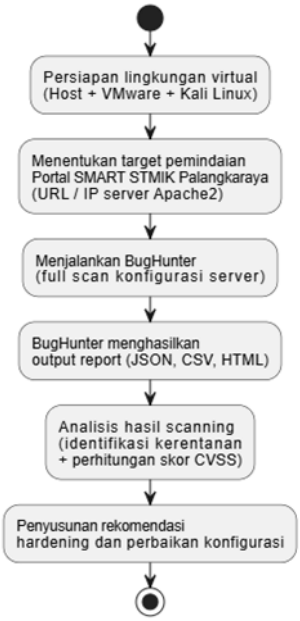
dilakukan audit konfigurasi menggunakan BugHunter agar institusi memiliki dasar teknis yang kuat untuk memperbaiki dan menstandarkan keamanan *server* mereka.

3.6 Desain

Desain penelitian digunakan untuk menggambarkan alur kerja, struktur tahapan, serta pendekatan yang digunakan dalam proses audit keamanan *server Apache2* pada Portal Akademik SMART STMIK Palangkaraya. Desain ini memastikan bahwa setiap langkah penelitian dilakukan secara sistematis, terukur, dan dapat direplikasi. Penelitian ini menggunakan desain penelitian eksperimen kuantitatif, di mana server diuji menggunakan *scanner BugHunter* sebelum dan sesudah dilakukan mitigasi konfigurasi. Hasil kedua proses tersebut kemudian dibandingkan untuk melihat peningkatan keamanan secara objektif berdasarkan nilai risiko *(CVSS v3.1).*

3.6.1 Desain Proses

Desain proses pada penelitian ini menggambarkan alur kerja audit keamanan *server Apache2* Portal Akademik SMART STMIK Palangkaraya menggunakan *BugHunter,* mulai dari persiapan lingkungan hingga penyusunan rekomendasi *hardening*. Alur proses tersebut divisualisasikan dalam sebuah flowchart pada Gambar 3.3 dan dapat dijelaskan sebagai berikut:



Gambar 3.3 Desain Proses Penelitian

1. Persiapan Lingkungan Virtual

   Tahap awal adalah menyiapkan lingkungan pengujian menggunakan satu laptop sebagai host yang menjalankan VMware Workstation. Di dalam VMware dibuat mesin virtual Kali Linux yang akan berperan sebagai client pemindai *(scanner)* eksternal. Pada tahap ini juga dilakukan instalasi BugHunter beserta seluruh *dependency* yang diperlukan.

2. Penentuan Target Pemindaian

   Setelah lingkungan siap, peneliti menentukan target pemindaian yaitu *server Apache2* yang digunakan oleh Portal Akademik SMART STMIK Palangkaraya. Target diinput dalam bentuk URL atau alamat IP yang dapat diakses dari mesin Kali Linux.

3. Menjalankan *BugHunter* (Proses *Scanning*)

   BugHunter dijalankan dari Kali Linux untuk melakukan pemindaian konfigurasi keamanan *server.* Pada tahap ini dilakukan full scan terhadap beberapa aspek, seperti *HTTP Security Headers,* konfigurasi *TLS/SSL,* informasi server, serta potensi *misconfiguration* lainnya. Proses ini mensimulasikan sudut pandang pihak luar yang mencoba menganalisis kelemahan *server.*

4. Pengumpulan Output Scanning

   Setelah proses pemindaian selesai, *BugHunter* menghasilkan beberapa bentuk keluaran, antara lain file laporan berformat *JSON, CSV, dan HTML. Output* ini berisi daftar kerentanan yang ditemukan, tingkat keparahan *(severity),* serta deskripsi teknis masing-masing temuan. Seluruh file laporan disimpan sebagai bahan analisis lebih lanjut.

5. Analisis Hasil dan Perhitungan Risiko

   Pada tahap ini peneliti membaca setiap temuan dari *BugHunter,* mengelompokkan jenis kerentanan, lalu menghitung tingkat risiko menggunakan acuan *CVSS v3.1.* Hasil analisis digunakan untuk memetakan kerentanan mana yang paling kritis bagi Portal SMART serta menentukan prioritas perbaikan.

6. Penyusunan Rekomendasi Konfigurasi

   Tahap terakhir adalah menyusun rekomendasi *hardening server* berdasarkan temuan dan nilai risiko. Rekomendasi meliputi aktivasi *HTTP Security Headers,* penguatan konfigurasi *TLS/SSL,* pengaturan *access control,* serta

langkah-langkah lain yang relevan. Rekomendasi ini kemudian dirangkum sebagai output utama penelitian dan akan dijadikan bahan evaluasi bagi tim IT STMIK Palangkaraya.

3.7 Jadwal Penelitian

Jadwal penelitian ditunjukkan dalam bentuk tabel yang berisi tahapan dalam penelitian dengan contoh seperti Tabel 3.1 dengan penyesuaian tahapan dari penelitian mahasiswa. Jadwal penelitian dapat dirincikan dengan menyertakan tahapan, tahun, bulan, dan minggu.

