TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

Deteksi Port Scanning pada Jaringan IoT dengan Pemantauan Web Real-Time

Bilal AlHafidz, Andri Firmansyah, Suherman

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Pelita Bangsa, Bekasi, Indonesia
Email: [1,*]bilalhafidz17@mhs.pelitabangsa.ac.id, [2]andrifirmansyah@pelitabangsa.ac.id, [3]suherman@pelitabangsa.ac.id
Email Penulis Korespondensi: bilalhafidz17@mhs.pelitabangsa.ac.id

Abstrak–Port scanning merupakan aktivitas rekonaisans yang dapat mendahului eksploitasi layanan pada perangkat Internet of Things (IoT). Penelitian ini merancang dan mengimplementasikan purwarupa deteksi port scanning TCP dengan pemantauan web real-time untuk jaringan lokal PT WIMISEC, Bekasi. Pengembangan dilakukan dengan pendekatan Research and Development dan tahapan Waterfall, sedangkan sistem mengintegrasikan aplikasi Laravel, basis data MySQL, serta mesin deteksi Python. Deteksi menggunakan aturan ambang berdasarkan jumlah port tujuan unik yang diakses oleh satu alamat IP dalam suatu jendela waktu. Evaluasi mencakup pengujian fungsional, lima TCP SYN Scan, lima TCP Connect Scan, tiga aktivitas browsing, dan tiga login SSH normal. Seluruh 10 percobaan scanning menghasilkan peringatan (detection rate teramati 100%), sedangkan tidak ada peringatan pada enam percobaan normal (false positive rate teramati 0% pada sampel terbatas). Rata-rata latensi dari identifikasi oleh mesin deteksi hingga peringatan tampil pada dashboard adalah 1,24 detik. Hasil ini menunjukkan kelayakan fungsional purwarupa untuk pemantauan awal pada LAN, bukan akurasi umum terhadap seluruh pola serangan. Keterbatasan evaluasi meliputi ukuran sampel kecil, belum adanya trafik campuran dan stress test, serta belum diujinya UDP scan, slow scan, pemindaian terdistribusi, dan teknik evasi.

Kata Kunci: Deteksi Anomali; Internet of Things; Keamanan Jaringan; Pemantauan Real-Time; Port Scanning

Abstract–Port scanning is a reconnaissance activity that may precede service exploitation on Internet of Things (IoT) devices. This study designs and implements a prototype for detecting TCP port scanning with real-time web monitoring in the local network of PT WIMISEC, Bekasi. The system was developed using a Research and Development approach and Waterfall stages, integrating a Laravel application, a MySQL database, and a Python detection engine. Detection applies a threshold rule based on the number of unique destination ports accessed by one source IP address within a time window. The evaluation covered functional testing, five TCP SYN scans, five TCP Connect scans, three browsing activities, and three normal SSH logins. All 10 scan trials generated alerts (an observed detection rate of 100%), while none of the six normal trials generated an alert (an observed false-positive rate of 0% in the limited sample). The mean latency from detection-engine identification to dashboard alert display was 1.24 seconds. These results demonstrate the prototype's functional feasibility for initial LAN monitoring rather than general accuracy across attack patterns. The evaluation remains limited by its small sample, the absence of mixed-traffic and stress testing, and the lack of tests involving UDP, slow, distributed, and evasive scans.

Keywords: Anomaly Detection; Internet of Things; Network Security; Port Scanning; Real-Time Monitoring

1. PENDAHULUAN

Perangkat Internet of Things (IoT) memperluas permukaan serangan jaringan melalui bertambahnya alamat IP, layanan, protokol, dan port aktif. Keterbatasan pembaruan serta variasi konfigurasi perangkat dapat meningkatkan risiko, tetapi penelitian ini tidak membebankan proses deteksi pada perangkat IoT. Pemantauan ditempatkan pada mesin terpisah agar target hanya berperan sebagai objek trafik, sehingga persoalan yang ditangani adalah visibilitas administrator terhadap aktivitas jaringan, bukan optimasi komputasi pada endpoint (Arshad et al., 2020; Suryawan et al., 2024).

Port scanning mengirimkan permintaan ke sejumlah port untuk mengidentifikasi host dan layanan yang aktif. Teknik ini sah untuk audit, tetapi pola yang sama dapat digunakan sebagai rekonaisans sebelum eksploitasi atau brute force (Muhyidin et al., 2022). Karena port terbuka belum dengan sendirinya membuktikan serangan, sistem pemantauan perlu membedakan akses rutin pada satu layanan dari akses terhadap banyak port dalam interval tertentu (Pebrianto, 2024). Fokus penelitian ini adalah menyediakan indikator awal berdasarkan pola tersebut, bukan menetapkan setiap pemindaian sebagai intrusi yang telah berhasil.

IDS dapat menggunakan signature, penyimpangan perilaku, atau kombinasi keduanya. Snort dan sistem berbasis pembelajaran mesin menyediakan cakupan yang lebih luas, tetapi membawa kebutuhan pengelolaan aturan, dataset, pelatihan, atau sumber daya komputasi yang berbeda (Arshad et al., 2020; Aldhaheri et al., 2024; Khan et al., 2023). Sistem berbasis aliran dan IDS ringan menunjukkan bahwa pemilihan fitur yang terbatas dapat mendukung pemantauan dengan kompleksitas lebih rendah (Javed et al., 2024; Mutambik, 2024). Konsekuensinya, kesederhanaan harus dibayar dengan cakupan deteksi dan ketahanan terhadap evasi yang lebih terbatas.

