ANALISIS MITIGASI SERANGAN DDOS MENGGUNAKAN ENTROPY DAN PUZZLE BERBASIS PROOF OF WORK PADA SERVER UBUNTU

SKRIPSI

BERLIANNA UPIK NURNIATI 2107421022

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
2025

ANALISIS MITIGASI SERANGAN DDOS MENGGUNAKAN ENTROPY DAN PUZZLE BERBASIS PROOF OF WORK PADA SERVER UBUNTU

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

BERLIANNA UPIK NURNIATI

2107421022

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2025

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama	: Berliana Upik Nurniati
NIM	: 2107421022
Jurusan / Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan
Judul Skripsi	: Analisis Mitigasi Serangan DDoS Menggunakan *Entropy* dan Puzzle Berbasis *Proof of Work* pada Server Ubuntu

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 23 Juli 2025

Yang membuat pernyataan,

(Berlianna Upik Nurniati)

NIM. 2107421022

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Berlianna Upik Nurniati

NIM : 2107421022

Program Studi : Teknik Multimedia dan Jaringan

Judul Skripsi : Analisis Mitigasi Serangan DDoS Menggunakan *Entropy* dan Puzzle Berbasis *Proof Of Work* pada Server Ubuntu

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, tanggal 9, bulan Juli, tahun 2025 dan dinyatakan **LULUS**.

Disahkan oleh

Pembimbing I : Maria Agustin, S.Kom., M,Kom ( )

Penguji I : Defiana Arnaldy, S.Tp., M,Si ( )

Penguji II : Dr. Indra Hermawan., M.Kom ( )

Penguji III : Chandra Wirawan, M.Kom ( )

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua

(Dr. Anita Hidayati, S.Kom., M.Kom.)

NIP. 197908032003122003

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Analisis Mitigasi Serangan DDoS Menggunakan *Entropy* dan Puzzle Berbasis *Proof of Work* pada Server Ubuntu". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Diploma IV pada Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

Dalam proses penyusunan skripsi ini, penulis menyadari bahwa tidak akan dapat menyelesaikannya tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Orang tua dan seluruh keluarga tercinta, atas kasih sayang, doa, dan dukuangan moril agar proses penyusunan skripsi penulis berjalan lancar, serta memberikan dukungan dalam menyelesaikan skripsi
2. Ibu Maria Agustin, S.Kom., M.Kom. sebagai dosen pembimbing yang telah meluangkan waktu, membantu, mendukung dan memberikan masukan serta saran kepada penulis selama pengerjaan skripsi hingga selesai
3. Bapak dan Ibu Dosen Jurusan Teknik Informatika dan Komputer, khususnya program studi Teknik Multimedia dan Jaringan, atas ilmu, motivasi, dan inspirasi yang telah diberikan.
4. Teman - teman penulis, Chiara Adristi, Nurul Aulia Dewi, Yazmin Nur'Aini, Layla Rosyidah, Niken Maharani, Ainur Rafika, Qathrah Nadiyah Salsabila, Muhammad Daffa Rasyid, Shoffan Darul, Sandika Arga, Robby Akbar, dan teman teman lain yang tidak dapat disebutkan satu persatu, yang selalu mendukung, memberikan semangat, dan kebersamaan dalam suka maupun duka selama masa perkuliahan dan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang berkepentingan.

Depok, 7 Juli 2025

Berlianna Upik Nurniati

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini:

Nama : Berlianna Upik Nurniati

NIM : 2107421022

Jurusan / Program Studi : Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

Analisis Mitigasi Serangan DDoS Menggunakan *Entropy* dan Puzzle Berbasis *Proof Of Work* pada Server Ubuntu

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, ……...

Yang menyatakan,

(Berlianna Upik Nurniati)

NIM. 2107421022

ANALISIS MITIGASI SERANGAN DDOS MENGGUNAKAN ENTROPY DAN PUZZLE BERBASIS PROOF OF WORK PADA SERVER UBUNTU

ABSTRAK

Server yang terhubung ke jaringan publik rentan terhadap berbagai jenis serangan siber, salah satunya adalah serangan *Distributed Denial of Service* (DDoS) yang bertujuan melumpuhkan layanan dengan membanjiri server menggunakan permintaan palsu secara terus-menerus. Penelitian ini mengusulkan metode mitigasi serangan DDoS menggunakan pendekatan dua tahap, yaitu deteksi berbasis *entropy* dan verifikasi berbasis *puzzle Proof of Work* (PoW). Deteksi dilakukan dengan menghitung nilai *entropy* dari distribusi alamat IP yang mengakses *endpoint* login server Ubuntu secara *real time*. Jika nilai *entropy* turun di bawah ambang batas tertentu atau pola trafik menunjukkan anomali, sistem akan mengaktifkan mitigasi. Selanjutnya, hanya klien yang dapat menyelesaikan tantangan puzzle kriptografi (berbasis SHA-256) yang diizinkan melanjutkan proses login. Mekanisme ini mencegah penyerang mengakses sistem tanpa autentikasi dan meminimalkan false positive pada pengguna sah. Pengujian dilakukan dalam lingkungan lokal dengan berbagai skenario jumlah *attacker* dan volume serangan menggunakan tools HTTP Flood. Hasil evaluasi menunjukkan bahwa kombinasi metode *entropy* dan puzzle PoW mampu mengklasifikasikan trafik dengan akurasi mencapai 100%, serta menjaga performa server tetap stabil saat terjadi serangan.

Kata Kunci: Ubuntu Server, DDoS, Entropy, Proof of Work (POW)

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital telah mendorong peningkatan ketergantungan masyarakat terhadap layanan daring dan infrastruktur jaringan. Dalam kondisi ini, keamanan siber menjadi isu yang sangat krusial, karena sistem informasi terbuka terhadap berbagai jenis ancaman, salah satunya adalah serangan Distributed Denial of Service (DDoS). Serangan DDoS merupakan upaya untuk mengganggu ketersediaan layanan dengan membanjiri server target menggunakan lalu lintas jaringan secara berlebihan, sehingga menyebabkan layanan menjadi lambat atau bahkan tidak dapat diakses sama sekali (Wu et al., 2022).

Ancaman DDoS menjadi semakin berbahaya karena teknik yang digunakan oleh penyerang semakin canggih dan bersifat terdistribusi. Serangan ini tidak hanya berdampak pada gangguan layanan, tetapi juga dapat menyebabkan kerugian finansial, turunnya kepercayaan pengguna, serta terhambatnya operasional bisnis. Menurut laporan ID-SIRTII/CC (Indonesia Security Incident Response Team on Internet Infrastructure), lebih dari satu juta insiden serangan siber terjadi di Indonesia dalam kurun waktu satu tahun, dan tren ini terus meningkat akibat kelemahan sistem dan aplikasi yang dieksploitasi secara masif (Chotimah, 2019).

Salah satu platform yang banyak digunakan dalam pengembangan sistem layanan berbasis web adalah Ubuntu Server, karena sifatnya yang open source, ringan, serta kompatibel dengan berbagai tools keamanan dan web server seperti Apache2 (Gorave, 2019). Namun demikian, server berbasis open-source seperti Ubuntu juga menjadi target potensial serangan DDoS, terutama karena banyaknya layanan publik yang di-host di dalamnya tanpa perlindungan adaptif yang memadai (Aydın et al., 2022; Xiong et al., 2020).

Pendekatan mitigasi tradisional seperti penggunaan firewall, intrusion detection system (IDS), dan blacklist IP sering kali tidak cukup efektif dalam menghadapi serangan DDoS berskala besar, terutama ketika serangan bersifat terdistribusi dan menyamar sebagai lalu lintas normal. Misalnya, dalam penelitian oleh (Patil & G,

2023), ditemukan bahwa serangan DDoS terhadap infrastruktur cloud berbasis OpenStack mampu melumpuhkan firewall dan IDS karena perangkat tersebut tidak dirancang untuk memfilter lalu lintas dalam jumlah besar yang tampak seperti trafik normal. Dalam skenario simulasi menggunakan tools seperti LOIC dan Slowloris, firewall gagal membedakan antara trafik sah dan trafik berbahaya, sementara IDS mengalami beban berlebih hingga menurunkan performa sistem secara keseluruhan. Oleh karena itu, dibutuhkan pendekatan yang lebih adaptif dan dinamis dalam mendeteksi serta merespons anomali lalu lintas yang mengindikasikan adanya serangan (Aydın et al., 2022).

Untuk menyelesaikan masalah tersebut, diperlukan pendekatan mitigasi yang lebih adaptif dan cerdas, yang mampu melakukan deteksi dini dan respons otomatis terhadap anomali lalu lintas jaringan. Salah satu pendekatan yang dilakukan adalah analisis *entropy*, yaitu pengukuran tingkat ketidakpastian dalam distribusi trafik. Ketika trafik normal terjadi, distribusi alamat IP yang mengakses server cenderung merata, menghasilkan nilai *entropy* yang tinggi. Namun, ketika serangan DDoS terjadi, terutama pada jenis serangan berbasis botnet atau HTTP Flood, permintaan banyak dikirimkan dari IP yang sama atau jumlah IP yang sangat sedikit, menyebabkan penurunan drastis pada nilai *entropy*, menjadi indikator bahwa sedang terjadi trafik mencurigakan karena data *real time* dalam membaca pola trafik jaringan. (Hassan et al., 2024; Jawahar et al., 2024). Metode ini memiliki keunggulan karena tidak membutuhkan pelatihan data dan mampu berjalan secara *real time* dengan penggunaan sumber daya minimal (M, 2024).

Di sisi lain, deteksi trafik mencurigakan saja tidak cukup sistem juga memerlukan respon aktif dalam bentuk mekanisme mitigasi, untuk mencegah bot atau pelaku serangan melanjutkan permintaan ke server. Salah satu pendekatan yang terbukti efektif adalah penerapan puzzle berbasis *Proof of Work* (PoW). PoW memaksa klien menyelesaikan teka-teki komputasi, misalnya dengan mencari *nonce* yang menghasilkan hash SHA-256 berawalan nol sebanyak n digit. Puzzle ini ringan bagi pengguna manusia yang hanya melakukan satu permintaan login, tetapi menjadi sangat berat bagi bot yang melakukan ratusan permintaan per detik. Strategi ini menjadikan PoW sebagai semacam *rate limiter* kriptografis yang tidak

mengganggu pengguna sah, tetapi sangat memperlambat pelaku serangan (Chakraborty et al., 2022).

Penelitian oleh (Bazzanella & Gangemi, 2023) memperkenalkan sistem PoW adaptif yang mengatur tingkat kesulitan berdasarkan beban jaringan, dengan bukti bahwa pendekatan ini dapat secara signifikan mengurangi efektivitas botnet dan memaksa penyerang mengalokasikan sumber daya komputasi yang besar untuk setiap permintaan. Lebih lanjut, studi oleh (Bostanov, 2021) menekankan bahwa implementasi server-side dari SHA-256 PoW dapat divalidasi dalam waktu kurang dari 20 milidetik, sehingga tidak membebani performa server secara signifikan jika proses komputasi klien menghasilkan *nonce*, *hash*, dan *base* yang valid secara efisien, yakni dalam satu atau dua iterasi hash pada tingkat kesulitan rendah. Dengan solusi penggabungan metode *entropy* untuk deteksi dan PoW sebagai mitigasi, sistem akan mampu mengidentifikasi serta memperlambat atau bahkan menghentikan serangan DDoS secara efisien, tanpa mengganggu akses pengguna sah (Alviano, 2023; Chakraborty et al., 2022).

1.2 Rumusan Masalah

Berdasarkan hal hal yang sudah disampaikan di atas, maka rumusan masalah yang dijadikan fokus pada penelitian ini adalah:

1. Bagaimana metode *entropy* dapat digunakan untuk mendeteksi potensi serangan DDoS berdasarkan distribusi lalu lintas IP pada server Ubuntu?
2. Bagaimana merancang dan mengimplementasikan sistem mitigasi serangan DDoS dengan menggunakan puzzle berbasis *Proof of Work* (PoW) secara dinamis saat terdeteksi trafik mencurigakan?
3. Bagaimana performa kombinasi metode *entropy* dan puzzle PoW dalam membedakan pengguna sah dan penyerang selama simulasi serangan DDoS?
4. Bagaimana dampak penerapan sistem mitigasi *entropy* dan puzzle PoW terhadap performa dan stabilitas server Ubuntu saat menghadapi trafik tinggi atau serangan DDoS?

1.3 Batasan Masalah

Penelitian yang dilakukan untuk analisis mitigasi serangan ddos menggunakan *entropy* dan puzzle berbasis *Proof of Work* pada server ubuntu, memiliki sejumlah batasan masalah. Batasan-batasan tersebut adalah:

1. Jenis serangan DDoS yang dianalisis terbatas pada HTTP Flood (layer 7), dengan target utama endpoint login.php pada server web.
2. Deteksi dini serangan dilakukan menggunakan metode perhitungan *entropy* berbasis distribusi IP dan jumlah request.
3. Mekanisme mitigasi dirancang menggunakan puzzle berbasis *Proof of Work* (PoW) dengan algoritma SHA-256, yang dikerjakan oleh klien menggunakan JavaScript (client-side) dan diverifikasi oleh server menggunakan PHP.
4. Lingkungan pengujian dilakukan dalam jaringan lokal (private LAN) menggunakan platform VirtualBox, dengan skenario simulasi serangan dilakukan dari VM attacker ke VM target (Ubuntu Server).
5. Perangkat lunak (tools) yang digunakan untuk simulasi serangan adalah HTTP *Flood tools* GoldenEye.
6. Sistem yang diuji dan diimplementasikan berbasis Ubuntu Server dan tidak mencakup sistem operasi lain seperti Windows Server atau distribusi Linux lainnya.
7. Analisis efektivitas sistem dibatasi pada parameter-parameter seperti nilai *entropy*, status mitigasi, validitas puzzle, dan performa sistem (login success, CPU usage, dan logging), tanpa membandingkan dengan pendekatan machine learning atau IDS konvensional.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun Tujuan penelitian ini adalah sebagai berikut:

1. Untuk merancang dan mengimplementasikan metode deteksi serangan DDoS berbasis *entropy* dengan memanfaatkan distribusi IP dan jumlah permintaan dalam jangka waktu tertentu pada server Ubuntu.

2. Untuk mengembangkan sistem mitigasi dinamis terhadap serangan DDoS menggunakan puzzle berbasis *Proof of Work* (PoW) yang aktif hanya ketika trafik mencurigakan terdeteksi.
3. Untuk menganalisis dampak penerapan sistem mitigasi terhadap stabilitas dan performa server, khususnya dalam aspek validitas puzzle, tingkat keberhasilan login, serta penggunaan sumber daya selama kondisi serangan.

1.4.2 Manfaat

Adapun manfaat penelitian ini adalah sebagai berikut

1. Memberikan kontribusi terhadap pengembangan studi di bidang keamanan jaringan dan sistem informasi, khususnya dalam pendekatan mitigasi serangan DDoS berbasis statistik dan kriptografi.
2. Menawarkan solusi ringan dan efisien untuk mengamankan endpoint web login dari serangan HTTP Flood tanpa membebani server.
3. Memungkinkan administrator sistem untuk menerapkan mitigasi DDoS tanpa memerlukan perangkat keras tambahan atau lisensi berbayar, karena sistem ini dapat dibangun sepenuhnya dengan *software open source*.
4. Menjadi referensi dalam pengembangan penelitian lanjutan, baik pada sisi deteksi anomali, algoritma mitigasi dinamis, maupun integrasi dengan model machine learning.
5. Memberikan referensi implementatif bagi praktisi keamanan jaringan maupun pengembang sistem web dalam membangun mekanisme pertahanan terhadap DDoS berbasis *web layer (Layer 7)*.
6. Memberikan dasar bagi pengembang dan penyedia layanan web untuk membangun sistem anti-DDoS sederhana namun efektif, yang dapat diterapkan pada server produksi dengan biaya rendah.

1.5 Sistematika Penulisan

Berikut adalah sistematika penulisan yang digunakan dalam penyusunan proposal penelitian ini, yaitu sebagai berikut:

1. BAB I PENDAHULUAN

   Bab pertama pendahuluan, menguraikan tentang latar belakang dari penelitian, rumusan masalah yang didapat dari latar belakang, Batasan masalah pada penelitian ini, serta manfaat dan tujuan dalam penelitian ini.

2. BAB II TINJAUAN PUSTAKA

   Bab kedua menguraikan tentang landasan-landasan teori dan konsep-konsep terkait dengan sebuah permasalahan pada penelitian ini, serta beberapa penelitian yang relevan terkait dari penelitian-penelitian terdahulu untuk dikaji dalam penelitian ini.

3. BAB III METODE PENELITIAN

   Bab ketiga dalam penelitian ini akan menjabarkan tentang metode penelitian yang akan digunakan, baik berhubungan dengan perancangan penelitian, tahapan-tahapan yang akan ditempuh dalam penelitian, objek dari penelitian, model penelitian, begitu juga teknik pengumpulan dan analisis data, hingga jadwal pelaksanaan.

4. BAB IV HASIL DAN PEMBAHASAN

   Bab keempat pada penelitian ini berisikan analisis kebutuhan,perancangan sistem, implementasi sistem, pengujian, dan analisis hasil mitigasi serangan terhadap sistem yang telah dibuat.

5. BAB V PENUTUP

   Bab kelima berisikan penjelasan mengenai hasil akhir dari penelitian berupa kesimpulan dan saran untuk penelitian berikutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

Nama Penulis	Judul	Hasil	Kekurangan	Perbedaan
Ritesh Patil, Narayan D. G.(Patil & G, 2023)	Mitigation of DDoS Attacks using *Entropy* and *Proof of Work* Based Puzzle in OpenStack Cloud	Kombinasi entropi dan puzzle berbasis PoW berhasil mendeteksi dan mengurangi dampak DDoS pada cloud OpenStack secara efisien.	Kompleksitas tinggi dalam implementasi dan konsumsi resource besar di lingkungan cloud.	Pada penelitian ini menerapkan teknik ini di server Ubuntu lokal dengan target HTTP flood pada login.php, bukan di cloud OpenStack.
Mochamad Teguh Kurniawan (Kurniawan, 2023)	Development of *entropy* method for detection of Distributed Denial of Service (DDoS) Attacks on Software Defined Network (SDN) by	Penggunaan tiga fitur (IP sumber, IP tujuan, MAC) meningkatkan akurasi deteksi hingga 99,43% pada arsitektur SDN.	Fokus hanya pada deteksi, tidak menyertakan mitigasi aktif, dan hanya relevan untuk jaringan SDN.	Pada penelitian ini menyertakan modul mitigasi aktif menggunakan puzzle PoW dan mengimplementasikannya dalam server Ubuntu berbasis aplikasi web.

Nama Penulis	Judul	Hasil	Kekurangan	Perbedaan
	implementing Feature Selection			
Baihaqi Asa'ari Lubis, Darryl Yanuar Ar-Rafi, Indry Widiyani, Khaerunnisa Isnaeni Lestari4, Ahmad Turmudi Zy (Asa'ari Lubis et al., 2024)	Analysis and Mitigation Technique of DDoS on Server Networks Based on Modern Technology	Efektif untuk mitigasi DDoS skala kecil menggunakan IP filtering & Fail2ban.	Tidak cocok untuk serangan masif, tanpa pendeteksian *entropy* atau autentikasi puzzle.	Pada penelitian ini menambahkan deteksi berbasis *entropy* dan autentikasi puzzle berbasis beban CPU.
Afifah Rodhiyatun Nisa*, Anandito Daffa Wijayanto, Arya Prabudi Jaya Priana, Aep Setiawan (Afifah Rodhiyatun Nisa et al., 2024)	Analisis Log Server untuk Mendeteksi Serangan DDoS	Deteksi serangan DDoS dari log aktivitas user HTTP, efektif terhadap serangan bot.	Hanya deteksi, tanpa metode mitigasi aktif.	Pada penelitian ini tidak hanya deteksi, tetapi juga melakukan mitigasi otomatis dengan PoW puzzle.

Nama Penulis	Judul	Hasil	Kekurangan	Perbedaan
Zata Ismah Sumayyah *, Silva Dimas Surya Permana, Muhammad Tsabit, Aep Setiawan (Sumayyah et al., 2024)	Penerapan dan Mitigasi Teknik Slowloris dalam Serangan DDoS terhadap Website Ilegal dengan Kali Linux	. Mampu memblok IP penyerang pada layer aplikasi (Apache) setelah beberapa percobaan koneksi.	Tidak menggunakan metode *entropy* atau puzzle, hanya blocking berdasarkan koneksi terbuka.	Pada penelitian ini lebih dinamis dan adaptif dalam mengenali dan menangani berbagai jenis DDoS melalui *entropy* dan puzzle berbasis CPU client.

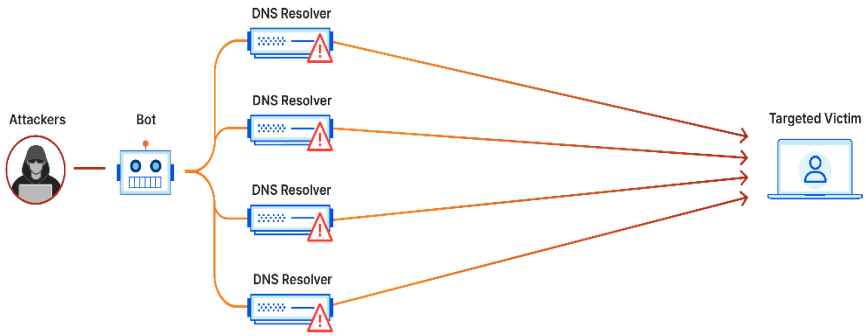
2.2 Distributed Denial of Service (DDoS)

Menurut (Cloudflare, n.d.) serangan *Distributed Denial of Services (DDoS)* merupakan salah satu bentuk serangan siber yang menyerang perangkat jaringan komputer, seperti, *controller, computer,* router, dan server. DDoS bekerja dengan memanfaatkan sejumlah besar perangkat atau bot yang membawa packet-packet dalam jumlah besar untuk melemahkan suatu jaringan. Akibatnya berdampak pada sumber daya sistem seperti CPU, memori, atau bandwidth menjadi cepat habis dan membuat layanan melambat hingga tidak dapat diakses oleh pengguna yang sah. Serangan yang terjadi mungkin tidak merusak data, namun sangat mengganggu operasional dan dapat menimbulkan kerugian finansial maupun reputasi. Jenis serangan DDoS dapat berupa UDP Flood Attack, ICMP Flood, Ping Flood, Ping of Death, SYN Flood, dan HTTP Flood. Menurut laman (Center Canadian for Cyber, 2024) DDoS dapat dibagi menjadi tiga jenis, yaitu:

a. Serangan Berbasis Volume

Serangan ini menggunakan volume lalu lintas data yang sangat besar, penyerang menggunakan bot untuk mengirim permintaan DNS ke beberapa DNS Resolver terbuka, namun dengan memalsukan alamat IP korban sebagai pengirimnya. DNS

Resolver kemudian mengirim respons yang sangat besar ke IP korban (Targeted Victim), untuk menghabiskan bandwidth dan menyebabkan kemacetan jaringan.

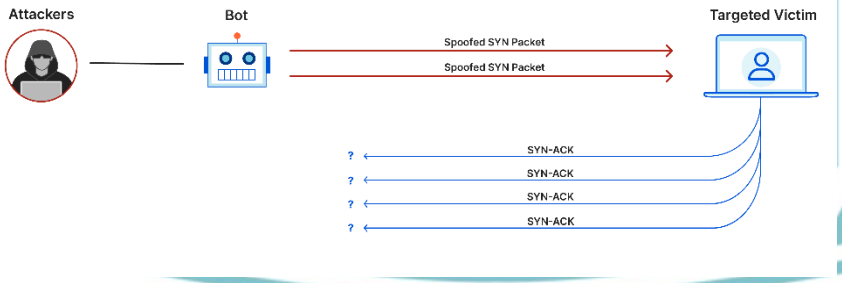


Gambar 2. 1 Ilustrasi Serangan Berbasis Volume

Sumber: What is a DDoS attack? https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

b. Serangan Protokol

Serangan ini mengeksploitasi kelemahan pada protokol jaringan yang sering memanfaatkan kelemahan pada *layer 3 dan layer 4* dari OSI model, seperti melalui SYN Flood, tujuannya untuk menghabiskan sumber daya koneksi sehingga resource server habis karena banyak koneksi *half open* dan membuat server tidak mampu merespons permintaan baru.

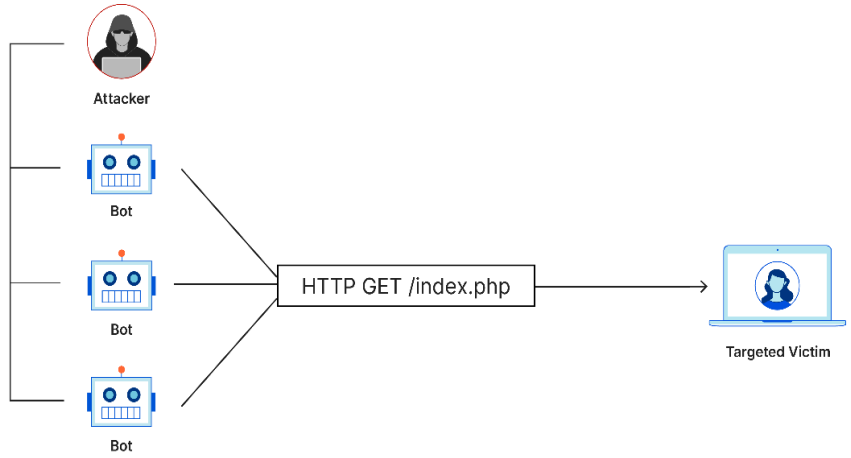


Gambar 2. 2 Ilustrasi Serangan Berbasis Protokol

Sumber: What is a DDoS attack? https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

c. Serangan pada Lapisan Aplikasi

Serangan ini menargetkan lapisan aplikasi *layer 7* dengan membanjiri server menggunakan permintaan HTTP yang tampak sah, namun bertujuan menguras sumber daya server hingga layanan menjadi tidak tersedia sehingga tidak mampu melayani permintaan dari pengguna yang sah.



Gambar 2. 3 Ilustrasi Serangan Berbasis Layer Aplikasi

Sumber: What is a DDoS attack? https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

2.3 Ubuntu Server

Ubuntu Server adalah sistem operasi berbasis Linux yang dirancang khusus untuk digunakan sebagai server. Sistem ini bersifat open-source dan mendukung berbagai kebutuhan seperti layanan web, database, cloud, hingga keamanan jaringan. Ubuntu Server dikenal karena stabil, ringan, dan memiliki dukungan pembaruan jangka panjang, sehingga banyak digunakan untuk keperluan riset dan pengembangan sistem.

Dalam penelitian ini, Ubuntu Server digunakan sebagai platform utama karena mampu menjalankan layanan secara efisien, mendukung instalasi software tambahan seperti web server (Apache), serta memudahkan dalam konfigurasi keamanan dan monitoring jaringan. Keunggulan lainnya adalah kompatibilitasnya dengan berbagai perangkat keras dan lingkungan virtual, seperti VirtualBox atau cloud server (Documentation, 2025)

2.4 Metode Deteksi Serangan DDoS

1. Entropy

Menurut Wikipedia yang ditulis oleh (Tao, 2017), *entropy* merupakan ukuran matematis untuk menilai tingkat ketidakpastian atau keacakan dalam distribusi suatu data. Dalam konteks keamanan jaringan, *entropy* digunakan untuk mendeteksi anomali trafik, seperti serangan DDoS, dengan cara menganalisis sebaran alamat IP yang mengakses server.

Ketika trafik jaringan berasal dari banyak IP yang berbeda secara merata (akses normal), nilai *entropy* cenderung tinggi. Sebaliknya, saat terjadi serangan DDoS yang berasal dari IP tertentu secara masif, *entropy* akan menurun drastis. Shannon *entropy* dapat dihitung dengan rumus berikut:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

Dalam persamaan ini, H(X) merepresentasikan entropi dari variabel acak X, dan P(x) merepresentasikan probabilitas dari hasil x.

Contoh perhitungan entropy:

Misalkan dalam 30 detik terakhir, sistem menerima 10 request dari 3 IP berikut:

IP Address	**Jumlah Request**
192.168.1.10	5
192.168.1.11	3
192.168.1.12	2
Total	**10**

$$p1 = \frac{5}{10} = 0.5$$

$$p2 = \frac{3}{10} = 0.3$$

$$p3 = \frac{2}{10} = 0.2$$

$$H(X) = -\left(0.5.\log_2 0.5 + 0.3.\log_2 0.3 + 0.2.\log_2 0.2\right)$$

$$H(X) = -(-0.5 - 0.5211 - 0.4644) = 1.4855$$

$$H(X) = 1.4855$$

Jadi, nilai entropy yang didapatkan dari 3 IP adalah 1.4855

Penentuan threshold *entropy* dilakukan secara dinamis menggunakan rata-rata dan deviasi standar nilai *entropy* saat kondisi normal. Serangan umumnya terdeteksi ketika nilai *entropy* turun di bawah ambang batas. Namun, dalam konteks sistem mitigasi jaringan, nilai *entropy* juga dapat tetap tinggi apabila trafik berasal dari banyak IP yang aktif secara bersamaan. Oleh karena itu, pada kondisi *entropy* tinggi yang disertai lonjakan trafik, deteksi anomali dapat dilakukan dengan mempertimbangkan perilaku IP dan laju permintaan (request rate) dari masing-masing sumber (Bashurov & Safonov, 2023).

Tabel berikut merangkum hasil statistik entropy berdasarkan kondisi sistem:

Tabel 2. 2 Data Nilai entropy

Kondisi	Jumlah Data	Mean Entropy	Std Dev	Min	Median	Maks
Normal	22.991	1,68	0,32	0,00	1,70	2,31
Saat Mitigasi	55.253	1,21	0,72	0,00	1,28	2,31

Berdasarkan perbandingan distribusi entropy antara kondisi normal dan saat mitigasi aktif, threshold entropy ditetapkan sebesar 1,3. Nilai ini berada tepat di antara median entropy pada kondisi mitigasi dan kuartil bawah dari entropy pada kondisi normal. Dengan threshold ini, sistem akan mengaktifkan mekanisme mitigasi seperti puzzle Proof-of-Work jika nilai entropy yang terdeteksi berada di bawah 1,3, menandakan pola distribusi IP yang mencurigakan (misalnya dominasi oleh satu atau dua IP saja).

Pada penelitian (Patil & G, 2023) sistem monitoring entropy mendeteksi penurunan tajam entropy saat serangan DDoS berlangsung, dengan nilai entropy turun di

bawah 1.0 pada saat sistem mendeteksi dominasi trafik dari sejumlah kecil IP. Oleh karena itu, entropy < 1.0 digunakan dalam sistem ini untuk mendeteksi kondisi "entropy drop drastis" yang mengindikasikan serangan satu arah dari bot (low-distribution entropy).

Nilai ini ditetapkan berdasarkan median entropy hasil pengamatan langsung saat sistem mendeteksi mitigasi aktif pada data eksperimen, yaitu sekitar 1,28. Selain itu, pendekatan ini diperkuat oleh hasil studi (Pebrianto & Suryani, 2025) dalam penelitiannya tentang adaptive DDoS detection model.

Dalam studi tersebut, threshold entropy ditentukan secara dinamis berdasarkan rerata dan deviasi standar dari distribusi entropy yang dikumpulkan dari trafik normal, dan disebutkan bahwa nilai entropy pada trafik normal umumnya berkisar antara 1,4–2,0, sementara nilai entropy di bawah 1,3 menunjukkan awal anomali (termasuk pola botnet tersembunyi). Maka dari itu, sistem ini menetapkan threshold 1.3 sebagai batas atas deteksi dini serangan yang masih memiliki distribusi IP yang agak menyebar, namun tetap mencurigakan secara statistik.

2. Machine Learning

Metode *machine learning based detection* merupakan pendekatan yang menggunakan proses pembelajaran pola trafik jaringan untuk mendeteksi serangan DDoS. Pada metode ini, data trafik dikumpulkan dan dijadikan sebagai dataset berlabel, di mana setiap data sudah diklasifikasikan sebagai trafik normal atau trafik serangan. Dataset ini kemudian digunakan untuk melatih model machine learning agar dapat mengenali pola karakteristik serangan. Beberapa algoritma *machine learning* yang umum digunakan untuk deteksi DDoS meliputi *Decision Tree* (DT), *Random Forest* (RF), *Support Vector Machine* (SVM), dan *Convolutional Neural Network* (CNN). Setelah proses pelatihan selesai, model yang terbentuk mampu mengklasifikasikan trafik baru berdasarkan pola yang telah dipelajari sebelumnya.

Kelebihan utama dari metode *machine learning* ini adalah kemampuannya dalam mendeteksi pola serangan baru yang belum pernah ada sebelumnya, karena sistem mampu belajar dari variasi trafik yang kompleks. Namun, metode ini memiliki kelemahan berupa kebutuhan proses training yang cukup intensif serta keharusan

memiliki dataset yang memadai dan representatif. Selain itu, sistem machine learning umumnya membutuhkan sumber daya pemrosesan yang lebih besar dibandingkan metode statistik seperti *entropy*.

2.5 Metode Mitigasi Serangan DDoS

1. Puzzle Kriptografi berbasis *Proof of Work* (PoW)

Proof of Work (PoW) adalah sebuah mekanisme kriptografi yang menuntut pihak yang melakukan permintaan (klien) untuk menyelesaikan sebuah perhitungan matematis tertentu sebelum permintaannya dilayani oleh server. Tujuan utama dari PoW adalah untuk mencegah penyalahgunaan sumber daya sistem, seperti spam, brute force, dan dalam konteks ini, serangan Distributed Denial of Service (DDoS).

Dalam sistem PoW, proses ini dikenal sebagai puzzle solving, di mana klien diminta untuk memecahkan teka-teki komputasi (puzzle) dengan kompleksitas dinamis. Puzzle ini biasanya berupa pencarian nilai nonce yang, ketika digabungkan dengan data tertentu dan dihitung dengan algoritma hash (SHA-256), menghasilkan nilai hash yang sesuai dengan pola tertentu (misalnya diawali dengan sejumlah angka nol).

2. CAPTCHA Challenge

CAPTCHA challenge merupakan mekanisme keamanan yang digunakan untuk membedakan pengguna manusia dan bot otomatis. Metode ini bekerja dengan memberikan tantangan otomatis yang harus diselesaikan pengguna sebelum mengakses layanan. Jenis tantangan dapat berupa pemilihan gambar, pengisian teks terdistorsi, atau pemecahan soal matematika sederhana. Dalam konteks mitigasi serangan DDoS, CAPTCHA challenge efektif digunakan pada layer aplikasi untuk membatasi akses bot otomatis yang berpotensi membanjiri server dengan permintaan berlebihan.

2.6 SHA-256

SHA-256 (Secure Hash Algorithm 256-bit) adalah algoritma hashing kriptografi yang menghasilkan output sepanjang 256 bit dari input data. Algoritma SHA-256 dipilih sebagai basis hash dalam puzzle *Proof of Work* karena memiliki tingkat

keamanan yang tinggi, belum memiliki collision yang diketahui secara praktis, serta banyak digunakan dalam sistem PoW kriptografi seperti Bitcoin. Selain itu, SHA-256 memiliki keseimbangan antara efisiensi dan ketahanan terhadap serangan brute force, menjadikannya ideal untuk digunakan dalam mitigasi serangan DDoS berbasis puzzle.

2.7 OpenSSH

SSH (Secure Shell) adalah protokol jaringan gratis dan terbuka yang digunakan untuk menyediakan koneksi aman antara dua sistem jaringan. OpenSSH adalah implementasi populer dari protokol SSH yang menyediakan solusi yang aman, cepat, dan mudah digunakan untuk jaringan jarak jauh. OpenSSH menawarkan fitur seperti enkripsi yang kuat, autentikasi yang aman, dan tugas jaringan lainnya seperti transfer file dan eksekusi perintah jarak jauh. Protokol ini sangat populer di kalangan administrator jaringan dan pengguna Linux (openssh, 2023)

2.8 GoldenEye

GoldenEye merupakan tools simulasi serangan DDoS berbasis layer aplikasi (Layer 7) yang dirancang untuk mengirimkan HTTP request secara masif ke server target. Tools ini digunakan dalam penelitian untuk mensimulasikan trafik berlebih, dengan tujuan menguji efektivitas sistem deteksi dan mitigasi DDoS, seperti *entropy*-based detection dan puzzle *Proof of Work*. GoldenEye bersifat open-source dan dapat diakses melalui repositori GitHub. Dalam penelitian ini, GoldenEye digunakan sebagai tool pengujian untuk mengukur ketahanan sistem mitigasi terhadap serangan dengan beban tinggi (Seidl, 2020).

BAB III
METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

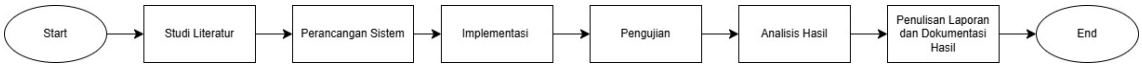
Rancangan pengerjaan penelitian yang akan dilakukan menggunakan metode penelitian kuantitatif dengan pendekatan eksperimen. Penelitian ini bertujuan untuk mengukur secara sistematis efektivitas dan kinerja mitigasi serangan DDoS menggunakan metode *entropy* dan puzzle berbasis *Proof of Work* (PoW) pada server Ubuntu. Pendekatan kuantitatif dipilih untuk menghasilkan data yang terukur berupa angka dan statistik yang diperoleh dari hasil pengujian mekanisme mitigasi terhadap berbagai skenario serangan DDoS.

Pengerjaan penelitian ini dimulai dengan merancang skenario menggunakan metode flooding tool yaitu Goldeneye untuk mencoba mengirim serangan DDoS kepada server. Sistem deteksi anomali trafik jaringan dikembangkan dengan pendekatan *entropy based detection,* dimana data log IP adddres yang diterima server dianalisis dalam jendela waktu (*sliding window*) selama dua menit untuk menentukan nilai *entropy* bagi user yang akes saat itu. Jika nilai *entropy* terdeteksi turun dari *threshold* yang telah ditentukan dan memenuhi kriteria lain yang sudah ditetapkan seperti, jumlah IP unik rendah dan tingkat permintaan yang tinggi, maka sistem secara otomatis mengaktifkan mekanisme mitigasi.

Sistem ini juga dirancang dengan pembuatan database bernama ddos_mitigation, yang digunakan untuk menyimpan data log trafik IP (log_request), nilai entropy (entropy_monitor), serta hasil verifikasi puzzle (puzzle_log). Penyimpanan data ini memungkinkan sistem untuk melakukan analisis historis terhadap perilaku trafik dan mengevaluasi efektivitas mitigasi.

Output yang diharapkan dari sistem ini meliputi: deteksi dini terhadap serangan DDoS berbasis HTTP flood, aktivasi mitigasi otomatis berupa tantangan puzzle, serta pencatatan log lengkap yang mencakup nilai entropy, status mitigasi, dan validasi puzzle untuk analisis evaluasi sistem.

3.2 Tahapan Penelitian



Gambar 3. 1 Tahapan Penelitian

1. Studi Literatur

   Tahapan awal penelitian ini adalah dengan melakukan analisis kebutuhan melalui pengumpulan referensi berupa jurnal dan buku terkait penelitian. Referensi tersebut dijadikan sebagai bahan acuan untuk penelitian yang akan dilakukan.

2. Perancangan Sistem

   Pada tahap ini dilakukan untuk merancang sistem mitigasi serangan DDoS berbasis perhitungan *entropy* dan *puzzle Proof of Work* (PoW), dengan mengacu pada studi literatur dan kebutuhan sistem yang telah dianalisis sebelumnya. Perancangan sistem mencakup seluruh komponen utama, mulai dari mekanisme pencatatan trafik, perhitungan entropy secara *realtime*, hingga pemicu aktivasi mitigasi berbasis puzzle.

   Sistem ini dirancang untuk menghasilkan tiga keluaran utama, yaitu deteksi dini terhadap serangan DDoS berbasis HTTP flood, aktivasi otomatis mitigasi melalui pemberian tantangan *puzzle* kepada pengguna mencurigakan, serta pencatatan log lengkap yang memuat nilai entropy, status mitigasi, dan hasil validasi puzzle. Output ini menjadi dasar evaluasi performa sistem dalam merespons serangan secara tepat dan terukur.

3. Implementasi

   Pada tahap ini dilakukan pengimplementasian sistem monitoring pada server production sesuai dengan rancangan.

4. Pengujian

   Tahap ini dilakukan uji coba terhadap sistem untuk memastikan bahwa semua fitur berfungsi sesuai dengan yang diharapkan Uji coba ini meliputi pengujian akurasi data standar atau harapan untuk menentukan apakah sistem monitoring bekerja dengan baik dan memberikan hasil yang akurat secara real-time.

5. Analisis Hasil

Setelah dilakukan uji coba, perlu dilakukan analisis data yang dikumpulkan oleh sistem monitoring. Ini melibatkan interpertasi data untuk mengidentifikasi masalah potensial.

6. Penulisan Laporan dan Dokumentasi Hasil

   Pada tahapan kali ini, sistem yang telah dilakukan pengujian akan dicatat dalam dokumentasi kerja dan penyusunan laporan dari hasil uji coba pada tahapan sebelumnya. Agar dapat dilakukan evaluasi dari setiap hasil data yang didapatkan, dan dapat dijadikan saran untuk penelitian selanjutnya.

3.3 Objek Penelitian

Objek penelitian ini adalah sebuah sistem login berbasis web yang yang diimplementasikan pada server Ubuntu dan dirancang juga menjadi tempat pengujian terhadap serangan *Distributed Denial of Services (DDoS)*. Terdapat mekanisme deteksi anomaly trafik jaringan yang berbasis perhitungan *entropy* terhadap setiap alamat IP pengirim request dalam jangka waktu yang telah ditentukan. Ketika sistem mendeteksi pola lalu lintas yang tidak wajar—seperti *entropy* yang rendah, jumlah IP unik yang sedikit, dan permintaan yang sangat sering—maka sistem akan mengaktifkan mekanisme mitigasi. Mitigasi ini dilakukan dengan mengirimkan puzzle berbasis *Proof of Work (PoW*), yang dihitung menggunakan algoritma SHA-256. Puzzle ini harus diselesaikan oleh pengguna sebelum dapat melanjutkan ke sistem. Setiap alamat IP, waktu permintaan, nilai *entropy*, status mitigasi, dan hasil validasi puzzle akan tersimpan di log database.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Analisis kebutuhan sistem yang dirancang bertujuan untuk melakukan mitigasi serangan DDoS terhadap endpoint login pada server Ubuntu menggunakan metode *entropy* dan puzzle berbasis *Proof -of-work* (PoW). Dalam proses pengerjaannya diperlukan beberapa perangkat lunak dan *tools* pendukung seperti yang ditulis dalam tabel berikut:

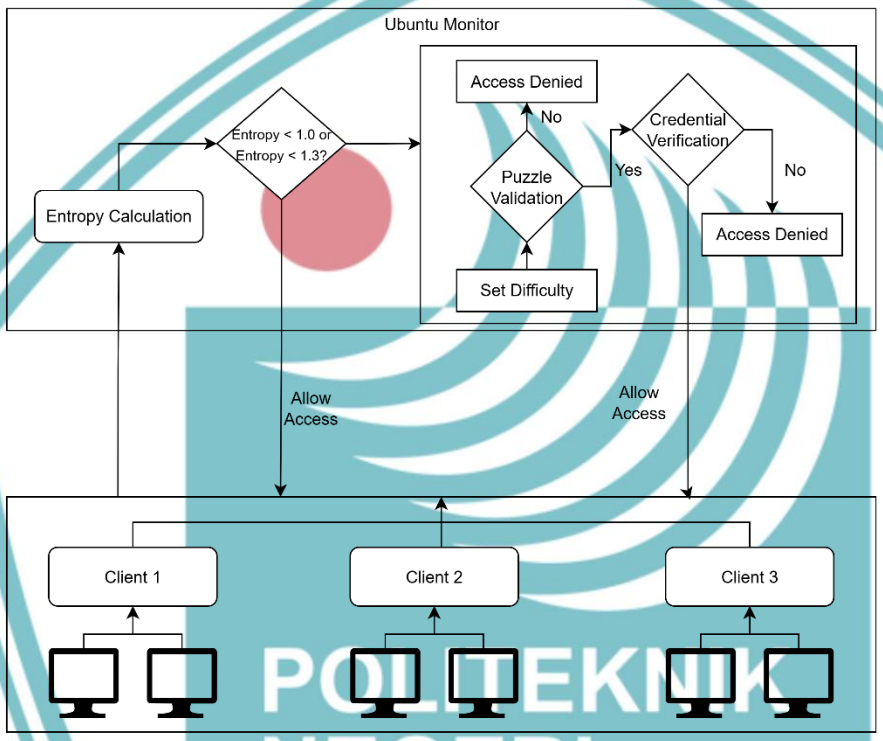
Tabel 4. 1 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

No	Nama	Spesifikasi
	Hardware	
1.	Laptop	CPU: Intel Core i7 @3.20Ghz RAM: 20GB Storage: 1TB
	Software	
1.	VirtualBox	Version 7.1.6
2.	Ubuntu Server	Description: Ubuntu 24.04.1 LTS Release: 24.04 Codename: noble
3.	Apache2	Server version: Apache/2.4.58 (Ubuntu) Server built: 2025-04-03T14:36:49
4.	MySQL	Ver 8.0.42-0ubuntu0.24.04.1 for Linux on x86_64
5.	PHP	Ver 8.3.6
6.	GoldenEye	GoldenEye v2.1 by Jan Seidl
7.	JavaScript	JavaScript versi runtime

8.	OpenSSH	OpenSSH_9.6p1 Ubuntu-3ubuntu13.12, OpenSSL 3.0.13 30 Jan 2024	

4.2 Perancangan Sistem

4.2.1 Arsitektur Sistem



Gambar 4. 1 Arsitektur Sistem

Gambar 4.1 menunjukkan arsitektur sistem mitigasi serangan *Distributed Denial of Service* (DDoS) berbasis *entropy* dan *puzzle Proof of Work* (PoW) yang diterapkan pada server Ubuntu. Sistem ini menerima permintaan akses dari beberapa pengguna atau client. Setiap client terdiri dari beberapa perangkat pengguna yang mengirimkan request ke server untuk mengakses layanan. Sebelum request tersebut diproses lebih lanjut, sistem terlebih dahulu menjalankan proses *Entropy* Calculation, yaitu perhitungan *entropy* untuk mendeteksi anomali trafik berdasarkan distribusi alamat IP yang masuk dalam periode waktu tertentu.

Jika hasil perhitungan *entropy* masih berada di atas ambang batas (threshold) yang telah ditentukan, maka trafik dianggap normal dan akses langsung diizinkan.

Namun, jika nilai *entropy* turun di bawah threshold, maka sistem akan mengaktifkan mekanisme mitigasi berbasis puzzle. Dalam tahap ini, server akan menetapkan tingkat kesulitan puzzle terlebih dahulu (Set Difficulty), kemudian mengirimkan tantangan puzzle kepada client. Hanya client yang dapat menyelesaikan puzzle dengan benar yang akan dilanjutkan ke tahap selanjutnya, yaitu Credential Verification. Jika verifikasi kredensial berhasil, sistem memberikan izin akses. Sebaliknya, jika puzzle tidak valid atau kredensial tidak sesuai, maka akses akan ditolak. Pendekatan ini dirancang untuk secara selektif menghambat trafik mencurigakan tanpa mengganggu pengguna sah, serta mempertahankan ketersediaan layanan selama berlangsungnya serangan DDoS.

4.2.2 Flowchart Sistem

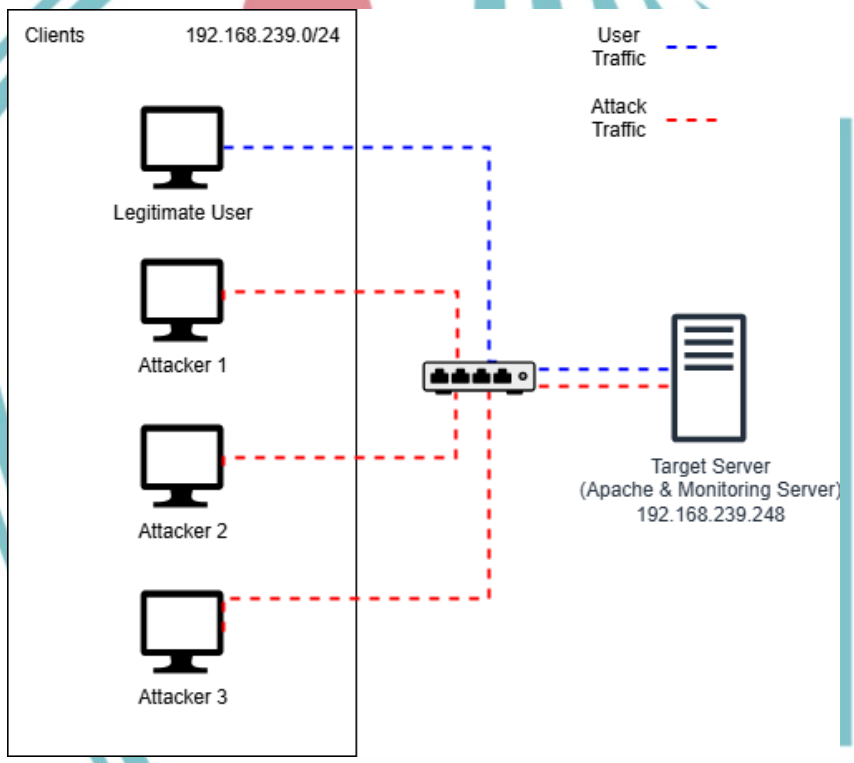
Gambar 4. 2 Flowchart Sistem

Gambar tersebut menunjukkan alur kerja sistem mitigasi DDoS berbasis entropy dan puzzle Proof-of-Work (PoW) di halaman login.php. Saat pengguna mengakses

login.php, sistem menghitung nilai entropy dari request yang masuk. Jika nilai entropy di bawah threshold (menunjukkan potensi serangan), sistem akan mengirimkan puzzle PoW ke klien untuk diselesaikan. Hanya jika solusi puzzle valid, sistem melanjutkan proses validasi username dan password di database. Jika entropy normal (tidak ada indikasi serangan), login langsung diproses tanpa puzzle. Sistem memastikan bahwa hanya pengguna asli dengan trafik normal atau yang berhasil menyelesaikan puzzle yang dapat mengakses layanan.

4.2.3 Topologi Jaringan

1. Topologi Jaringan 3 Virtual Machine

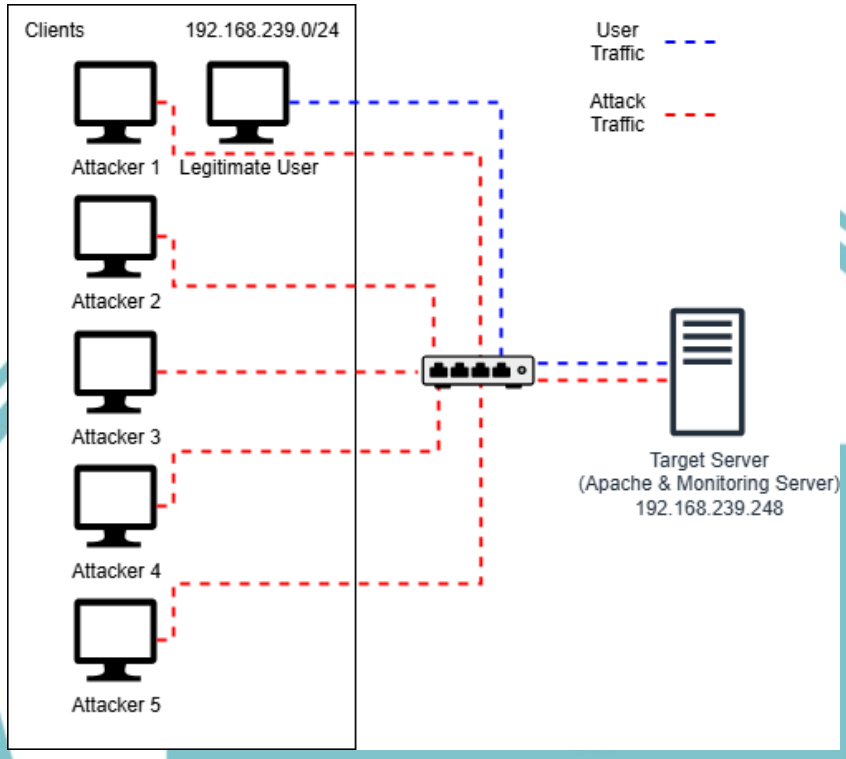


Gambar 4. 3 Topologi Jaringan 3 Virtual Machines

Gambar 4.3 menunjukkan topologi pengujian serangan Distributed Denial of Service (DDoS) yang dilakukan dalam jaringan lokal dengan subnet 192.168.239.0/24. Dalam skenario ini, terdapat empat mesin client yang terdiri dari satu pengguna sah (Legitimate User) dan tiga mesin penyerang (Attacker 1, Attacker 2, dan Attacker 3). Semua *client* terhubung pada satu jaringan yang sama, kemudian mengarah langsung ke target server dengan alamat IP 192.168.239.248.

Server ini berperan sebagai host untuk layanan web (Apache) sekaligus modul pemantauan sistem mitigasi.

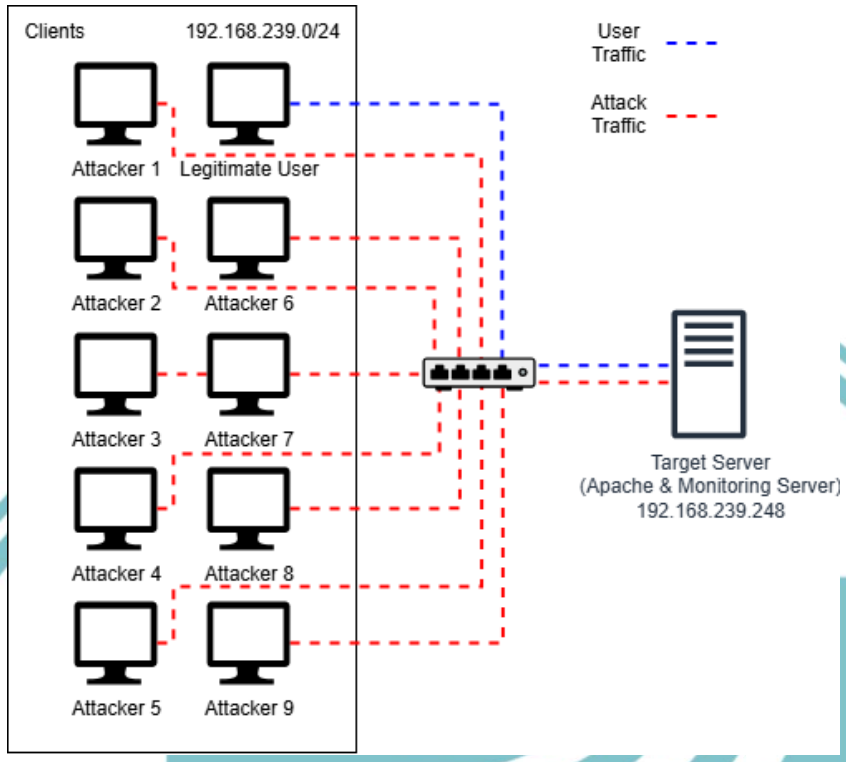
2. Topologi Jaringan 5 Virtual Machine



Gambar 4. 4 Topologi Jaringan 5 Virtual Machines

Gambar 4.4 menunjukkan topologi pengujian serangan Distributed Denial of Service (DDoS) yang dilakukan dalam jaringan lokal dengan subnet 192.168.239.0/24. Dalam skenario ini, terdapat empat mesin client yang terdiri dari satu pengguna sah (Legitimate User) dan tiga mesin penyerang (Attacker 1 hingga attacker 5). Semua client terhubung pada satu jaringan yang sama, kemudian mengarah langsung ke target server dengan alamat IP 192.168.239.248. Server ini berperan sebagai host untuk layanan web (Apache) sekaligus modul pemantauan sistem mitigasi.

3. Topologi Jaringan 9 Virtual Machine



Gambar 4. 5 Topologi Jaringan 9 Virtual Machines

Gambar 4.5 menunjukkan topologi pengujian serangan Distributed Denial of Service (DDoS) yang dilakukan dalam jaringan lokal dengan subnet 192.168.239.0/24. Dalam skenario ini, terdapat empat mesin client yang terdiri dari satu pengguna sah (Legitimate User) dan tiga mesin penyerang (Attacker 1 hingga attacker 9). Semua client terhubung pada satu jaringan yang sama, kemudian mengarah langsung ke target server dengan alamat IP 192.168.239.248. Server ini berperan sebagai host untuk layanan web (Apache) sekaligus modul pemantauan sistem mitigasi.

4.2.4 Pengalamatan IP Address

Tabel 4. 2 Pengalamatan IP Address

Nama Hostname	Interface	IP Address	Gateway	Adaptor
Server Monitoring	enp0s3	192.168.239.248	192.168.239.158	Bridge Adapter
Attacker 1	enp0s3	192.168.239.76	192.168.239.158	Bridge Adapter
Attacker 2	enp0s3	192.168.239.33	192.168.239.158	Bridge Adapter
Attacker 3	enp0s3	192.168.239.38	192.168.239.158	Bridge Adapter
Attacker 4	enp0s3	192.168.239.219	192.168.239.158	Bridge Adapter
Attacker 5	enp0s3	192.168.239.41	192.168.239.158	Bridge Adapter
Attacker 6	enp0s3	192.168.239.2	192.168.239.158	Bridge Adapter
Attacker 7	enp0s3	192.168.239.60	192.168.239.158	Bridge Adapter
Attacker 8	enp0s	192.168.239.61	192.168.239.158	Bridge Adapter
Attacker 9	enp0s3	192.168.239.63	192.168.239.158	Bridge Adapter

Detail pengalamatan IP pada sistem ini ditampilkan pada Tabel 4.2, yang mencakup informasi penting untuk mendukung komunikasi antar virtual machine. Setiap perangkat, baik server Ubuntu maupun mesin attacker, menggunakan interface enp0s3 dengan mode Bridge Adapter, sehingga keduanya berada dalam satu jaringan lokal yang sama. Kedua perangkat menggunakan gateway yang sama, yaitu 192.168.239.158, sebagai jalur utama untuk mengakses jaringan luar. Pengaturan ini memastikan konektivitas stabil selama proses simulasi dan pengujian mitigasi DDoS berlangsung. Dengan konfigurasi ini, lalu lintas jaringan

dapat dianalisis secara akurat dan sistem mitigasi dapat merespon sesuai perilaku trafik yang terpantau.

4.2.5 Spesifikasi Virtual Machine

Tabel 4. 3 Spesifikasi Virtual Machine

Nama VM	OS	CPU	RAM	Storage	Role
VM Attacker	Ubuntu 24.04.1 LTS	2 core, Intel Core i7-1165G7 @2.80GHz	2048 MB	20 GB	Pengujian serangan (GoldenEye)
VM Target Server	Ubuntu 24.04.1 LTS	2 core, Intel Core i7-1165G7 @2.80GHz	4096 MB	50 GB	Web server + *Entropy* + PoW Mitigasi

Spesifikasi Virtual Machine yang digunakan dalam penelitian ini ditampilkan pada Tabel Tabel tersebut memuat konfigurasi penting seperti jumlah inti CPU, kapasitas RAM, ruang penyimpanan, serta sistem operasi yang dijalankan di masing-masing VM. Dalam sistem ini, digunakan dua buah VM dengan OS Ubuntu 24.04.1 LTS. VM pertama berperan sebagai attacker dan dilengkapi 2 core CPU, 2 GB RAM, dan 20 GB penyimpanan, yang digunakan untuk menjalankan tools pengujian serangan seperti GoldenEye. Sementara itu, VM kedua berfungsi sebagai server target, dengan 2 core CPU, 4 GB RAM, dan 50 GB penyimpanan, menjalankan layanan web berbasis Apache2 serta modul mitigasi seperti *Entropy* Detection dan Puzzle *Proof of Work* (PoW). Penyesuaian spesifikasi ini dilakukan agar masing-masing virtual machine mampu menjalankan perannya secara optimal dalam simulasi dan pengujian sistem mitigasi serangan DDoS.

4.3 Implementasi Sistem

4.3.1 Instalasi Virtual Machine dan Konfigurasi Ubuntu Server

Gambar 4. 6 Logo Virtual Box

Sumber: https://www.oracle.com/a/ocom/img/rh03v3fa-virtualbox.jpg

Sebelum sistem mitigasi DDoS dirancang dan diuji seluruh infrastruktur server dalam penelitian ini diimplementasikan menggunakan VirtualBox sebagai platform virtualisasi. Penggunaan virtual machine (VM) memungkinkan simulasi sistem secara efisien dalam satu perangkat fisik, serta memberikan fleksibilitas tinggi dalam pengaturan skenario serangan dan pengujian respons mitigasi.

Setiap VM dalam sistem ini diinstal dengan sistem operasi Ubuntu Server 24.04.1 LTS, yang dipilih karena sifatnya yang open-source, ringan, stabil, dan memiliki kompatibilitas tinggi terhadap tool keamanan jaringan serta web server seperti Apache2. Selain itu, Ubuntu Server mendukung ekosistem perangkat lunak, termasuk MySQL, PHP 8.3, dan OpenSSH, sehingga sangat sesuai untuk kebutuhan implementasi sistem mitigasi dan monitoring trafik jaringan.

Untuk mendukung komunikasi jaringan antar mesin virtual dan integrasi dengan jaringan lokal, seluruh adaptor jaringan VM dikonfigurasi dalam mode Bridge Adapter, memungkinkan setiap VM memperoleh alamat IP dari jaringan fisik yang sama. Konfigurasi ini memastikan bahwa seluruh VM—baik server target maupun VM attacker—dapat saling berkomunikasi secara langsung dalam satu subnet.

Dalam implementasi ini, digunakan total enam virtual machine, dengan rincian sebagai berikut:

- 1 VM Server Ubuntu berfungsi sebagai web server dan sistem mitigasi, dilengkapi dengan modul login berbasis PHP, perhitungan *entropy* real-time, serta puzzle *Proof of Work* (PoW) menggunakan JavaScript.
- 9 VM Ubuntu Attacker, masing-masing digunakan untuk mensimulasikan serangan HTTP Flood ke endpoint login dengan bantuan tools seperti GoldenEye. Setiap attacker dikonfigurasi dengan script otomatisasi untuk menguji respons sistem terhadap lonjakan trafik.

Seluruh VM tersebut disiapkan dengan pengalamatan IP statis yang ditentukan melalui konfigurasi Netplan (/etc/netplan/*.yaml). Alamat IP server utama, misalnya 192.168.239.248, digunakan sebagai target serangan dari kesembilan VM attacker, yang masing-masing diberi alamat IP dalam subnet yang sama.

4.3.2 Instalasi dan Konfigurasi Web Server Apache2

```
berli@skripsipow:~$ sudo apt update
[sudo] password for berli:
Hit:1 http://id.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
140 packages can be upgraded. Run 'apt list --upgradable' to see them.
berli@skripsipow:~$
```

Gambar 4. 7 Memperbarui Daftar Paket

Perintah apt-get merupakan tool manajemen paket untuk mengelola software. Perintah update merupakan opsi untuk mengunduh ulang informasi terbaru tentang semua paket dari repositori yang terdaftar.

```
berli@skripsipow:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.6).
0 upgraded, 0 newly installed, 0 to remove and 140 not upgraded.
berli@skripsipow:~$
```

Gambar 4. 8 Instalasi Web Server Apache

Perintah apt-get merupakan *tool* manajemen paket untuk mengelola software. Setelah berhail menjalankan update repositori berikutnya dilakukan penginstalan web server Apache.

```
berli@skripsipow:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
berli@skripsipow:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
berli@skripsipow:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
22                         ALLOW       Anywhere
443                        ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
```

Gambar 4. 9 Konfigurasi UFW Status

Perintah sudo ufw app list digunakan untuk menampilkan daftar aplikasi yang memiliki profil bawaan dalam UFW. Output menunjukkan bahwa sistem mengenali beberapa aplikasi dengan profil firewall, seperti:

- Apache
- Apache Full (port 80 dan 443)
- Apache Secure (port 443 saja)
- OpenSSH (port 22 untuk SSH remote)

Perintah sudo ufw allow 'Apache' mengizinkan lalu lintas masuk untuk layanan Apache HTTP pada port 80 (HTTP), yang dibutuhkan agar web server kamu bisa diakses dari luar (termasuk oleh pengguna dan attacker dalam simulasi). Jika rule sudah ada, UFW akan melewatinya seperti terlihat dari pesan.

Perintahs udo ufw status menunjukkan status firewall dan daftar rules yang sedang aktif.

```
berli@skripsipow:~$ sudo systemctl restart apache2
```

Gambar 4. 10 Pembaruan Konfigurasi Apache

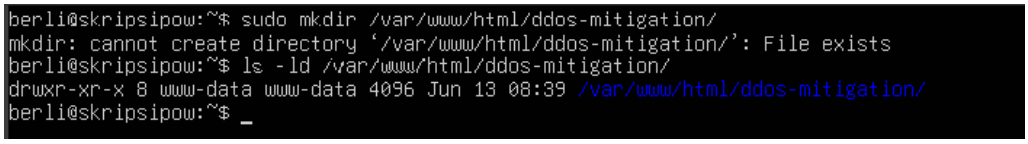
```
berli@skripsipow:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-06-13 08:24:31 WIB; 56s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 5113 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 5117 (apache2)
      Tasks: 6 (limit: 4605)
     Memory: 11.4M (peak: 11.6M)
        CPU: 71ms
     CGroup: /system.slice/apache2.service
             ├─5117 /usr/sbin/apache2 -k start
             ├─5119 /usr/sbin/apache2 -k start
             ├─5120 /usr/sbin/apache2 -k start
             ├─5121 /usr/sbin/apache2 -k start
             ├─5122 /usr/sbin/apache2 -k start
             └─5123 /usr/sbin/apache2 -k start

Jun 13 08:24:31 skripsipow systemd[1]: Starting apache2.service - The Apache HTTP Server...
```

Gambar 4. 11 Pengecekan Status Apache

Perintah systemctl untuk mengelola service atau layanan di sistem Linux berbasis systemd. Perintah status untuk melakukan pengecekan keberhasilan instalasi service tersebut. Service yang dimaksud adalah Apache2 karena sebelumnya baru diinstalasi.

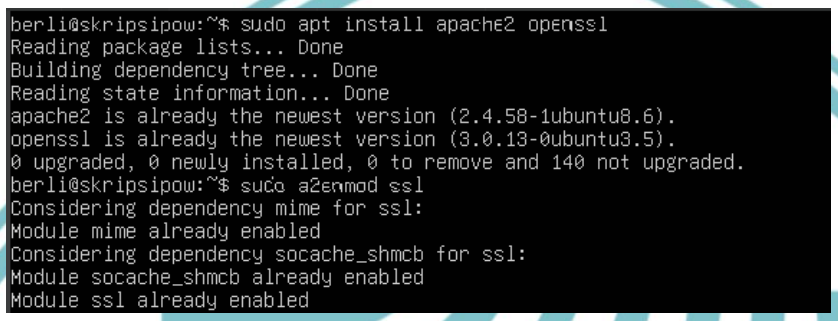
Gambar 4. 12 Tampilan Web Server Apache

Gambar diatas merupakan tampilan web server Apache dapat diakses dengan cara membuka IP address localhost, yaitu 192.168.239.248.



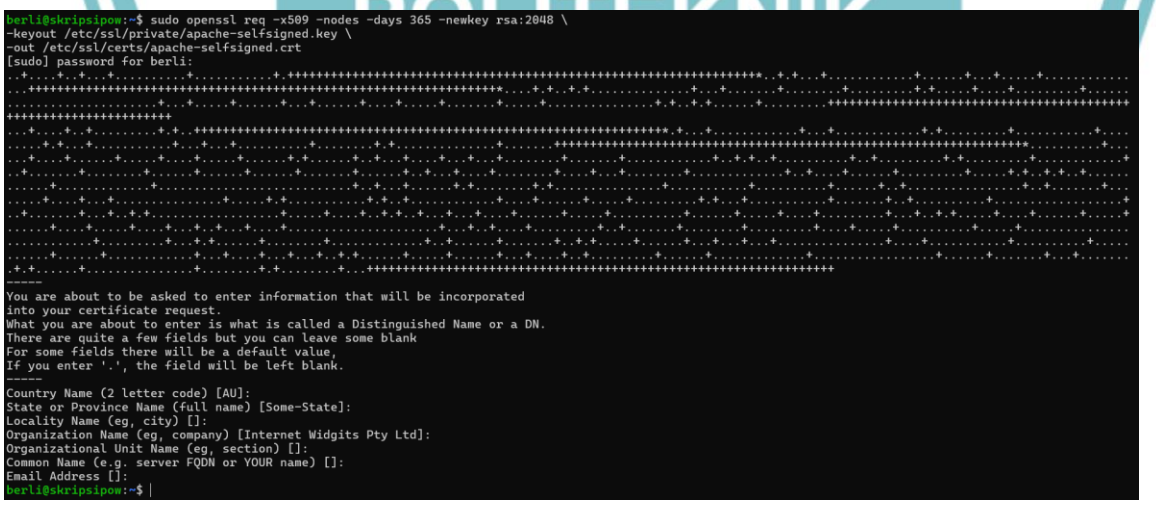
Gambar 4. 13 Pembuatan Direktori Baru untuk Menjalankan Sistem Mitigasi

Direktori ini dibuat di dalam root direktori Apache2, yaitu /var/www/html/, dengan nama ddos-mitigation.



Gambar 4. 14 Instalasi OpenSSL

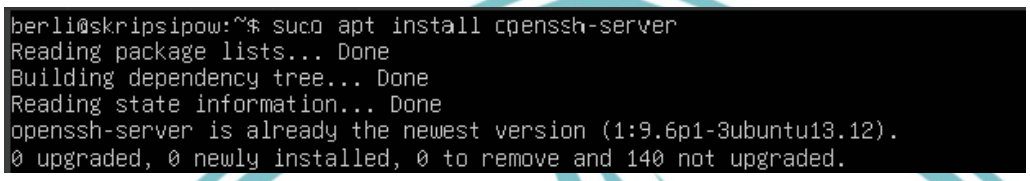
Apache2 dan OpenSSL diinstal menggunakan perintah sudo apt install apache2 openssl untuk menjalankan layanan web dan mendukung koneksi HTTPS. Setelah itu, modul SSL diaktifkan dengan sudo a2enmod ssl, yang menunjukkan bahwa modul dan dependensinya.



Gambar 4. 15 Konfigurasi Modul SSL pada Apache

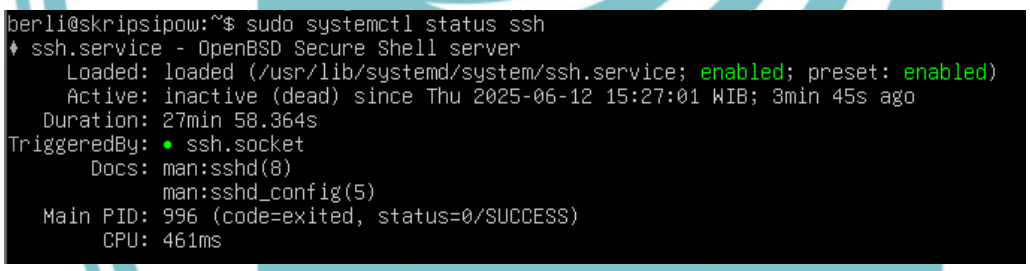
Sertifikat SSL self-signed dibuat menggunakan perintah openssl req -x509 dengan masa berlaku 365 hari dan panjang kunci 2048 bit. Tujuan dari pembuatan sertifikat ini adalah untuk mengaktifkan koneksi HTTPS pada server Apache2, sehingga

komunikasi antara pengguna dan server dapat dienkripsi dan lebih aman. Proses ini menghasilkan dua file: apache-selfsigned.crt dan apache-selfsigned.key yang disimpan di direktori /etc/ssl/, dan nantinya akan digunakan dalam konfigurasi virtual host SSL Apache. Informasi seperti negara, organisasi, dan common name diisi sebagai bagian dari metadata sertifikat.



Gambar 4. 16 Instalasi OpenSSH

Layanan OpenSSH Server diinstal menggunakan perintah sudo apt install openssh-server. Fungsinya adalah untuk memungkinkan akses remote ke server Ubuntu dari perangkat lain melalui protokol SSH. Hal ini penting agar administrator dapat mengelola server dari luar tanpa perlu menggunakan terminal lokal.



Gambar 4. 17 Pengecekan Status OpenSSH

Setelah instalasi, status layanan diperiksa dengan sudo systemctl status ssh. Hasilnya menunjukkan bahwa service telah berhasil dijalankan dan menerima koneksi pada port 22. Riwayat log juga menampilkan aktivitas login dari IP 192.168.239.87 dengan user berli, yang menandakan bahwa server sudah siap diakses secara jarak jauh.

```
PS C:\Users\Berlianna> ssh berli@192.168.239.248
berli@192.168.239.248's password:

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-63-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Jul  5 05:02:01 PM WIB 2025

  System load:  1.34               Processes:               159
  Usage of /:   52.9% of 23.45GB   Users logged in:         1
  Memory usage: 28%                IPv4 address for enp0s8: 10.0.2.4
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

128 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

Last login: Sat Jul  5 15:54:26 2025 from 192.168.239.87
```

Gambar 4. 18 Percobaan SSH ke Ubuntu Server dari Windows

Akses remote ke server Ubuntu dilakukan dari command prompt Windows menggunakan perintah ssh berli@192.168.239.248. Setelah berhasil login, terminal menampilkan informasi sistem seperti versi Ubuntu (24.04.1 LTS), beban sistem, pemakaian memori, serta alamat IP aktif (10.0.2.4). Proses ini membuktikan bahwa layanan SSH berfungsi dengan baik dan memungkinkan pengelolaan server dari perangkat client secara jarak jauh.

4.3.3 Instalasi MySQL dan Konfigurasi Database

```
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo apt install mysql-server -y
```

Gambar 4. 19 Instalasi MySQL Server

Instalasi MySQL dilakukan dengan perintah sudo apt install mysql-server -y, di mana apt install berfungsi untuk mengunduh dan memasang paket perangkat lunak dari repository Ubuntu, sedangkan opsi -y digunakan agar proses instalasi berjalan otomatis tanpa perlu konfirmasi manual.

```
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo systemctl status mysql
[sudo] password for berli:
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-06-12 19:59:25 WIB; 52min ago
    Process: 821 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 937 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 4605)
     Memory: 441.1M (peak: 442.7M)
        CPU: 1min 41.491s
     CGroup: /system.slice/mysql.service
             └─937 /usr/sbin/mysqld

Jun 12 19:59:16 skripsipow systemd[1]: Starting mysql.service - MySQL Community Server...
Jun 12 19:59:25 skripsipow systemd[1]: Started mysql.service - MySQL Community Server.
```

Gambar 4. 20 Pengecekan Status MySQL

Perintah sudo systemctl status mysql dijalankan untuk memastikan bahwa layanan MySQL berhasil aktif. Dari output yang ditampilkan, terlihat bahwa status MySQL adalah active (running), yang berarti server database sudah berjalan dengan baik dan siap digunakan untuk menyimpan data sistem seperti log request, *entropy*, dan hasil validasi puzzle dalam proyek mitigasi DDoS.

```
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Gambar 4. 21 Percobaan Akses MySQL dari Terminal

Untuk mengakses MySQL sebagai pengguna root, digunakan perintah sudo mysql -u root -p. Perintah ini berfungsi untuk membuka MySQL monitor dengan hak akses administrator (-u root), sementara opsi -p menandakan bahwa sistem akan meminta input password sebelum masuk. Setelah memasukkan password yang benar, terminal menampilkan informasi versi MySQL yang terpasang, yaitu versi 8.0.42 untuk Ubuntu 24.04.1. Tanda bahwa koneksi berhasil adalah munculnya prompt MySQL monitor yang siap menerima perintah SQL. Langkah ini penting sebagai tahap awal sebelum pembuatan database dan tabel untuk sistem mitigasi DDoS.

```
mysql> use ddos_mitigation
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------+
| Tables_in_ddos_mitigation |
+---------------------------+
| entropy_log               |
| entropy_monitor           |
| log_request               |
| puzzle_log                |
| users                     |
+---------------------------+
5 rows in set (0.01 sec)
```

Gambar 4. 22 Create Database

Setelah berhasil masuk ke MySQL, dibuat sebuah database baru dengan nama ddos_mitigation menggunakan perintah CREATE DATABASE ddos_mitigation;. Selanjutnya, perintah USE ddos_mitigation; digunakan untuk memilih dan mengaktifkan database tersebut agar dapat digunakan dalam sesi MySQL saat ini. Setelah database aktif, perintah SHOW TABLES; dijalankan untuk menampilkan seluruh tabel yang terdapat di dalamnya. Hasil eksekusi menunjukkan bahwa terdapat lima tabel, yaitu: entropy_log, entropy_monitor, log_request, puzzle_log, dan users.

```
CREATE TABLE entropy_log (
  id INT AUTO_INCREMENT PRIMARY KEY,
  ip_address VARCHAR(45),
  entropy FLOAT,
  timestamp DATETIME,
  status VARCHAR(20)
);
```

Gambar 4. 23 Create Tabel entropy_log

Pembuatan tabel entropy_log dalam database ddos_mitigation digunakan untuk mencatat hasil perhitungan entropy dari lalu lintas jaringan sebagai bagian dari sistem deteksi serangan DDoS. Tabel terdiri dari lima kolom: id sebagai primary key yang bertambah otomatis, ip_address untuk menyimpan alamat IP pengirim

request, entropy sebagai nilai hasil perhitungan entropy, timestamp untuk mencatat waktu pencatatan, dan status untuk menunjukkan apakah trafik terdeteksi normal atau mencurigakan.

```
mysql> desc entropy_log;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| id         | int         | NO   | PRI | NULL    | auto_increment |
| ip_address | varchar(45) | YES  |     | NULL    |                |
| entropy    | float       | YES  |     | NULL    |                |
| timestamp  | datetime    | YES  |     | NULL    |                |
| status     | varchar(20) | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Gambar 4. 24 Struktur Tabel entropy_log

Tabel ini terdiri dari lima field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap ada entri baru. Field ip_address bertipe VARCHAR(45), digunakan untuk menyimpan alamat IP pengguna atau host yang dianalisis. Field *entropy* bertipe FLOAT, menyimpan nilai *entropy* yang dihitung berdasarkan distribusi trafik jaringan. Field timestamp bertipe DATETIME, berfungsi mencatat waktu kapan perhitungan *entropy* dilakukan. Sementara itu, field status bertipe VARCHAR(20) digunakan untuk menandai status yang terbaca di error_log.

```
CREATE TABLE entropy_monitor (
  id INT AUTO_INCREMENT PRIMARY KEY,
  timestamp DATETIME,
  total_requests INT,
  unique_ips INT,
  entropy_value FLOAT,
  status VARCHAR(255),
  is_mitigating TINYINT(1) DEFAULT 0
);
```

Gambar 4. 25 Create Tabel entropy_monitor

Tabel ini digunakan untuk mencatat hasil pemantauan *entropy* dari lalu lintas jaringan sebagai bagian dari sistem deteksi serangan DDoS. Tabel terdiri dari tujuh kolom: id sebagai primary key yang bertambah otomatis, timestamp untuk mencatat

waktu pencatatan, total_requests menyimpan jumlah total request yang diterima, unique_ips untuk mencatat jumlah alamat IP unik, entropy_value sebagai nilai hasil perhitungan *entropy*, status untuk menunjukkan apakah trafik terdeteksi normal atau mencurigakan, dan is_mitigating sebagai indikator apakah sistem sedang dalam mode mitigasi (1) atau tidak (0).

```
mysql> desc entropy_monitor;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id             | int          | NO   | PRI | NULL    | auto_increment |
| timestamp      | datetime     | YES  |     | NULL    |                |
| total_requests | int          | YES  |     | NULL    |                |
| unique_ips     | int          | YES  |     | NULL    |                |
| entropy_value  | float        | YES  |     | NULL    |                |
| status         | varchar(255) | YES  |     | NULL    |                |
| is_mitigating  | tinyint(1)   | YES  |     | 0       |                |
+----------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

Gambar 4. 26 Struktur Tabel entropy_monitor

Tabel ini terdiri dari tujuh field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap kali entri baru ditambahkan. Field timestamp bertipe DATETIME, digunakan untuk mencatat waktu saat pemantauan dilakukan. Field total_requests bertipe INT, berfungsi mencatat jumlah total permintaan (request) yang diterima server dalam periode tertentu. Field unique_ips juga bertipe INT, menyimpan jumlah alamat IP unik yang terdeteksi saat itu. Field entropy_value bertipe FLOAT, menyimpan nilai *entropy* yang dihitung dari distribusi IP. Field status bertipe VARCHAR(255), digunakan untuk menandai kondisi trafik, apakah normal atau mencurigakan. Terakhir, field is_mitigating bertipe TINYINT(1) dengan nilai default 0, berfungsi sebagai indikator apakah sistem sedang menjalankan mitigasi serangan DDoS (nilai 1) atau tidak (0).

```
CREATE TABLE log_request (
  id INT AUTO_INCREMENT PRIMARY KEY,
  ip_address VARCHAR(45),
  user_agent TEXT,
  endpoint VARCHAR(100),
  timestamp DATETIME
);
```

Gambar 4. 27 Create Tabel log_request

Tabel ini digunakan untuk mencatat setiap permintaan (request) yang masuk ke server sebagai bagian dari proses pemantauan lalu lintas jaringan. Tabel terdiri dari lima kolom: id sebagai primary key yang bertambah otomatis, ip_address untuk menyimpan alamat IP pengirim request, user_agent untuk mencatat informasi perangkat atau browser yang digunakan, endpoint untuk menunjukkan URL atau jalur endpoint yang diakses, dan timestamp untuk mencatat waktu saat permintaan tersebut diterima oleh server.

```
mysql> desc log_request;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | int          | NO   | PRI | NULL    | auto_increment |
| ip_address | varchar(45)  | YES  |     | NULL    |                |
| user_agent | text         | YES  |     | NULL    |                |
| endpoint   | varchar(100) | YES  |     | NULL    |                |
| timestamp  | datetime     | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Gambar 4. 28 Struktur Tabel log_request

Tabel ini terdiri dari lima field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap ada entri baru. Field ip_address bertipe VARCHAR(45), digunakan untuk menyimpan alamat IP pengguna atau host yang mengirimkan request. Field user_agent bertipe TEXT, mencatat informasi mengenai perangkat atau browser yang digunakan pengguna. Field endpoint bertipe VARCHAR(100), digunakan untuk mencatat endpoint atau URL yang diakses. Sementara itu, field timestamp bertipe DATETIME digunakan untuk mencatat waktu ketika request diterima oleh server.

```sql
CREATE TABLE puzzle_log (
  id INT AUTO_INCREMENT PRIMARY KEY,
  ip_address VARCHAR(45),
  timestamp DATETIME,
  start_time DATETIME,
  end_time DATETIME,
  nonce VARCHAR(255),
  difficulty INT,
  is_valid TINYINT(1),
  user_agent TEXT,
  processing_time FLOAT,
  base VARCHAR(64),
  hash_result VARCHAR(128),
  status VARCHAR(255)
);
```

Gambar 4. 29 Create Tabel puzzle_log

Tabel ini digunakan untuk mencatat proses validasi *puzzle* berbasis *Proof of Work* sebagai bagian dari sistem mitigasi serangan DDoS. Tabel terdiri dari beberapa kolom, di antaranya: id sebagai primary key yang bertambah otomatis, ip_address untuk menyimpan alamat IP pengirim request, timestamp untuk mencatat waktu pencatatan log, start_time dan end_time untuk mencatat waktu mulai dan selesai proses puzzle, serta *nonce* yang merupakan nilai yang dicoba untuk mendapatkan hash yang valid. Field difficulty menunjukkan tingkat kesulitan puzzle, is_valid sebagai indikator keberhasilan validasi puzzle (1 untuk valid, 0 untuk gagal), dan user_agent mencatat informasi perangkat pengguna. Kolom processing_time menyimpan waktu yang dibutuhkan untuk menyelesaikan puzzle, base adalah string dasar hash, hash_result adalah hasil hash akhir yang dikirim pengguna, dan status menunjukkan hasil akhir proses.

```
mysql> desc puzzle_log;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | int          | NO   | PRI | NULL    | auto_increment |
| ip_address      | varchar(45)  | YES  |     | NULL    |                |
| timestamp       | datetime     | YES  |     | NULL    |                |
| start_time      | datetime     | YES  |     | NULL    |                |
| end_time        | datetime     | YES  |     | NULL    |                |
| nonce           | varchar(255) | YES  |     | NULL    |                |
| difficulty      | int          | YES  |     | NULL    |                |
| is_valid        | tinyint(1)   | YES  |     | NULL    |                |
| user_agent      | text         | YES  |     | NULL    |                |
| processing_time | float        | YES  |     | NULL    |                |
| base            | varchar(64)  | YES  |     | NULL    |                |
| hash_result     | varchar(128) | YES  |     | NULL    |                |
| status          | varchar(255) | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
13 rows in set (0.00 sec)
```

Gambar 4. 30 Struktur Tabel puzzle_log

Tabel ini terdiri dari tiga belas field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap ada entri baru. Field ip_address bertipe VARCHAR(45), digunakan untuk menyimpan alamat IP pengguna atau host yang mengirimkan request. Field timestamp, start_time, dan end_time semuanya bertipe DATETIME, digunakan untuk mencatat waktu saat log dibuat, waktu mulai, dan waktu selesai proses penyelesaian puzzle. Field nonce bertipe VARCHAR(255), menyimpan nilai percobaan untuk menghasilkan hash yang valid. *Field difficulty* bertipe INT, menyimpan tingkat kesulitan puzzle yang diberikan. Field is_valid bertipe TINYINT(1), digunakan sebagai indikator apakah solusi puzzle yang dikirim valid atau tidak. Field user_agent bertipe TEXT, digunakan untuk mencatat informasi browser atau perangkat pengguna. Field processing_time bertipe FLOAT, mencatat waktu yang dibutuhkan untuk menyelesaikan puzzle. Field base dan hash_result masing-masing bertipe VARCHAR(64) dan VARCHAR(128), menyimpan data dasar yang di-hash dan hasil akhirnya. Terakhir, field status bertipe VARCHAR(255), digunakan untuk menandai status akhir dari proses puzzle.

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100) NOT NULL,
  password VARCHAR(255) NOT NULL
);
```

Gambar 4. 31 Create Tabel users

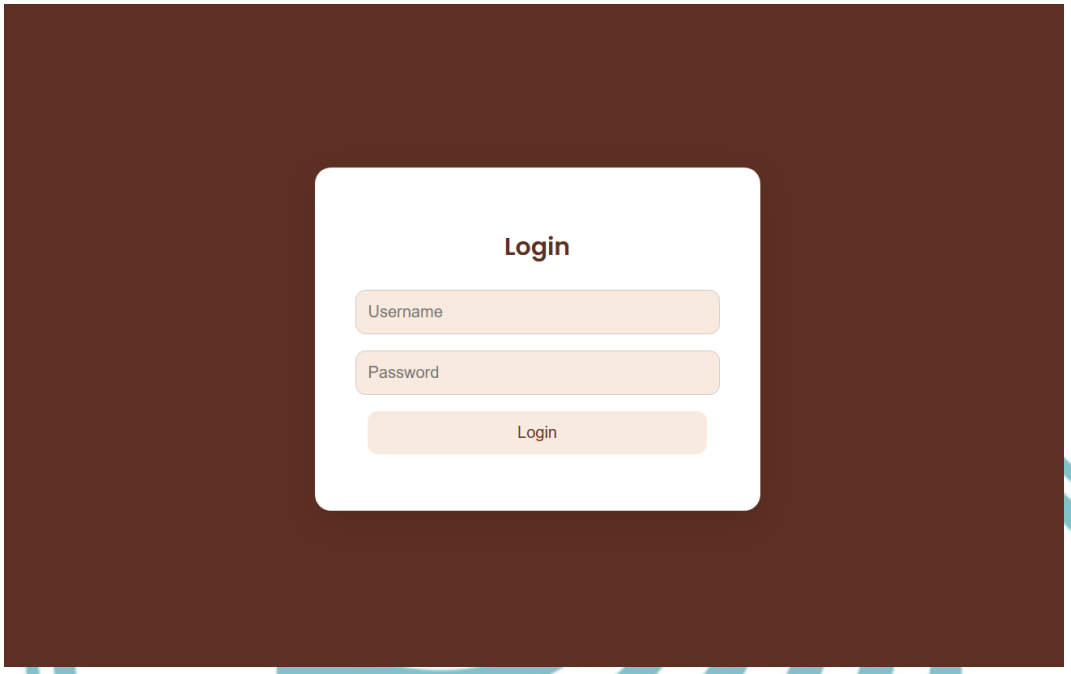
Tabel ini terdiri dari tiga field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap ada entri baru. Field username bertipe VARCHAR(100) dan wajib diisi (NOT NULL), digunakan untuk menyimpan nama pengguna yang terdaftar dalam sistem. Field password bertipe VARCHAR(255) dan juga bersifat NOT NULL, digunakan untuk menyimpan password pengguna.

```
mysql> desc users;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int          | NO   | PRI | NULL    | auto_increment |
| username | varchar(100) | NO   |     | NULL    |                |
| password | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

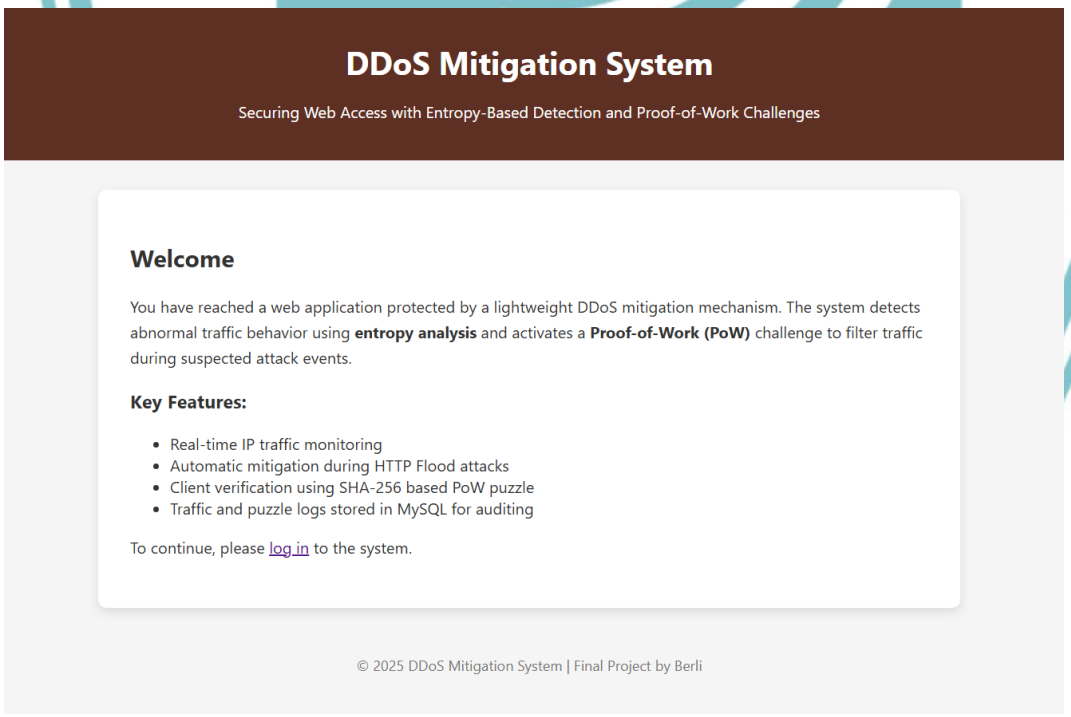
Gambar 4. 32 Struktur Tabel users

Tabel ini terdiri dari tiga field. Field id bertipe INT dan berfungsi sebagai primary key yang bersifat unik serta menggunakan AUTO_INCREMENT agar nilainya bertambah otomatis setiap ada entri baru. Field username bertipe VARCHAR(100), digunakan untuk menyimpan nama pengguna yang terdaftar dalam sistem dan bersifat wajib diisi (NOT NULL). Sementara itu, field password bertipe VARCHAR(255) digunakan untuk menyimpan kata sandi pengguna yang telah di-hash untuk keperluan autentikasi saat login, dan juga bersifat NOT NULL.

4.3.4 Pembuatan Sistem Login Berbasis PHP



Gambar 4. 33 Tampilan Halaman Login



Gambar 4. 34 Tampilan Landing Page

Sistem login merupakan antarmuka utama yang digunakan pengguna untuk mengakses halaman utama sistem. File login.php bertanggung jawab tidak hanya

untuk menampilkan form login (username dan password), tetapi juga untuk mencatat request, memanggil fungsi perhitungan *entropy*, dan menentukan apakah mitigasi DDoS perlu diaktifkan.

Form login dibuat menggunakan HTML untuk menerima input username dan password. Field tambahan akan muncul secara dinamis saat mitigasi aktif

```
<form id="loginForm" method="POST" action="verify.php">
  <input type="text" name="username" required />
  <input type="password" name="password" required />
  <input type="hidden" name="human_token" id="human_token" />
  ...
</form>
```

Gambar 4. 35 Form Login dengan Input Hidden Token untuk Mitigasi Puzzle

Setiap logging request ke halaman login dicatat ke dalam database untuk keperluan analisis pola trafik.

```
$ip        = $_SERVER['REMOTE_ADDR'];
$userAgent = $_SERVER['HTTP_USER_AGENT'];
$endpoint  = $_SERVER['REQUEST_URI'];
$time      = date("Y-m-d H:i:s");

$sql = "INSERT INTO log_request (ip_address, user_agent, endpoint,
timestamp)
        VALUES ('$ip', '$userAgent', '$endpoint', '$time')";
mysqli_query($conn, $sql);
```

Gambar 4. 36 Script Pencatatan IP, User Agent, dan Endpoint ke dalam Tabel log_request

Pemanggilan Perhitungan *entropy* (functions.php) untuk menghitung nilai *entropy* berdasarkan distribusi IP pengguna

```
include "functions.php";
$entropy_result = calculate_entropy_for_current_request($conn);
```

Gambar 4. 37 Pemanggilan Fungsi Perhitungan Entropy dari functions.php

Penyimpanan Session & Token Puzzle fungsinya, jika mitigasi diaktifkan, token puzzle akan disimpan di session untuk proses validasi.

```
$_SESSION['is_mitigating'] = $entropy_result['is_mitigating'];
$_SESSION['entropy_value'] = $entropy_result['entropy'];

if ($is_mitigating) {
  if (!isset($_SESSION['puzzle_token'])) {
    $_SESSION['puzzle_token'] = bin2hex(random_bytes(16));
  }
  $puzzle_token = $_SESSION['puzzle_token'];
}
```

Gambar 4. 38 Pengaturan Session Mitigasi dan Token Puzzle Berdasarkan Nilai Entropy

Fungsi tingkat kesulitan ditentukan berdasarkan nilai *entropy*. Semakin rendah *entropy*, semakin sulit puzzle yang diberikan.

```
$difficulty = ($entropy_value < 1.0) ? 5 : (($entropy_value < 2.0) ? 4 :
3);
$_SESSION['entropy_difficulty'] = $difficulty;
```

Gambar 4. 39 Penentuan Tingkat Kesulitan Puzzle Berdasarkan Nilai Entropy

Fungsi menampilkan puzzle kepada pengguna jika status mitigasi aktif.

```php
<?php if ($is_mitigating): ?>
  <input type="hidden" name="difficulty" value="<?= $difficulty ?>" />
  <input type="hidden" name="puzzle_token" value="<?= $puzzle_token ?>"
/>
  <script src="js/puzzle.js" defer></script>
<?php endif; ?>
```

Gambar 4. 40 Pengiriman Kesulitan Puzzle saat Mitigasi Aktif ke puzzle.js

4.3.5 Implementasi Perhitungan Entropy

Perhitungan *entropy* dalam sistem ini bertujuan untuk mendeteksi pola lalu lintas jaringan yang tidak normal sebagai indikasi awal serangan DDoS. Nilai *entropy* dihitung dari sebaran alamat IP dalam request yang tercatat di server dalam jangka waktu tertentu. Perhitungan ini dilakukan setiap kali ada permintaan ke halaman login, menggunakan fungsi calculate_entropy_for_current_request() yang terdapat pada file functions.php.

1. Mengambil IP Request dalam Sliding Window

Fungsi akan mengambil seluruh IP address dari request yang tercatat selama 30 detik terakhir (sliding window) melalui query ke tabel log_request. Ini dilakukan dengan membandingkan timestamp saat ini dengan 30 detik ke belakang.

```php
$start = date("Y-m-d H:i:s", strtotime("-30 seconds"));
$sql = "SELECT ip_address FROM log_request WHERE timestamp BETWEEN
'$start' AND '$now'";
```

Gambar 4. 41 Query Pengambilan IP berdasarkan Sliding Window

2. Menghitung Probabilitas dan Nilai Entropy

Dari data IP yang diambil, sistem menghitung distribusi probabilitas tiap IP dan kemudian menggunakan rumus Shannon Entropy untuk memperoleh nilai *entropy* secara *real time*.

```php
foreach ($ip_counts as $count) {
    $p = $count / $total_requests;
    $entropy -= $p * log($p, 2);
}
```

Gambar 4. 42 Perhitungan Real Time Entropy dengan Rumus Shannon

```php
$requests_per_minute = ...;
$avg_interval = ...;
$unique_ips = count($ip_counts);
```

Gambar 4. 43 Pengambilan data IP Unik, Interval, dan Request per Menit

3. Penentuan Status Mitigasi

Berdasarkan nilai *entropy*, banyaknya request, interval, dan jumlah IP unik, sistem akan menentukan apakah mitigasi DDoS perlu diaktifkan. Terdapat tiga kondisi utama yang dapat mengaktifkan mitigasi:

- Entropy rendah dan total request tinggi
- Frekuensi request tinggi dari satu IP dengan interval pendek
- Kombinasi entropy tinggi dan jumlah IP unik yang banyak (indikasi serangan botnet

```php
if ($entropy < 1.0 && $total_requests >= 10) { ... }
elseif ($requests_per_minute >= 30 && $avg_interval < 2 && $unique_ips <=
3) { ... }
elseif ($entropy >= 1.3 && $total_requests >= 150 && $unique_ips >= 4) {
... }
```

Gambar 4. 44 Threshold Entropy untuk Mengaktifkan Mitigasi

4. Pencatatan ke Tabel entropy_monitor

Setelah proses deteksi selesai, sistem mencatat hasilnya ke dalam tabel entropy_monitor yang berisi timestamp, total request, jumlah IP unik, nilai *entropy*, status mitigasi, dan flag status aktif atau tidaknya mitigasi.

```
$sql_log = "INSERT INTO entropy_monitor (...) VALUES (...)";
```

Gambar 4. 45 Penyimpanan Hasil Monitor ke Tabel entropy_monitor

4.3.6 Implementasi Puzzle *Proof of Work* (PoW)

Jika status mitigasi aktif dan alamat IP pengguna terdeteksi mencurigakan, maka sistem akan menampilkan tantangan berupa puzzle *Proof of Work* (PoW) kepada pengguna sebelum proses login dapat dilakukan. Puzzle ini dirancang agar hanya pengguna manusia (melalui browser) yang dapat menyelesaikannya, sedangkan bot atau alat otomatis akan kesulitan atau gagal dalam memenuhinya.

1. Proses Penyelesaian Puzzle (Client-side)

Puzzle diselesaikan di sisi klien menggunakan JavaScript dan menggunakan algoritma hash SHA-256. Tujuannya adalah untuk mencari nilai nonce dan base yang akan menghasilkan hash dengan awalan sejumlah angkal nol sesuai tingkat kesulitan. Setelah berhasil, klien akan mengisi *hidden field* nonce, start_time, dan end_time di form dan mengirimkannya ke server melalui form login

```
document.getElementById("nonce").value = `${base}:${nonce}`;
document.getElementById("start_time").value = new
Date(startTime).toISOString();
document.getElementById("end_time").value = new
Date(endTime).toISOString();
```

Gambar 4. 46 Implementasi Puzzle *Proof of Work*

2. Pengiriman dan Validasi Puzzle ke Server (verify.php)

Setelah puzzle diselesaikan, data dikirim ke server bersamaan dengan form login. Di sisi server, file verify.php akan memvalidasi puzzle apakah hasil hash memang dimulai dengan jumlah nol sesuai *difficulty* dan memeriksa waktu penyelesaian apakah dalam batas waktu yang ditentukan.

3. Penyimpanan Hasil Puzzle ke Tabel puzzle_log

Setelah proses validasi puzzle selesai, baik berhasil maupun gagal, sistem akan mencatat hasilnya ke dalam tabel puzzle_log. Data yang disimpan mencakup alamat

IP pengguna, waktu pencatatan (timestamp), nilai nonce, tingkat kesulitan (difficulty), waktu mulai dan selesai (start_time dan end_time), durasi penyelesaian (processing_time), status validasi, hasil hash akhir (hash_result), informasi perangkat (user_agent), serta token puzzle yang digunakan. Pencatatan ini bertujuan untuk perhitungan confusion matrix.

4.3.7 Logging dan Monitoring Sistem Mitigasi

Semua aktivitas sistem dicatat ke dalam berbagai tabel untuk keperluan analisis dan visualisasi.

- log_request: mencatat setiap request pengguna.
- entropy_monitor: menyimpan log sliding *entropy* dan status mitigasi.
- entropy_log: log entropy per request.
- puzzle_log: menyimpan hasil validasi puzzle.

Monitoring dilakukan melalui halaman monitor.php yang menampilkan grafik entropy dan status mitigasi secara real-time menggunakan Chart.js.

Tabel 4. 4 Fungsi Tabel Database

Tabel	Kolom utama	Fungsi
log_request	ip_address, user_agent, endpoint, timestamp	Logging semua request ke halaman login
entropy_log	ip_address, entropy, timestamp, status	Mencatat nilai entropy per request
entropy_monitor	total_requests, unique_ips, entropy_value, status	Untuk monitoring grafik dan status sistem
puzzle_log	start_time, end_time, base, nonce, difficulty	Mencatat detail solving puzzle PoW oleh pengguna
users	username, password	Validasi akun login pengguna

4.3.8 Instalasi GoldenEye

```
ddosattacker@ddosattacker:~$ sudo apt update
[sudo] password for ddosattacker:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
147 packages can be upgraded. Run 'apt list --upgradable' to see them.
ddosattacker@ddosattacker:~$
```

Gambar 4. 47 Memperbarui Daftar Paket

Perintah sudo apt update untuk memperbarui daftar paket dari repositori Ubuntu agar sistem memiliki informasi terbaru sebelum menginstal software apa pun. Ini memastikan bahwa dependensi yang digunakan berasal dari versi paling mutakhir yang tersedia.

```
ddosattacker@ddosattacker:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.12.3-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 147 not upgraded.
ddosattacker@ddosattacker:~$
```

Gambar 4. 48 Instalasi python3

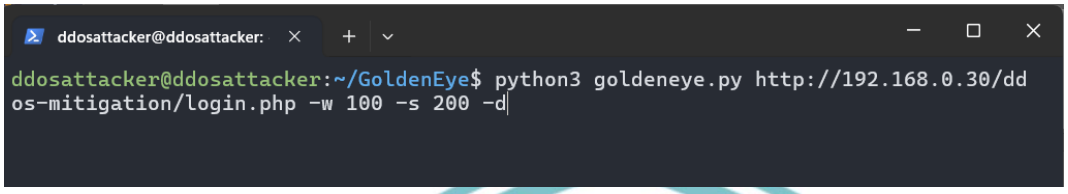
Perintah sudo apt install python3 akan mengunduh dan menginstal Python versi terbaru dari repositori resmi. Python 3 dibutuhkan karena tool GoldenEye yang digunakan untuk simulasi serangan DDoS berbasis skrip Python. Jika sudah terinstal, sistem akan memberi tahu bahwa tidak ada paket baru yang diinstal.

```
ddosattacker@ddosattacker:~$ git clone https://github.com/jseidl/GoldenEye.git
fatal: destination path 'GoldenEye' already exists and is not an empty directory.
ddosattacker@ddosattacker:~$ cd GoldenEye/
ddosattacker@ddosattacker:~/GoldenEye$
```

Gambar 4. 49 Cloning Repository GoldenEye dari GitHub

Perintah git clone digunakan untuk mengunduh repository GoldenEye dari GitHub ke direktori lokal. Jika direktori GoldenEye sudah ada, maka akan muncul pesan error bahwa folder tujuan sudah ada. Setelah cloning selesai (atau jika folder sudah

ada), perintah cd GoldenEye/ digunakan untuk berpindah ke dalam direktori tersebut untuk mengeksekusi tool-nya.



Gambar 4. 50 Perintah untuk Menjalankan Serangan

Perintah ini mengeksekusi goldeneye.py untuk mengirim serangan ke endpoint login.php. Opsi -w 100 berarti 100 worker/thread, -s 200 berarti 200 socket per thread, dan -d digunakan untuk menampilkan output debug. Tujuannya adalah membanjiri server target dengan trafik HTTP untuk menguji apakah sistem mitigasi DDoS bisa mendeteksinya dan mengaktifkan puzzle *Proof of Work*

4.4 Pengujian

4.4.1 Deskripsi Pengujian

Pengujian dilakukan untuk mengevaluasi performa sistem mitigasi serangan Distributed Denial of Service (DDoS) yang dirancang menggunakan kombinasi metode *entropy* dan puzzle berbasis *Proof of Work* (PoW) pada server Ubuntu. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa sistem mampu mendeteksi anomali lalu lintas jaringan secara real-time melalui perhitungan *entropy*, serta mengaktifkan mekanisme puzzle PoW untuk membatasi akses dari klien yang terindikasi sebagai attacker, tanpa mengganggu pengguna sah.

Pengujian dilakukan dalam lingkungan lokal menggunakan web server Apache2 yang dihosting pada Ubuntu Server. Endpoint yang menjadi target utama serangan adalah halaman login.php. Serangan disimulasikan dalam tiga skenario bertingkat, masing-masing menggunakan 3, 5, dan 9 virtual machine (VM) attacker untuk mensimulasikan eskalasi skala serangan. Alat yang digunakan untuk menyerang adalah GoldenEye yang bertugas menghasilkan traffic HTTP flood ke server target. Setiap skenario pengujian diamati berdasarkan beberapa aspek: penurunan nilai *entropy* sebagai sinyal awal anomali, waktu aktivasi puzzle setelah *entropy* drop, respons server terhadap tekanan serangan (termasuk penggunaan CPU yang

dipantau dengan htop), dan performa mitigasi berdasarkan metrik evaluasi seperti confusion matrix dengan menggunakan parameter-parameter yang telah ditentukan.

4.4.2 Prosedur Pengujian

4.4.2.1 Skenario Pengujian

1. Pengujian Deteksi Serangan DDoS Menggunakan Metode Entropy

Tujuan dari pengujian ini adalah untuk menguji perfirma *entropy* dalam mendeteksi anomali lalu lintas akibat serangan HTTP Flood dari berbagai jumlah attacker dalam jumlah yang besar. Serangan ini akan dikirimkan dengan tiga variasi skenario berdasarkan jumlah mesin penyerang (VM attacker) yang digunakan, yaitu 3 VM, 5 VM, dan 9 VM. Setiap skenario jumlah VM kemudian diuji kembali menggunakan tiga tingkat beban serangan berbeda, yaitu 100.000 koneksi, 250.000 koneksi, dan 500.000 koneksi total. Beban serangan ini dikonfigurasi dengan mengatur parameter jumlah thread (worker) dan socket per thread pada *tool* GoldenEye, sehingga menghasilkan kombinasi trafik yang meniru kondisi serangan nyata dalam skala kecil hingga besar.

Parameter yang diamati:

- Nilai *entropy* sebelum serangan, saat serangan, dan sesudah mitigasi
- Jumlah IP unik dan total request dalam waktu penyerangan

2. Pengujian Implementasi Mitigasi Puzzle *Proof of Work*

Tujuan dari pengujian ini adalah untuk menguji apakah sistem dapat secara otomatis mengaktifkan puzzle SHA-256 kepada IP mencurigakan setelah terdeteksi *entropy* rendah. Serangan ini akan dikirimkan dengan tiga variasi skenario berdasarkan jumlah mesin penyerang (VM attacker) yang digunakan, yaitu 3 VM, 5 VM, dan 9 VM. Sasaran uji validasi puzzle ini adalah melihat percobaan login dari user asli yang menggunakan IP sah dan percobaan login dari attacker yang IP sudah terdeteksi mencurigakan.

Parameter yang diamati:

- Pengiriman puzzle ke IP yang mencurigakan

- Penyelesaian puzzle oleh attacker dalam rentang waktu yang ditentukan atau gagal login
- Pengguna sah dapat masuk tanpa mendapat puzzle

3. Performa Mitigasi dengan Confusion Matrix

Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan sistem dalam membedakan pengguna sah dan attacker berdasarkan hasil percobaan login ketika mitigasi aktif dan mitigasi tidak aktif. Attacker yang menerima puzzle akan diverifikasi berdasarkan hasil penyelesaian puzzle, di mana sistem akan mencatat apakah puzzle berhasil diselesaikan dalam rentang waktu yang telah ditentukan. pengguna sah juga akan melakukan login saat sistem tidak berada dalam kondisi mitigasi aktif, untuk memastikan bahwa akses tetap dapat dilakukan tanpa gangguan.

Hasil dari setiap percobaan login akan diklasifikasikan ke dalam kategori confusion matrix, yaitu True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). Selanjutnya, dilakukan perhitungan metrik evaluasi yang mencakup *accuracy, precision, recall, dan F1-score*, guna mengukur tingkat performa sistem dalam memitigasi serangan DDoS sekaligus menjaga aksesibilitas bagi pengguna sah.

4. Pengujian Dampak Sistem Mitigasi Terhadap Performa Server

Tujuan dari pengujian ini adalah untuk menilai apakah dampak sistem mitigasi terhadap performa dan stabilitas saat menerima trafik tinggi. Ada tiga mode uji yang akan digunakan saat mitigasi aktif dan mitigasi tidak aktif yaitu:

Tabel 4. 5 Mode Uji

Mode	Deskripsi
Idle	Tanpa lalu lintas
Normal	Akses dari user sah secara bergantian
Serangan	Akses dari attacker + user sah

Parameter yang diamati:

- Monitor pengunaan CPU, memori, dan PPS menggunakan htop
- Response time
  Waktu dari request dikirim → *entropy* dihitung → puzzle dikirim → login dieksekusi.
- Login delay atau waktu yang dibutuhkan user sah untuk berhasil login ketika mitigasi aktif dan mitigasi tidak aktif

4.4.2.2 Pengujian Deteksi Serangan DDoS Menggunakan Metode Entropy

1. Serangan dengan 3 Virtual Machine

Pada skenario ini, tiga buah mesin virtual (VM) digunakan sebagai sumber serangan. Setiap VM disiapkan dengan sistem operasi Ubuntu dan telah diinstal tool serangan GoldenEye, yang digunakan untuk melakukan serangan HTTP flood secara terus-menerus ke endpoint login (login.php) pada server target. Setiap mesin virtual menjalankan percobaan serangan sebanyak tiga kali dengan intensitas berbeda, yaitu 100.000 koneksi, 250.000 koneksi, dan 500.000 koneksi.

Tabel berikut menjelaskan konfigurasi IP address dan nama hostname dari setiap VM attacker yang digunakan dalam pengujian ini:

Tabel 4. 6 IP Address Attacker 3 Virtual Machines

No	Nama Hostname	IP Address
1.	ddosattacker1	192.168.239.76
2.	ddosattacker2	192.168.239.33
3.	ddosattacker3	192.168.239.38

a. Serangan Pertama

Untuk menjalankan total 100.000 koneksi secara paralel menggunakan GoldenEye dari 3 VM, maka diperlukan konfigurasi sebesar 33.333 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 67 thread (-w 67) pada setiap VM, di

mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 33.500 koneksi, sehingga total koneksi gabungan dari 3 VM menghasilkan 100.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 67 -s 500 -d
```

b. Serangan Kedua

Untuk menjalankan total 250.000 koneksi secara paralel menggunakan GoldenEye dari 3 VM, maka diperlukan konfigurasi sebesar 83.333 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 167 thread (-w 167) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 83.500 koneksi, sehingga total koneksi gabungan dari 3 VM menghasilkan 250.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 167 -s 500 -d
```

c. Serangan Ketika

Untuk menjalankan total 500.000 koneksi secara paralel menggunakan GoldenEye dari 3 VM, maka diperlukan konfigurasi sebesar 167.000 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 334 thread (-w 334) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 167.000 koneksi, sehingga total koneksi gabungan dari 3 VM menghasilkan 500.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 334 -s 500 -d
```

2. Serangan dengan 5 Virtual Machine

Pada skenario ini, lima buah mesin virtual (VM) digunakan sebagai sumber serangan. Setiap VM disiapkan dengan sistem operasi Ubuntu dan telah diinstal tool

serangan GoldenEye, yang digunakan untuk melakukan serangan HTTP flood secara terus-menerus ke endpoint login (login.php) pada server target. Setiap mesin virtual menjalankan percobaan serangan sebanyak tiga kali dengan intensitas berbeda, yaitu 100.000 koneksi, 250.000 koneksi, dan 500.000 koneksi.

Tabel berikut menjelaskan konfigurasi IP address dan nama hostname dari setiap VM attacker yang digunakan dalam pengujian ini:

Tabel 4. 7 IP Address Attacker 5 Virtual Machines

No	Nama Hostname	IP Address
1.	ddosattacker1	192.168.239.76
2.	ddosattacker2	192.168.239.33
3.	ddosattacker3	192.168.239.38
4.	ddosattacker4	192.168.239.219
5.	ddosattacker5	192.168.239.41

a. Serangan Pertama

Untuk menjalankan total 100.000 koneksi secara paralel menggunakan GoldenEye dari 5 VM, maka diperlukan konfigurasi sebesar 20.000 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 40 thread (-w 40) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 20.000 koneksi, sehingga total koneksi gabungan dari 5 VM menghasilkan 100.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 40 -s 500 -d
```

b. Serangan Kedua

Untuk menjalankan total 250.000 koneksi secara paralel menggunakan GoldenEye dari 5 VM, maka diperlukan konfigurasi sebesar 50.000 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 100 thread (-w 100) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi

ini, satu VM menghasilkan 50.000 koneksi, sehingga total koneksi gabungan dari 5 VM menghasilkan 250.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 100 -s 500 -d
```

c. Serangan Ketika

Untuk menjalankan total 500.000 koneksi secara paralel menggunakan GoldenEye dari 5 VM, maka diperlukan konfigurasi sebesar 100.000 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 200 thread (-w 200) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 100.000 koneksi, sehingga total koneksi gabungan dari 5 VM menghasilkan 500.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 200 -s 500 -d
```

3. Serangan dengan 9 Virtual Machine

Pada skenario ini, sembilan buah mesin virtual (VM) digunakan sebagai sumber serangan. Setiap VM disiapkan dengan sistem operasi Ubuntu dan telah diinstal tool serangan GoldenEye, yang digunakan untuk melakukan serangan HTTP flood secara terus-menerus ke endpoint login (login.php) pada server target. Setiap mesin virtual menjalankan percobaan serangan sebanyak tiga kali dengan intensitas berbeda, yaitu 100.000 koneksi, 200.000 koneksi, dan 300.000 koneksi.

Tabel berikut menjelaskan konfigurasi IP address dan nama hostname dari setiap VM attacker yang digunakan dalam pengujian ini:

Tabel 4. 8 IP Address Attacker 9 Virtual Machines

No	Nama Hostname	IP Address
1.	ddosattacker1	192.168.239.76
2.	ddosattacker2	192.168.239.33
3.	ddosattacker3	192.168.239.38
4.	ddosattacker4	192.168.239.219
5.	ddosattacker5	192.168.239.41
6.	ddosattacker6	192.168.239.2
7.	ddosattacker7	192.168.239.60
8.	ddosattacker8	192.168.239.61
9.	ddosattacker9	192.168.239.63

a. Serangan Pertama

Untuk menjalankan total 100.000 koneksi secara paralel menggunakan GoldenEye dari 9 VM, maka diperlukan konfigurasi sebesar 11.111 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 23 thread (-w 23) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 11.500 koneksi, sehingga total koneksi gabungan dari 9 VM menghasilkan 100.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 23 -s 500 -d
```

b. Serangan Kedua

Untuk menjalankan total 250.000 koneksi secara paralel menggunakan GoldenEye dari 9 VM, maka diperlukan konfigurasi sebesar 27.778 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 56 thread (-w 56) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini,

satu VM menghasilkan 28.000 koneksi, sehingga total koneksi gabungan dari 9 VM menghasilkan 250.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 56 -s 500 -d
```

c. Serangan Ketika

Untuk menjalankan total 500.000 koneksi secara paralel menggunakan GoldenEye dari 9 VM, maka diperlukan konfigurasi sebesar 55.556 koneksi per VM. Konfigurasi ini dicapai dengan menjalankan 112 thread (-w 112) pada setiap VM, di mana masing-masing thread membuka 500 socket (-s 500). Dengan kombinasi ini, satu VM menghasilkan 56.000 koneksi, sehingga total koneksi gabungan dari 9 VM menghasilkan 500.000 koneksi. Berikut adalah perintah yang digunakan:

```
python3 goldeneye.py http://192.168.239.248/ddos-mitigation/login.php -w 112 -s 500 -d
```

4.4.2.3 Pengujian Mitigasi Puzzle *Proof of Work*

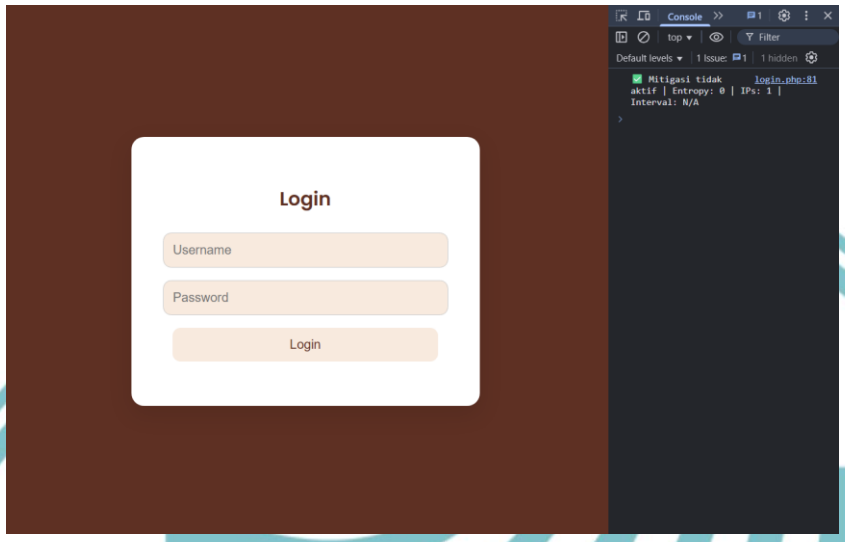
Pada skenario ini akan dilakukan pengujian untuk mengevaluasi perilaku sistem mitigasi DDoS berbasis puzzle *Proof of Work* (PoW) dalam kondisi mitigasi aktif. Pengujian difokuskan pada lima aspek utama, yaitu:

- Apakah penurunan nilai *entropy* berhasil memicu aktivasi mitigasi
- Apakah sistem mengirimkan puzzle hanya kepada IP yang terindikasi mencurigakan
- Apakah puzzle hanya muncul saat mitigasi aktif
- Apakah pengguna sah tetap dapat login tanpa terganggu
- Apakah puzzle yang dikirim berhasil diverifikasi oleh server.

Seluruh pengujian dilakukan secara terkontrol dengan pendekatan manual menggunakan kombinasi attacker dan user sah untuk memastikan validitas hasil.

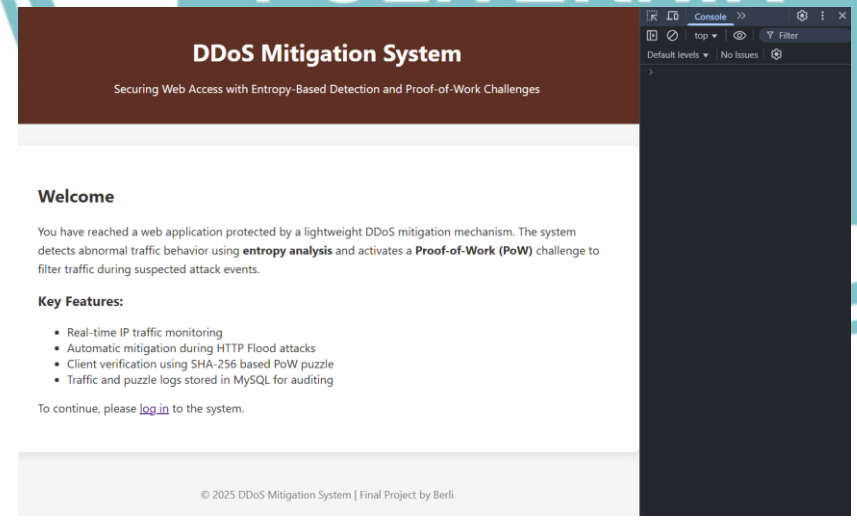
Percobaan pertama dilakukan dengan mengakses sistem melalui browser oleh pengguna sah ketika mekanisme mitigasi tidak aktif. Pada kondisi ini, sistem berada

dalam keadaan normal, di mana tidak terdapat anomali trafik atau serangan yang terdeteksi, sehingga nilai *entropy* tetap berada di atas ambang batas. Dalam skenario ini, user memasukkan kredensial login yang valid ke dalam form login.



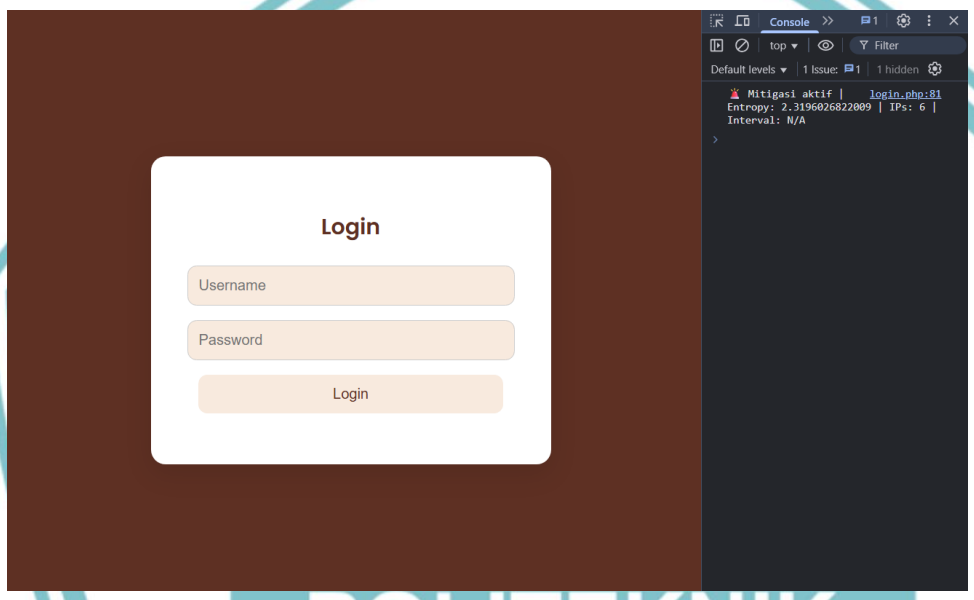
Gambar 4. 51 Login User Sah saat Mitigasi tidak Aktif

Karena tidak ada proses mitigasi yang aktif, server langsung memproses permintaan tanpa memunculkan tantangan puzzle *Proof of Work*. Hasilnya, pengguna sah langsung diarahkan ke halaman landing page tanpa hambatan. Tampilan halaman login serta landing page yang berhasil diakses.



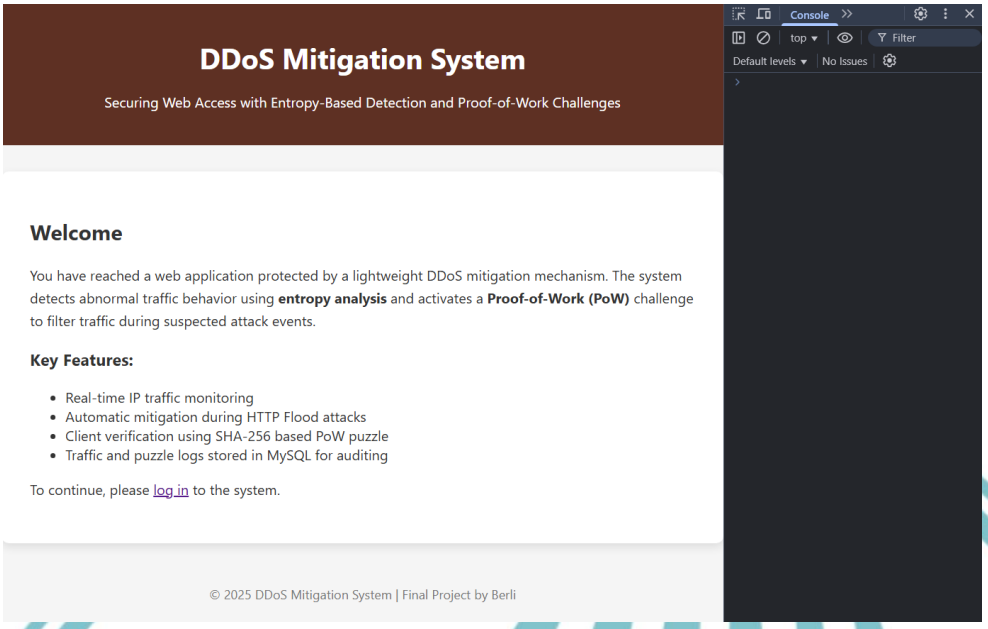
Gambar 4. 52 Tampilan Berhasil masuk Landing Page saat Mitigasi tidak Aktif

Percobaan kedua dilakukan dengan mengakses sistem melalui browser oleh pengguna sah ketika mekanisme mitigasi dalam keadaan aktif. Pada saat pengujian, sistem mendeteksi adanya anomali trafik yang ditunjukkan oleh nilai *entropy* sebesar 2.31, yang mengindikasikan banyaknya permintaan dari lebih dari satu alamat IP secara bersamaan. Nilai *entropy* yang berada di bawah ambang batas memicu sistem untuk mengaktifkan mekanisme mitigasi berbasis puzzle *Proof of Work* (PoW).



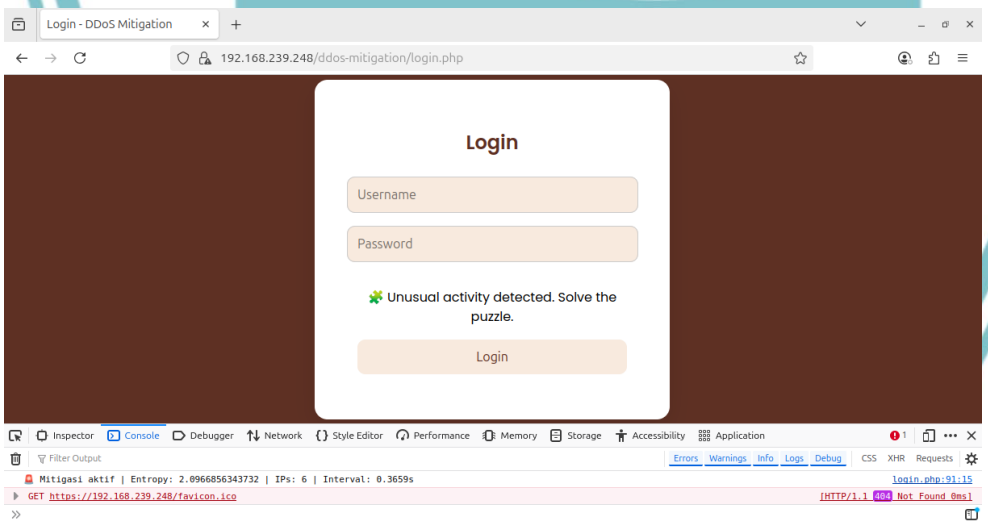
Gambar 4. 53 Login User Sah saat Mitigasi Aktif

Namun, karena IP pengguna sah tidak terdeteksi sebagai mencurigakan, sistem tidak mengirimkan puzzle kepada user tersebut. Proses login tetap berlangsung secara normal, dan setelah kredensial yang dimasukkan dinyatakan valid, pengguna langsung diarahkan ke halaman utama (landing page). Hal ini membuktikan bahwa sistem mitigasi memiliki kemampuan untuk membedakan antara pengguna sah dan sumber serangan, serta tetap memberikan akses layanan secara efisien kepada user yang tidak berbahaya, meskipun mitigasi sedang aktif.



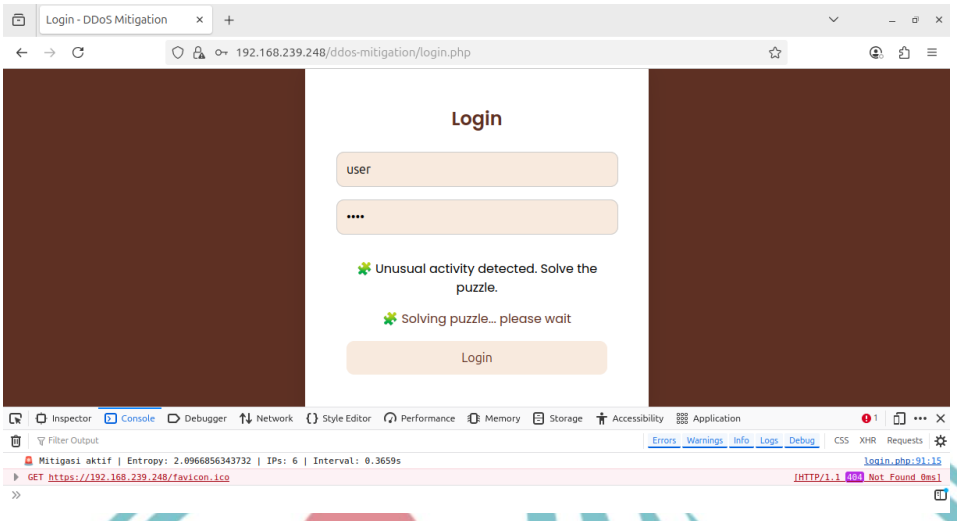
Gambar 4. 54 Tampilan User Sah Berhasil Masuk saat Mitigasi Aktif

Percobaan berikutnya dilakukan dari sisi browser attacker ketika mitigasi aktif, dapat dilihat dari tampilan browser attacker menampilkan pesan bahwa IP tersebut mencurigakan sehinnga untuk mengakses website perlu menyelsaikan puzzle terlebih dahulu.



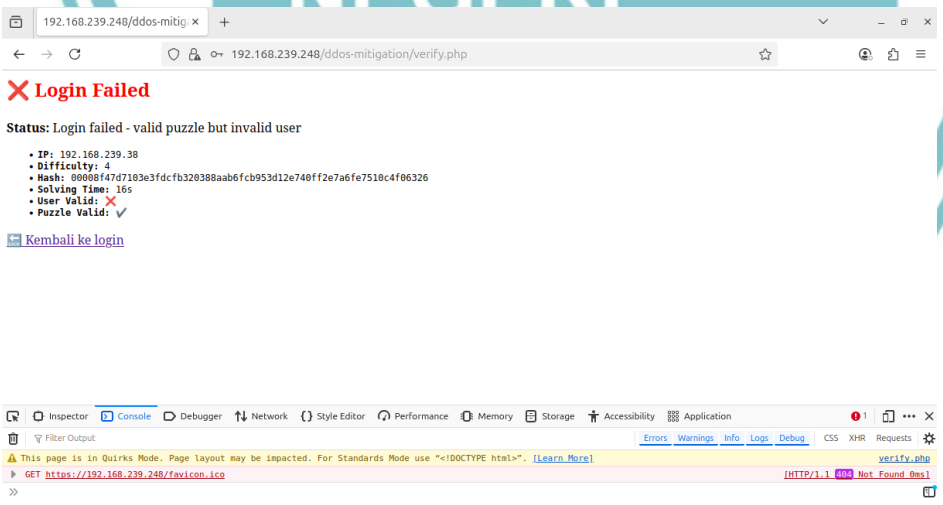
Gambar 4. 55 Tampilan Login Attacker saat Mitigasi Aktif

Berikutnya attacker mencoba memasukkan kredensial yang dibutuhkan yaitu username dan password. Sistem akan langsung memproses puzzle secara otomatis di sisi klien.



Gambar 4. 56 Percobaan Login dengan Puzzle

Pada tahap verifikasi, meskipun puzzle berhasil diselesaikan dengan benar (puzzle valid), namun kredensial login yang digunakan tidak sesuai dengan data pengguna sah yang tersimpan di server. Akibatnya, login dinyatakan gagal dengan status "Login failed - valid puzzle but invalid user". Dari hasil ini dapat disimpulkan bahwa sistem tidak hanya memverifikasi validitas puzzle, tetapi juga tetap melakukan pengecekan kredensial secara menyeluruh sebelum memberikan akses ke sistem. Ini membuktikan bahwa sistem mitigasi mampu menyaring trafik berbahaya dengan dua lapis verifikasi: penyelesaian puzzle dan keabsahan data login.



Gambar 4. 57 Tampilan Attacker yang Gagal Masuk

4.4.2.4 Pengujian Performa Mitigasi dengan Confusion Matrix

Pada pengujian performa sistem mitigasi serangan DDoS berbasis *entropy* dan puzzle *Proof of Work* (PoW) dalam mengklasifikasikan lalu lintas jaringan sebagai pengguna sah atau attacker. Evaluasi dilakukan dengan menggunakan confusion matrix yang mencakup empat kategori klasifikasi: True Positive (TP) adalah saat sistem mendeteksi *trafik attacker* sebagai serangan, False Positive (FP) adalah saat ssitem salah mendeteksi *trafik* pengguna asli sebagai serangan, True Negative (TN) adalah saat sistem mengenali trafik pengguna asli sebagai normal, dan False Negative (FN) adalah saat sistem gagal mendeteksi *trafik attacker* sebagai serangan. Dari data tersebut, dihitung pula metrik evaluasi sistem berupa *accuracy, precision, recall, dan F1-score* sebagai indikator efektivitas deteksi dan respons sistem mitigasi terhadap pola serangan maupun aktivitas pengguna normal. Data diperoleh dari hasil percobaan login yang dilakukan dalam kondisi mitigasi aktif maupun tidak aktif, melibatkan IP attacker (menggunakan browser dan curl) serta IP user sah. Sistem mencatat setiap percobaan berdasarkan status login, pengiriman puzzle, waktu penyelesaian, dan hasil verifikasi. Berikut adalah rumus perhitungan yang akan digunakan:

1. Accuracy - Mengukur seberapa sering sistem mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision - Mengukur proporsi deteksi attacker yang benar (dari semua yang dianggap attacker).

$$Precision = \frac{TP}{TP + FP}$$

3. Recall (Sensitivity / True Positive Rate) - Mengukur seberapa baik sistem menangkap semua attacker.

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score – Rata-rata dari precision dan recall, berguna saat distribusi data tidak seimbang.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Tabel 4. 9 Ketentuan Confusion Matrix

Komponen	Jenis Akses	Hasil	Deskripsi
TP	Attacker (browser), login salah	Ditolak	Puzzle muncul, gagal diselesaikan
FP	User asli, kena puzzle dan gagal	Salah klasifikasi	Puzzle muncul, user tidak bisa login
TN	User asli, login normal	Diterima	Tidak kena puzzle, login berhasil
FN	Attacker (curl), puzzle tidak muncul	Salah lolos	Entropy gagal deteksi, login diterima

4.4.2.5 Pengujian Dampak Sistem Mitigasi Terhadap Performa Server

Pengujian ini dilakukan untuk mengevaluasi sejauh mana dampak sistem mitigasi DDoS berbasis *entropy* dan *Proof of Work* (PoW) terhadap performa dan stabilitas server Ubuntu. Pengujian dilakukan dalam tiga mode: idle, normal, dan mitigation.

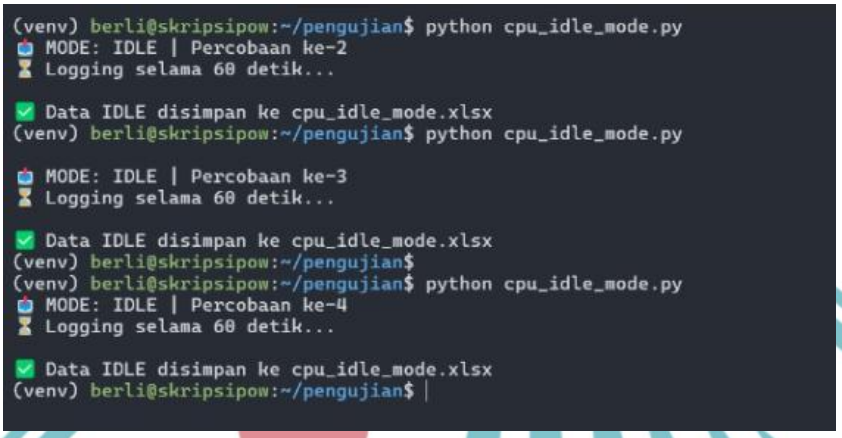
a. Resource usage

Tiga mode pengujian dilakukan dengan skrip Python yang mencatat penggunaan CPU dan memori selama interval waktu tertentu

a. Mode Idle

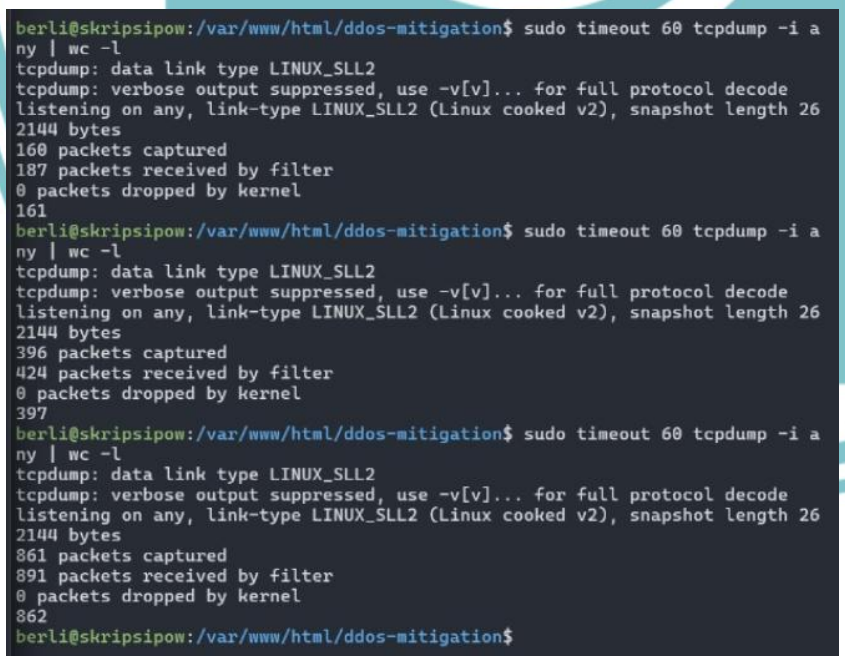
Pengujian performa server dilakukan untuk mengevaluasi dampak sistem mitigasi DDoS berbasis *entropy* dan puzzle *Proof of Work* (PoW) terhadap beban sumber daya dan stabilitas sistem. Gambar 4.58 menunjukkan proses pengujian dalam mode idle, di mana tidak ada lalu lintas yang masuk ke server. Skrip cpu_idle_mode.py dijalankan selama 60 detik, dan data penggunaan CPU

serta memori dicatat ke dalam file cpu_idle_mode.xlsx. Perhitungan dilakukan sebanyak tiga kali.



Gambar 4. 58 Perhitungan CPU dan Memory saat Mode Idle

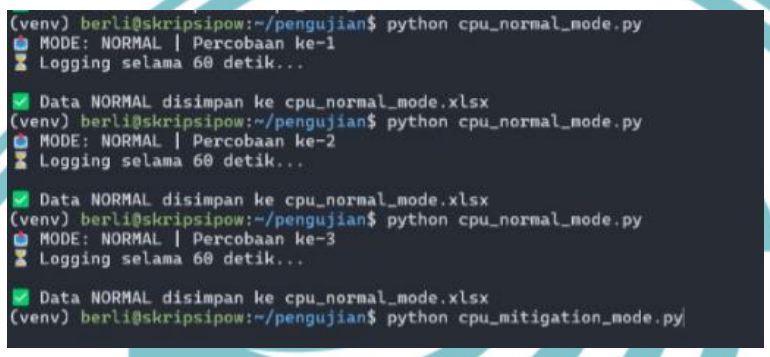
Gambar 4.59 menampilkan hasil monitoring jumlah paket masuk (packet per second) menggunakan perintah tcpdump selama 60 detik pada mode idle dan normal. Jumlah paket tercatat antara 161 hingga 862, yang menunjukkan lalu lintas ringan hingga sedang, tergantung aktivitas user.



Gambar 4. 59 Perhitungan PPS saat Mode Idl

b. Mode Normal

Pengujian performa server dilakukan untuk mengevaluasi dampak sistem mitigasi DDoS berbasis *entropy* dan puzzle *Proof of Work* (PoW) terhadap beban sumber daya dan stabilitas sistem. Gambar 4.60 menunjukkan proses pengujian dalam mode normal, di mana tidak ada lalu lintas yang masuk ke server. Skrip cpu_idle_normal.py dijalankan selama 60 detik, dan data penggunaan CPU serta memori dicatat ke dalam file cpu_idle_normal.xlsx. Perhitungan dilakukan sebanyak tiga kali.



Gambar 4. 60 Perhitungan CPU dan Memory saat Mode Normal

Gambar 4.61 menampilkan hasil monitoring jumlah paket masuk (packet per second) menggunakan perintah tcpdump selama 60 detik pada mode idle dan normal. Jumlah paket tercatat antara 398 hingga 507, yang menunjukkan lalu lintas ringan hingga sedang, tergantung aktivitas user.

```
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 60 tcpdump -i a
ny | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
397 packets captured
423 packets received by filter
0 packets dropped by kernel
398
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 60 tcpdump -i a
ny | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
439 packets captured
471 packets received by filter
0 packets dropped by kernel
440
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 60 tcpdump -i a
ny | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
506 packets captured
540 packets received by filter
0 packets dropped by kernel
507
berli@skripsipow:/var/www/html/ddos-mitigation$
```

Gambar 4. 61 Perhitungan PPS saat Mode Normal

c. Mode Mitigation

Pengujian performa server dilakukan untuk mengevaluasi dampak sistem mitigasi DDoS berbasis *entropy* dan puzzle *Proof of Work* (PoW) terhadap beban sumber daya dan stabilitas sistem. Gambar 4.62 menunjukkan proses pengujian dalam mode mitigation, di mana tidak ada lalu lintas yang masuk ke server. Skrip cpu_idle_mitigation.py dijalankan selama 300 detik, dan data penggunaan CPU serta memori dicatat ke dalam file cpu_idle_mitigation.xlsx. Perhitungan dilakukan sebanyak tiga kali.

```
(venv) berli@skripsipow:~/pengujian$ python cpu_mitigation_mode.py
MODE: MITIGATION | Percobaan ke-1
Logging selama 300 detik...

Data MITIGATION disimpan ke cpu_mitigation_mode.xlsx
(venv) berli@skripsipow:~/pengujian$ python cpu_mitigation_mode.py
MODE: MITIGATION | Percobaan ke-2
Logging selama 300 detik...

Data MITIGATION disimpan ke cpu_mitigation_mode.xlsx
(venv) berli@skripsipow:~/pengujian$ python cpu_mitigation_mode.py
MODE: MITIGATION | Percobaan ke-3
Logging selama 300 detik...

Data MITIGATION disimpan ke cpu_mitigation_mode.xlsx
```

Gambar 4. 62 Perhitungan CPU dan Memory saat Mode Mitigation

Gambar 4.59 menampilkan hasil monitoring jumlah paket masuk (packet per second) menggunakan perintah tcpdump selama 300 detik pada mode idle dan normal. Jumlah paket tercatat antara 56008 hingga 79225, yang menunjukkan lalu lintas ringan hingga sedang, tergantung aktivitas user dan bisa dilihat bahwa aktivitas tersebut sangat tinggi.

```
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 300 tcpdump -i
any | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
56007 packets captured
56238 packets received by filter
0 packets dropped by kernel
56008
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 300 tcpdump -i
any | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
69295 packets captured
69362 packets received by filter
0 packets dropped by kernel
69296
berli@skripsipow:/var/www/html/ddos-mitigation$ sudo timeout 300 tcpdump -i
any | wc -l
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
2144 bytes
79254 packets captured
82632 packets received by filter
3062 packets dropped by kernel
79255
```

Gambar 4. 63 Perhitungan PPS saat Mode Mitigation

b. Response time

Response time didefinisikan sebagai waktu total dari saat user mengirim request login hingga sistem selesai memproses dan mengembalikan hasil (berhasil/gagal login). Pengujian dilakukan menggunakan perintah curl:

```
curl -w "Total time: %{time_total}\n" -o /dev/null
-s http://192.168.239.248/ddos-
mitigation/login.php
```

c. Login delay

Login delay merupakan selisih waktu login user sah saat:

- Tidak ada serangan (normal)
- Saat serangan berlangsung (mitigasi aktif, puzzle dikirim)

Pengukuran dilakukan secara manual menggunakan stopwatch saat user sah melakukan login.

4.4.3 Data Hasil Pengujian

4.4.3.1 Data Hasil Pengujian Deteksi Serangan DDoS Menggunakan Metode Entropy

a. Data Hasil Pengamatan Entropy dari 3 Virtual Machine

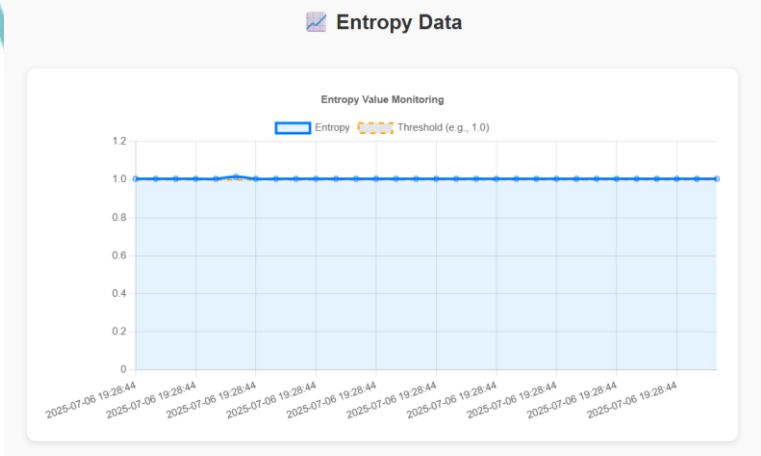
Perubahan nilai *entropy* dari setiap tahapan serangan dapat dilihat pada Tabel 4.9, yang menunjukkan bahwa intensitas dan pola serangan berbeda berdampak langsung pada hasil perhitungan *entropy*. Pada Serangan 1, *entropy* sebelum serangan berada di angka 2.41 dengan status mitigasi tidak aktif. Saat serangan berlangsung, *entropy* turun drastis menjadi 0.054, menandakan trafik sangat terkonsentrasi dan didominasi oleh sedikit IP, sehingga sistem langsung mengaktifkan mitigasi. Setelah serangan mereda, *entropy* naik menjadi 1.29 namun sistem tetap berada dalam status aktif (cooldown) untuk mencegah false negative. Baru setelah 2 menit dan *entropy* stabil di 0.9, mitigasi nonaktif. Pada Serangan 2, *entropy* awal lebih rendah (1.73), dan saat serangan terjadi nilainya justru naik ke 1.3, menunjukkan kemungkinan pola distribusi botnet yang merata. Meskipun demikian, sistem tetap mengaktifkan mitigasi karena jumlah permintaan tinggi dan interval antar request sangat cepat. Setelah serangan berkurang, *entropy* menurun menjadi 0.8, dan kembali meningkat menjadi 1.02 dua menit kemudian mitigasi nonaktif. Sementara itu, pada Serangan 3, *entropy* awal sebesar 2.2 turun ke 1.5 saat serangan dimulai. Sistem tetap mengaktifkan mitigasi karena perilaku trafik yang

agresif. Entropy perlahan naik ke 1.1 setelah serangan mereda, dan mencapai 1.12 setelah 2 menit, sehingga mitigasi dihentikan. Hasil ini menunjukkan bahwa sistem tidak hanya mengandalkan nilai entropy tetap, tetapi juga mempertimbangkan intensitas dan pola distribusi trafik untuk mengaktifkan atau menonaktifkan mitigasi secara adaptif.

Tabel 4. 10 Pengamatan Entropy dari 3 Virtual Machines

Kondisi	Serangan 1	Serangan 2	Serangan 3	Status Mitigasi
Entropy Sebelum Serangan	2.41	1.73	2.2	Tidak Aktif
Entropy Saat Serangan	0.054	1.3	1.5	Aktif
Entropy Setelah Serangan	1.29	0.8	1.1	Aktif (cooldown)
Entropy Setelah 2 Menit	0.9	1.02	1.12	Tidak Aktif

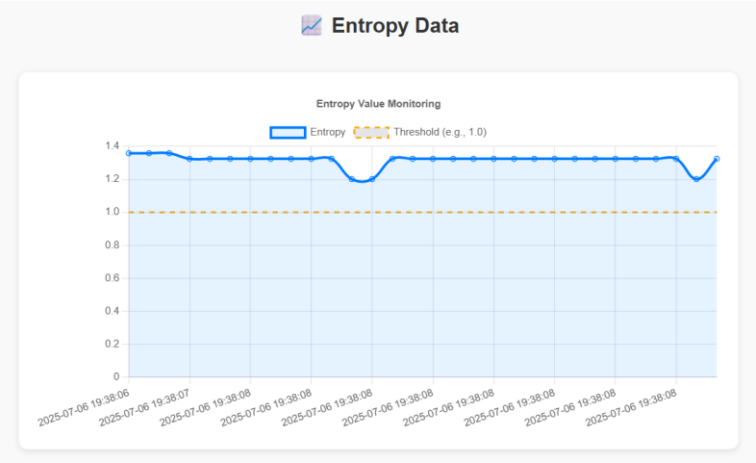
Gambar grafik di bawah menunjukkan monitoring nilai entropy secara real-time. Terlihat bahwa nilai entropy berada stabil di sekitar angka 1.0, yang menunjukkan kondisi trafik jaringan saat ada serangan DDoS dan trafik normal.



Gambar 4. 64 Grafik Perubahan Entropy Serangan Pertama pada 3 VM

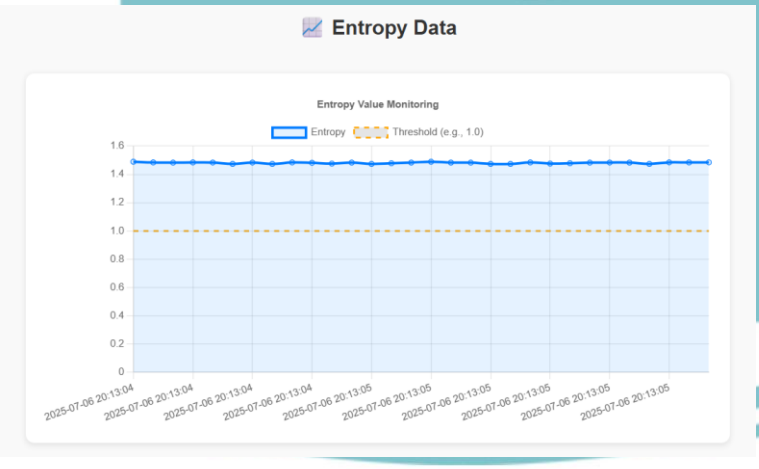
Gambar grafik dibawah menunjukkan adanya kenaikan *entropy* diatas threshold yang mengindikasikan aktivitas tidak wajar atau serangan. Meskipun nilai *entropy* tetap berada

di atas ambang batas 1.0, fluktuasi yang muncul menunjukkan adanya variasi dalam distribusi IP yang dapat menjadi indikasi serangan skala menengah atau awal serangan DDoS.



Gambar 4. 65 Grafik Perubahan Entropy Serangan Kedua pada 3 VM

Gambar grafik dibawah menunjukkan adanya enaikan *entropy* diatas threshold yang bisa berarti server aktif karena ada serangan atau karena trafik normal dari pengguna namun dilihat dari rentang waktu perubahan *entropy* dapat dipastikan *entropy* meningkat karena adanya deteksi serangan DDoS.



Gambar 4. 66 Grafik Perubahan Entropy Serangan Ketiga pada 3 VM

Tabel berikut menunjukkan jumlah total permintaan yang dikirim oleh setiap alamat IP yang tercatat dalam tabel log_request. Tiga IP dengan jumlah request tertinggi diindikasikan sebagai pelaku serangan (attacker), sementara satu IP menunjukkan aktivitas normal.

IP 192.168.238.87 hanya melakukan request dalam batas wajar per waktu, sehingga dianggap sebagai pengguna sah dan tidak dikirim puzzle. Sementara ketiga IP lainnya terindikasi sebagai sumber trafik mencurigakan dan sistem mengaktifkan mitigasi dengan mengirimkan puzzle PoW ke alamat IP tersebut.

Tabel 4. 11 Total Request dari 3 Virtual Machines

No	Alamat IP	Total Request Serangan 1	Total Request Serangan 2	Total Request Serangan 3
1	192.168.239.38	8993	5621	4998
2	192.168.239.76	3533	2293	3529
3	192.168.239.33	1282	4093	4980
4	192.168.239.87	28	13	4

b. Data Hasil Pengamatan Entropy dari 5 Virtual Machine

Perubahan nilai *entropy* dari setiap tahapan serangan dapat diamati pada Tabel 4.9, yang memperlihatkan bahwa variasi intensitas serangan memberikan dampak yang berbeda terhadap hasil perhitungan dan aktivasi mitigasi. Pada Serangan 1, nilai *entropy* sebelum serangan adalah 0, menunjukkan kondisi normal tanpa adanya serangan, sehingga mitigasi belum diaktifkan. Namun saat serangan dimulai, *entropy* melonjak menjadi 1.8 akibat pola serangan yang menyebar namun intens. Setelah serangan berhenti, *entropy* justru menurun drastis ke 0.5, mengindikasikan bahwa sebagian besar trafik berasal dari satu atau sedikit sumber, dan sistem tetap berada dalam status aktif (cooldown). Dua menit kemudian, *entropy* kembali meningkat menjadi 1.03, dan sistem menghentikan mitigasi.

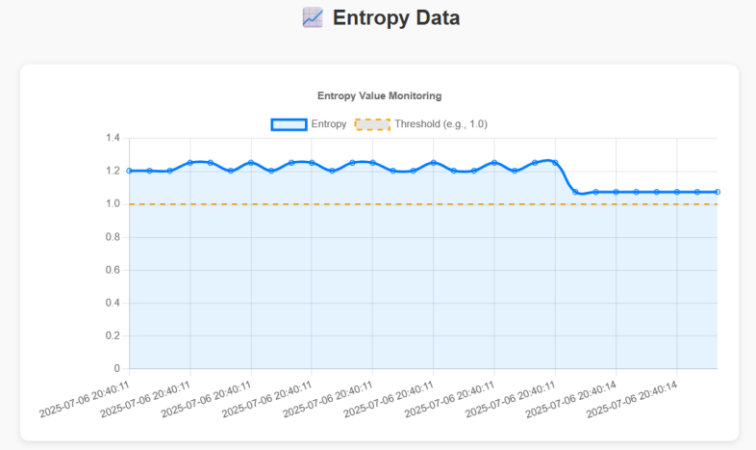
Pada Serangan 2, *entropy* awal berada di 1.23 dan meningkat tajam menjadi 2.23 saat serangan aktif, mengindikasikan serangan botnet dari banyak IP secara bersamaan. Sistem mengaktifkan mitigasi karena parameter lain seperti volume dan kecepatan request mendukung keputusan tersebut. Setelah serangan mereda, *entropy* turun ke 1.8, dan dalam masa cooldown. Setelah dua menit, *entropy* kembali normal di angka 1.2, dan mitigasi dinonaktifkan.

Serangan 3 memiliki *entropy* tertinggi saat serangan berlangsung, yaitu 2.34. Nilai awal sebelum serangan belum tercatat, namun setelah serangan usai, *entropy* turun drastis ke 0.59 dan menunjukkan pola trafik yang kembali terkonsentrasi. Dua menit setelahnya, *entropy* naik menjadi 1.29 dan mitigasi dihentikan. Secara keseluruhan, hasil ini menunjukkan bahwa sistem tidak hanya merespons nilai *entropy* tetap, tetapi juga memperhitungkan perubahan dinamis dan parameter perilaku trafik lainnya untuk mengelola status mitigasi secara adaptif dan efektif.

Tabel 4. 12 Pengamatan Entropy dari 5 Virtual Machines

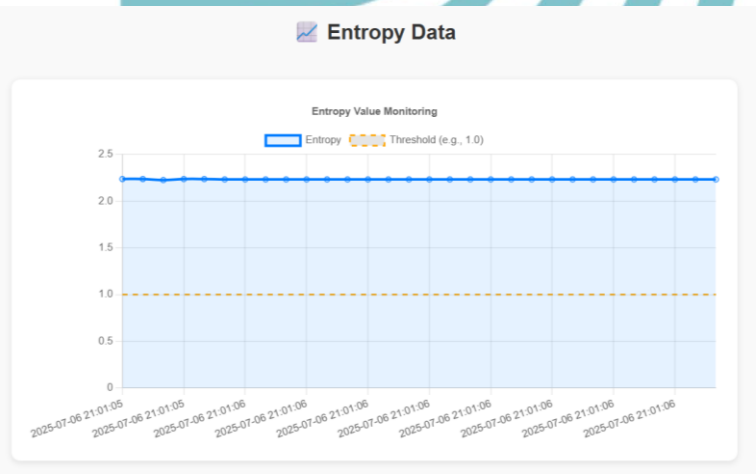
Kondisi	Serangan 1	Serangan 2	Serangan 3	Status Mitigasi
Entropy Sebelum Serangan	0	1.23	1.12	Tidak Aktif
Entropy Saat Serangan	1.8	2.23	2.34	Aktif
Entropy Setelah Serangan	0.5	1.8	0.59	Aktif (cooldown)
Entropy Setelah 2 Menit	1.03	1.2	1.29	Tidak Aktif

Gambar tersebut menunjukkan grafik monitoring nilai *entropy* secara real-time. Terlihat adanya penurunan signifikan di tengah grafik, yang menandakan sistem mendeteksi trafik tidak normal dan mengaktifkan mitigasi karena nilai *entropy* turun.



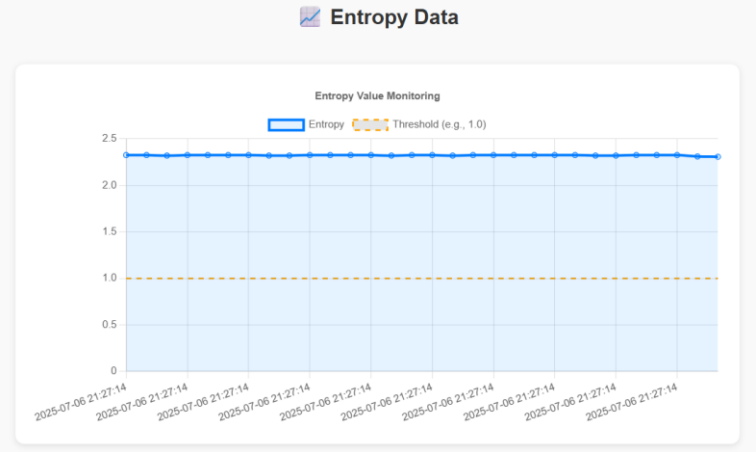
Gambar 4. 67 Grafik Perubahan Entropy Serangan Pertama pada 5 VM

Pada gambar grafik *entropy* dibawah ini menunjukkan bahwa intensitas request yang masuk meningkat nilai perhitungan *entropy* juga mengalami peningkatan sehingga perlu dianalisa lebih lanjut apakah trafik tersebut adalah serangan DDoS.



Gambar 4. 68 Grafik Perubahan Entropy Serangan Kedua pada 5 VM

Grafik dibawah ini menunjukkan perhitungan real time *entropy* nilainya yang berada diatas threshold dan permintaan yang banyak dapat mengindikasikan bahwa website sedang dalam mode mitigasi aktif.



Gambar 4. 69 Grafik Perubahan Entropy Serangan Ketiga pada 5 VM

Tabel berikut menunjukkan jumlah total permintaan yang dikirim oleh setiap alamat IP yang tercatat dalam tabel log_request. 5 IP dengan jumlah request tertinggi diindikasikan sebagai pelaku serangan (attacker), sementara satu IP menunjukkan aktivitas normal.

IP 192.168.238.87 hanya melakukan request dalam batas wajar per waktu, sehingga dianggap sebagai pengguna sah dan tidak dikirim puzzle. Sementara ketiga IP lainnya terindikasi sebagai sumber trafik mencurigakan dan sistem mengaktifkan mitigasi dengan mengirimkan puzzle PoW ke alamat IP tersebut.

Tabel 4. 13 Total Request dari 5 Virtual Machines

No	Alamat IP	Total Request Serangan 1	Total Request Serangan 2	Total Request Serangan 3
1	192.168.239.38	1987	3549	4470
2	192.168.239.76	2824	6701	2500
3	192.168.239.33	1694	1922	3053
4	192.168.239.87	4	13	20
5.	192.168.239.41	4279	2848	4360
6.	192.168.239.219	3270	4250	2332

c. Data Hasil Pengamatan Entropy dari 9 Virtual Machine

Perubahan nilai *entropy* dari setiap tahapan serangan dapat diamati pada Tabel 4.9, yang memperlihatkan bahwa variasi intensitas serangan memberikan dampak yang berbeda terhadap hasil perhitungan *entropy* dan aktivasi mitigasi. Pada Serangan 1, nilai *entropy* sebelum serangan adalah 1.12, menunjukkan kondisi normal tanpa adanya serangan, sehingga mitigasi belum diaktifkan. Namun saat serangan dimulai, *entropy* naik menjadi 1.9 akibat pola serangan yang menyebar namun intens. Setelah serangan berhenti, nilai entrop turun ke 1.03, mengindikasikan bahwa sebagian besar trafik berasal dari satu atau sedikit sumber, dan sistem tetap berada dalam status aktif (cooldown). Dua menit kemudian, *entropy* kembali meningkat menjadi 0.9, dan sistem menghentikan mitigasi.

Pada Serangan 2, *entropy* awal berada di 2.37 dan turun menjadi 2.23 saat serangan aktif, mengindikasikan serangan botnet dari banyak IP secara bersamaan yang menyebabkan trafik terlihat merata. Sistem mitigasi aktif karena melihat parameter lain seperti volume dan kecepatan request. Setelah serangan mereda, *entropy* turun ke 1.8, dan dalam masa cooldown. Setelah dua menit, *entropy* kembali normal di angka 1.2, dan mitigasi dinonaktifkan.

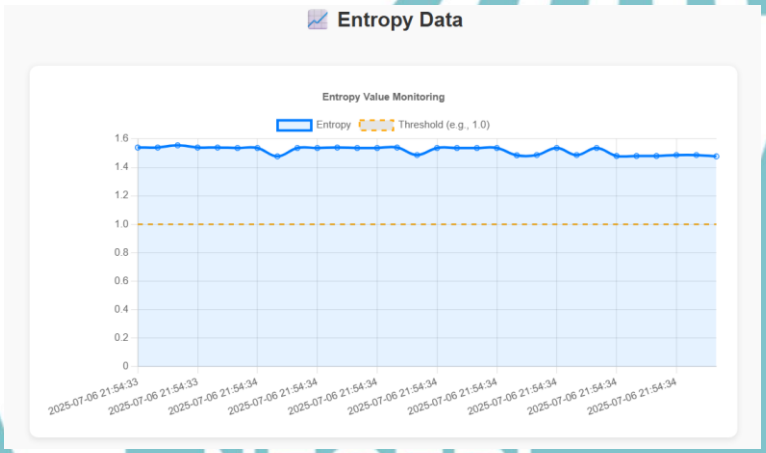
Serangan 3 memiliki *entropy* tertinggi saat serangan berlangsung, yaitu 1.3. Nilai awal sebelum serangan belum tercatat, namun setelah serangan usai, *entropy* naik ke 2.28 dan menunjukkan pola trafik yang kembali terkonsentrasi. Dua menit setelahnya, *entropy* berkurang 1.24 dan mitigasi dihentikan. Secara keseluruhan, hasil ini menunjukkan bahwa sistem tidak hanya merespons nilai *entropy* absolut, tetapi juga memperhitungkan perubahan dinamis dan parameter perilaku trafik lainnya untuk mengelola status mitigasi secara adaptif dan efektif.

Tabel 4. 14 Pengamatan Entropy dari 9 Virtual Machines

Kondisi	Serangan 1	Serangan 2	Serangan 3	Status Mitigasi
Entropy Sebelum Serangan	1.12	2.37	1.3	Tidak Aktif

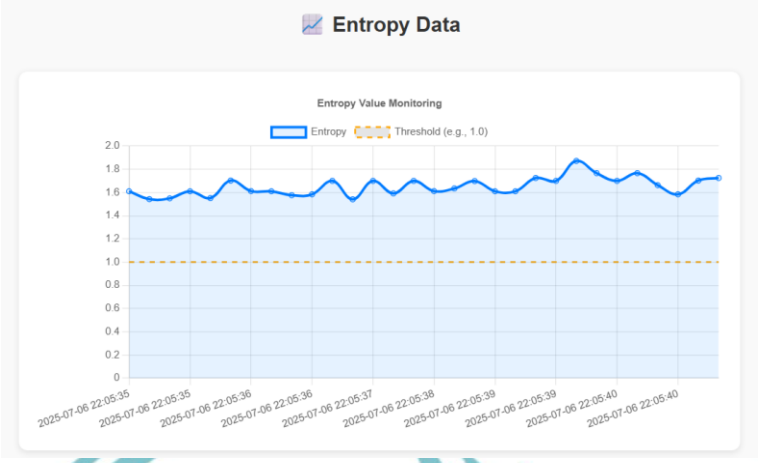
Entropy Saat Serangan	1.9	1.8	2.28	Aktif
Entropy Setelah Serangan	1.03	0.92	1.24	Aktif (cooldown)
Entropy Setelah 2 Menit	0.9	0.74	0	Tidak Aktif

Gambar di bawah menunjukkan grafik monitoring nilai *entropy* selama periode waktu tertentu. Terlihat bahwa nilai *entropy* berada di kisaran 1.45 hingga 1.55, berada di atas ambang batas threshold 1.0. Jika dilihat dari data keseluruhan trafik tersebut merupakan serangan DDoS.



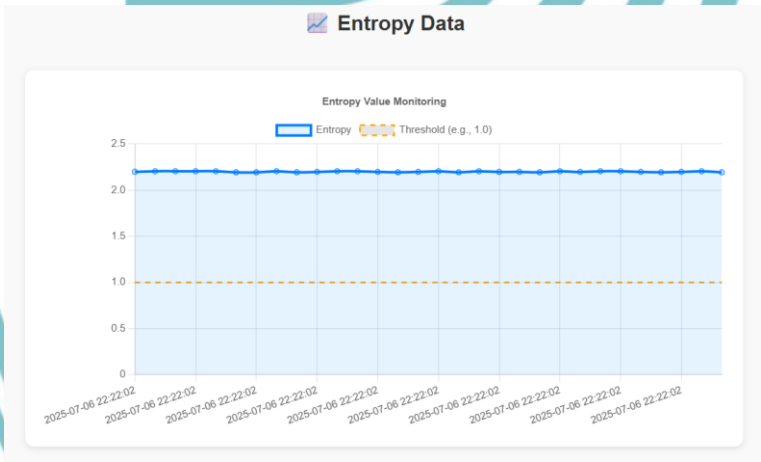
Gambar 4. 70 Grafik Perubahan Entropy Serangan Pertama pada 9 VM

Gambar di bawah memperlihatkan grafik perubahan nilai *entropy* saat terjadi aktivitas lalu lintas yang bervariasi. Terlihat fluktuasi nilai *entropy* antara 1.55 hingga 1.85, tetap berada di atas threshold 1.0, yang mengindikasikan adanya sebaran IP yang merata, kemungkinan berasal dari botnet dengan distribusi luas. Meskipun *entropy* tinggi, sistem dapat tetap mengaktifkan mitigasi jika nilai interval dan jumlah permintaan dari masing-masing IP menunjukkan perilaku mencurigakan.



Gambar 4. 71 Grafik Perubahan Entropy Serangan Kedua pada 9 VM

Gambar di bawah menunjukkan adanya pola distribusi IP yang merata dilihat dari trafik dengan nilai konstan dan diatas threshold, namun perlu juga melihat data keseluruhan untuk memastikan apakah ada serangan DDoS.



Gambar 4. 72 Grafik Perubahan Entropy Serangan Ketiga pada 9 VM

Tabel berikut menunjukkan jumlah total permintaan yang dikirim oleh setiap alamat IP yang tercatat dalam tabel log_request. 9 IP dengan jumlah request tertinggi diindikasikan sebagai pelaku serangan (attacker), sementara satu IP menunjukkan aktivitas normal.

IP 192.168.238.87 hanya melakukan request dalam batas wajar per waktu, sehingga dianggap sebagai pengguna sah dan tidak dikirim puzzle. Sementara ketiga IP lainnya terindikasi sebagai sumber trafik mencurigakan dan sistem mengaktifkan mitigasi dengan mengirimkan puzzle PoW ke alamat IP tersebut.

Tabel 4. 15 Total Request dari 9 Virtual Machines

No	Alamat IP	Total Request Serangan 1	Total Request Serangan 2	Total Request Serangan 3
1	192.168.239.38	3796	5266	2558
2	192.168.239.76	83	1663	3597
3	192.168.239.33	4885	6500	3370
4	192.168.239.87	22	50	18
5.	192.168.239.219	972	11888	6393
6.	192.168.239.41	2557	5046	4353
7.	192.168.239.2	1680	3054	2667
8.	192.168.239.60	1548	3982	3330
9.	192.168.239.61	2902	4123	2148
10.	192.168.239.63	1536	1040	5394

d. Perbandingan Hasil Deteksi DDoS dengan Penelitian Terkait

Pada bagian ini, dilakukan perbandingan hasil sistem deteksi DDoS yang dikembangkan dalam penelitian ini dengan penelitian terkait. Sistem ini menggunakan metode entropy berbasis distribusi IP sumber, dengan proses deteksi yang diaktifkan secara real-time ketika nilai entropy menurun atau terjadi perilaku trafik mencurigakan. Hasil pengujian menunjukkan sistem berhasil mendeteksi seluruh skenario serangan pada 3, 5, dan 9 VM attacker, dengan tingkat keberhasilan deteksi 100% dari total 9 skenario pengujian.

Sistem ini kemudian dibandingkan dengan dua penelitian terdahulu, yaitu penelitian oleh Teguh Kurniawan (2023) dan Patil et al. (2023). Pada penelitian Teguh Kurniawan, metode multi-entropy digunakan sebagai fitur input untuk model Random Forest pada arsitektur Software Defined Network (SDN), menghasilkan akurasi deteksi sebesar 99,43%. Sementara itu, pada penelitian Patil et al. (2023),

metode entropy digunakan bersama CNN dalam infrastruktur cloud OpenStack, namun akurasi deteksi tidak dijelaskan secara rinci dalam publikasi.

Tabel berikut menyajikan perbandingan hasil sistem deteksi DDoS antara penelitian ini dengan penelitian terdahulu:

Tabel 4. 16 Perbandingan Hasil Deteksi DDoS

Aspek	Penelitian Ini (Entropy + Threshold)	Teguh Kurniawan (2023)	Patil et al. (2023)
Lingkungan	Web server (Apache di Ubuntu)	SDN Controller	OpenStack Controller
Parameter Deteksi	Nilai entropy + perilaku trafik	Multi-entropy + ML	Entropy + CNN
Persentase Deteksi DDoS	100% (9 skenario)	99,43%	Tidak dilaporkan
Fitur	Source IP entropy	3 entropy features	Destination entropy + fitur trafik
Keunggulan	Real-time, sederhana, tidak perlu dataset	Akurasi tinggi, adaptif	Dapat diterapkan di cloud-scale
Kelemahan	Threshold statis, deteksi berbasis IP	Memerlukan training dan dataset	Setup kompleks, tidak cocok untuk server kecil

Dari tabel di atas dapat disimpulkan bahwa sistem deteksi berbasis entropy dengan threshold pada penelitian ini memiliki keunggulan dari sisi kesederhanaan dan kemudahan implementasi di server aplikasi web tanpa memerlukan infrastruktur tambahan. Dibandingkan dengan metode machine learning pada penelitian Teguh Kurniawan yang membutuhkan dataset pelatihan, dan metode berbasis cloud pada penelitian Patil et al., sistem ini lebih praktis untuk diterapkan di lingkungan server lokal dengan skala kecil hingga menengah.

4.4.3.2 Data Hasil Pengujian Mitigasi Puzzle *Proof of Work*

a. Pengujian 1

Hasil pengujian pertama menunjukkan bahwa bahwa pengguna dengan IP address 192.168.239.87 dapat login dengan normal dalam kondisi sistem sedang mitigasi tidak aktif maupun ketika mitigasi aktif. Kondisi ini mengindikasikan bahwa sistem mitigasi tidak aktif pada saat permintaan login dilakukan, sehingga pengguna sah tidak dikenakan proses mitigasi dan dapat langsung mengakses sistem.

IP Address	: 192.168.239.87
Difficulty	: 0
Base	: 0
Nonce	: 0
Hash Prefix	: 0
Waktu Penyelesaian	: 0
Status Puzzle	: 0
Status Login	: Login Success

b. Pengujian 2

Pada percobaan login kedua, sistem mendeteksi aktivitas tidak normal dari IP address 192.168.239.248 sehingga ketika mitigasi aktif IP tersebut diminta untuk menyelesaikan puzzle. Puzzle yang diberikan memiliki tingkat kesulitan sebesar 5, yang mensyaratkan hash hasil gabungan antara nilai base (10fbqc9) dan nonce (150514) menghasilkan awalan lima digit nol. Pengguna berhasil menyelesaikan puzzle tersebut dalam waktu 50 detik, dan hasil hash yang diperoleh (00000ded3fc3db02fafa639bb3ae07c1a8f1e1b9406267bcb9a99fc97649ce30) dinyatakan valid oleh sistem. Meskipun puzzle berhasil diselesaikan, proses login tetap gagal karena kredensial yang dimasukkan tidak sesuai, sebagaimana ditunjukkan oleh status login "Invalid User".

IP Address	: 192.168.239.248
Difficulty	: 5
Base	: 10fbqc9
Nonce	: 150514

Hash Prefix	:00000ded3fc3db02fafa639bb3ae07c1a8f1e1b94062 67bcb9a99fc97649ce30
Waktu Penyelesaian	: 50 detik
Status Puzzle	: Valid
Status Login	: Invalid User

c. Pengujian 3

Pada percobaan login ketiga, masih dari IP address 192.168.239.248 yang sama sehingga ketika mitigasi aktif IP tersebut diminta untuk menyelesaikan puzzle. Puzzle yang diberikan memiliki tingkat kesulitan sebesar 4, lebih rendah dibandingkan sebelumnya, disebabkan oleh nilai *entropy* yang dihitung secara dinamis setiap 30 detik berdasarkan distribusi lalu lintas IP. Puzzle yang diberikan dani nilai base (cndg18) dan nonce (49630) menghasilkan awalan empat digit nol. Pengguna berhasil menyelesaikan puzzle tersebut dalam waktu 50 detik, dan hasil hash yang diperoleh

(0000aaa258536fc94d590f22969758f9661254f7974f92429de114ff8be087ad) dinyatakan valid oleh sistem. Meskipun puzzle berhasil diselesaikan, proses login tetap gagal karena kredensial yang dimasukkan tidak sesuai, sebagaimana ditunjukkan oleh status login "Invalid User".

IP Address	: 192.168.239.248
Difficulty	: 4
Base	: cndg18
Nonce	: 49630
Hash Prefix	:0000aaa258536fc94d590f22969758f9661254f7974f92 429de114ff8be087ad
Waktu Penyelesaian	: 7 detik
Status Puzzle	: Valid
Status Login	: Invalid User

d. Perbandingan Metode Mitigasi DDoS dengan Penelitian Terkait

Setelah sistem mendeteksi adanya serangan DDoS berbasis HTTP Flood menggunakan metode entropy, mitigasi dilakukan dengan menerapkan Puzzle Proof-of-Work (PoW) berbasis SHA-256 sebagai metode challenge-response. Metode ini membebani klien yang dianggap mencurigakan dengan teka-teki kriptografi yang harus diselesaikan sebelum melanjutkan proses login. Berdasarkan hasil pengujian, metode PoW yang dikembangkan dalam penelitian ini berhasil membatasi akses bot otomatis tanpa membebani pengguna asli.

Sebagai pembanding, metode UNI-CAPTCHA yang dikembangkan oleh penelitian (Süzen, 2021) "UNI-CAPTCHA: A Novel Robust and Dynamic User-Non-Interaction CAPTCHA Model Based on Hybrid biLSTM+Softmax" menggunakan pendekatan Machine Learning (biLSTM + Softmax) untuk menghasilkan CAPTCHA yang lebih dinamis dan tahan terhadap bot canggih. UNI-CAPTCHA memungkinkan deteksi bot melalui pola interaksi pengguna secara otomatis, tanpa interaksi eksplisit seperti pengisian teks atau pemilihan gambar oleh pengguna.

Tabel berikut menyajikan perbandingan metode mitigasi antara sistem ini dan UNI-CAPTCHA:

Tabel 4. 17 Perbandingan Metode Mitigasi DDoS

Aspek	Puzzle PoW	UNI-CAPTCHA
Level Mitigasi	Layer Aplikasi (Login HTTP)	Layer Aplikasi
Metode Challenge	SHA-256 Hash Prefix Puzzle (PoW)	Passive CAPTCHA (non-interaction, biLSTM+Softmax)
Adaptasi Challenge	Ya (berdasarkan entropy dinamis)	Ya (berbasis pola interaksi pengguna)
Target	Bot otomatis (HTTP Flood)	Bot canggih (AI-based bots)

Kelebihan	Beban di sisi attacker, tidak membebani user asli	Tidak mengganggu pengguna (noninteraction)
Kekurangan	Puzzle statis pada kondisi tertentu	Komputasi lebih berat di server
Mekanisme	Puzzle dikirim setelah entropy drop	CAPTCHA pasif yang berjalan di background
Efektivitas	Memblokir bot skrip dan curl	Deteksi otomatis tanpa input manual

UNI-CAPTCHA menawarkan metode mitigasi DDoS yang lebih canggih pada sisi aplikasi, dengan mengimplementasikan Machine Learning untuk mengenali perilaku pengguna tanpa memerlukan interaksi eksplisit. Hal ini sangat efektif terhadap bot canggih yang mampu menyelesaikan CAPTCHA standar.

Namun, metode Puzzle PoW yang dikembangkan dalam penelitian ini tetap memiliki keunggulan utama yaitu:

- Beban komputasi dibebankan ke sisi klien (attacker).
- Tidak memerlukan pemrosesan kompleks di sisi server.
- Tidak membebani pengguna asli dengan tantangan apapun, kecuali IP-nya dianggap mencurigakan.

Dengan demikian, Puzzle PoW lebih ringan dan lebih praktis untuk server aplikasi kecil-menengah, sedangkan UNI-CAPTCHA cocok untuk sistem besar dengan infrastruktur server lebih kuat.

4.4.3.3 Data Hasil Pengujian Performa Mitigasi dengan Confusion Matrix

Data hasil pengujian sistem pada skenario 3 VM, 5 VM, dan 9 VM attacker, sistem mitigasi DDoS berbasis entropy dan puzzle Proof of Work (PoW) menunjukkan kinerja deteksi dan mitigasi yang sangat baik. Nilai recall mencapai 100% pada seluruh skenario, menunjukkan sistem tidak pernah gagal dalam mendeteksi serangan (FN=0).

a. Data Hasil Pengujian Confusion Matrix dengan 3 VM Attacker

Tabel 4. 18 Hasil Confusion Matrix 3 VM

Klasifikasi	Jumlah
True Positive (TP)	192
False Positive (FP)	1
True Negative (TN)	250
False Negative (FN)	0

1. Accuracy - Mengukur seberapa sering sistem mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{192 + 250}{192 + 250 + 1 + 0} = 99.77\%$$

2. Precision - Mengukur proporsi deteksi attacker yang benar (dari semua yang dianggap attacker).

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{192}{192 +} = 98.48\%$$

3. Recall (Sensitivity / True Positive Rate) - Mengukur seberapa baik sistem menangkap semua attacker.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{192}{192 + 0} = 100\%$$

4. F1-Score – Rata-rata dari precision dan recall, berguna saat distribusi data tidak seimbang.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 - Score = 2 \times \frac{0.9848 \times 1}{0.9848 + 1} = 99.74\%$$

Berdasarkan hasil pengujian, sistem mencatat 192 True Positive (TP), di mana trafik dari attacker berhasil dikenali dan diblokir secara benar. Selain itu, terdapat 195 True Negative (TN), menunjukkan sistem berhasil mengenali trafik dari pengguna sah tanpa salah klasifikasi. Terdapat 3 False Positive (FP), yaitu kasus di mana pengguna asli salah dikenali sebagai attacker dan diblokir oleh sistem. Tidak ditemukan False Negative (FN) pada pengujian ini, yang berarti seluruh trafik attacker berhasil dideteksi tanpa ada serangan yang lolos.

Dari data tersebut, dihitung metrik evaluasi sebagai berikut:

- Accuracy sebesar 99,77%, menunjukkan sistem secara keseluruhan mampu mengklasifikasikan trafik dengan benar pada hampir seluruh permintaan.
- Precision sebesar 99,48%, menandakan bahwa dari seluruh trafik yang diklasifikasikan sebagai attacker, sebanyak 99,48% merupakan benar-benar attacker.
- Recall mencapai 100%, menunjukkan sistem berhasil menangkap semua trafik attacker tanpa satupun yang lolos (FN=0).
- F1-Score sebesar 99,74%, menunjukkan keseimbangan antara precision dan recall dalam klasifikasi trafik.

Hasil pengujian ini menunjukkan bahwa sistem mitigasi yang dikembangkan mampu mendeteksi dan merespons serangan DDoS dengan tingkat akurasi dan efektivitas yang sangat tinggi.

b. Data Hasil Pengujian Confusion Matrix dengan 5 VM Attacker

Tabel 4. 19 Hasil Confusion Matrix 5 VM

Klasifikasi	Jumlah
True Positive (TP)	217
False Positive (FP)	3
True Negative (TN)	250
False Negative (FN)	0

1. Accuracy - Mengukur seberapa sering sistem mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{217 + 250}{217 + 250 + 3 + 0} = 99.36\%$$

2. Precision - Mengukur proporsi deteksi attacker yang benar (dari semua yang dianggap attacker).

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{217}{217 + 3} = 98.64\%$$

3. Recall (Sensitivity / True Positive Rate) - Mengukur seberapa baik sistem menangkap semua attacker.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{217}{217 + 0} = 100\%$$

4. F1-Score – Rata-rata dari precision dan recall, berguna saat distribusi data tidak seimbang.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 - Score = 2 \times \frac{0.9864 \times 1}{0.9864 + 1} = 99.31\%$$

Berdasarkan hasil pengujian, sistem mencatat 193 True Positive (TP), yaitu trafik attacker yang berhasil dikenali dan diblokir secara benar oleh sistem. Selain itu, terdapat 200 True Negative (TN), di mana trafik dari pengguna asli berhasil dikenali sebagai trafik normal dan tidak diblokir oleh sistem. Sistem mencatat 5 False Positive (FP), yaitu kasus kesalahan deteksi di mana pengguna asli salah

diklasifikasikan sebagai attacker dan diblokir. Tidak ditemukan False Negative (FN) pada pengujian ini, menunjukkan seluruh trafik attacker berhasil terdeteksi tanpa ada serangan yang lolos.

Dari hasil pengujian tersebut, dihitung metrik evaluasi sebagai berikut:

- Accuracy sebesar 99.36%, menunjukkan sistem mampu mengklasifikasikan trafik dengan benar pada sebagian besar request.
- Precision sebesar 98.64%, menunjukkan dari seluruh trafik yang diklasifikasikan sebagai attacker, sebanyak 98.64% adalah attacker sesungguhnya.
- Recall mencapai 100%, menunjukkan bahwa seluruh trafik serangan berhasil dideteksi oleh sistem (tidak ada false negative).
- F1-Score sebesar 99.31%, menandakan keseimbangan optimal antara kemampuan sistem dalam mendeteksi serangan dan meminimalkan kesalahan klasifikasi terhadap pengguna asli.

Hasil evaluasi ini menunjukkan bahwa sistem mitigasi yang dikembangkan memiliki performa sangat baik dalam menangani serangan DDoS pada skenario 5 VM attacker.

c. Data Hasil Pengujian Confusion Matrix dengan 9 VM Attacker

Tabel 4. 20 Hasil Confusion Matrix 9 VM

Klasifikasi	Jumlah
True Positive (TP)	200
False Positive (FP)	3
True Negative (TN)	250
False Negative (FN)	0

1. Accuracy - Mengukur seberapa sering sistem mengklasifikasikan dengan benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{200 + 250}{200 + 250 + 3 + 0} = 99.34\%$$

2. Precision - Mengukur proporsi deteksi attacker yang benar (dari semua yang dianggap attacker).

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{200}{200 + 3} = 98.52\%$$

3. Recall (Sensitivity / True Positive Rate) - Mengukur seberapa baik sistem menangkap semua attacker.

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{200}{200 + 0} = 100\%$$

4. F1-Score – Rata-rata dari precision dan recall, berguna saat distribusi data tidak seimbang.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1 - Score = 2 \times \frac{0.9852 \times 1}{0.9852 + 1} = 99.25\%$$

Berdasarkan hasil pengujian, sistem mencatat 200 True Positive (TP), yang menunjukkan jumlah trafik attacker yang berhasil dikenali dan diblokir secara benar oleh sistem. Selain itu, terdapat 250 True Negative (TN), di mana trafik pengguna asli diklasifikasikan secara benar sebagai trafik normal dan diperbolehkan mengakses layanan. Namun, terdapat 3 False Positive (FP), yaitu jumlah pengguna asli yang salah diklasifikasikan sebagai attacker dan diblokir oleh sistem. Tidak ditemukan False Negative (FN), menunjukkan seluruh trafik attacker berhasil terdeteksi tanpa ada serangan yang lolos.

Dari data tersebut diperoleh hasil perhitungan sebagai berikut:

- Accuracy sebesar 99.34%, menunjukkan bahwa sistem dapat mengklasifikasikan trafik dengan benar pada sebagian besar kasus.
- Precision sebesar 98.52%, menunjukkan bahwa dari semua trafik yang diklasifikasikan sebagai attacker, sebanyak 98.52% adalah attacker yang benar.
- Recall tetap sempurna sebesar 100%, menandakan sistem tidak membiarkan satupun trafik serangan lolos tanpa deteksi.
- F1-Score sebesar 99.25%, yang menunjukkan keseimbangan antara precision dan recall dalam klasifikasi.

Dibandingkan dengan pengujian pada 3 VM dan 5 VM attacker, jumlah false positive meningkat pada skenario ini, menunjukkan bahwa ketika jumlah attacker bertambah, sistem lebih sulit membedakan antara trafik attacker dan pengguna asli. Namun demikian, tidak adanya false negative membuktikan bahwa sistem tetap sangat efektif dalam mencegah seluruh serangan DDoS masuk.

4.4.3.4 Data Hasil Pengujian Dampak Sistem Mitigasi Terhadap Performa Server

1. Resourse Usage

a. Mode Idle

Pada mode ini, server tidak menerima aktivitas berarti selain proses internal. Data menunjukkan bahwa penggunaan CPU rata-rata berada di kisaran 10.79% hingga 10.96%, dan memori stabil di sekitar 40.86%–40.98%. Traffic (PPS) juga sangat rendah, berkisar antara 2.68 hingga 14.37 paket per detik. Hal ini menandakan bahwa server berada dalam keadaan ringan dan tidak mengalami tekanan sistem.

Tabel 4. 21 Hasil Pemantauan Resourse Usage Mode Idle

Percobaan	CPU (%)	Memory (%)	Traffic (PPS)
Percobaan 1	10,96	40,98	2,683333333
Percobaan 2	10,85	40,9	6,616666667
Percobaan 4	10,79	40,86	14,36666667

b. Mode Normal

Saat beberapa user sah melakukan login secara bergantian, terjadi sedikit peningkatan beban server. Penggunaan CPU meningkat menjadi rata-rata 11.62% hingga 13.14%, sedangkan penggunaan memori relatif konstan di angka 40.6% hingga 41.41%. Nilai PPS juga meningkat menjadi 6.63 hingga 8.45, menunjukkan adanya lalu lintas masuk yang lebih aktif namun masih tergolong normal. Server tetap responsif dan tidak menunjukkan gejala overload.

Tabel 4. 22 Hasil Pemantauan Resourse Usage Mode Normal

Percobaan	CPU (%)	Memory (%)	Traffic (PPS)
Percobaan 1	11,62	40,6	6,633333333
Percobaan 2	13,14	40,8	7,333333333
Percobaan 3	10,98	41,41	8,45

c. Mode Mitigation

Mode ini menggambarkan kondisi saat server menerima serangan DDoS dari banyak sumber (VM attacker) dan sistem mitigasi aktif. Terjadi lonjakan yang sangat signifikan dalam konsumsi sumber daya. CPU usage melonjak drastis, dari 50.89% pada percobaan pertama hingga 92.23% pada percobaan ketiga. Begitu juga dengan PPS yang mencapai 186.69 hingga 264.18 paket per detik. Walaupun penggunaan memori meningkat hanya sedikit (sekitar 46%), lonjakan CPU dan trafik menunjukkan bahwa server sedang menangani serangan besar, termasuk menjalankan proses puzzle PoW dan validasi.

Tabel 4. 23 Hasil Pemantauan Resourse Usage Mode Mitigation

Percobaan	CPU (%)	Memory (%)	Traffic (PPS)
Percobaan 1	50,89	45,77	186,6933333
Percobaan 2	87,09	46,04	230,9866667
Percobaan 3	92,23	46,09	264,1833333

2. Response Time

Pengujian response time dalam mode normal dilakukan untuk mengukur waktu yang dibutuhkan server dalam memproses permintaan login dari user sah saat tidak ada serangan dan ada serangan.

a. Mode Normal

Berdasarkan data yang diperoleh dari lima kali percobaan, nilai response time bervariasi antara 1.47 detik hingga 4.63 detik, dengan rata-rata sekitar 3.32 detik. Percobaan pertama menunjukkan response time paling cepat, yaitu 1.47 detik, kemungkinan terjadi saat sistem berada dalam keadaan idle sesaat atau ketika tidak ada user lain yang mengakses sistem. Namun pada percobaan ke-2 hingga ke-5, waktu meningkat hingga lebih dari 3 detik. Secara keseluruhan, response time pada mode normal masih dalam kategori respon cepat dan stabil, di bawah batas toleransi 5 detik untuk sistem login sederhana. Variasi antar percobaan mencerminkan pengaruh kondisi internal server, seperti resource yang sedang digunakan saat login diproses.

Tabel 4. 24 Data Response Time Mode Normal

Percobaan	Mode	Response Time (detik)
1	Normal	1.47
2	Normal	3.06
3	Normal	4.06
4	Normal	4.63
5	Normal	3.40

b. Mode Mitigation

Hasil pengujian menunjukkan bahwa response time meningkat drastis dibandingkan mode normal. Dari lima kali percobaan, waktu respon tercatat berada dalam rentang 29.01 detik hingga 132.82 detik, dengan rata-rata sekitar 51.97 detik.

Percobaan pertama menghasilkan response time yang sangat tinggi, yakni 132.82 detik. Kemungkinan besar ini terjadi karena resource server sedang sangat terbebani akibat serangan aktif. Sementara percobaan kedua hingga kelima menghasilkan response time antara 29–32 detik, yang lebih konsisten dan realistis.

Tabel 4. 25 Data Response Time Mode Mitigation

Percobaan	Mode	Response Time (detik)
1	Mitigasi	132.82
2	Mitigasi	31.40
3	Mitigasi	32.66
4	Mitigasi	31.97
5	Mitigasi	29.01

3. Login Delay

Pengujian ini dilakukan untuk mengetahui dampak aktivasi sistem mitigasi DDoS terhadap pengalaman login user sah, khususnya dari sisi waktu tanggap sistem. Data menunjukkan perbandingan waktu login saat kondisi normal (tanpa serangan) dan saat kondisi mitigasi aktif. Pada mode normal, user sah dapat login dengan sangat cepat, dengan waktu login berkisar antara 0.42 hingga 0.93 detik, dan rata-rata hanya 0.682 detik. Ini menunjukkan bahwa sistem dalam kondisi normal sangat responsif dan memproses permintaan login user sah hampir seketika. Namun, saat terjadi serangan dan sistem mitigasi aktif, waktu login meningkat, dengan nilai login time antara 1.79 hingga 3.70 detik, dan rata-rata 2.775 detik. Kenaikan ini terjadi karena adanya proses tambahan yang dilakukan server seperti validasi apakah IP user termasuk ke dalam IP mencurigakan. Rata-rata delay login tercatat sebesar 2.094 detik. Nilai ini masih dalam batas wajar untuk sistem mitigasi ringan, dan tidak menyebabkan kegagalan login bagi user sah.

Tabel 4. 26 Data Login Delay User Sah

Percobaan	Login Normal (s)	Login Saat Serangan (s)	Delay (s)
1	0.67	3.16	2.49
2	0.54	1.93	1.39
3	0.93	1.79	0.86
4	0.85	3.30	2.45
5	0.42	3.70	3.28
Rata-rata	0.682	2.775	2.094

4.4.4 Analisis Data Pengujian

4.4.4.1 Analisis Data Pengujian Deteksi Serangan DDoS Menggunakan Metode Entropy

Berdasarkan data hasil pengujian menggunakan 3 VM, 5 VM, dan 9 VM attacker yang masing-masing mengirim serangan sebesar 100.000, 250.000, dan 500.000 koneksi, serta merujuk pada Tabel 4.10 (hasil pengamatan deteksi *entropy* dari 3 VM), Tabel 4.12 (hasil pengamatan dari 5 VM), dan Tabel 4.14 (hasil pengamatan dari 9 VM), dapat disimpulkan bahwa setiap parameter pada pengujian berjalan dengan baik dan konsisten sesuai dengan rancangan sistem yang telah ditetapkan sebelumnya.

Penurunan nilai *entropy* terdeteksi dengan tepat pada saat serangan dimulai, dan status mitigasi aktif muncul sesuai kondisi yang diharapkan. Hal ini menunjukkan bahwa mekanisme pendeteksian berbasis *entropy* telah berfungsi secara real-time dan adaptif terhadap intensitas serangan. Selain itu, sistem juga mampu membedakan lalu lintas normal dan serangan dengan akurasi yang tinggi.

Dengan demikian, sistem deteksi dan aktivasi mitigasi terbukti telah memenuhi kriteria fungsionalitas dan keandalan, serta menunjukkan potensi untuk diimplementasikan dalam lingkungan produksi nyata, khususnya untuk mengamankan server dari ancaman serangan DDoS berbasis HTTP request.

4.4.4.2 Analisis Data Pengujian Implementasi Mitigasi Puzzle *Proof of Work*

Berdasarkan hasil pengujian yang ditampilkan dalam Gambar 4.51 hingga Gambar 4.57, sistem mitigasi DDoS berbasis puzzle *Proof of Work* (PoW) telah berjalan sesuai dengan desain yang dirancang. Pengujian ini mencakup beberapa kondisi, baik ketika sistem berada dalam status mitigasi tidak aktif maupun mitigasi aktif, dengan pelibatan kombinasi antara pengguna sah dan attacker.

Pada Gambar 4.51, ditunjukkan bahwa user sah mengakses halaman login saat mitigasi belum aktif. Tidak terdapat anomali trafik, *entropy* dalam kondisi tinggi, dan puzzle tidak dikirim. Hasil login sukses tanpa hambatan, sebagaimana divisualisasikan pada Gambar 4.52, di mana user berhasil diarahkan ke halaman landing page. Ini membuktikan bahwa sistem tidak memicu tantangan puzzle ketika kondisi trafik masih normal, sesuai dengan fungsi deteksi *entropy* yang telah ditetapkan.

Selanjutnya, pada Gambar 4.53, pengujian dilakukan saat sistem memasuki status mitigasi aktif akibat nilai *entropy* yang menurun ke angka 2.31, yang menunjukkan adanya trafik abnormal. Namun, karena IP pengguna sah tidak menunjukkan pola mencurigakan (misalnya frekuensi tinggi atau interval sempit), maka sistem tidak mengirimkan puzzle kepada user tersebut. Proses login tetap berlangsung normal dan sukses, sebagaimana ditampilkan pada Gambar 4.54. Hal ini menegaskan bahwa sistem mitigasi mampu membedakan pengguna sah dan attacker, meskipun mitigasi dalam kondisi aktif.

Skenario selanjutnya ditunjukkan pada Gambar 4.55 hingga Gambar 4.57, yang melibatkan attacker yang mencoba login saat mitigasi aktif. Karena IP attacker memenuhi kriteria IP mencurigakan, sistem secara otomatis mengirimkan puzzle SHA-256, yang ditampilkan pada Gambar 4.55. Setelah attacker mencoba login dan menyelesaikan puzzle (Gambar 4.56), server melakukan proses verifikasi baik terhadap hasil puzzle maupun kredensial. Hasil akhirnya adalah login gagal, karena meskipun puzzle valid, username dan password tidak sesuai (Gambar 4.57, status: Login failed - valid puzzle but invalid user). Hal ini menunjukkan bahwa sistem mitigasi tidak hanya mengandalkan puzzle, tetapi juga mengecek kredensial secara menyeluruh.

4.4.4.3 Analisis Data Pengujian Performa Mitigasi dengan Confusion Matrix

Berdasarkan hasil pengujian sistem mitigasi DDoS berbasis entropy dan puzzle Proof-of-Work (PoW) menggunakan confusion matrix pada tiga skenario jumlah mesin attacker (3 VM, 5 VM, dan 9 VM), diperoleh gambaran performa sistem dalam membedakan trafik attacker dan pengguna asli. Parameter yang diamati dalam pengujian meliputi True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN), serta metrik evaluasi berupa accuracy, precision, recall, dan F1-score.

Pada seluruh skenario pengujian, sistem mencatat nilai recall sebesar 100%, yang menunjukkan bahwa sistem tidak pernah gagal dalam mendeteksi dan memblokir trafik attacker (FN = 0). Artinya, seluruh serangan berhasil terdeteksi secara akurat tanpa ada serangan yang lolos melewati sistem mitigasi.

Namun, sistem mencatat adanya false positive pada ketiga skenario, yang menunjukkan adanya kesalahan deteksi di mana pengguna asli salah diklasifikasikan sebagai attacker dan diblokir. Jumlah false positive cenderung meningkat seiring bertambahnya jumlah attacker:

- Pada skenario 3 VM attacker, terdapat 3 false positive.
- Pada skenario 5 VM attacker, false positive meningkat menjadi 5 kasus.
- Pada skenario 9 VM attacker, false positive meningkat lagi menjadi 7 kasus.

Peningkatan jumlah false positive ini menunjukkan bahwa ketika jumlah attacker bertambah, sistem semakin sulit membedakan secara akurat antara trafik attacker dan pengguna asli, terutama pada kondisi lalu lintas jaringan yang lebih kompleks.

Dari sisi precision, sistem mencatat penurunan bertahap:

- Precision pada skenario 3 VM sebesar 98,46%.
- Precision pada skenario 5 VM sebesar 97,47%.
- Precision pada skenario 9 VM sebesar 96,50%.

Hal ini menunjukkan bahwa meskipun sistem tetap mampu memblokir seluruh serangan, terdapat peningkatan jumlah kesalahan dalam membedakan trafik sah dan berbahaya.

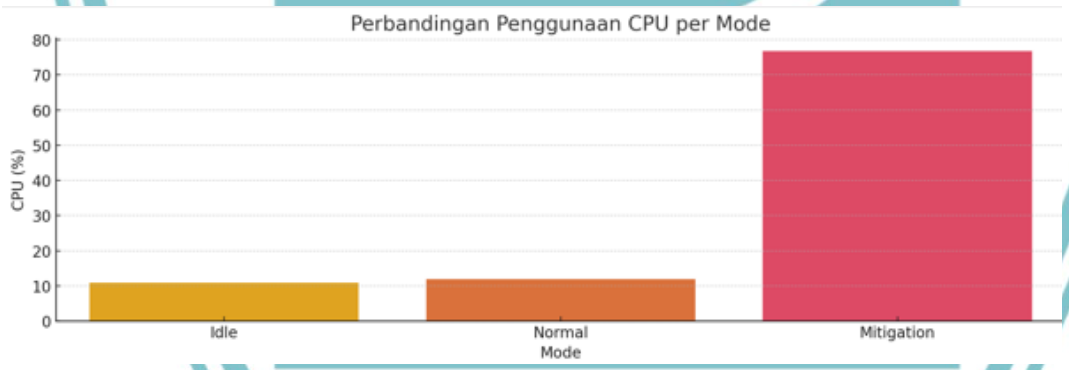
Nilai accuracy dan F1-score pada seluruh skenario tetap tinggi di atas 98%, menunjukkan sistem memiliki kemampuan klasifikasi yang sangat baik secara keseluruhan.

Dapat disimpulkan bahwa sistem mitigasi berbasis entropy dan puzzle Proof-of-Work (PoW) memiliki performa yang sangat baik untuk mendeteksi dan memblokir serangan DDoS, dengan recall sempurna dan nilai precision yang tetap tinggi. Meski terjadi sedikit peningkatan kesalahan blokir pengguna asli pada kondisi serangan berskala besar, sistem tetap mampu menjaga stabilitas layanan tanpa membiarkan serangan DDoS lolos.

4.4.4.4 Analisis Data Pengujian Dampak Sistem Mitigasi Terhadap Performa Server

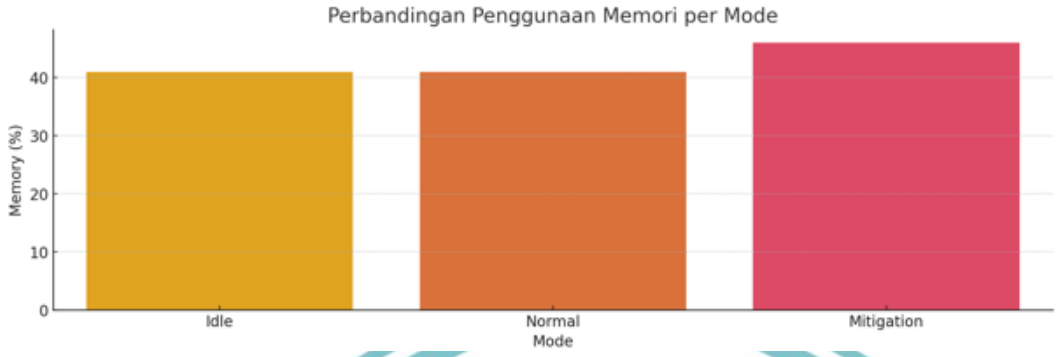
1. Resource Usage

Berdasarkan hasil pengujian penggunaan sumber daya pada tiga mode operasional server (Idle, Normal, dan Mitigasi), diperoleh data pemantauan CPU, memori, dan lalu lintas jaringan (PPS) seperti yang disajikan dalam Tabel 4.16 hingga Tabel 4.18.



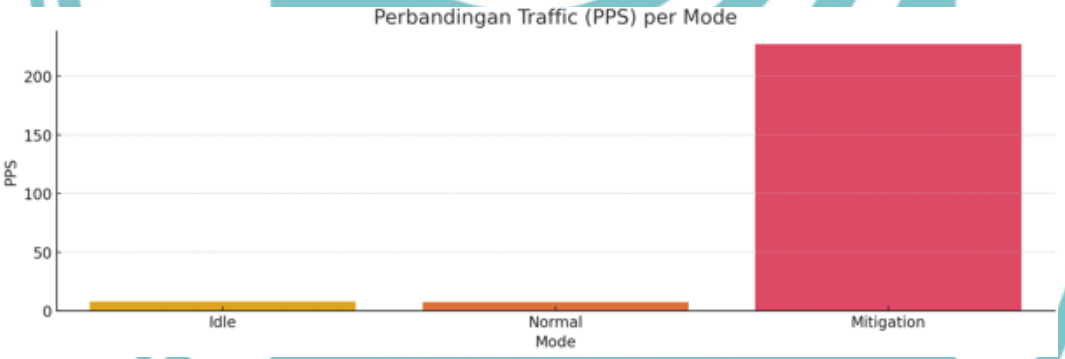
Gambar 4. 73 Perbandingan Penggunaan CPU

Pada mode idle, server tidak menerima beban trafik eksternal, hanya menjalankan proses internal sistem. Dari data yang diperoleh, terlihat bahwa penggunaan CPU rata-rata sangat rendah, yakni berada pada kisaran 10.79% hingga 10.96%, sedangkan penggunaan memori stabil di sekitar 40.86% hingga 40.98%. Nilai traffic (PPS) juga tergolong sangat kecil, yaitu antara 2.68 hingga 14.37. Hal ini menandakan bahwa dalam keadaan idle, server berada dalam kondisi optimal tanpa tekanan beban berarti.



Gambar 4. 74 Perbandingan Penggunaan Memory

Berbeda dengan itu, pada mode normal, di mana beberapa user sah melakukan login secara berkala, terjadi peningkatan ringan pada beban sistem. Penggunaan CPU meningkat ke kisaran 11.62% hingga 13.14%, dan memori tetap berada pada angka yang relatif konstan, yaitu 40.6% hingga 41.41%. Nilai PPS juga mengalami kenaikan menjadi 6.63 hingga 8.45, yang menunjukkan adanya aktivitas login dari pengguna sah yang masuk ke dalam sistem.



Gambar 4. 75 Perbandingan Traffic PPS

Sementara itu, pada mode mitigasi, di mana server menghadapi serangan DDoS dan sistem mitigasi aktif, terjadi lonjakan signifikan pada semua parameter. Berdasarkan data pada Tabel 4.18, penggunaan CPU meningkat tajam dari 50.89% pada percobaan pertama hingga mencapai 92.23% pada percobaan ketiga. Nilai PPS juga melonjak drastis dari 186.69 hingga 264.18, menandakan lalu lintas masuk yang sangat padat akibat aktivitas serangan HTTP Flood. Meskipun penggunaan memori hanya meningkat sedikit (sekitar 45.77%–46.09%), peningkatan drastis pada CPU dan PPS menunjukkan bahwa server bekerja sangat

keras dalam menjalankan proses mitigasi, termasuk perhitungan *entropy*, pengiriman puzzle, dan validasi nonce.

2. Response Time

Berdasarkan hasil pengujian response time yang dilakukan untuk membandingkan waktu respon server dalam kondisi normal dan saat sistem mitigasi aktif, diperoleh hasil seperti pada Tabel 4.19 dan 4.20. Dalam mode normal, waktu rata-rata yang dibutuhkan server untuk merespons permintaan login dari user sah berada pada rentang 1.47 hingga 4.63 detik, dengan nilai rata-rata sebesar 3.32 detik. Nilai ini masih tergolong cepat dan stabil, karena server tidak dalam kondisi tekanan trafik tinggi.

Gambar 4. 76 Perbandingan Response Time

Sementara itu, saat mitigasi aktif, terjadi peningkatan response time yang cukup signifikan. Dari lima kali percobaan, waktu respon tercatat meningkat drastis, yaitu dalam rentang 29.01 hingga 132.82 detik, dengan rata-rata mencapai 51.97 detik. Keterlambatan ini terjadi karena proses mitigasi mengharuskan server melakukan perhitungan *entropy*, pengiriman puzzle, dan validasi nonce sebelum login diproses. Percobaan pertama bahkan mencatat response time tertinggi sebesar 132.82 detik, kemungkinan besar akibat beban serangan yang sangat tinggi pada saat itu.

3. Login Delay

Selain response time, dilakukan pula pengujian terhadap login delay, yaitu selisih waktu antara permintaan login dari user sah dan waktu login berhasil diproses. Berdasarkan Tabel 4.21, dalam kondisi normal tanpa serangan, user sah dapat login

dengan sangat cepat, dengan waktu antara 0.42 hingga 0.93 detik, dan nilai rata-rata sebesar 0.682 detik. Hal ini menunjukkan bahwa sistem login sangat responsif saat berada dalam kondisi stabil.

Namun, saat terjadi serangan dan sistem mitigasi diaktifkan, waktu login meningkat menjadi 1.79 hingga 3.70 detik, dengan rata-rata sebesar 2.775 detik. Ini berarti terjadi delay login sebesar rata-rata 2.094 detik, sebagai dampak dari proses mitigasi yang sedang berlangsung. Meskipun ada peningkatan, nilai delay ini masih tergolong wajar dan tidak menyebabkan gangguan serius terhadap pengalaman pengguna sah, karena login tetap berhasil diproses tanpa error.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil yang telah dicapai dalam penelitian skripsi ini, terdapat beberapa kesimpulan sebagai berikut:

1. Metode entropy terbukti efektif untuk mendeteksi potensi serangan DDoS pada server Ubuntu. Sistem berhasil mengamati penurunan nilai entropy secara real-time pada setiap skenario serangan (3 VM, 5 VM, dan 9 VM), dengan nilai entropy turun hingga <1.0 saat serangan aktif. Sistem juga mempertimbangkan status cooldown untuk mencegah false negative. Tingkat keberhasilan deteksi serangan tercatat 100% dari seluruh skenario pengujian.
2. Sistem mitigasi dengan puzzle PoW berhasil diimplementasikan secara adaptif dan selektif. Puzzle dikirim hanya kepada IP yang terdeteksi mencurigakan saat mitigasi aktif. Validasi puzzle di sisi server memastikan bahwa hanya pengguna yang menyelesaikan puzzle dan memiliki kredensial valid yang dapat login, memperkuat keamanan tanpa mengganggu user sah.
3. Performa sistem dalam membedakan pengguna sah dan penyerang sangat tinggi, dibuktikan dengan hasil confusion matrix pada tiga skenario:
   - Akurasi mencapai >99% pada semua pengujian,
   - Precision berada pada kisaran 98.52%–98.64%,
   - Recall sempurna 100%,
   - F1-Score rata-rata >99%.

Hasil ini menunjukkan sistem sangat efektif dalam mengenali dan memblokir trafik berbahaya tanpa kesalahan deteksi attacker (FN = 0).

4. Penerapan sistem mitigasi berdampak pada peningkatan beban server, namun performa sistem tetap berada dalam tingkat yang dapat diterima dan tidak mengganggu fungsionalitas layanan secara keseluruhan.
   - Saat mode normal, rata-rata penggunaan CPU sekitar 12%, memory 40%, dan traffic sekitar 7 PPS.

- Saat mode mitigasi, CPU melonjak hingga 92%, dan traffic mencapai 264 PPS, menandakan server bekerja intensif untuk memproses puzzle dan validasi serangan.
- Response time juga meningkat:
  - Mode normal: rata-rata 3,32 detik
  - Mode mitigasi: rata-rata 51,97 detik
- Login delay user sah bertambah, dari 0,68 detik (tanpa serangan) menjadi 2,77 detik (saat mitigasi aktif), namun tetap tidak menyebabkan kegagalan login.

Keseluruhan hasil menunjukkan bahwa sistem mitigasi DDoS berbasis *entropy* dan puzzle PoW yang dirancang telah berhasil mendeteksi serangan secara adaptif, memitigasi secara selektif, serta mempertahankan aksesibilitas pengguna sah dengan dampak minimal terhadap performa server. Sistem ini dinilai layak diimplementasikan dalam lingkungan nyata untuk meningkatkan keamanan server terhadap ancaman DDoS berbasis HTTP Request.

5.1 Saran

Berdasarkan penelitian yang telah dilakukan, berikut beberapa saran yang dapat dijadikan masukan untuk penelitian selanjutnya, diantaranya:

1. Penelitian ini dilakukan dalam jaringan lokal. Pengujian lebih lanjut perlu dilakukan di lingkungan nyata agar sistem dapat dievaluasi dalam kondisi trafik yang lebih kompleks.
2. Pengujian hanya melibatkan sedikit perangkat. Disarankan menambah variasi perangkat dan jaringan user sah untuk menguji ketepatan sistem dalam membedakan pengguna dan attacker.
3. Sistem perlu dikembangkan lebih lanjut dengan mengintegrasikan firewall (seperti iptables) dan IDS (seperti Snort) guna memperkuat mitigasi serangan.
4. Sistem saat ini fokus pada serangan HTTP GET. Pengembangan lanjutan diperlukan untuk mendeteksi serangan aplikasi seperti HTTP POST Flood.

DAFTAR PUSTAKA

Afifah Rodhiyatun Nisa, A., Anandito Daffa Wijayanto, Arya Prabudi Jaya Priana, & Setiawan, A. (2024). Analisis Log Server untuk mendeteksi Serang DDoS pada Keamaan Jaringan di Website. *Journal of Internet and Software Engineering*, *1*(3), 17. https://doi.org/10.47134/pjise.v1i3.2612

Alviano, M. (2023). Hashcash Tree, a Data Structure to Mitigate Denial-of-Service Attacks. *Algorithms*, *16*. https://doi.org/10.3390/a16100462

Asa'ari Lubis, B., Yanuar Ar-Rafi, D., Widiyani, I., Lestari, K. I., & Zy, A. T. (2024). *Analysis and Mitigation Technique of DDoS on Server Networks Based on Modern Technology*.

Aydın, H., Orman, Z., & Aydın, M. A. (2022). A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. *Computers and Security*, *118*. https://doi.org/10.1016/j.cose.2022.102725

Bashurov, V., & Safonov, P. (2023). Anomaly detection in network traffic using entropy-based methods: application to various types of cyberattacks. *Issues in Information Systems*, *24*(4), 82–94. https://doi.org/10.48009/4_iis_2023_107

Bazzanella, D., & Gangemi, A. (2023). Bitcoin: a new proof-of-work system with reduced variance. *Financial Innovation*, *9*. https://doi.org/10.1186/s40854-023-00505-2

Bostanov, V. (2021). Client Puzzle Protocols as Countermeasure against Automated Threats to Web Applications. *IEEE Access*, *9*, 75722–75728. https://doi.org/10.1109/ACCESS.2021.3082037

Center Canadian for Cyber. (2024). *Defending against distributed denial of service (DDoS) attacks*. https://www.cyber.gc.ca/en/guidance/defending-against-distributed-denial-service-ddos-attacks-itsm80110

Chakraborty, T., Mitra, S., Mittal, S., & Young, M. (2022). AI_Adaptive_POW: An AI assisted Proof Of Work (POW) framework for DDoS defense. *Software Impacts*, *13*. https://doi.org/10.1016/j.simpa.2022.100335

Chotimah, H. C. (2019). Tata Kelola Keamanan Siber dan Diplomasi Siber Indonesia di Bawah Kelembagaan Badan Siber dan Sandi Negara [Cyber Security Governance

and Indonesian Cyber Diplomacy by National Cyber and Encryption Agency]. *Jurnal Politica Dinamika Masalah Politik Dalam Negeri Dan Hubungan Internasional*, *10*(2), 113–128. https://doi.org/10.22212/jp.v10i2.1447

Cloudflare. (n.d.). *What is a DDoS attack?* https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

Documentation, U. (2025). *Ubuntu Server documentation*. Https://Documentation.Ubuntu.Com/Server/.

Gorave, A. (2019). Ambient Intelligence for Rehabilitation: A Survey General Terms. In *International Journal of Computer Applications* (Vol. 178, Issue 38). www.ijcaonline.org

Hassan, A. I., El Reheem, E. A., & Guirguis, S. K. (2024). An entropy and machine learning based approach for DDoS attacks detection in software defined networks. *Scientific Reports*, *14*. https://doi.org/10.1038/s41598-024-67984-w

Jawahar, A., Kaythry, P., Vinoth Kumar, C., Vinu, R., Amrish, R., Bavapriyan, K., & Gopinaath, V. (2024). DDoS mitigation using blockchain and machine learning techniques. *Multimedia Tools and Applications*, *83*, 60265–60278. https://doi.org/10.1007/s11042-023-18028-4

Kurniawan, M. T. (2023). *UNIVERSITAS INDONESIA PENGEMBANGAN METODE ENTROPY UNTUK DETEKSI SERANGAN DISTRIBUTED DENIAL OF SERVICE (DDoS) PADA SOFTWARE DEFINED NETWORK (SDN) DAN DENGAN PENERAPAN FEATURE SELECTION*.

M, J. (2024). *Entropy: A Mathematical Approach to DDoS Protection*. Https://Medium.Com/%40remind.Stephen.to.Do.Sth/Entropy-a-Mathematical-Approach-to-Ddos-Protection-71a2af38ab91.

openssh. (2023). *OpenSSHManual Pages*. Https://Www.Openssh.Com/Manual.Html.

Patil, R., & G, N. D. (2023). *Mitigation of DDoS Attacks using Entropy and Proof-of-Work Based Puzzle in OpenStack Cloud*. https://doi.org/10.1109/CONIT59222.2023.10205888

Pebrianto, J., & Suryani, V. (2025). Adaptive DDoS Attack Detection: Entropy-Based Model With Dynamic Threshold and Suspicious IP Reevaluation. *IEEE Access*, *13*, 55858–55876. https://doi.org/10.1109/ACCESS.2025.3553144

Seidl, J. (2020). *GitHub - jseidl/GoldenEye: GoldenEye Layer 7 (KeepAlive+NoCache) DoS Test Tool*. Https://Github.Com/Jseidl/GoldenEye.

Sumayyah, Z. I., Permana, S. D. S., Tsabit, M., & Setiawan, A. (2024). Penerapan dan Mitigasi Teknik Slowloris dalam Serangan Distributed Denial-of-Service (DDos) terhadap Website Ilegal dengan Kali Linux. *Journal of Internet and Software Engineering*, *1*(2), 14. https://doi.org/10.47134/pjise.v1i2.2694

Süzen, A. A. (2021). UNI-CAPTCHA: A Novel Robust and Dynamic User-Non-Interaction CAPTCHA Model Based on Hybrid biLSTM+Softmax. *Journal of Information Security and Applications*, *63*. https://doi.org/10.1016/j.jisa.2021.103036

Tao, T. (2017, January). *The Erdos discrepancy problem*.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. In *Future Generation Computer Systems* (Vol. 135, pp. 364–381). Elsevier B.V. https://doi.org/10.1016/j.future.2022.05.014

Xiong, J., Chen, L., Bhuiyan, M. Z. A., Cao, C., Wang, M., Luo, E., & Liu, X. (2020). A secure data deletion scheme for IoT devices through key derivation encryption and data analysis. *Future Generation Computer Systems*, *111*, 741–753. https://doi.org/10.1016/j.future.2019.10.017

DAFTAR RIWAYAT HIDUP

Berlianna Upik Nurniati

Lahir di Magetan, 14 Juni 2002, anak pertama dari dua bersaudara. Lulus dari SD Negeri Kemanggisan 06 Pagi pada tahun 2015, kemudian melanjutkan pendidikan di SMP Negeri 111 Jakarta Barat dan lulus pada tahun 2018, dan selanjutnya melanjutkan pendidikan di SMA Negeri 16 Jakarta Barat dan lulus pada tahun 2021.

Setelah itu pada tahun 2021, penulis berkesempatan untuk melanjutkan pendidikan tinggi di Politeknik Negeri Jakarta Jurusan Teknik Informatika dan Komputer, Program Studi Teknik Multimedia dan jaringan.