Tabel 3.1. Jadwal Penelitian Tugas Akhir

No.	Tahapan	September 2025				Oktober 2025				November 2025				Desember 2025			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1.	Identifikasi masalah																
2.	Penyusunan Latar Belakang & Kajian Awal																
3.	Studi Pustaka & Referensi																
4.	Perumusan Masalah & Tujuan Penelitian																
5.	Penyusunan Teori Pendukung & Penelitian Terdahulu																
6.	Penyusunan Kesimpulan Bab II																
7.	Desain Sistem / Arsitektur Penelitian																
8.	Penentuan Alat, Bahan, & Lingkungan Uji																
9.	Penyusunan Diagram																
10.	Finalisasi Bab III																
11.	Pengujian Sistem																

12.	Analisis & Evaluasi																
13.	Penyusunan Laporan Proposal Tugas Akhir																

DAFTAR PUSTAKA

Abdulghaffar, K., Elmrabit, N. and Yousefi, M. (2023) "Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners," *Computers*, 12(11), pp. 1–17. Available at: https://doi.org/10.3390/computers12110235.

Altulaihan, E.A., Alismail, A. and Frikha, M. (2023) "A Survey on Web Application Penetration Testing," *Electronics (Switzerland)*, 12(5). Available at: https://doi.org/10.3390/electronics12051229.

Ajayi, W., Ibeto, O., Olomola, T., & Madewa, M. (2022). Analysis of modern cybersecurity threat techniques and available mitigating methods. International Journal of Advanced Research in Computer Science, 13(2), 49–55. Available at: https://doi.org/10.26483/ijarcs.v13i2.6815

Apache Software Foundation. (2021). Apache HTTP Server Documentation. https://httpd.apache.org/docs

BugHunter Security Project. (2023). BugHunter: Lightweight web vulnerability scanner for security misconfiguration detection. GitHub. https://github.com

Chaleshtari, N.B. *et al.* (2023) "Metamorphic Testing for Web System Security," *IEEE Transactions on Software Engineering*, 49(6), pp. 3430–3471. Available at: https://doi.org/10.1109/TSE.2023.3256322.

Cruz, D.B., Almeida, J.R. and Oliveira, J.L. (2023) "Open Source Solutions for Vulnerability Assessment: A Comparative Analysis," *IEEE Access*, 11, pp. 100234–100255. Available at: https://doi.org/10.1109/ACCESS.2023.3315595.

FIRST. (2021). Common Vulnerability Scoring System v3.1: Specification Document. https://www.first.org/cvss

Mubaiwa, T.G. and Mukosera, M. (2022) "a Hybrid Approach To Detect Security Vulnerabilities in Web Applications," *International Journal of Computer Science and Mobile Computing*, 11(2), pp. 89–98. Available at: https://doi.org/10.47760/ijcsmc.2022.v11i02.011.

Qadir, S. *et al.* (2025) "Comparative evaluation of approaches & tools for effective security testing of Web applications," *PeerJ Computer Science*, 11, pp. 1–42. Available at: https://doi.org/10.7717/peerj-cs.2821.

Syarifudin, N.A. and Setiyani, L. (2023) "Analysis of Higher Education SIAKAD Website Security Gaps Using the Vulnerability Assessment Method," *International Journal of Multidisciplinary Approach Research and Science*, 1(03), pp. 332–344. Available at: https://doi.org/10.59653/ijmars.v1i03.177.

Wartschinski, L. *et al.* (2022) "VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Python," *Information and Software*

Technology, 144(i). Available at: https://doi.org/10.1016/j.infsof.2021.106809.

Wijaya, I.G.A.S.P., Sasmita, G.M.A. and Pratama, I.P.A.E. (2024) "Web Application Penetration Testing on Udayana University's OASE E-learning Platform Using Information System Security Assessment Framework (ISSAF) and Open Source Security Testing Methodology Manual (OSSTMM)," *International Journal of Information Technology and Computer Science*, 16(2), pp. 45–56. Available at: https://doi.org/10.5815/ijitcs.2024.02.04.

LAMPIRAN

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS PEMBIMBING TUGAS AKHIR

NO. 258/STMIK-3.C.2/AU/X/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama-nama tersebut di bawah ini :

1. Nama : Elia Zakharia, M.T
   NIK : 199205262016104
   Sebagai Pembimbing I dalam **PEMBUATAN PROGRAM**
2. Nama : Rommi Kaestria, M.Kom
   NIK : 198605242011103
   Sebagai Pembimbing II dalam **PENULISAN TUGAS AKHIR**

Untuk membimbing Tugas Akhir mahasiswa :

Nama	:	Faryan Sandhika
NIM	:	C2255201097
Program Studi	:	TEKNIK INFORMATIKA (55201)
Tanggal Daftar	:	17/10/2025
Judul Tugas Akhir	:	Deteksi Kerentanan Jaringan pada Server Apache2 Portal Akademik SMART STMIK Palangkaraya Menggunakan BugHunter