Penelitian terdahulu telah mendeteksi port scanning menggunakan Snort, firewall, notifikasi Telegram, dan visualisasi ELK Stack. Razzanda dan Koprawi (2024) menggabungkan Snort, iptables, dan Telegram untuk deteksi sekaligus pencegahan; Sabila et al. (2025) menggunakan aturan Snort untuk beberapa teknik Nmap; sedangkan Robbani et al. (2025) dan Zain et al. (2023) menyediakan analitik log melalui ELK Stack. Pendekatan tersebut unggul dalam cakupan aturan, respons, atau kedalaman visualisasi. Celah praktis yang dituju penelitian ini bukan menggantikan IDS tersebut, melainkan menyediakan purwarupa dengan alur operasional lebih sempit: registrasi target IoT, deteksi perilaku TCP berbasis port unik, log, dashboard, dan daftar hitam dalam satu aplikasi. Dengan demikian, kontribusinya terletak

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

pada integrasi dan kemudahan penerapan awal, bukan pada algoritma deteksi baru atau keunggulan akurasi atas sistem eksisting.

Pada lingkungan penelitian PT WIMISEC, pemantauan port scanning membutuhkan antarmuka terpusat yang menghubungkan identitas perangkat dengan kejadian jaringan. Kebutuhan diperoleh melalui observasi alur kerja dan wawancara informal dengan staf teknis. Purwarupa ditujukan untuk LAN skala kecil hingga menengah yang belum memerlukan fungsi SIEM atau IPS lengkap. Posisi ini membatasi klaim penelitian: sistem berfungsi sebagai lapisan observasi awal, sedangkan validasi, pemblokiran, dan investigasi lanjutan tetap menjadi tanggung jawab administrator.

Penelitian ini mengintegrasikan Laravel, MySQL, dan mesin deteksi Python. Mesin deteksi menghitung jumlah port tujuan unik yang diakses satu IP sumber dalam jendela waktu tertentu; kejadian yang melewati ambang disimpan dan ditampilkan pada dashboard. Tujuan penelitian adalah (1) merancang arsitektur pemantauan terpisah dari endpoint IoT, (2) mengimplementasikan deteksi TCP berbasis ambang, dan (3) mengevaluasi fungsi serta latensi notifikasi pada skenario LAN terbatas. Kontribusi yang diuji adalah kelayakan integrasi tersebut. Penelitian tidak mengklaim kebaruan algoritmik, superioritas terhadap IDS umum, ataupun generalisasi di luar skenario pengujian.

2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Penelitian dilaksanakan di PT WIMISEC, Bekasi, menggunakan pendekatan Research and Development karena menghasilkan purwarupa perangkat lunak berdasarkan kebutuhan pengguna. Kebutuhan dihimpun melalui observasi lingkungan jaringan, wawancara informal dengan staf teknis, dan studi literatur. Tahapan Waterfall—analisis, perancangan, implementasi, pengujian, dan evaluasi—digunakan untuk menjaga keterlacakan kebutuhan terhadap artefak dan hasil uji (Pressman & Maxim, 2020). Pemilihan ini bukan karena Waterfall dianggap paling adaptif untuk sistem keamanan, melainkan karena ruang lingkup purwarupa dibekukan pada deteksi TCP dan dashboard LAN selama penelitian. Untuk penerapan operasional yang menghadapi perubahan ancaman dan tuning aturan secara berkala, pengembangan iteratif lebih sesuai dan menjadi batasan pendekatan ini.

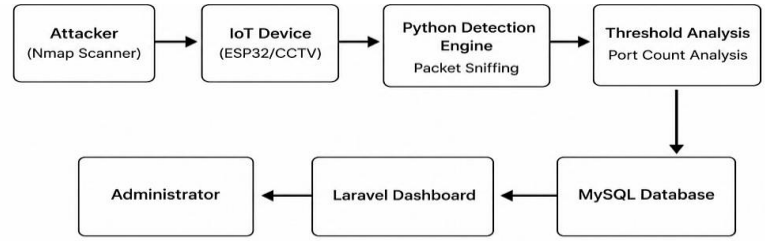
Objek teknis penelitian adalah lalu lintas jaringan yang menuju perangkat IoT atau perangkat dummy yang diregistrasikan pada aplikasi. Aktor sistem dibagi menjadi administrator, aplikasi web, basis data, mesin deteksi, perangkat target, dan perangkat penguji yang bertindak sebagai attacker. Administrator mengelola data perangkat dan membaca hasil pemantauan. Mesin deteksi mengamati koneksi jaringan dan meneruskan kejadian yang memenuhi kriteria pemindaian ke aplikasi. Basis data menyimpan identitas perangkat, waktu kejadian, alamat IP sumber, port tujuan, status deteksi, dan data blacklist. Batasan penelitian mencakup protokol TCP, jaringan lokal skala kecil, dan skenario pemindaian menggunakan Nmap. Sistem tidak dirancang sebagai IPS yang melakukan pemblokiran otomatis pada level jaringan. Ringkasan kebutuhan fungsional, nonfungsional, dan batasan sistem disajikan pada Tabel 1.

Tabel 1. Ringkasan kebutuhan sistem

Jenis	Kebutuhan utama
Fungsional	Login administrator, registrasi perangkat IoT, deteksi pola koneksi mencurigakan, penyimpanan log, peringatan dashboard, pencarian riwayat, dan pengelolaan blacklist.
Nonfungsional	Antarmuka responsif, waktu pembaruan cepat, dapat dijalankan pada server standar, mendukung jaringan lokal, dan mudah dikembangkan.
Batasan	Deteksi difokuskan pada TCP port scanning; pengujian dilakukan di LAN dengan perangkat IoT/dummy dan Nmap sebagai alat simulasi.

2.2 Arsitektur dan Tahapan Pengembangan

Arsitektur sistem menggunakan pola klien-server sebagaimana ditunjukkan pada Gambar 1. Perangkat penguji mengirimkan permintaan koneksi ke perangkat target yang alamat IP-nya telah terdaftar. Mesin deteksi berbasis Python menangkap dan mengelompokkan aktivitas koneksi, kemudian menjalankan analisis ambang. Ketika pola memenuhi kriteria pemindaian, data kejadian dikirim ke aplikasi Laravel dan disimpan dalam MySQL. Administrator mengakses dashboard untuk melihat ringkasan, log, perangkat, dan blacklist. Pemisahan antara mesin deteksi dan aplikasi web memudahkan pengembangan komponen deteksi tanpa mengubah seluruh modul antarmuka.



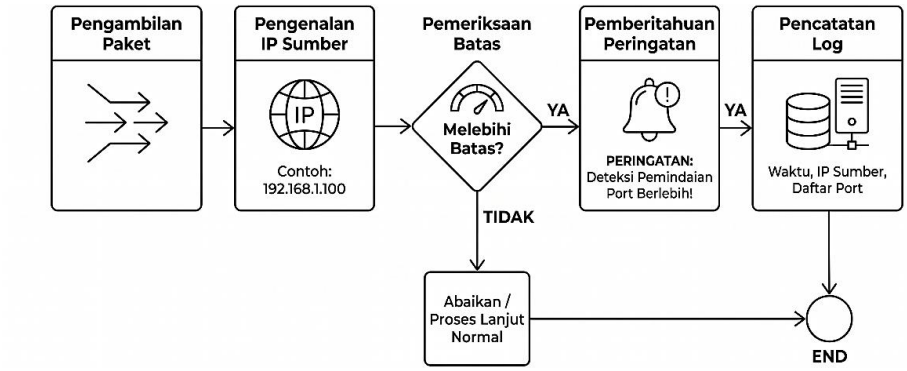
Gambar 1. Arsitektur sistem deteksi dan monitoring

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

Tahap analisis kebutuhan memetakan masalah, aktor, data, dan fitur. Tahap perancangan menghasilkan flowchart, use case, activity diagram, rancangan basis data, serta rancangan antarmuka. Tahap implementasi menggunakan PHP dan Laravel pada lapisan aplikasi web, Tailwind CSS pada tampilan, MySQL sebagai RDBMS, dan Python pada mesin deteksi. Laravel dipilih karena pola Model-View-Controller memisahkan logika, data, dan tampilan sehingga modul autentikasi, perangkat, log, dan blacklist dapat dikelola secara terstruktur. Implementasi aplikasi berbasis PHP-Laravel pada sistem inventori juga menunjukkan bahwa framework ini memadai untuk pengembangan web terstruktur dan cepat (Munada & Kurniawan, 2024; Puspitasari et al., 2022). MySQL digunakan karena mendukung relasi data dan operasi transaksi yang dibutuhkan oleh log monitoring (Sitanggang et al., 2022).

2.3 Mekanisme Deteksi Port Scanning dan Batasannya

Untuk setiap alamat IP sumber, mesin deteksi membentuk himpunan port tujuan unik yang diakses dalam jendela waktu Δt. Nilainya dinyatakan sebagai $N\_ports(IP\_s, \Delta t)$. Peringatan dibangkitkan ketika $N\_ports(IP\_s, \Delta t) \geq \theta$. Parameter yang digunakan pada implementasi adalah θ = [ISI NILAI AMBANG] port dan Δt = [ISI JENDELA WAKTU] detik. Nilai tersebut perlu dicantumkan karena menentukan sensitivitas: θ yang rendah meningkatkan peluang false positive, sedangkan θ yang tinggi atau Δt yang pendek dapat melewatkan pemindaian lambat. Aktivitas berulang pada satu port tidak menaikkan jumlah port unik, sehingga akses rutin ke satu layanan tidak langsung diklasifikasikan sebagai scanning. Data peringatan mencakup IP sumber, target, port, waktu, metode, dan status.



Gambar 2. Alur deteksi port scanning

Alur pada Gambar 2 mencakup pengambilan paket, pemeriksaan protokol dan target terdaftar, pengelompokan berdasarkan IP sumber, penghitungan port unik, serta pembandingan dengan ambang. Setelah jendela waktu berakhir, state pengamatan perlu dihapus atau diperbarui agar hitungan lama tidak memengaruhi kejadian baru. Desain ini ringan dan mudah diaudit, tetapi tidak memiliki mekanisme adaptasi otomatis. Slow scan dapat tetap berada di bawah θ pada setiap Δt; pemindaian terdistribusi membagi aktivitas ke banyak IP; sedangkan spoofing dapat mengaburkan identitas sumber. Karena itu, mekanisme ini hanya layak sebagai indikator awal. Penanganan teknik tersebut memerlukan korelasi lintas-jendela, agregasi berbasis target, baseline per perangkat, dan fitur paket atau aliran tambahan (Mutambik, 2024).

2.4 Skenario Pengujian

Evaluasi terdiri atas pengujian fungsional dan pengujian deteksi. Black-box testing memeriksa autentikasi, validasi perangkat, penyimpanan data, dashboard, log, dan blacklist. Pengujian deteksi menggunakan Nmap dari perangkat attacker ke target terdaftar, meliputi lima TCP SYN Scan dan lima TCP Connect Scan. Kontrol normal terdiri atas tiga aktivitas browsing dan tiga login SSH. Detection rate dihitung sebagai jumlah percobaan scanning yang menghasilkan alert dibagi seluruh percobaan scanning. False positive rate (FPR) teramati dihitung sebagai jumlah aktivitas normal yang salah diberi alert dibagi seluruh aktivitas normal. Latensi diukur sejak aktivitas dikenali mesin deteksi hingga alert tampil pada dashboard. Desain ini memvalidasi fungsi purwarupa, tetapi belum mencakup stress test, trafik campuran, pembanding sistem eksisting, interval kepercayaan yang bermakna, maupun generalisasi berbasis dataset.

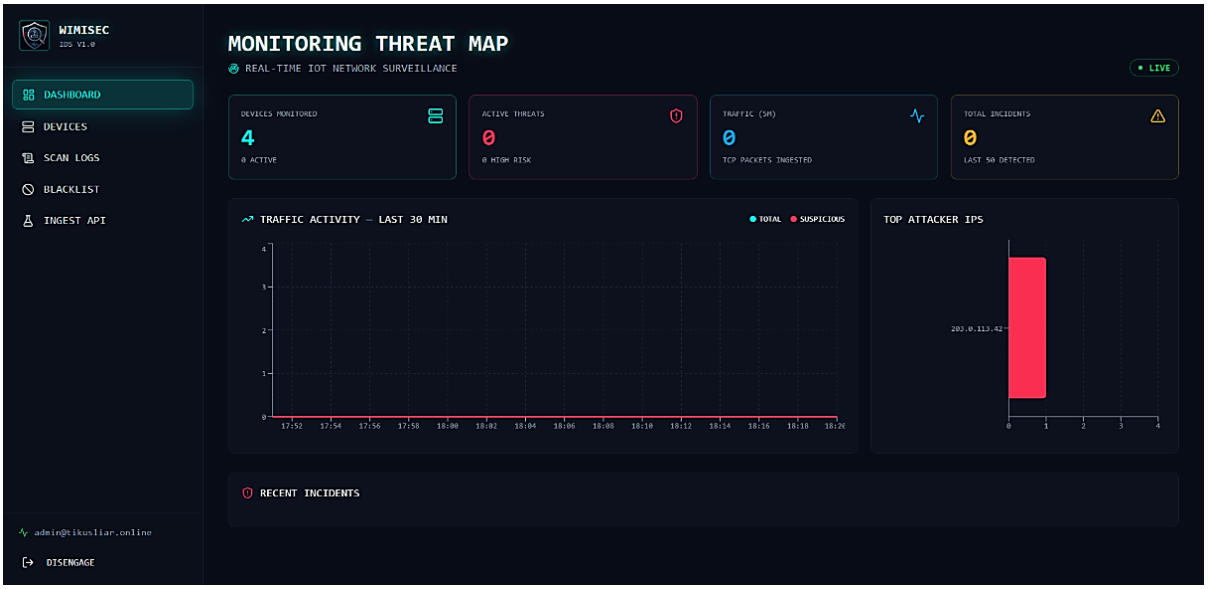
3. HASIL DAN PEMBAHASAN

3.1 Implementasi Antarmuka dan Integrasi Sistem

Implementasi menghasilkan aplikasi web dengan modul autentikasi, dashboard, registrasi perangkat, log aktivitas, dan blacklist. Setelah pengguna berhasil masuk, dashboard menampilkan jumlah perangkat yang dipantau, status aktivitas, informasi ringkas kejadian, serta visualisasi yang membantu administrator melihat kondisi jaringan dari satu halaman. Data pada dashboard berasal dari basis data yang diperbarui setelah mesin deteksi mengirimkan hasil analisis. Dengan mekanisme tersebut, pengguna tidak perlu menjalankan perintah terminal untuk membaca log dasar, sehingga proses pemantauan menjadi lebih mudah bagi administrator yang membutuhkan informasi cepat.

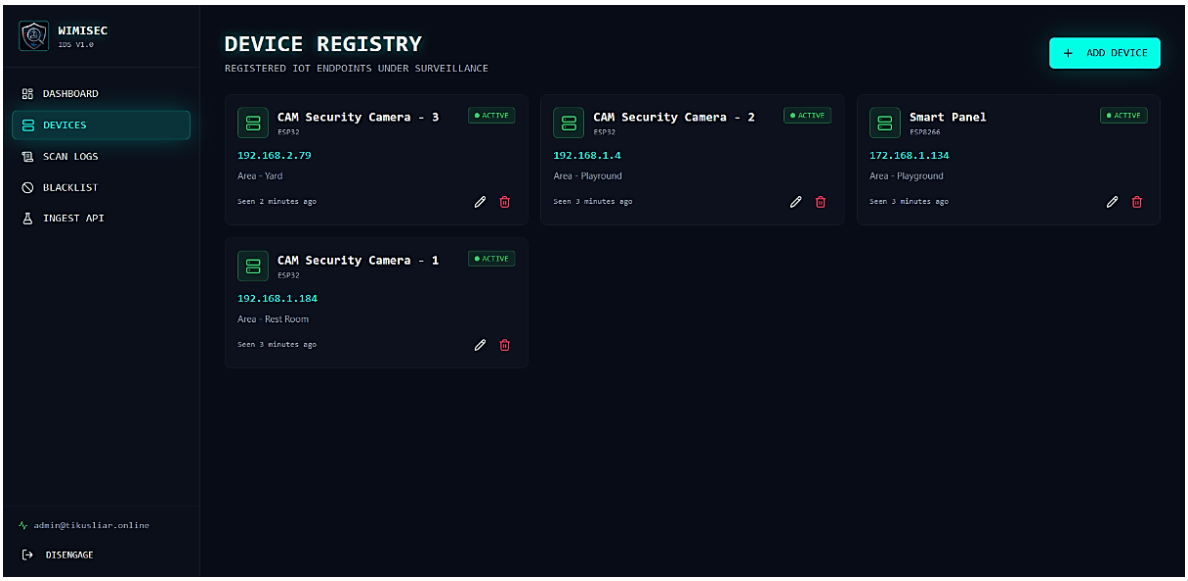
TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

Tampilan dashboard pada Gambar 3 menggunakan desain gelap dengan kartu ringkasan dan grafik aktivitas. Pemilihan struktur visual tersebut bertujuan memisahkan informasi prioritas, seperti jumlah perangkat, jumlah deteksi, dan status kejadian, dari rincian log. Dashboard tidak menggantikan analisis forensik mendalam, tetapi berfungsi sebagai lapisan situational awareness. Ketika aktivitas port scanning terdeteksi, administrator dapat menelusuri IP sumber dan port tujuan melalui halaman log, kemudian menambahkan alamat yang mencurigakan ke daftar hitam sebagai bahan tindakan lanjutan.



Gambar 3. Dashboard monitoring real-time

Modul registrasi perangkat menjadi penghubung antara objek yang dipantau dan mesin deteksi. Administrator memasukkan identitas perangkat, kategori, serta alamat IP yang valid melalui halaman registrasi yang ditunjukkan pada Gambar 4. Setiap data dapat diperbarui apabila perangkat berpindah alamat atau tidak lagi digunakan. Pemisahan data perangkat dari data log membuat riwayat kejadian tetap dapat ditelusuri tanpa harus menghapus identitas sumber pemantauan. Struktur tersebut juga memudahkan penambahan metadata pada penelitian lanjutan, misalnya lokasi perangkat, sistem operasi, protokol yang diizinkan, dan pemilik perangkat.



Gambar 4. Halaman registrasi perangkat IoT

Halaman perangkat menyimpan nama, tipe, dan alamat IP target. Validasi alamat IP diperlukan agar mesin deteksi hanya memproses target yang sesuai dengan konfigurasi. Halaman log menampilkan waktu, alamat IP penyerang, port target, dan status aktivitas. Contoh tampilan log pada Gambar 5 memperlihatkan bahwa kejadian dapat dikelompokkan dan dibaca kembali tanpa membuka berkas log mentah. Fitur ini mendukung audit sederhana, identifikasi pola berulang, serta verifikasi apakah kejadian berasal dari proses pengujian atau sumber lain di jaringan. Keterhubungan antara data perangkat dan catatan aktivitas membantu administrator mengenali target yang menerima koneksi mencurigakan secara lebih cepat. Informasi waktu kejadian dapat digunakan untuk mencocokkan alert dengan

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

aktivitas operasional, sedangkan alamat IP sumber dan port tujuan memberikan konteks awal untuk menentukan apakah pola akses memerlukan pemeriksaan lanjutan. Penyajian log dalam antarmuka web juga mengurangi ketergantungan pada pembacaan keluaran mesin deteksi secara langsung dan membuat proses penelusuran kejadian lebih terstruktur. Meskipun demikian, log tersebut hanya berfungsi sebagai bukti awal. Penetapan suatu sumber sebagai ancaman tetap memerlukan korelasi dengan inventaris layanan, jadwal pengujian, dan informasi jaringan lain agar aktivitas administratif yang sah tidak keliru ditafsirkan sebagai serangan.

Gambar 5. Tampilan log aktivitas pemindaian

Blacklist pada Gambar 6 berfungsi sebagai daftar administrasi alamat IP yang telah dinilai mencurigakan. Pada versi penelitian, daftar tersebut belum langsung mengubah aturan firewall. Pemisahan ini sengaja dilakukan agar fungsi deteksi dan keputusan pemblokiran tidak tercampur. Administrator tetap dapat memverifikasi kejadian sebelum menerapkan kebijakan jaringan. Pengembangan berikutnya dapat menghubungkan blacklist dengan iptables, MikroTik, atau perangkat firewall lain, sebagaimana konsep proteksi dan pembatasan akses pada penelitian terdahulu (Aryantho et al., 2022; Razzanda & Koprawi, 2024).

Gambar 6. Halaman pengelolaan blacklist

3.2 Hasil Pengujian Fungsional

Pengujian fungsional menunjukkan bahwa fitur utama bekerja sesuai kebutuhan. Login dengan kredensial benar mengarahkan pengguna ke dashboard, sedangkan password salah menghasilkan pesan kesalahan. Penambahan perangkat dengan data lengkap tersimpan dalam basis data, sementara input alamat IP kosong ditolak oleh validasi. Dashboard, log, dan blacklist dapat diakses setelah autentikasi. Ringkasan hasil pengujian disajikan pada Tabel 2.

Tabel 2. Hasil pengujian fungsional Aplikasi

No	Skenario	Hasil yang diharapkan	Hasil	Status
1	Login data benar	Dashboard admin ditampilkan	Sesuai	Valid
2	Login password salah	Sistem menolak dan menampilkan pesan	Sesuai	Valid
3	Tambah perangkat baru	Data perangkat tersimpan	Sesuai	Valid
4	Alamat IP kosong	Validasi menolak input	Sesuai	Valid
5	Buka halaman log	Riwayat deteksi ditampilkan	Sesuai	Valid
6	Tambah IP ke blacklist	Data blacklist tersimpan	Sesuai	Valid

Hasil tersebut menunjukkan keterhubungan antarmuka, logika aplikasi, dan basis data dapat berjalan. Validasi input penting karena kesalahan alamat target akan menghasilkan pemantauan yang tidak relevan. Autentikasi juga mencegah halaman monitoring diakses secara terbuka. Meskipun pengujian black-box tidak membuktikan keamanan kode secara menyeluruh, hasil valid menegaskan bahwa alur operasional dasar telah memenuhi kebutuhan yang dirumuskan pada tahap analisis.

3.3 Hasil Pengujian Deteksi Port Scanning

Pengujian deteksi menggunakan Nmap dilakukan dari perangkat attacker menuju target yang telah diregistrasikan. TCP SYN Scan dipilih karena teknik ini mengirim paket SYN dan menilai respons tanpa selalu menyelesaikan *three-way handshake*. TCP Connect Scan menyelesaikan koneksi melalui fungsi sistem operasi sehingga menghasilkan pola

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

koneksi yang lebih lengkap. Selain itu, pemindaian beberapa port digunakan untuk menguji kemampuan pengelompokan aktivitas dari satu IP sumber. Hasil pada Tabel 3 menunjukkan ketiga skenario utama berhasil terdeteksi dan peringatan ditampilkan pada dashboard.

Tabel 3. Pengujian skenario port scanning

No	Skenario	Alat/metode	Keluaran yang diharapkan	Hasil	Status
1	TCP SYN Scan	Nmap SYN scan	Aktivitas pemindaian TCP terdeteksi	Terdeteksi	Valid
2	Multiple Port Scan	Nmap pada beberapa port	Akses ke banyak port dikenali	Terdeteksi	Valid
3	Real-time Alert	Scanning dan monitoring dashboard	Peringatan muncul langsung	Muncul	Valid

Keberhasilan deteksi menunjukkan bahwa penghitungan port tujuan unik dapat mengenali pola pemindaian yang jelas. Sistem tidak bergantung pada tanda tangan Nmap tertentu, melainkan pada perilaku akses dari IP sumber dalam interval pengamatan. Oleh karena itu, alat pemindai yang berbeda tetap berpotensi terdeteksi selama menghasilkan pola yang melewati ambang. Akan tetapi, pemindaian yang sangat lambat, terdistribusi, atau menyamarkan sumber dapat berada di bawah ambang dan memerlukan teknik analisis tambahan. Batasan tersebut serupa dengan tantangan IDS anomali yang harus menyeimbangkan sensitivitas dan false positive (Arshad et al., 2020; Aldhaheri et al., 2024).

3.4 Akurasi pada Skenario Pengujian Terbatas

Pengujian berulang dilakukan terhadap dua jenis aktivitas scanning dan dua aktivitas normal. Masing-masing TCP SYN Scan dan TCP Connect Scan dilakukan lima kali. Normal browsing dan login SSH masing-masing dilakukan tiga kali. Semua percobaan scanning menghasilkan peringatan, sedangkan aktivitas normal tidak menghasilkan status port scanning. Tabel 4 menyajikan hasil tersebut.

Tabel 4. Hasil deteksi aktivitas scanning dan normal

No	Jenis aktivitas	Total uji	Terdeteksi sebagai scanning	Hasil
1	TCP SYN Scan	5	5	100% terdeteksi
2	TCP Connect Scan	5	5	100% terdeteksi
3	Normal browsing	3	0	Tidak terindikasi
4	SSH login normal	3	0	Tidak terindikasi

Pada 10 percobaan scanning, seluruhnya menghasilkan alert sehingga detection rate teramati adalah 10/10 atau 100%. Pada enam percobaan normal, tidak terdapat alert sehingga FPR teramati adalah 0/6 atau 0%. Nilai tersebut hanya merupakan proporsi empiris pada sampel uji, bukan estimasi kinerja populasi. Ukuran sampel yang kecil dan aktivitas normal yang homogen belum cukup untuk menilai precision, recall, F1-score, atau ketahanan terhadap perubahan trafik. Evaluasi lanjutan perlu mencakup trafik campuran berdurasi panjang, lebih banyak perangkat, slow scan, pemindaian terdistribusi, UDP dan teknik evasi, serta dataset IoT berlabel (Javed et al., 2024; Khan et al., 2023).

3.5 Waktu Respons dan Pembaruan Real-Time

Latensi notifikasi pada lima percobaan berada pada rentang 1,1–1,4 detik dengan rata-rata 1,24 detik, sedangkan rincian setiap percobaan disajikan pada Tabel 5. Nilai ini menggambarkan waktu dari identifikasi mesin deteksi hingga alert tersedia pada dashboard dalam konfigurasi uji. Karena penelitian tidak menguji sistem pembanding atau batas layanan yang disyaratkan, hasil tersebut tidak dapat digunakan untuk menyatakan bahwa sistem lebih cepat daripada pendekatan lain. Interpretasi yang dapat dipertahankan adalah bahwa pembaruan berlangsung dalam orde detik dan dapat digunakan untuk pemantauan manusia pada skenario pengujian.

Tabel 5. Waktu respons notifikasi sistem

Percobaan	Waktu deteksi
1	1,2 detik
2	1,4 detik
3	1,1 detik
4	1,3 detik
5	1,2 detik
Rata-rata	1,24 detik

Variasi antarpercobaan dapat berasal dari pemrosesan paket, transaksi basis data, dan siklus pembaruan antarmuka, tetapi kontribusi setiap komponen belum diukur secara terpisah. Oleh sebab itu, angka 1,24 detik disebut latensi notifikasi aplikasi, bukan waktu pencegahan atau respons jaringan. Pengujian produksi perlu mencatat timestamp pada tahap capture, analisis, penyimpanan, pengiriman, dan render, kemudian melaporkan distribusi latensi, bukan hanya rata-rata, di bawah beberapa tingkat beban.

TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

3.6 Pembahasan

Perbandingan karakteristik sistem dengan tiga penelitian terdahulu disajikan pada Tabel 6. Dimensi yang dibandingkan meliputi mesin atau pendekatan deteksi, media visualisasi atau notifikasi, dan fokus utama. Perbandingan ini digunakan untuk menunjukkan posisi serta batas kontribusi penelitian, bukan untuk menyatakan keunggulan kinerja karena seluruh sistem tidak diuji pada trafik, konfigurasi, dan perangkat keras yang sama.

Tabel 6. Perbandingan karakteristik penelitian

Penelitian	Mesin/pendekatan	Visualisasi/notifikasi	Fokus utama
Razzanda & Koprawi (2024)	Snort dan iptables	Telegram	Deteksi dan pencegahan TCP scan/ICMP flood
Sabila et al. (2025)	Snort rules	Alert Snort	Beragam teknik scan pada jaringan virtual
Robbani et al. (2025)	Snort dan ELK	Kibana	Deteksi dan visualisasi multi-serangan
Penelitian ini	Python threshold-based	Dashboard Laravel, log, blacklist	TCP port scanning pada perangkat IoT/LAN

Berdasarkan Tabel 6, Razzanda dan Koprawi (2024) memiliki cakupan respons paling lengkap karena menggabungkan Snort, iptables, dan Telegram untuk mendeteksi sekaligus mencegah TCP scan dan ICMP flood. Sabila et al. (2025) juga memanfaatkan Snort, tetapi berfokus pada pengujian beragam teknik pemindaian dalam jaringan virtual. Robbani et al. (2025) menggabungkan Snort dan ELK Stack sehingga menawarkan visualisasi multi-serangan yang lebih kaya melalui Kibana. Berbeda dari ketiga penelitian tersebut, sistem yang dikembangkan menggunakan aturan ambang berbasis Python dan menyatukan dashboard Laravel, log, registrasi perangkat IoT, serta blacklist dalam satu aplikasi. Pendekatan ini mengurangi jumlah komponen dan memudahkan penerapan awal pada LAN, tetapi cakupan deteksinya lebih sempit, belum menyediakan notifikasi eksternal, dan belum melakukan pemblokiran otomatis. Dengan demikian, kontribusi penelitian ini terletak pada integrasi pemantauan perangkat IoT yang sederhana, bukan pada superioritas akurasi, kelengkapan serangan, atau kemampuan pencegahan dibandingkan sistem terdahulu. Secara konseptual, sistem menggunakan aturan perilaku statis: tidak mencari signature payload dan tidak mempelajari baseline secara otomatis. Transparansi aturan memudahkan audit, tetapi θ dan Δt menciptakan trade-off langsung antara sensitivitas dan alarm palsu. Selain itu, agregasi hanya berdasarkan IP sumber membuat sistem rentan terhadap pemindaian lambat dan terdistribusi. Pengembangan yang lebih kuat memerlukan beberapa skala jendela waktu, korelasi banyak sumber terhadap satu target, profil normal per perangkat, dan evaluasi perubahan konsep trafik. Pendekatan berbasis aliran dapat menjadi rujukan untuk menambah fitur tanpa memindahkan beban komputasi ke endpoint IoT (Mutambik, 2024).

Implikasi praktis hasil penelitian adalah tersedianya satu alur untuk mengaitkan target IoT dengan kejadian yang terstruktur. Namun, dashboard tidak boleh diperlakukan sebagai bukti final serangan. Alert tetap memerlukan verifikasi terhadap inventaris layanan, aktivitas pemeliharaan, dan konteks sumber. Pada penerapan nyata, aplikasi juga membutuhkan HTTPS, pembatasan akses berbasis peran, proteksi sesi dan API, retensi log, sinkronisasi waktu, audit trail, pemantauan kesehatan sensor, serta pembaruan dependensi.

Keterbatasan penelitian mencakup cakupan TCP, LAN kecil, sampel uji terbatas, tidak adanya stress test dan benchmark, parameter statis, serta blacklist yang belum terhubung ke firewall. Tantangan pengembangan bukan sekadar menambah fitur, melainkan mencegah pemblokiran salah dan menjaga integritas data. Integrasi firewall memerlukan masa berlaku blokir, allowlist, mekanisme rollback, dan persetujuan administrator. Notifikasi eksternal memerlukan deduplikasi dan pengendalian laju agar alert storm tidak mengganggu operasi. Ambang adaptif harus diuji terhadap perubahan pola trafik agar tidak mengalami concept drift. Prioritas berikutnya adalah memperluas desain evaluasi sebelum mengotomatisasi respons.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa mesin deteksi Python berbasis jumlah port unik dapat diintegrasikan dengan Laravel dan MySQL untuk menyediakan registrasi target, log, dashboard, dan blacklist pada jaringan lokal. Dalam 10 percobaan scanning, seluruh aktivitas menghasilkan alert; pada enam aktivitas normal tidak terdapat alert; dan rata-rata latensi notifikasi lima percobaan adalah 1,24 detik. Temuan tersebut membuktikan kelayakan fungsional pada konfigurasi uji, tetapi belum membuktikan akurasi umum atau keunggulan terhadap IDS eksisting. Implikasi utamanya adalah bahwa arsitektur terpisah dapat menambah visibilitas tanpa membebani endpoint IoT, dengan konsekuensi ketergantungan pada sensor jaringan dan parameter statis. Sistem belum mampu menangani slow scan, pemindaian terdistribusi, UDP, teknik evasi, atau beban tinggi secara tervalidasi. Sebelum digunakan sebagai mekanisme respons otomatis, penelitian lanjutan perlu menetapkan dan menguji θ serta Δt secara sistematis, memakai trafik campuran dan dataset berlabel, mengukur FPR pada observasi panjang, membandingkan kinerja pada lingkungan yang sama, dan merancang kontrol aman untuk integrasi firewall serta notifikasi eksternal.


TIN: Terapan Informatika Nusantara
Vol 7, No 2, July 2026, page 877-884
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v7i2.10521

REFERENCES

Aldhaheri, A., Alwahedi, F., Ferrag, M. A., & Battah, A. (2024). Deep learning for cyber threat detection in IoT networks: A review. *Internet of Things and Cyber-Physical Systems, 4*, 110–128. https://doi.org/10.1016/j.iotcps.2023.09.003

Ardiyansyah, F., Setiawan, K., & Sutisna, N. (2024). Implementation of IDS on computer networks using Snort based on Telegram chatbot. *MALCOM: Indonesian Journal of Machine Learning and Computer Science, 4*(4), 1614–1623. https://doi.org/10.57152/malcom.v4i4.1561

Arshad, J., Azad, M. A., Amad, R., Salah, K., Alazab, M., & Iqbal, R. (2020). A review of performance, energy and privacy of intrusion detection systems for IoT. *Electronics, 9*(4), 629. https://doi.org/10.3390/electronics9040629

Aryantho, A., Kusumaningsih, R. Y. R., & Triyono, J. (2022). Analisa dan penerapan keamanan jaringan Mikrotik menggunakan metode deteksi port scanning berbasis firewall. *Jurnal Jarkom, 10*(1).

Febrianti, F., Wibowo, S. A., & Vendyansyah, N. (2021). Implementasi IoT (Internet of Things) monitoring kualitas air dan sistem administrasi pada pengelola air bersih skala kecil. *JATI (Jurnal Mahasiswa Teknik Informatika), 5*(1). https://doi.org/10.36040/jati.v5i1.3249

Javed, A., Ehtsham, A., Jawad, M., Awais, M. N., Qureshi, A.-u.-H., & Larijani, H. (2024). Implementation of lightweight machine learning-based intrusion detection system on IoT devices of smart homes. *Future Internet, 16*(6), 200. https://doi.org/10.3390/fi16060200

Kakihary, N. L. (2021). Pieces framework for analysis of user saticfaction Internet of Things-based devices. *Journal of Information Systems and Informatics, 3*(2), 243–252. https://doi.org/10.33557/journalisi.v3i2.119

Khan, N. W., Alshehri, M. S., Khan, M. A., Almakdi, S., Moradpoor, N., Alazeb, A., Ullah, S., Naz, N., & Ahmad, J. (2023). A hybrid deep learning-based intrusion detection system for IoT networks. *Mathematical Biosciences and Engineering, 20*(8), 13491–13520. https://doi.org/10.3934/mbe.2023602

Muhyidin, Y., Totohendarto, M. H., Undamayanti, E., & Choerunnisa, S. (2022). Perbandingan tingkat keamanan website menggunakan Nmap dan Nikto dengan metode ethical hacking. *Jurnal Teknologika, 12*(1), 80–89. https://doi.org/10.51132/teknologika.v12i1.143

Munada, M. S., & Kurniawan, R. (2024). Pengembangan aplikasi sales order menggunakan bahasa pemrograman PHP pada PT Sinar Sosro. *JATI (Jurnal Mahasiswa Teknik Informatika), 8*(2), 2236–2242. https://doi.org/10.36040/jati.v8i2.9227

Mutambik, I. (2024). An efficient flow-based anomaly detection system for enhanced security in IoT networks. *Sensors, 24*(22), 7408. https://doi.org/10.3390/s24227408

Pebrianto, J. (2024). Perlindungan port dari pemindaian NMAP dan meningkatkan efisiensi IPv4 dengan MikroTik dan Nginx. *Media Jurnal Informatika, 16*(1), 54. https://doi.org/10.35194/mji.v16i1.4012

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Puspitasari, W. A., Turmudizy, A., & Edora. (2022). Penerapan metode RAD untuk pengembangan sistem inventori obat berbasis web menggunakan framework Laravel. *Pelita Teknologi, 17*(1), 12–28. https://doi.org/10.37366/pelitatekno.v17i1.1018

Razzanda, I. M., & Koprawi, M. (2024). Implementasi IDS dan IPS terhadap serangan TCP port scanning dan ICMP flooding. *The Indonesian Journal of Computer Science, 13*(4). https://doi.org/10.33022/ijcs.v13i4.4212

Robbani, F. D., Haryatmi, E., Riyadi, T. A., Supono, R. A., Kurniawan, A. B., & Rosdiana. (2025). Implementation of intrusion detection system using Snort and log visualization using ELK Stack. *International Journal of Engineering, Science and Information Technology, 5*(3), 220–228. https://doi.org/10.52088/ijesty.v5i3.901

Sabila, M. I., Tahir, M., Mardania, S. D., & Arifin, R. I. (2025). Implementasi Snort sebagai IDS dalam mendeteksi serangan port scanning NMAP pada simulasi jaringan virtual. *JATI (Jurnal Mahasiswa Teknik Informatika), 9*(4). https://doi.org/10.36040/jati.v9i4.14340

Sitanggang, R., Dachi, T. U., & Manurung, I. H. G. (2022). Rancang bangun sistem penjualan tanaman hias berbasis web menggunakan PHP dan MySQL. *Jurnal Teknologi Kesehatan dan Ilmu Sosial, 4*(1), 84–90.

Suryawan, A. F. D., Putra, F. G. D., Lovely, V. A., & Setiawan, A. (2024). Keamanan IoT dan sistem terdistribusi. *Journal of Internet and Software Engineering, 1*(3), 10. https://doi.org/10.47134/pjise.v1i3.2619

Zain, A. R., Oktivasari, P., Soelaiman, N. F., & Umam, F. W. (2023). Implementasi intrusion detection system (IDS) Suricata dan management log ELK Stack untuk pendeteksian kegiatan mining. *Jurnal Poli-Teknologi, 22*(1), 23–29. https://doi.org/10.32722/pt.v22i1.4974