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya, dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 28 Oktober 2025

Ketua Program Studi,

Ir. Abdul Hadi, S.T., M.Kom
NIK. 198505102021101

TEMBUSAN :
1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip Prodi TI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No 114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikpik.ac.id - website : www.stmikpik.ac.id

KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : Fariyan Gandhika
NIM : C2255201097
Tanggal Persetujuan Judul : 28-10-2025
Judul Tugas Akhir : Deteksi kerentanan Jaringan Pada server Apache2 Portal Akademik Smart Politeknik menggunakan Pughunter

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	2-Des-2025	3-Des-2025	Revisi dari Bab 1, 2, dan 3	
2.	4-Des-2025	9-Des-2025	revisi dari Daftar Lampiran dan Bab 1, 2, dan 3	
3.	9-Des-2025	10-Des-2025	konsultasi revisi Perumusan masalah dan Bab 3	
4.	10-Des-2025	10-Des-2025	ACC Seminar	

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
	10-Des-2025	10 Des 25	perbaikan penulisan Bab I & II Terkait dengan kesalahan penulisan kata dan sistematik penulisan. Sesuaikan catatan perbaikan.	
	10-Des-2025	11 Des 25	perbaikan penulisan Bab I & II, III - terkait penulisan kata-kata - kekeliruan landasan teori - kesalahan sistematika penulisan, sesuaikan pedoman	
	11-Des-2025	12 Des 25	- perbaikan penulisan Bab I, II, III untuk - penggunaan istilah asing - penulisan referensi	
	12-Des-2025	13 Des 25	- Perbaikan untuk melengkapi kelengkapan berkas mulai cover sampai lampiran - lengkapi seluruh naskah mulai cover sampai lampiran	
		13 Des 25	acc Seminar proposal	

Menyetujui :

Dosen Pembimbing I,

ELIS LASMARIA

Dosen Pembimbing II,

PUTRI KAESTRA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G.Obos No.114 Telp.(0536) 3224593, 3225515 Fax. (0536) 3225515 Palangka Raya
email : humas@stmikplk.ac.id - website:www.stmikplk.ac.id

Nomor : 022/STMIK-C.2/AK/I/2026
Lampiran : -
Perihal : **Pemberian Ijin Penelitian Untuk Penyusunan Tugas Akhir**

Kepada :
Yth. Saudara (i) **FARYAN SANDHIKA**
Di –
Palangka Raya

Berdasarkan surat permohonan Saudara (i) tanggal 14 Januari 2026, mengenai permohonan ijin penelitian untuk penyusunan Tugas Akhir, maka dengan ini Ketua STMIK Palangkaraya memberikan ijin penelitian sebagaimana dimaksud sebagai berikut:

Nama	:	**FARYAN SANDHIKA**
NIM	:	C2255201097
Prodi (Jenjang)	:	Teknik Informatika (S-1)
Tahun Akad. (Semester)	:	2025/2026 (7)
Tempat Penelitian	:	STMIK Palangkaraya
Lama Penelitian	:	15 Januari s.d. 15 Februari 2026
Judul Tugas Akhir	:	

"Deteksi Kerentanan Jaringan Pada Server Apache2 Portal Akademik Smart STMIK Palangkaraya Menggunakan Bughunter"

Demikian disampaikan untuk dapat dipergunakan sebagaimana mestinya.

Palangka Raya, 15 Januari 2026
Ketua,

Hj. Siti Maryamah, MM.
NIK. 196902021995004

Tembusan :

1. Yth. Kaprodi Teknik Informatika;
2. Yth. Kepala Unit Lab Komputer;
3. Arsip.

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS
PENGUJI SEMINAR PROPOSAL TUGAS AKHIR

367/STMIK-3.C.2/AU/XII/2025

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama-nama berikut :

1. Nama : Sam'ani, S.T., M.Kom.
   NIK : 197703252005105
   Sebagai Ketua
2. Nama : Elia Zakharia, M.T
   NIK : 199205262016104
   Sebagai Sekretaris
3. Nama : Rommi Kaestria, M.Kom
   NIK : 198605242011103
   Sebagai Anggota

Untuk menguji seminar proposal tugas akhir mahasiswa :

Nama Mahasiswa	:	Faryan Sandhika
N I M	:	C2255201097
Hari / Tanggal Ujian	:	Selasa, 13 Januari 2026
Waktu	:	10.00 WIB
Ruangan	:	Seminar 2
Judul TA	:	DETEKSI KERENTANAN JARINGAN PADA SERVER APACHE2 PORTAL AKADEMIK SMART STMIK PALANGKARAYA MENGGUNAKAN BUGHUNTER

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 27 Desember 2025

Ketua Program Studi Teknik Informatika,

Ir. Abdul Hadi, S.T., M.Kom
NIK. 198505102021101

Tembusan :
1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip