



Desain dan Analisis Metode Pengkondisian Clear and Less Malicious Internet Traffic

Muhti Subiyantoro
21917014

*Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2025*

Lembar Pengesahan Pembimbing

Desain dan Analisis Pengkondisian Clear and Less Malicious Internet Traffic

Muhti Subiyantoro
21917014



Pembimbing

A handwritten signature in blue ink, appearing to read 'Mukhammad Andri Setiawan', is placed over the name of the supervisor.

Mukhammad Andri Setiawan S.T., M.SC., PH.D.

Lembar Pengesahan Penguji

Desain dan Analisis Pengkondisian Clear and Less Malicious Internet Traffic

Muhti Subiyantoro
21917014

Yogyakarta, Juli 2025

Tim Penguji,

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.
Ketua

Dr. Yudi Prayudi, S.Si., M. Kom.
Anggota I

Ir. Irving Vitra Paputungan, S.T., M.Sc., Ph.D.
Anggota II

15/08/2025

الجامعة الإسلامية
الاندونيسية

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

Abstrak

Desain dan Analisis Pengkondisian Clear and Less Malicious Internet Traffic

Peningkatan penggunaan *internet* telah meningkatkan risiko serangan siber melalui *domain* berbahaya yang digunakan untuk distribusi *malware*, *phishing*, dan operasi *botnet*. Ancaman ini dapat merugikan individu maupun organisasi, sebagaimana terbukti dari insiden tahun 2023 yang menyebabkan *downtime* layanan perbankan dan kebocoran data *e-commerce*. Meskipun penyaringan *domain* pada *firewall perimeter* telah menjadi strategi defensif yang umum diterapkan, efektivitasnya dalam skala besar, khususnya pada lingkungan *Internet Service Provider (ISP)*, masih memerlukan evaluasi lebih lanjut. Penelitian ini bertujuan mengevaluasi efektivitas penyaringan *domain* berbahaya yang diterapkan sebagai bagian dari layanan keamanan *firewall perimeter* pada *ISP*. Metodologi penelitian menggunakan analisis empiris berbasis data dari jaringan yang telah menerapkan *firewall filtering*, dengan fokus pada analisis kinerja teknik penyaringan berbasis *blacklist domain* berbahaya dan pengaruhnya terhadap *traffic flow* jaringan. Hasil penelitian menunjukkan dampak signifikan dari implementasi *firewall filtering domain* berbahaya. Dari aspek *traffic flow*, terjadi peningkatan total *flow* sebesar 2.49% akibat mekanisme *retry connection* ketika *domain* diblokir. Namun, dampak ini tidak signifikan terhadap *bandwidth* karena *filtering* berhasil mengurangi total *bytes* sebesar 24.5% dan total paket sebesar 10.5%, disertai penurunan rata-rata *bandwidth* sebesar 22.58% dan *packet rate* sebesar 8.81%. Efektivitas keamanan terbukti dari terdeteksinya 1,090 *IP address* pengguna yang mencoba mengakses *domain* berbahaya dan keberhasilan memblokir 1,919 *IP address malicious*. Penelitian ini menyimpulkan bahwa metode *filtering domain* berbahaya terbukti efektif dalam meningkatkan keamanan pengguna dan mengoptimalkan penggunaan *bandwidth* dengan membersihkan *traffic* yang tidak diinginkan. Selain itu, metode ini mudah diterapkan pada produk *existing* yang digunakan oleh *Internet Service Provider*. Meskipun terjadi peningkatan *flow* akibat *retry connection*, dampaknya tidak signifikan terhadap *bandwidth* karena *filtering* berhasil mengurangi total *bytes* dan paket secara substansial.

Kata kunci

malicious domain filtering, ISP cybersecurity, blacklist-based security, network performance metrics, perimeter defense, traffic flow analysis

Abstract

Design and Conditioning Analysis of Clear and Less Malicious Internet Traffic

The growing dependence on internet connectivity has heightened cybersecurity threats through malicious domains that facilitate malware, phishing, and botnet operations. These threats can harm both individuals and organizations, as evidenced by incidents in 2023 that caused banking service downtime and e-commerce customer data breaches. While domain filtering on perimeter firewalls has become a commonly implemented defensive strategy, its effectiveness at a large scale, particularly in Internet Service Provider (ISP) environments, still requires further evaluation. This research aims to evaluate the effectiveness of malicious domain filtering implemented as part of perimeter firewall security services in ISPs. The research methodology employs empirical analysis based on data from networks that have implemented firewall filtering, focusing on performance analysis of blacklist-based malicious domain filtering techniques and their impact on network traffic flow. The research results demonstrate a significant impact from the implementation of malicious domain firewall filtering. From the traffic flow perspective, there was a 2.49% increase in total flow due to retry connection mechanisms when domains were blocked. However, this impact was not significant on bandwidth because filtering successfully reduced total bytes by 24.5% and total packets by 10.5%, accompanied by an average bandwidth decrease of 22.58% and packet rate reduction of 8.81%. Security effectiveness was proven by the detection of 1,090 user IP addresses attempting to access malicious domains and the successful blocking of 1,919 malicious IP addresses. This research concludes that malicious domain filtering methods are proven effective in enhancing user security and optimizing bandwidth utilization by cleaning unwanted traffic. Additionally, this method is easily *implementable* on existing products used by Internet Service Providers. Although there was an increase in flow due to retry connections, its impact on bandwidth was not significant because filtering successfully reduced total bytes and packets substantially.

Keywords

malicious domain filtering, ISP cybersecurity, blacklist-based security, network performance metrics, perimeter defense, traffic flow analysis

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Juli 2025



Muhti Subiyantoro, ST

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Publikasi berikut menjadi bagian dari Bab 1, 3, 4, dan 5

Subiyantoro, M., & Setiawan, A. (2025). Quantitative Assessment of Blacklist-Based Malicious Domain Filtering for ISP Security: Balancing Protection and Performance ARTICLE INFO ABSTRACT. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI)*, 11(2), 322–345. <https://doi.org/10.26555/jiteki.v11i2.30815>
<https://journal.uad.ac.id/index.php/JITEKI/article/view/30815>

Kontributor	Jenis Kontribusi
Muhti Subiyantoro S.T.	Mendesain eksperimen (70%) Menulis <i>paper</i> (70%)
Mukhammad Andri Setiawan S.T., M.SC., PH.D.	Mendesain eksperimen (30%) Menulis dan mengedit <i>paper</i> (30%)

Halaman Kontribusi

Ucapan terima kasih tak terhingga atas kontribusi dalam terselesaikannya tesis ini. Pertama Bapak Mukhammad Andri Setiawan S.T., M.SC., PH.D., selaku Pembimbing yang telah memberikan arahan-arahannya kepada penulis, sehingga penulisan tesis ini bisa selesai dengan baik. Bapak Teguh Wijaya, S.T., selaku Direktur PT. Spectrum Indowibawa, yang memberikan dukungan dan saran serta motivasi atas terselesaikannya Program Magister. Bapak Hendriansyah, selaku owner dari PT. Sekar Solusi Abadi dan Team yang telah memberikan ijin sebagai tempat untuk melakukan penelitian dan membantu dalam pengambilan data, sehingga tesis ini bisa terselesaikan.

Halaman Persembahan

1. Almarhumah Ibu Subarmini dan Almarhum Bapak Munasir selaku Ibu dan Ayah yang telah memberikan dukungan moral dalam menempuh studi.
2. Dian Septiana selaku istri tercinta, Zahwa Alifi Zakyah dan Muazzama Nur Amira selaku anak-anak yang selalu memberikan semangat dan dukungan dari mulai awal studi hingga terselesaikannya tesis ini.
3. Mukhammad Andri Setiawan, S.T., M.SC., PH.D., yang telah memberikan bimbingan dan motivasi selama pelaksanaan dan penyusunan penelitian
4. Seluruh jajaran Manajemen Magister Informatika yang telah memberikan fasilitas dan dukukung dalam studi.
5. Seluruh teman-teman Angkatan FD-21 dan teman-teman di Magister Informatika UII lainnya yang telah meluangkan waktu untuk berbagi dan diskusi terkait tahapan penyelesaian tesis.
6. Semua pihak yang telah berkontribusi secara langsung maupun tak langsung sehingga tesis ini terselesaikan dengan baik.

Kata Pengantar

Puji syukur kehadiran Allah s.w.t., karena atas rahmat dan karunia-Nya saya dapat menyelesaikan penelitian ini. Penelitian yang berjudul “**Desain dan Analisis Metode Pengkondisian *Clear and Less Malicious Internet Traffic***” disusun sebagai salah satu syarat untuk memperoleh gelar Magister Teknik pada Program Studi Teknik Industri Konsentrasi Forensika Digital dari Universitas Islam Indonesia.

Saya menyadari tanpa bantuan dan bimbingan dari berbagai pihak akan sulit bagi saya untuk menyelesaikan tugas akhir dari program magister ini. Oleh sebab itu, saya mengucapkan terim kasih kepada pihak yang telah membantu saya diantaranya:

1. Almarhum Bapak Munasir dan Almarhumah Subarmini selaku orang tua yang telah memberi bimbingan dari kecil dan memberikan seluruh kemampuan dan dukungan moral serta finansial sehingga menumbuhkan semangat sampai terselesaikannya tesis ini
2. Sugito Prajitno selaku Mertua yang selalu memberi motivasi dan doá
3. Dian Septiana selaku istri, Zahwa Alifi Zakyah dan Muazzaama Nur Amira selaku anak-anak yang selalu memberikan semangat, motivasi dan do’a selama menempuh studi
4. Mukhammad Andri Setiawan, S.T., M.SC., PH.D., yang telah memberikan bimbingan dan motivasi selama pelaksanaan dan penyusunan penelitian
5. Seluruh Staf Pengajar Program Magister Teknik Industri Universitas Islam Indonesia yang telah memberikan fasilitas, ilmu pengetahuan, saran, nasehat dan berbagi pengalaman selama ini
6. Seluruh Staf Tata Usaha Program Magister Teknik Industri Universitas Islam Indonesia yang telah memberikan layanan administrasi selama studi
7. Kawan-kawan Mahasiswa Magister Teknologi Industri yang telah menemani dan memberikan motivasi selama studi

Tak ada gading yang tak retak, saya menyadari sepenuhnya dengan segala kerendahan dan keterbatasan daya dan kemampuan yang saya miliki, penelitian ini masih jauh dari sempurna, namun hal itu tidak akan menghambat kelanjutan proses belajar untuk mendapatkan pengalaman berharga.

Oleh karena itu, masukan, saran, dan dukungan yang membangun akan menjadi sumber sangat berharga dalam menyempurnakan penelitian ini. Besar harapan saya, jauh dari sempurna, penelitian ini bisa memberikan manfaat bagi berbagai pihak

Yogyakarta, Juli 2025

Muhti Subiyantoro, S.T.

Daftar Isi

Lembar Pengesahan Pembimbing	i
Lembar Pengesahan Penguji.....	ii
Abstrak.....	iii
Abstract.....	iv
Pernyataan Keaslian Tulisan	v
Daftar Publikasi	vi
Halaman Kontribusi.....	vii
Halaman Persembahan	viii
Kata Pengantar.....	ix
Daftar Isi	xi
Daftar Tabel.....	xiv
Daftar Gambar	xv
BAB 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Batasan Masalah	7
1.4 Tujuan Penelitian	7
1.5 Manfaat Penelitian	7
1.6 Metodologi Penelitian	8
1.7 <i>Review</i> Penelitian	8
1.8 Sistematika Penulisan	8
BAB 2 Tinjauan Pustaka.....	10
2.1 Forensik dan Forensik Jaringan	10
2.2 Arsitektur Jaringan <i>Internet</i>	10
2.3 <i>Domain Name</i>	13

2.3.1 Arsitektur <i>Domain Name</i>	13
2.3.2 Proses Resolusi <i>DNS (Query DNS)</i>	14
2.4 <i>Malicious Internet Traffic</i>	16
2.4.1 <i>Malicious Domain</i>	17
2.4.2 VTeknik dan Taktik yang Digunakan oleh <i>Malicious Domain</i>	19
2.5 <i>Domain Black List</i>	20
2.6 <i>Firewall</i>	22
2.6.1 Metode <i>Firewall</i>	23
2.6.2 Metode <i>Filtering</i>	25
2.6.3 Teknologi <i>Filtering</i>	25
2.6.4 Algoritma <i>Filtering</i>	25
2.6.5 Sumber Data	25
2.7 <i>MikroTik Device Landscape</i>	26
2.7.1 <i>Firewall</i> pada <i>MikroTik</i>	27
2.7.2 <i>Blocking</i> dengan <i>MikroTik</i>	27
2.7.3 <i>Address List</i> pada <i>MikroTik</i>	27
2.7.4 <i>Import File .rsc</i> ke dalam <i>Address List</i>	28
2.7.5 <i>Traffic Flow</i> pada <i>MikroTik</i>	28
2.8 <i>NFDUMP Open Source Tools</i>	29
2.8.1 Konfigurasi	30
2.8.2 <i>Netflow Processing</i>	30
2.8.3 Kompatibilitas dan Dukungan <i>Protocol</i>	32
2.8.4 Fitur Analisis dan <i>Filtering</i>	33
2.8.5 Kemampuan Statistik dan <i>Reporting</i>	34
2.8.6 Performa dan Efisiensi	34
2.8.7 Integrasi dengan Sistem <i>Monitoring</i>	34
BAB 3 Metodologi Penelitian	35

3.1 <i>Literature Review</i>	35
3.2 Lokasi dan Waktu Penelitian	36
3.3 Identifikasi Sistem	36
3.4 Analisis Kebutuhan Sistem	38
3.5 Perancangan dan Implementasi.....	40
3.6 Analisis dan Evaluasi.....	45
3.7 Rekomendasi Implementasi	45
BAB 4 Hasil dan Pembahasan	46
4.1 <i>Malicious Domain Address List</i>	46
4.1.1 Analisis Forensik Artefak Domain Berbahaya.....	49
4.1.2 Jejak Digital dan Metadata Serangan	50
4.2 <i>Malicious Domain IP Traffic</i>	50
4.3 Analisis Forensik Sistem <i>Firewall</i> dan Deteksi Ancaman	53
4.4 Validasi Ancaman Berdasarkan Karakteristik <i>Domain</i> Berbahaya	54
4.5 <i>Traffic Flow Data Analysis</i>	56
4.5.1 Artefak <i>Flow Records and Analysis Behavioral</i>	59
4.5.2 Interpretasi Forensik <i>Network Flow Metrics</i>	60
4.5.3 <i>Chain of Custody</i> dan Integritas Artefak	61
BAB 5 Kesimpulan dan Saran	81
5.1 Kesimpulan	81
5.2 Saran	82
Daftar Pustaka.....	84

Daftar Tabel

Tabel 4. 1 Daftar 10 Baris Terakhir dari 84765 <i>Malicious Domain Address List File .csv</i>	48
Tabel 4. 2 Daftar 10 Baris Terakhir dari 1090 <i>IP Client Request</i> ke <i>Malicious Domain</i>	51
Tabel 4. 3 Daftar 10 Baris Terakhir dari 1919 <i>IP Malicious Domain</i>	52
Tabel 4. 4 Data Harian Periode I Selama 9 Hari - <i>FILTER OFF</i>	62
Tabel 4. 5 Data Harian Periode II Selama 9 Hari - <i>FILTER ON</i>	62
Tabel 4. 6 Data Harian Periode III Selama 9 Hari - <i>FILTER OFF</i>	63
Tabel 4. 7 Data Harian Periode IV Selama 9 Hari - <i>FILTER ON</i>	64
Tabel 4. 8 Data Periodik - <i>FILTER OFF</i> dan <i>ON</i>	69
Tabel 4. 9 Total Data – Persentase Kenaikan/Penurunan Saat <i>FILTER OFF</i> dan <i>ON</i>	74

Daftar Gambar

Gambar 1.1 Tahapan metode penelitian.	8
Gambar 2.1 Prinsip operasi <i>NFDUMP</i>	29
Gambar 2.2 Langkah pemrosesan <i>Netflow</i>	31
Gambar 2.3 Opsional <i>output</i> berdasarkan <i>command</i> pemrosesan.....	32
Gambar 3.1 <i>Client traffic communication domain request</i>	39
Gambar 3.2 Tahapan dalam <i>live experiment</i>	42
Gambar 3.3 Prototipe diagram penelitian.....	42
Gambar 3.4 NFDUMP sebagai <i>software</i> kolektor dan <i>analyzer</i>	43
Gambar 4.1 Hasil konversi <i>address list .rsc</i> dari <i>.csv file</i>	49
Gambar 4.2 Data yang dikumpulkan dari <i>server</i> kolektor.	57
Gambar 4.3 Data <i>file nfcapd</i> dari <i>server</i> kolektor.	58
Gambar 4.4 Pengecekan kode <i>hash file</i> data	61
Gambar 4.5 Grafik <i>daily total flows</i>	65
Gambar 4.6 Grafik <i>daily total bytes</i>	66
Gambar 4.7 Grafik <i>daily total packets</i>	66
Gambar 4.8 Grafik <i>daily average BPS</i>	67
Gambar 4.9 Grafik <i>daily average PPS</i>	68
Gambar 4.10 Grafik <i>daily average BPP</i>	68
Gambar 4.11 Grafik fluktuasi <i>periodic total flows</i>	70
Gambar 4.12 Grafik fluktuasi <i>periodic total bytes</i>	70
Gambar 4.13 Grafik fluktuasi <i>periodic total packets</i>	71
Gambar 4.14 Grafik fluktuasi <i>periodic total average BPS</i>	72
Gambar 4.15 Grafik fluktuasi <i>periodic total average PPS</i>	72
Gambar 4.16 Grafik fluktuasi <i>periodic average BPP</i>	73
Gambar 4.17 Kenaikan <i>total flows</i> 2.49%.....	74
Gambar 4.18 Penurunan <i>total bytes</i> 24.50%.	75
Gambar 4.19 Penurunan <i>total packets</i> 10.15%.....	76
Gambar 4.20 Penurunan <i>total average BPS</i> 22.58%.....	76
Gambar 4.21 Penurunan <i>total average PPS</i> 8.81%.....	77
Gambar 4.22 Penurunan <i>total average BPP</i> 17.25%.	78

BAB 1

Pendahuluan

1.1 Latar Belakang

Di era digital yang semakin maju, *internet* telah menjadi bagian integral dari kehidupan sehari-hari, memfasilitasi komunikasi, transaksi keuangan, pendidikan, dan berbagai aspek penting lainnya (Babari *et al.*, 2023; Dabić *et al.*, 2023; Shaengchart & Kraiwanit, 2023; Song *et al.*, 2024; Yaqoob *et al.*, 2023; M. Zeng *et al.*, 2023). Namun, kemajuan ini juga membawa tantangan besar dalam bentuk ancaman keamanan siber (Dimitriadis *et al.*, 2020; Erola *et al.*, 2022; Guarascio *et al.*, 2022; Y. Li & Liu, 2021; Meland *et al.*, 2022; Y. Zeng, 2022). Salah satu ancaman yang paling signifikan adalah penggunaan *domain* berbahaya atau *malicious domain* oleh aktor jahat untuk melakukan berbagai aktivitas kriminal siber, seperti distribusi *malware*, *phishing*, penipuan *online*, serta pengendalian *botnet* (Aljabri & Mohammad, 2023; Böck *et al.*, 2019; Cucchiarelli *et al.*, 2021; Ghosh *et al.*, 2023; Karthikeyan *et al.*, 2023; Lazarus *et al.*, 2023; Marques *et al.*, 2021a; Mizerek *et al.*, 2023; Mvula *et al.*, 2022; Nanath & Olney, 2023; Palaniappan *et al.*, 2020a; Steinebach *et al.*, 2021; Suchacka *et al.*, 2021; Takefuji, 2023; Zheng *et al.*, 2022).

Malicious domain adalah nama *domain* yang digunakan untuk tujuan berbahaya oleh individu atau kelompok dengan niat jahat. *Domain* ini sering kali digunakan sebagai infrastruktur untuk menjalankan serangan siber, mengarahkan pengguna yang tidak curiga ke situs berbahaya, atau untuk mencuri informasi pribadi dan finansial (Ahammad *et al.*, 2022; Dong *et al.*, 2022; Manasrah *et al.*, 2022; Sugawara *et al.*, 2023). Ancaman ini semakin kompleks karena teknik-teknik canggih yang digunakan oleh pelaku, seperti *DNS hijacking*, *domain shadowing*, dan pemanfaatan *domain* dinamis, membuat deteksi dan mitigasi menjadi semakin sulit (X. J. Li *et al.*, 2019; Qu *et al.*, 2021; Shen *et al.*, 2022; Zhao *et al.*, 2020).

Keberadaan *malicious domain* telah menjadi perhatian utama bagi para profesional keamanan siber dan organisasi di seluruh dunia. Hal ini disebabkan oleh dampak yang luas dan berpotensi merugikan bagi individu, perusahaan, dan bahkan keamanan nasional (Hausken, 2020a, 2020b; Mishra *et al.*, 2022). Misalnya, serangan *phishing* yang menggunakan *domain* palsu telah menyebabkan kerugian finansial yang sangat besar, sementara *domain* yang digunakan untuk menyebarkan *malware* dapat menginfeksi jutaan perangkat dalam waktu singkat (Gusev, 2022; Varga *et al.*, 2021; Yuryna Connolly &

Borrion, 2022). Selain itu, regulasi dan kebijakan internasional terkait keamanan siber dan penggunaan *domain* juga semakin ketat, mendorong pentingnya penelitian di bidang ini. Pemerintah dan organisasi global seperti ICANN (*Internet Corporation for Assigned Names and Numbers*) terus berupaya memperkuat kerangka kerja untuk mengawasi dan mengelola pendaftaran *domain* agar tidak disalahgunakan (Mota & Cilento, 2020).

Pertumbuhan *Internet Service Provider (ISP)* dan pengembang jaringan komputer di Indonesia memberikan kontribusi yang sangat besar dalam mendukung perkembangan *landscape* transformasi digital di Indonesia. Menurut Badan Pusat Statistik Indonesia (Badan Pusat Statistik, 2022), pertumbuhan *ISP* dan pelanggan *internet* menunjukkan peningkatan yang pesat di tahun 2022. Dari 331 *ISP* di tahun 2018, berkembang menjadi 828 *ISP*. Dalam survei tersebut disampaikan 66.48% penduduk Indonesia telah memiliki akses *internet*, dimana 91.95% diantaranya didominasi oleh pelanggan perorangan. Perkembangan jaringan *internet* ke pelosok-pelosok tanah air menjadikan tantangan tersendiri untuk semua kalangan.

Keamanan siber saat ini sudah menjadi tanggung jawab bersama antara pengguna *internet* dan *ISP* (APJII - UMKM & Korporasi, 2023). Namun, tantangan utama yang dihadapi adalah *ISP* yang sedang berkembang rata-rata memiliki kemampuan finansial yang kurang memadai untuk membuat sebuah perlindungan yang kompleks bagi pelanggan. Kondisi ini semakin diperparah oleh fakta bahwa menurut survei Asosiasi Penyelenggara Jasa *Internet* Indonesia (APJII) (APJII - UMKM & Korporasi, 2023), 70.28% UMKM dan Korporasi belum memiliki prosedur mitigasi terhadap ancaman siber, dengan alasan terbanyak dikarenakan tidak memiliki dana untuk membuat prosedur mitigasi. Urgensi masalah ini terlihat dari laporan Survei APNIC (APNIC, 2022) yang menyajikan data bahwa ancaman *phishing*, spam, *malware*, dan *ransomware* menduduki posisi 42%. Adapun kendala utama untuk membangun *system firewall* dan sistem mitigasi serangan siber meliputi biaya aplikasi, lisensi dan mesin yang sangat tinggi, serta sumber daya manusia yang kurang memadai. Hal ini dikonfirmasi oleh survei APJII (APJII - ISP, 2023) yang mendapatkan data 53.40% responden menyatakan tidak memiliki sumber daya manusia untuk menangani mitigasi serangan siber. Selain masalah SDM, mahalnya dan kompleksnya *system firewall* yang ada di pasaran juga menjadi kendala tersendiri bagi *ISP* yang tidak memiliki sumber daya manusia yang memadai. Kondisi ini menjadi semakin krusial mengingat survei dari 562 *responder ISP* di atas juga memberikan informasi bahwa dilihat dari modal, 85% *ISP* di Indonesia tergolong Usaha Mikro dan Kecil, yang hanya memiliki modal usaha maksimum 1M atau pendapatan maksimum per tahun 2M, serta 65.20% hanya

didukung karyawan yang berjumlah kurang dari 10 orang. Sementara itu, kebutuhan pengamanan pelanggan dalam bentuk perimeter layanan keamanan sudah tidak bisa ditawarkan lagi, mengingat tingginya ancaman siber yang ada dan dampak kerugian yang dapat ditimbulkan. Kesenjangan antara kebutuhan keamanan yang mendesak dengan keterbatasan sumber daya yang dimiliki oleh UMKM, Korporasi, dan ISP di Indonesia menciptakan celah keamanan yang perlu segera diatasi melalui solusi yang terjangkau dan mudah diimplementasikan.

Dalam penelitiannya Team tanggap insiden pertama yang dibentuk di Polandia *CERT Polska*, dibawah naungan *NASK (Scientific and Academic Computer National)*, sebuah lembaga penelitian ilmiah yang melakukan kegiatan ilmiah, beralamat *web <https://cert.pl>* menyediakan layanan database *malicious domain* yang dilaporkan oleh pengguna *internet* di seluruh dunia. Salah satu fokus penelitiannya adalah mengumpulkan data *malicious domain* di seluruh dunia, memverifikasi, dan mengkonfirmasi untuk dijadikan sebuah *black-list*. *Domain* dikategorikan sebagai sebuah *malicious* karena di dalam *domain* tersebut tersembunyi aktifitas *bothnet*, *fake investments online*, *fake email attachment*, *request for quick transfer*, *fake online store*, *online game phishing*, judi *online*, dan lain-lain. Kontribusi dari penelitian ini, menyediakan database yang bisa digunakan oleh siapa saja yang membutuhkan. *Database* ini dibuat dalam format yang bisa dijalankan di beberapa *platform router* sebagai kelengkapan atribut *Firewall*.

Penelitian yang dilakukan oleh Emara. A (Emara *et al.*, 2019) mengusulkan tentang pendekatan baru dalam pembuatan *Real Time Firewall Policy Object* serta membuat *tools Firewall Policy Manager (FPM)*. Penelitian ini mengembangkan pendekatan baru untuk mengakuisisi kebijakan *firewall* secara *real-time* tanpa mengubah konfigurasi *firewall* yang sedang berjalan. Menghasilkan *tool FPM* yang dapat bekerja lintas *vendor firewall* menggunakan arsitektur *REST API*.

Jaidi F (Jaïdi, 2019) dalam penelitiannya memberikan kontribusi praktis meningkatkan kualitas layanan *firewall filtering* melalui mekanisme *self-adaptive* yang dapat merespons perubahan kondisi keamanan secara *real-time*. Hal ini menghadirkan solusi untuk masalah fundamental dalam manajemen *firewall* yaitu ketergantungan pada konfigurasi manual yang sering kali tidak optimal dan tidak responsif terhadap ancaman yang berkembang, sehingga menghasilkan sistem keamanan yang lebih proaktif dan *intelligent*.

Han S (Han & Yu, 2021) dalam penelitiannya memberikan solusi untuk implementasi sistem analisis *log firewall* otomatis yang lebih efisien dan *scalable*. Dengan

mengurangi kompleksitas fitur melalui *optimized feature selection*, sistem dapat melakukan *monitoring* dan klasifikasi *traffic* jaringan dalam waktu yang lebih singkat, yang sangat penting untuk aplikasi *real-time security monitoring* dan *incident response* dalam lingkungan jaringan yang memproses volume *traffic* tinggi.

Dalam penelitiannya (Ismail *et al.*, 2022) Penelitian ini menghasilkan kesimpulan praktis berupa dua metode utama pencegahan *privacy violation*: penggunaan *computer security software* dan *end user awareness training*. Kontribusi ini memberikan *guidance* yang *actionable* bagi individu dan organisasi di Indonesia untuk mengimplementasikan strategi perlindungan privasi yang efektif, dengan mempertimbangkan konteks spesifik tingkat adopsi teknologi di Indonesia yang terus meningkat.

(Miloslavskaya *et al.*, 2022) dalam penelitiannya menghasilkan framework yang dapat membantu spesialis *Information Security (IS)* untuk memastikan *level* keamanan intranet yang dibutuhkan dan direkomendasikan. Model yang dikembangkan bersifat adaptif karena mengakui bahwa objek *IS* dalam berbagai *intranet* memiliki karakteristik yang berbeda (*vulnerabilities, threat models, acceptable risks, incident response structure*), sehingga *basic set of Information Protection Tools (IPTs)* dapat disesuaikan dengan kebutuhan spesifik organisasi, memberikan fleksibilitas implementasi yang praktis untuk berbagai jenis organisasi.

Penelitian (Yoshida *et al.*, 2020) mengembangkan metode klasifikasi baru yang berbasis *cardinality analysis* untuk mengidentifikasi pola akses *multiple clients* terhadap berbagai *domain names*. Teknik ini berupa pendekatan inovatif yang memanfaatkan analisis *behavioral pattern* dalam *DNS query-response* data untuk mendeteksi aktivitas *malicious*, berbeda dari pendekatan *blacklist* tradisional yang mengandalkan daftar *host* yang sudah dikenal berbahaya. Penelitian ini divalidasi menggunakan dataset besar berupa 26 juta *DNS query-response* data, menunjukkan bahwa metode yang diusulkan dapat bekerja pada skala *real-world* yang *massive*. Tidak hanya teoritis efektif tetapi juga praktis *implementable* untuk mengatasi tantangan *malicious hosts* baru yang muncul setiap hari.

Penelitian (Hadjadj *et al.*, 2019) mengoptimalkan aturan pemfilteran pada *firewall*, khususnya dalam konteks arsitektur *firewall paralel*. Dengan meningkatnya kompleksitas dan ukuran kebijakan pemfilteran, penelitian ini berfokus pada pengembangan arsitektur baru yang dapat mengurangi waktu pemrosesan dan meningkatkan efisiensi dalam pengelolaan kebijakan keamanan yang besar. Penelitian ini menawarkan arsitektur *firewall paralel* yang menggabungkan keuntungan dari pemrosesan fungsi paralel dan pengurangan duplikasi data. Arsitektur ini memungkinkan pemrosesan paket tanpa menduplikasi data,

yang mengurangi waktu pemrosesan secara signifikan. Secara keseluruhan, penelitian ini memberikan kontribusi penting dalam pengembangan arsitektur *firewall* yang lebih efisien dan efektif, dengan fokus pada pengelolaan kebijakan pemfilteran yang kompleks dan besar.

(Wang *et al.*, 2023) memperkenalkan *DDOFM* (*Dynamic Detection of Malicious Domain*), sebuah pendekatan berlapis yang menggabungkan analisis statistik, model probabilistik, dan algoritma divergensi untuk deteksi domain berbahaya. Penelitian ini mengombinasikan beberapa teknik canggih secara sinergis, *Hidden Markov Model (HMM)* untuk analisis probabilitas pembentukan domain, *Jensen-Shannon divergence* untuk mengukur perbedaan antar domain, *Local Iterative Threshold Finding Algorithm (LLTFA)* yang dikembangkan khusus untuk penelitian ini. Kontribusi praktis dari penelitian ini adalah akurasi tinggi, eksperimen menunjukkan tingkat deteksi yang melebihi 99% untuk berbagai jenis domain berbahaya, yang merupakan peningkatan signifikan dibanding metode tradisional. *Deteksi C&C Server*, sistem dapat mengidentifikasi koneksi ke *command and control server* lebih cepat dibanding metode serupa, yang sangat penting untuk *respons* keamanan *real-time*. Kemampuan generalisasi, mengatasi keterbatasan metode tradisional yang memiliki kemampuan generalisasi buruk terhadap *domain* berbahaya yang kompleks dan bervariasi.

Merujuk pada hasil penelitian sebelumnya, terdapat kesadaran yang semakin meningkat terhadap ancaman *malicious domain traffic*. Namun, beberapa kekurangan masih ditemukan dalam penelitian-penelitian terdahulu. Pertama keterbatasan implementasi dan integrasi terhadap *existing product* yang digunakan oleh *ISP*, seperti *firewall policy management* (Emara *et al.*, 2019), analisis risiko kebijakan *firewall* (Jaidi, 2019), dan *log analysis* berbasis *machine learning* (Han & Yu, 2021), tanpa menyediakan solusi langsung yang dapat diintegrasikan dalam sistem jaringan *ISP* untuk pengelolaan dan mitigasi ancaman secara *real-time*. Kedua, kurangnya fokus pada lingkungan *ISP*, penelitian cenderung berorientasi pada jaringan internal atau perusahaan (*intranet*), seperti yang dilakukan oleh (Miloslavskaya *et al.* 2022) dan (Hadjadj *et al.*, 2019), tanpa mempertimbangkan kebutuhan khusus *ISP* dalam menghadapi *traffic domain* berbahaya yang terjadi dalam skala besar dan heterogen. Ketiga, belum adanya solusi ekonomis dan skalabel. Penelitian yang menawarkan pendekatan inovatif, seperti *bee swarm optimization* atau *Software-Defined Perimeter* (Han & Yu, 2021; Moubayed *et al.*, 2019), sering kali menghadirkan tantangan implementasi dalam hal kompleksitas dan biaya, sehingga kurang sesuai untuk adopsi di lingkungan *ISP* di Indonesia. Dalam survei APJII (APJII - ISP, 2023), dilihat dari modal maksimum sebesar 1M atau pendapatan maksimum 2M per tahun, jumlah

karyawan dan sumber dayanya, 85% *ISP* di Indonesia tergolong Usaha Mikro dan Kecil. Keempat, minimnya integrasi *live filtering* system, meskipun beberapa penelitian telah berfokus pada pengelolaan *blacklist* atau pengembangan metode klasifikasi *domain* berbahaya (Yoshida *et al.*, 2020), belum menawarkan integrasi langsung ke sistem *live filtering* di lingkungan *ISP* untuk secara otomatis mendeteksi dan memblokir *traffic domain* berbahaya.

Berdasarkan kekurangan tersebut, penelitian ini mengusulkan desain dan analisis metode pengondisian lalu lintas *internet* yang bersih dan minim ancaman (*Clear and Less Malicious Internet Traffic*) melalui penerapan sistem penyaringan *domain* berbahaya (*malicious domain filtering*) yang terintegrasi dalam layanan keamanan *perimeter ISP*. Untuk menjawab tantangan keamanan jaringan yang semakin kompleks, penelitian ini bertujuan memberikan solusi yang ekonomis dan praktis bagi *ISP* untuk memanfaatkan infrastruktur yang sudah ada tanpa memerlukan investasi besar. Sistem yang diusulkan menawarkan pendekatan sederhana namun efektif untuk melindungi pelanggan *ISP* dari ancaman serangan siber.

Dalam upaya mencapai tujuan tersebut, penelitian ini mengajukan beberapa hipotesis kunci yang akan diuji secara komprehensif. Integrasi sistem penyaringan *domain* berbahaya secara waktu nyata (*real-time*) diharapkan dapat mengurangi lalu lintas berbahaya (*malicious traffic*), sehingga meningkatkan efisiensi dan keamanan jaringan *ISP*.

Secara spesifik, hipotesis yang diajukan dalam penelitian ini meliputi:

- Penggunaan teknik *filtering malicious domain* pada *firewall perimeter* dapat mengidentifikasi dan memblokir *domain* berbahaya secara efektif.
- Implementasi *filtering domain* berbahaya akan mengurangi volume *traffic* berbahaya secara signifikan tanpa mengorbankan performa jaringan secara keseluruhan.
- Metode *filtering* yang diusulkan akan meningkatkan efisiensi *traffic* jaringan *ISP* yang dibuktikan dengan pengurangan metrik *pps* (*packet per second*) dan *bps* (*byte per second*) yang tidak diperlukan.

Selain itu, penelitian ini diharapkan dapat memberikan kontribusi signifikan melalui rekomendasi praktis bagi praktisi keamanan siber dan kebijakan yang dapat diadopsi oleh organisasi. Dengan demikian, penelitian ini tidak hanya mengisi celah dari penelitian sebelumnya, tetapi juga menawarkan solusi aplikatif untuk tantangan nyata di lapangan, khususnya dalam konteks keamanan jaringan *ISP* yang semakin kompleks dan rentan terhadap ancaman dunia maya.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, beberapa rumusan masalah yang akan dicari solusinya adalah sebagai berikut:

- a. Dimana bisa ditemukan artefak dan informasi berupa apa?
- b. Apakah metode *firewall malicious domain filtering* yang mengadopsi *black-list* yang diusulkan sesuai untuk digunakan oleh *ISP*, dengan mempertimbangkan perangkat yang dimiliki, sumber daya manusia yang tersedia, dan kebutuhan investasi yang minimal?
- c. Bagaimana dampak dan seberapa efisien antara penerapan *firewall* yang dirancang dengan *traffic flow* dalam jaringan *ISP*?

1.3 Batasan Masalah

Penelitian ini memiliki batasan-batasan sebagai berikut:

- a. Menggunakan parameter *black list* yang telah ada dalam penelitian terdahulu.
- b. Metode *firewall* yang digunakan disesuaikan dengan fitur perangkat jaringan yang tersedia pada *ISP* yang menjadi objek penelitian.
- c. Korelasi yang akan diteliti berkaitan dengan penggunaan *firewall* terhadap metrik *packet per second* dan *byte per second* dalam *traffic flow* di jalur *internet*.

1.4 Tujuan Penelitian

Tujuan utama penelitian ini meliputi:

- a. Memberikan solusi yang ekonomis dan praktis untuk *ISP* guna memanfaatkan infrastruktur yang sudah ada tanpa memerlukan investasi besar.
- b. Menawarkan pendekatan sederhana namun efektif untuk melindungi pelanggan *ISP* dari ancaman *cyber attacks*.
- c. Mengintegrasikan *malicious domain filtering* secara *real-time* untuk mengurangi *malicious traffic*, sehingga meningkatkan efisiensi dan keamanan jaringan *ISP*.

1.5 Manfaat Penelitian

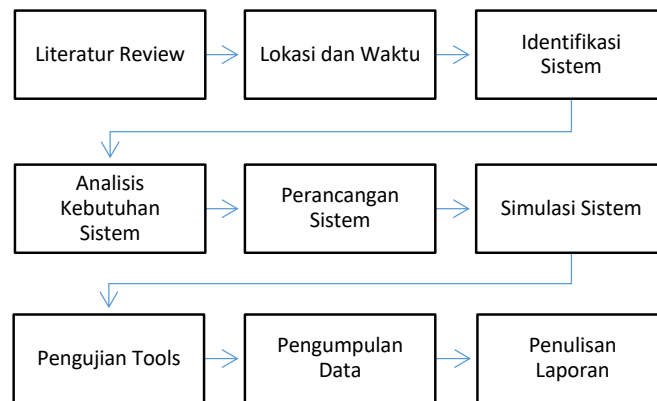
Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Memberikan kontribusi ekonomis serta menawarkan solusi *firewall* sederhana kepada *ISP* untuk melakukan *filtering malicious domain* dalam infrastruktur, guna melindungi pelanggan dari ancaman *cybercrime*.

- b. Membersihkan jalur *internet* antara *ISP* dan pelanggan, sehingga pelanggan dapat menikmati jalur yang bebas dari *malicious traffic* serta jalur yang lebih efisien.
- c. Memberikan rekomendasi praktis kepada praktisi keamanan siber dan kebijakan yang dapat diadopsi oleh organisasi untuk memperkuat pertahanan terhadap ancaman dari *malicious domain*.

1.6 Metodologi Penelitian

Metode diperlukan untuk menyusun dan melaksanakan langkah-langkah penelitian secara sistematis dan terstruktur. Gambar 1.1 Menggambarkan tahapan metode yang digunakan, mencakup berbagai proses yang dirancang untuk mencapai tujuan penelitian secara efektif.



Gambar 1.1 Tahapan metode penelitian.

1.7 Review Penelitian

Bagian ini mengulas penelitian-penelitian terdahulu yang relevan dengan topik yang diangkat. Penelitian sebelumnya menjadi referensi untuk memahami parameter, metode, atau pendekatan yang telah dilakukan, serta memberikan dasar dalam mengembangkan inovasi atau penyempurnaan pada penelitian ini.

1.8 Sistematika Penulisan

Pembahasan dalam penelitian ini disusun berdasarkan sistematika berikut:

BAB I PENDAHULUAN

Bab ini berisi pengantar penelitian yang mencakup latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, ulasan penelitian sebelumnya, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan dasar dan landasan teori yang relevan dengan penelitian. Teori yang dibahas meliputi konsep-konsep terkait forensik jaringan *internet* dan teknologi yang digunakan untuk mengatasi masalah yang diangkat.

BAB III METODOLOGI PENELITIAN

Bab ini memaparkan lokasi dan waktu penelitian, metode yang digunakan, serta tahapan-tahapan penelitian yang dilakukan untuk mencapai tujuan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan hasil uji coba, data-data penelitian yang diperoleh, analisis, serta solusi dari permasalahan yang dirumuskan dalam penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan berdasarkan hasil penelitian, saran untuk mengatasi keterbatasan penelitian, dan rekomendasi untuk pengembangan penelitian lebih lanjut di masa depan.

BAB 2

Tinjauan Pustaka

2.1 Forensik dan Forensik Jaringan

Forensik adalah ilmu yang diterapkan untuk mengumpulkan, menganalisis, dan menyajikan bukti terkait dengan tindak kejahatan yang dapat digunakan dalam sistem peradilan. Pengumpulan data dalam bidang ini didasarkan pada metode ilmiah yang dapat dipertanggungjawabkan secara akademis. Seiring dengan perkembangan teknologi komputer, muncul cabang baru dalam ilmu forensik, yaitu forensik digital. Forensik digital merupakan aplikasi ilmu pengetahuan dan teknologi komputer untuk mendukung proses pembuktian hukum, khususnya dalam mengungkap kejahatan melalui analisis data digital yang tersimpan dalam perangkat komputer.

Perkembangan jaringan komputer dan *internet* turut melahirkan cabang baru dari forensik digital, yaitu forensik jaringan. Forensik jaringan adalah sub-bidang yang secara khusus berfokus pada penyelidikan lalu lintas jaringan (*network traffic*), yaitu data yang bergerak di dalam jaringan, dengan tujuan mengidentifikasi ancaman atau aktivitas kejahatan yang terjadi melalui jaringan komputer. Penyelidikan ini mencakup analisis data yang ditransmisikan melalui jaringan, termasuk paket-paket data yang dikirimkan antar perangkat di *internet* maupun intranet. Langkah-langkah dalam forensik jaringan mencakup:

1. Pengumpulan data lalu lintas jaringan, meliputi *log server*, *log firewall*, dan data yang diperoleh dari alat pemantauan jaringan.
2. Menganalisis, untuk mendeteksi potensi serangan, kebocoran data, atau aktivitas tidak sah lainnya.
3. Merekonstruksi kejadian atau serangan guna memahami mekanisme pelanggaran yang terjadi, termasuk sumber dan metode serangan yang digunakan.

2.2 Arsitektur Jaringan *Internet*

Arsitektur jaringan *internet* adalah struktur atau rancangan yang mengatur bagaimana data dikirim, diterima, dan diproses antara perangkat-perangkat di dalam sebuah jaringan yang saling terhubung. *Internet* terdiri dari berbagai lapisan dan komponen yang bekerja secara sinergis untuk memungkinkan komunikasi global. Berikut ini adalah elemen-elemen utama dalam arsitektur jaringan *internet*:

1. Protokol dan Lapisan komunikasi

Arsitektur *internet* menggunakan pemodelan *TCP/IP* (*Transmission Control Protocol/Internet Protocol*) sebagai dasar komunikasinya. Model ini dibagi beberapa lapisan:

- a. Lapisan Aplikasi (*Application Layer*), merupakan tempat aplikasi *internet* seperti *web browser*, *email*, dan *transfer file* beroperasi. Protokol disini meliputi *HTTP*, *HTTPS*, *FTP*, *SMTP*, *DNS*, dan lainnya.
- b. Lapisan Transport (*Transport Layer*), lapisan ini bertanggungjawab untuk memastikan pengiriman data antar perangkat, biasanya melalui *TCP* (*reliable, connection oriented*) atau *UDP* (*Uni Datagram Protocol*) (*unreliable, connectionless*).
- c. Lapisan Internet (*Internet Layer*), merupakan Protokol *IP* (*IPv4* atau *IPv6*) beroperasi disini, untuk menentukan alamat *IP* untuk setiap perangkat dan mengarahkan paket data ke tujuannya
- d. Lapisan Network Access (*Physical Layer* dan *Data Link Layer*), berperan dalam transmisi data melalui media fisik seperti kabel *Ethernet*, *Fiber Optic* atau *Wireless*.

2. Router dan Switch

Router dan *switch* adalah perangkat yang digunakan mengarahkan lalu lintas data dalam jaringan. *Router* berfungsi untuk menghubungkan jaringan yang berbeda dan memutuskan jalur terbaik dalam mengirimkan data antar jaringan. *Switch* digunakan dalam jaringan lokal (*LAN*) untuk menghubungkan perangkat-perangkat dalam jaringan.

3. Topologi Jaringan

Topologi menggambarkan bagaimana perangkat di dalam jaringan saling terhubung secara fisik maupun secara logis. Beberapa topologi umum yang digunakan dalam jaringan *internet* sebagai berikut:

- a. Topologi *Star*, Semua perangkat terhubung ke satu perangkat pusat, misalnya *hub* atau *switch hub*.
- b. Topologi *Bus*, semua perangkat terhubung ke satu kabel utama
- c. Topologi *Mesh*, setiap perangkat terhubung secara langsung ke banyak perangkat lainnya sehingga menciptakan *redundancy*

4. *DNS (Domain Name System)*

DNS bertindak sebagai “buku telepon” *internet*, yang memungkinkan pengguna mengakses situs *web* menggunakan nama *domain* seperti www.example.com, bukan menggunakan alamat numerik.

5. *Internet Service Provider(ISP)*

ISP adalah organisasi yang menyediakan akses *internet* kepada pengguna. *ISP* menghubungkan pengguna ke infrastruktur jaringan global melalui *router* dan *backbone internet*.

6. *Firewall* dan keamanan Jaringan

Firewall dan keamanan jaringan digunakan untuk melindungi jaringan dari ancaman *external* dengan memfilter lalu lintas berbahaya dan memastikan hanya data yang diijinkan yang dapat masuk atau keluar jaringan.

7. Infrastruktur Jaringan Global (*Backbone Internet*)

Backbone internet adalah serangkaian koneksi data berkecepatan tinggi dan jalur yang menghubungkan berbagai wilayah dunia. Infrastruktur ini memungkinkan pertukaran data antar benua dan mencakup kabel bawah laut, pusat data dan *server-server*.

8. *Cloud Computing* dan Virtualisasi

Dalam tahun-tahun terakhir, penggunaan komputasi awan (*Cloud Computing*) menjadi semakin populer. Layanan *cloud* seperti *Amazon Web Services (AWS)*, *Microsoft Azure*, dan *Google Cloud Platform (GCP)* memungkinkan penyimpanan dan pemrosesan data di *server* yang dikelola di pusat data seluruh dunia.

9. *Peer to Peer (P2P) vs Client Server*

Dalam arsitektur jaringan, terdapat dua model komunikasi:

- a. *Client Server*, dimana *server* menyediakan layanan atau data dan *client* meminta data atau layanan ke *server*
- b. *Peer-to-Peer*, dimana semua perangkat dalam jaringan dapat bertindak sebagai *client* maupun *server*, berbagi data secara langsung tanpa perantara.

10. *IPv4* dan *IPv6*

- a. *IPv4* merupakan protokol *IP* yang paling banyak digunakan, namun dengan keterbatasan jumlah alamat yang dapat disediakan (sekitar 4,3 milyar alamat).

- b. *IPv6*, dibuat untuk menggantikan *IPv4* dengan kemampuan untuk menyediakan jauh lebih banyak alamat (sekitar 340 undecillion alamat).

2.3 *Domain Name*

Domain Name (Nama *Domain*) adalah alamat unik yang digunakan untuk mengidentifikasi situs *web* atau alamat *server* di *internet*. *Domain name* bertindak sebagai pengganti dari alamat *IP* numerik yang sulit diingat, sehingga memungkinkan pengguna untuk mengakses situs *web* dengan lebih mudah.

2.3.1 *Arsitektur Domain Name*

Arsitektur Domain Name System (DNS) adalah sistem yang terdistribusi dan hierarkis, dirancang untuk menerjemahkan nama *domain* yang mudah diingat oleh manusia menjadi alamat *IP* yang diperlukan oleh perangkat komputer untuk berkomunikasi di jaringan *internet*. Berikut adalah penjelasan lengkap tentang arsitektur *DNS*:

1. *Komponen Utama Arsitektur DNS*

a. *DNS Resolver (Client Side)*

DNS Resolver adalah komponen yang bertanggung jawab untuk memulai dan mengelola proses pencarian *DNS*. Ini biasanya dilakukan oleh perangkat pengguna, seperti komputer, smartphone, atau *router*. *DNS Resolver* menerima permintaan dari aplikasi (misalnya, *browser web*) dan memulai proses pencarian untuk menemukan alamat *IP* yang terkait dengan nama *domain*.

- b. *Caching Resolver* menyimpan hasil pencarian *DNS* sebelumnya untuk waktu tertentu (*TTL—time to live*) guna mempercepat permintaan selanjutnya.

2. *Root Name Server*

Root Name Server adalah *server* pada tingkat paling atas dari hierarki *DNS*. Saat *DNS Resolver* tidak dapat menemukan jawaban dalam *cache*-nya, ia akan mengirimkan permintaan ke *Root Server*. *Root Server* tidak memiliki jawaban langsung untuk nama *domain*, tetapi mengarahkan permintaan ke *Top-Level Domain (TLD) Server* yang relevan. Ada 13 grup *root server* di seluruh dunia yang dioperasikan oleh berbagai organisasi, dan mereka terdistribusi untuk keandalan dan ketersediaan tinggi.

3. *Top-Level Domain (TLD) Server*

TLD Server adalah *server* yang bertanggung jawab untuk mengelola *domain* tingkat atas seperti *.com*, *.org*, *.net*, atau *domain* negara seperti *.id* (untuk Indonesia) atau *.uk* (untuk Inggris). Setelah permintaan dikirim dari *Root Server*, *TLD Server* akan mengarahkan permintaan ke *Authoritative Name Server* yang relevan.

4. *Authoritative Name Server*

Authoritative Name Server adalah *server* yang menyimpan informasi *DNS* lengkap untuk *domain* yang spesifik, termasuk peta alamat *IP* dari nama *domain* yang diminta. Ketika *DNS Resolver* mencapai *Authoritative Name Server*, ia akan menerima informasi yang diperlukan untuk menyelesaikan permintaan (misalnya, alamat *IP* untuk nama *domain* tersebut). Setelah itu, *DNS Resolver* akan mengirim jawaban kembali ke perangkat pengguna.

5. Arsitektur Fisik *DNS*

- a. Desentralisasi dan Replikasi: *DNS* terdiri dari *server* yang tersebar di seluruh dunia, menjadikannya desentralisasi dan lebih tahan terhadap kegagalan atau serangan. *Server DNS* juga direplikasi untuk meningkatkan kecepatan dan ketahanan.
- b. *Caching*: Untuk mengurangi beban jaringan dan mempercepat proses pencarian, *DNS Resolver* dan *server DNS* menggunakan mekanisme *caching*. Informasi *DNS* yang pernah dicari akan disimpan sementara waktu sehingga pencarian berikutnya lebih cepat.

2.3.2 Proses Resolusi *DNS* (*Query DNS*)

Mekanisme *DNS request* (permintaan *DNS*) merujuk pada proses yang dilakukan oleh komputer atau perangkat jaringan untuk mengonversi nama *domain* yang dapat dibaca manusia (seperti *www.example.com*) menjadi alamat *IP* yang dapat dikenali oleh komputer atau jaringan (misalnya, 192.0.2.1). *DNS* (*Domain Name System*) berfungsi sebagai buku telepon *internet*, menghubungkan nama *domain* ke alamat *IP* sehingga pengguna dapat mengakses situs *web* dan layanan lainnya. Berikut adalah tahapan mekanisme *DNS request*:

1. Pengguna Mengirim Permintaan *DNS* (*Client Request*)

Ketika seorang pengguna atau perangkat mencoba mengakses situs *web* (misalnya, mengetikkan *www.example.com* di *browser*), permintaan *DNS* dikirim dari *resolver*

DNS yang ada di perangkat pengguna ke *server DNS* untuk mendapatkan alamat *IP* yang sesuai dengan nama *domain*.

Resolver ini adalah bagian dari sistem operasi perangkat (biasanya bagian dari *DNS client*) yang berperan sebagai penghubung antara aplikasi (misalnya *browser*) dengan *server DNS*.

2. *Cache Resolver DNS Lokal*

Langkah pertama yang dilakukan *resolver* adalah memeriksa apakah alamat *IP* untuk *domain* yang diminta sudah tersimpan di *cache* perangkat (hasil permintaan *DNS* sebelumnya). Jika ada dan masih *valid* (belum kadaluwarsa), *resolver* akan mengembalikan alamat *IP* tersebut ke aplikasi tanpa perlu melakukan permintaan tambahan ke *server DNS* eksternal. Jika *cache* tidak ada atau sudah kadaluwarsa, proses berlanjut ke langkah berikutnya.

3. *Recursive DNS Query ke Resolver DNS (DNS Recursor)*

Jika *cache* lokal kosong, *recursive DNS resolver* atau *DNS recursor* (biasanya disediakan oleh *ISP* atau pihak ketiga seperti *Google Public DNS*) akan menerima permintaan dari perangkat pengguna. *DNS resolver* ini bertugas untuk mencari alamat *IP* terkait dengan nama *domain* yang diminta. *DNS resolver* ini akan bertindak sebagai perantara dan melakukan pencarian berjenjang melalui berbagai *server DNS* untuk menemukan informasi yang diperlukan.

4. Pencarian di *Cache Resolver DNS*

Sebelum berkomunikasi dengan *server DNS* lainnya, *DNS resolver* juga akan memeriksa *cache*-nya sendiri. Jika alamat *IP domain* yang diminta ada dalam *cache resolver* dan masih *valid*, *DNS resolver* akan langsung mengembalikan jawaban ke *client*. Jika tidak ada dalam *cache*, *DNS resolver* akan melanjutkan pencarian ke *server DNS* lainnya.

5. Permintaan ke *Root Name Server*

Jika tidak ada hasil dari *cache*, *DNS resolver* akan memulai proses pencarian dari nol dengan mengirimkan permintaan (*query*) ke salah satu *root name server*. *Root name server* adalah *server DNS* yang berada di puncak hierarki *DNS* dan berfungsi sebagai titik awal pencarian untuk semua *domain*. *Root name server* tidak tahu langsung alamat *IP* dari *www.example.com*, tetapi ia tahu ke mana harus mengarahkan *resolver* untuk mencari lebih lanjut. *Root name server* akan mengarahkan *DNS resolver* ke top-level *domain (TLD) name server* yang sesuai berdasarkan ekstensi *domain* yang diminta (misalnya, *.com*, *.org*, *.net*).

6. Permintaan ke *TLD Name Server*

DNS resolver kemudian mengirimkan permintaan ke *TLD name server* yang bertanggung jawab atas *domain* tingkat atas. Misalnya, untuk *domain* *www.example.com*, *resolver* akan berkomunikasi dengan *TLD name server* yang mengelola *domain* *.com*. *TLD name server* tidak menyimpan informasi *IP* lengkap untuk *www.example.com*, tetapi ia tahu ke mana harus mengarahkan permintaan selanjutnya. *TLD name server* akan mengarahkan *resolver* ke *authoritative name server* untuk *domain* tersebut.

7. Permintaan ke *Authoritative DNS Server*

Tahap akhir pencarian adalah permintaan ke *authoritative DNS server*. Ini adalah *server DNS* yang secara langsung menyimpan informasi *DNS* (seperti alamat *IP*) untuk *domain* yang diminta. Misalnya, *authoritative DNS server* untuk *www.example.com* akan memiliki catatan A (*Address Record*) yang menghubungkan *www.example.com* dengan alamat *IP* spesifiknya, seperti 192.0.2.1. *Authoritative name server* kemudian mengirimkan jawaban ke *DNS resolver* dengan alamat *IP* yang diminta.

8. *Resolver* Mengirim Jawaban ke *Client*

Setelah *DNS resolver* menerima alamat *IP* dari *authoritative name server*, ia menyimpan informasi tersebut dalam *cache*-nya untuk jangka waktu tertentu (disebut *Time To Live*, atau *TTL*) sehingga jika ada permintaan *DNS* serupa di masa depan, *resolver* dapat langsung merespon dari *cache* tanpa harus mengulangi proses pencarian. *Resolver* kemudian mengirimkan alamat *IP* tersebut kembali ke *client* (browser atau aplikasi lainnya), yang kemudian menggunakan alamat *IP* tersebut untuk membuat koneksi ke *server* tujuan dan mengambil konten yang diminta.

2.4 *Malicious Internet Traffic*

Malicious internet traffic merujuk pada lalu lintas data yang dikirim melalui jaringan *internet* dengan tujuan yang bersifat merugikan, seperti mencuri informasi sensitif, merusak sistem, atau mengganggu operasional jaringan. Lalu lintas ini sering kali dimanfaatkan oleh pelaku kejahatan siber untuk meluncurkan berbagai jenis serangan, termasuk distribusi *malware*, *phishing*, serangan *denial-of-service (DoS)*, *command-and-control (C&C) botnet*, hingga eksploitasi kerentanan sistem. Karakteristik *malicious internet traffic* dapat mencakup pola data yang mencurigakan, koneksi ke *domain* atau *server* berbahaya, serta volume atau frekuensi lalu lintas yang tidak wajar. Pelaku sering menyamarkan lalu lintas

berbahaya ini agar terlihat seperti lalu lintas yang sah untuk menghindari deteksi oleh sistem keamanan, misalnya dengan menggunakan enkripsi atau layanan anonim seperti *Virtual Private Network (VPN)*. Oleh karena itu, upaya deteksi dan pencegahan menjadi tantangan yang terus berkembang seiring dengan semakin canggihnya metode yang digunakan oleh pelaku ancaman.

Dalam upaya mitigasi, berbagai pendekatan dan teknologi digunakan untuk mengidentifikasi dan memblokir *malicious internet traffic* secara efektif. Perangkat keamanan seperti *firewall*, sistem deteksi dan pencegahan *Intrusion Detection System / Intrusion Prevention System (IDS/IPS)*, serta solusi pemantauan berbasis *machine learning* atau kecerdasan buatan (*Artificial Intelligence*) memainkan peran penting dalam mengamankan jaringan. Selain itu, analisis *log* jaringan dan lalu lintas secara berkala dapat membantu dalam mendeteksi pola serangan dan mencegah ancaman di masa depan. Strategi keamanan yang efektif juga memanfaatkan teknologi *blacklist domain* untuk memblokir akses ke sumber berbahaya, serta metode reputasi *IP* untuk mengidentifikasi dan menanggulangi alamat *IP* yang terkait dengan aktivitas berbahaya. Namun, pendekatan teknis saja tidak cukup; pelatihan dan kesadaran keamanan bagi pengguna juga penting untuk mencegah serangan berbasis rekayasa sosial yang sering kali menjadi pintu masuk bagi lalu lintas berbahaya. Dengan kombinasi teknologi canggih, pemantauan yang proaktif, dan edukasi keamanan, ancaman dari *malicious internet traffic* dapat diminimalkan untuk melindungi data, jaringan, dan kepercayaan pengguna.

2.4.1 Malicious Domain

Malicious domain adalah nama *domain* yang dirancang dan digunakan oleh aktor jahat untuk melakukan aktivitas berbahaya di *internet*, seperti penyebaran *malware*, serangan *phishing*, pencurian data, atau pengelabuan pengguna. *Domain* berbahaya ini sering menjadi bagian dari serangan siber yang lebih luas dan digunakan untuk berbagai tujuan kriminal, seperti mencuri informasi sensitif, menyebarkan *ransomware*, atau menjalankan serangan *Distributed Denial of Service (DDoS)*.

Berikut adalah penjelasan lengkap tentang *malicious domain*, cara kerjanya, dan bagaimana *domain* ini digunakan oleh penyerang dalam berbagai aktivitas jahat:

1. Karakteristik *Malicious Domain*

- a. Tujuan Berbahaya

Malicious domain digunakan untuk tujuan jahat, termasuk penyebaran virus, *malware*, *ransomware*, atau mencuri kredensial pengguna (*phishing*). Beberapa

domain berbahaya mungkin tampak seperti *domain* asli dan terpercaya, tetapi sebenarnya dioperasikan oleh penjahat siber.

b. Penampilan yang Menipu

Banyak *malicious domain* dirancang agar terlihat mirip dengan situs *web* sah, menggunakan teknik seperti *typosquatting* atau *homograph attack* (memanfaatkan huruf yang mirip secara visual dari berbagai bahasa atau karakter *Unicode*). Misalnya, *domain* palsu seperti *g00gle.com* (menggunakan angka "0" alih-alih huruf "o") untuk meniru *Google*.

c. *Domain Disposable* atau Sementara

Banyak *malicious domain* hanya aktif untuk waktu yang singkat, karena digunakan dalam serangan kilat. Setelah serangan selesai atau *domain* terdeteksi sebagai berbahaya, penyerang sering menutup atau mengganti *domain* tersebut.

d. Penggunaan dalam Infrastruktur Serangan

Domain-domain ini sering digunakan sebagai bagian dari jaringan *command-and-control* (C2), yang memungkinkan penyerang untuk mengendalikan *malware* yang telah terinfeksi di perangkat korban. *Malicious domain* juga sering digunakan untuk *malvertising* (iklan berbahaya) atau sebagai titik distribusi *malware*.

2. Jenis-Jenis Serangan yang Melibatkan *Malicious Domain*

a. *Phishing*

Phishing domain adalah *domain* yang dirancang untuk mencuri informasi sensitif, seperti kredensial *login*, data kartu kredit, atau informasi pribadi.

Penyerang mengirim email atau pesan dengan tautan yang mengarahkan pengguna ke situs *web* palsu yang meniru situs sah (misalnya, bank atau layanan *email*). Pengguna tidak menyadari bahwa mereka mengunjungi situs berbahaya yang dirancang untuk mencuri informasi. Contoh: *paypal.com* (mengganti "l" dengan "1" agar terlihat seperti *domain* asli *Paypal*).

b. *Malware Distribution*

Malware domains digunakan untuk menyebarkan *malware*, seperti *virus*, *trojan*, *spyware*, atau *ransomware*. Pengguna diarahkan ke *domain* ini melalui berbagai cara, termasuk *email phishing*, iklan berbahaya (*malvertising*), atau unduhan perangkat lunak palsu. Begitu pengguna mengunjungi *domain* berbahaya ini, *malware* diunduh secara otomatis ke perangkat mereka.

c. *Command-and-Control* (C2)

Domain ini digunakan oleh penyerang untuk berkomunikasi dengan perangkat yang sudah terinfeksi *malware*. *C2 domains* berfungsi sebagai pusat kendali di mana penyerang bisa mengirim perintah ke perangkat yang dikendalikan, seperti mengumpulkan data yang dicuri atau meluncurkan serangan tambahan. Misalnya, *botnet* yang terinfeksi mungkin secara teratur menghubungi *server C2* melalui *domain* berbahaya untuk menerima instruksi baru dari penyerang.

d. *Typosquatting* dan *Homograph Attack*

Typosquatting memanfaatkan kesalahan ketik pengguna yang tidak disengaja saat mereka memasukkan nama *domain* di browser. Penyerang mendaftarkan *domain* yang mirip dengan *domain* populer tetapi dengan perbedaan kecil, seperti *goggle.com* (mengganti "o" dengan "a" atau "0").

Homograph attack adalah serangan di mana penyerang menggunakan karakter dari bahasa yang berbeda yang terlihat seperti huruf Latin untuk menipu pengguna agar mengunjungi situs berbahaya. Kedua metode ini dirancang untuk mengelabui pengguna agar tidak menyadari bahwa mereka mengunjungi situs berbahaya.

e. *Domain Generation Algorithm (DGA)*

Penyerang menggunakan *Domain Generation Algorithm (DGA)* untuk secara otomatis menghasilkan daftar besar nama *domain* yang digunakan sebagai *C2* atau untuk menyebarkan *malware*. *DGA* memungkinkan penyerang untuk menghindari deteksi karena *domain* berubah dengan cepat dan sulit untuk dilacak oleh otoritas keamanan. *Malware* yang terinfeksi akan mencoba terhubung ke salah satu dari banyak *domain* yang dihasilkan secara otomatis untuk menerima instruksi dari *server* penyerang.

2.4.2 Teknik dan Taktik yang Digunakan oleh *Malicious Domain*

a. *Fast-Flux DNS*

Teknik ini digunakan untuk menyembunyikan lokasi *server* sebenarnya yang mengendalikan *domain* berbahaya. Dengan *Fast-Flux*, alamat *IP* yang terkait dengan *domain* berubah dengan sangat cepat, sering kali setiap beberapa menit atau detik, sehingga sulit bagi penyedia keamanan untuk memblokir *domain* tersebut.

Penyerang memanfaatkan jaringan komputer yang telah diretas (*botnet*) untuk bertindak sebagai *proxy* dan menyamarkan lokasi *server* yang sebenarnya.

b. *Domain Parking* dan Pendaftaran Anonim

Domain berbahaya sering kali didaftarkan melalui layanan *domain parking* atau pendaftaran anonim yang menyembunyikan identitas asli pendaftar *domain*. Hal ini mempersulit penegak hukum atau penyedia layanan *internet* untuk melacak penyerang.

Domain parking juga memungkinkan penyerang untuk mengarahkan *domain* yang tidak aktif ke iklan atau halaman palsu.

c. *Short-Lived Domains*

Banyak *malicious domain* hanya beroperasi untuk waktu singkat, mungkin hanya beberapa jam atau hari, sebelum ditutup atau diganti oleh penyerang. Ini membantu penyerang menghindari deteksi dan pemblokiran oleh layanan keamanan. Penggunaan *domain* sementara ini umum dalam serangan *phishing* dan *malware distribution* yang bertujuan untuk mendapatkan keuntungan cepat sebelum *domain* terdeteksi.

2.5 *Domain Black List*

Domain Blacklist adalah daftar *domain* yang dianggap berbahaya, tidak aman, atau terlibat dalam aktivitas jahat seperti penyebaran *malware*, *phishing*, spam, atau serangan siber lainnya. *Domain-domain* yang termasuk dalam daftar ini dipantau dan diblokir oleh sistem keamanan, penyedia layanan *internet*, *firewall*, atau layanan *DNS* untuk mencegah pengguna mengakses situs *web* berbahaya yang dapat mengancam keamanan perangkat dan data mereka. *Domain* yang ada dalam daftar ini secara otomatis diblokir oleh berbagai layanan keamanan, seperti:

1. Penyedia Layanan *Email*: Untuk mencegah distribusi *email spam* atau *email* yang berisi *malware* dan *link* berbahaya.
2. Peramban *Web (Browser)*: *Browser* seperti *Google Chrome*, *Mozilla Firefox*, atau *Safari* sering kali memblokir akses ke *domain* yang ada dalam daftar *blacklist*, memberi peringatan kepada pengguna.
3. *Firewall* dan Sistem Keamanan Jaringan: Administrator jaringan dapat mengonfigurasi *firewall* untuk memblokir akses ke *domain* berbahaya yang masuk dalam *blacklist*.
4. Sistem *DNS* Aman: Layanan *DNS* bisa mencegah resolusi *domain-domain* yang dianggap berbahaya dengan memblokir permintaan ke *domain* dalam daftar hitam.

Domain bisa masuk ke dalam *blacklist* melalui berbagai cara, umumnya berdasarkan aktivitas jahat atau mencurigakan yang dilakukan oleh *domain* tersebut. Berikut adalah beberapa alasan umum mengapa *domain* bisa masuk ke dalam *blacklist*:

1. *Phishing*

Jika sebuah *domain* digunakan untuk mencuri informasi pengguna dengan berpura-pura menjadi situs yang sah (seperti meniru situs bank atau email populer), *domain* tersebut dapat masuk ke dalam daftar *blacklist*.

2. *Malware Distributio*

Domain yang menyebarkan *virus*, *trojan*, *ransomware*, atau jenis *malware* lainnya akan dengan cepat dimasukkan ke dalam *blacklist* untuk melindungi pengguna dari infeksi.

3. *Spamming*

Jika *domain* sering digunakan dalam kampanye email spam yang berisi tautan ke konten jahat atau produk tidak sah, *domain* tersebut bisa masuk ke dalam *blacklist*. Penyedia layanan email biasanya secara otomatis memfilter *domain spam* dari pesan masuk.

4. *Command-and-Control (C2) Servers*

Domain yang digunakan sebagai bagian dari infrastruktur *Command-and-Control (C2)* untuk mengendalikan perangkat yang terinfeksi *malware* juga akan diblokir.

5. Penggunaan *DNS* Berbahaya (*DNS Hijacking*, *Fast-Flux*)

Domain yang terlibat dalam teknik serangan seperti *DNS hijacking* atau *Fast-Flux* (di mana alamat *IP* yang terhubung dengan *domain* terus berubah dengan cepat untuk menyembunyikan lokasi sebenarnya) juga dianggap sebagai ancaman dan dapat dimasukkan ke dalam daftar hitam.

6. Aktivitas Pencurian Identitas (*Identity Theft*)

Situs *web* yang diduga terlibat dalam pencurian identitas atau penyebaran data pribadi tanpa izin akan dimasukkan ke daftar hitam oleh otoritas keamanan.

7. Keluhan Pengguna

Pengguna *internet* atau organisasi keamanan siber dapat melaporkan *domain* yang mereka anggap berbahaya atau mencurigakan, yang kemudian akan dievaluasi oleh penyedia *blacklist*.

Domain Black List bekerja sebagai berikut:

1. Pemantauan dan Penilaian

Penyedia keamanan, penyedia layanan *DNS*, dan sistem keamanan jaringan memantau *domain* secara terus-menerus. *Domain* yang menunjukkan aktivitas mencurigakan, seperti mengirimkan *malware*, *phishing*, atau serangan *brute force*, akan ditandai untuk penyelidikan lebih lanjut.

2. Deteksi Otomatis

Sistem otomatis, seperti anti-virus atau *firewall* berbasis *cloud*, menggunakan algoritma untuk mendeteksi *domain-domain* yang memiliki pola perilaku jahat. Jika *domain* memenuhi kriteria tertentu, sistem ini dapat secara otomatis menambahkannya ke *blacklist*.

3. Peringatan Pengguna

Ketika pengguna mencoba mengunjungi *domain* yang ada dalam *blacklist*, mereka akan menerima peringatan dari *browser*, penyedia *email*, atau sistem keamanan mereka yang mengindikasikan bahwa situs tersebut berbahaya. Contohnya, *Google Safe Browsing* akan menampilkan peringatan ketika pengguna mencoba mengakses situs yang dikenal menyebarkan *malware* atau melakukan *phishing*.

4. Pemblokiran Akses

Setelah *domain* terdaftar dalam *blacklist*, akses ke *domain* tersebut akan diblokir. Pemblokiran ini dapat dilakukan di level *DNS* (mencegah *domain* untuk diakses dari *internet*) atau di level aplikasi (seperti *browser* atau *email*).

5. Pemulihan *Domain*

Dalam beberapa kasus, pemilik *domain* yang masuk ke *blacklist* karena alasan yang tidak disengaja atau setelah memperbaiki masalah keamanan bisa meminta agar *domain* mereka dihapus dari *blacklist*. Ini bisa dilakukan dengan menghubungi penyedia *blacklist* dan memberikan bukti bahwa masalah yang teridentifikasi telah diatasi.

2.6 Firewall

Firewall adalah perangkat keras atau perangkat lunak yang dirancang untuk melindungi jaringan komputer dengan mengontrol lalu lintas data yang masuk dan keluar berdasarkan aturan keamanan yang telah ditentukan. *Firewall* bertindak sebagai penghalang antara jaringan internal yang aman dan jaringan eksternal seperti *internet*, sehingga hanya lalu lintas yang diizinkan sesuai kebijakan keamanan yang dapat melewati sistem. Secara umum,

firewall memeriksa paket data berdasarkan *header*, *port*, atau protokol, dan dapat memblokir akses ke sumber berbahaya, seperti *domain* yang teridentifikasi sebagai penyebar *malware* atau alamat *IP* yang dicurigai. Dengan fitur-fitur seperti penyaringan paket (*packet filtering*), inspeksi *stateful*, dan pemantauan aplikasi, *firewall* membantu mencegah akses tidak sah, serangan siber, serta kebocoran data.

Firewall modern sering kali dilengkapi dengan kemampuan canggih yang melampaui fungsi dasar penyaringan. Misalnya, *Next-Generation Firewall (NGFW)* dapat melakukan inspeksi mendalam terhadap konten data (*deep packet inspection*), mendeteksi ancaman berbasis aplikasi, dan mengintegrasikan fitur keamanan tambahan seperti sistem deteksi intrusi (*Intrusion Detection System*) dan pencegahan intrusi (*Intrusion Prevention System*). Beberapa *firewall* juga mendukung penyaringan berbasis *domain* atau *URL* untuk melindungi pengguna dari situs *web* berbahaya. Selain itu, *firewall* juga digunakan untuk membatasi akses antar segmen dalam jaringan internal, menciptakan zona keamanan yang lebih tinggi untuk melindungi aset penting. Dalam ekosistem keamanan siber yang semakin kompleks, *firewall* tetap menjadi salah satu komponen penting dalam strategi keamanan jaringan yang komprehensif, melengkapi alat-alat lain seperti antivirus, *VPN*, dan sistem manajemen keamanan informasi.

2.6.1 Metode Firewall

Metode *firewall* yang efektif tidak hanya mampu memblokir koneksi ke *domain* berbahaya, tetapi juga harus dapat beradaptasi dengan ancaman baru yang terus berkembang, meminimalkan *false positives*, dan beroperasi tanpa mengorbankan kinerja jaringan secara signifikan. Dalam lingkungan *ISP* yang menangani lalu lintas data dalam jumlah besar, pemilihan dan implementasi metode *firewall* yang tepat menjadi krusial untuk menyeimbangkan antara kebutuhan keamanan yang ketat dan pengalaman pengguna yang optimal. Metode *Firewall* untuk Memblokir *Domain* Berbahaya di *ISP*

1. DNS Filtering/DNS Sinkholing

DNS filtering adalah metode efektif yang banyak digunakan oleh *ISP*. Saat pengguna mencoba mengakses *domain* berbahaya, *DNS server ISP* akan mengarahkan permintaan ke alamat *null* atau halaman peringatan. Keuntungannya:

- a. Implementasi sederhana pada level *ISP*
- b. Konsumsi sumber daya rendah
- c. Mampu memblokir *domain* dalam jumlah besar secara sekaligus.

2. *Deep Packet Inspection (DPI)*

DPI memeriksa konten paket data secara mendalam, tidak hanya *header*. Metode ini:

- a. Mampu menganalisis dan memfilter *traffic* berdasarkan konten
- b. Dapat memblokir *domain* tertentu bahkan ketika menggunakan protokol terenkripsi
- c. Memberikan kontrol lebih terperinci, tetapi membutuhkan lebih banyak sumber daya

3. *Border Gateway Protocol (BGP) Blackholing*

Metode ini cocok untuk menangani serangan *DDoS* dengan memblokir *traffic* dari sumber serangan pada *level routing ISP* dapat:

- a. Membagikan rute *BGP* yang mengarahkan *traffic* berbahaya ke "*black hole*"
- b. Mencegah serangan menyebar ke jaringan *ISP*
- c. Efektif untuk mitigasi serangan berskala besar

4. *URL Filtering*

Teknik ini memungkinkan *ISP* memblokir akses ke *URL* spesifik bahkan dalam *domain* yang sama:

- a. Lebih presisi dibanding pemblokiran *domain* penuh
- b. Memungkinkan akses ke bagian *domain* yang aman sambil memblokir bagian berbahaya
- c. Membutuhkan lebih banyak *resource processing*

5. *IP Reputation-Based Filtering*

ISP dapat menggunakan database reputasi *IP* untuk memblokir komunikasi dengan alamat *IP* yang diketahui berbahaya:

- a. Dapat diperbarui secara *real-time*
- b. Memblokir *domain* berbahaya berdasarkan alamat *IP* yang terkait
- c. Efektif untuk mencegah koneksi ke infrastruktur *malware*

6. *Application Layer Firewall*

Firewall ini berfokus pada aplikasi tertentu dan dapat memblokir *domain* berbahaya berdasarkan karakteristik aplikasi:

- a. Mampu mengidentifikasi dan memblokir *traffic* aplikasi berbahaya
- b. Berfungsi pada *layer 7 model OSI*
- c. Dapat mendeteksi serangan yang berjalan melalui port dan protokol standar.

7. Hybrid Approach

Pendekatan terbaik untuk *ISP* sering kali adalah kombinasi metode di atas:

- a. *DNS filtering* untuk pemblokiran *domain* skala besar
- b. *DPI* untuk analisis mendalam pada *traffic* mencurigakan
- c. Sistem reputasi untuk respons dinamis terhadap ancaman baru

2.6.2 Metode Filtering

1. *DNS-Based Filtering*, menggunakan *DNS (Domain Name System)* untuk memblokir akses ke *domain malicious*.
2. *IP-Based Filtering*, memantau dan memblokir alamat *IP* yang dikaitkan dengan aktivitas *malicious*.
3. *URL Filtering*, menganalisis URL untuk mendeteksi pola *malicious*.
4. *Content Filtering*, menganalisis konten halaman *web* untuk mendeteksi *malicious code*.

2.6.3 Teknologi Filtering

1. *Firewall*, menggunakan *firewall* untuk memblokir akses ke *domain malicious*.
2. *Intrusion Detection System (IDS)*, mendeteksi dan memblokir serangan *malicious*.
3. *Web Application Firewall (WAF)*, melindungi aplikasi *web* dari serangan *malicious*.
4. *Secure Web Gateway (SWG)*, mengontrol akses ke situs *web* dan memblokir konten *malicious*.

2.6.4 Algoritma Filtering

1. *Machine Learning*, menggunakan algoritma *machine learning* untuk mendeteksi pola *malicious*.
2. *Deep Learning*, menggunakan algoritma *deep learning* untuk menganalisis konten halaman *web*.
3. *Rule-Based Filtering*, menggunakan aturan-aturan untuk memblokir akses ke *domain malicious*.
4. *Reputation-Based Filtering*: Menggunakan reputasi *domain* untuk menentukan apakah *domain* tersebut *malicious*.

2.6.5 Sumber Data

1. *Blacklist*, menggunakan daftar hitam *domain malicious*.
2. *Whitelist*, menggunakan daftar putih *domain* yang aman.

3. *Threat Intelligence*, menggunakan data intelijen ancaman untuk mendeteksi *domain malicious*.
4. *Open-Source Intelligence*, menggunakan data terbuka untuk mendeteksi *domain malicious*.

2.7 MikroTik Device Landscape

MikroTik adalah perusahaan teknologi yang berbasis di Latvia dan dikenal sebagai penyedia perangkat jaringan yang andal dan ekonomis. Produk-produk *MikroTik* mencakup *router*, *switch*, dan perangkat lunak manajemen jaringan yang digunakan secara luas dalam berbagai skala, mulai dari rumah tangga hingga perusahaan besar (Ceron *et al.*, 2020). Sistem operasi utama yang digunakan oleh perangkat *MikroTik* adalah *RouterOS*, yang menawarkan berbagai fitur jaringan seperti *routing*, *firewall*, pengelolaan *bandwidth*, *VPN*, dan manajemen pengguna. Selain itu, *MikroTik* juga mengembangkan *RouterBOARD*, yaitu perangkat keras yang dirancang untuk bekerja secara optimal dengan *RouterOS*.

Salah satu keunggulan utama perangkat *MikroTik* adalah fleksibilitas dan kemudahan penggunaannya. *RouterOS* memiliki antarmuka berbasis *Command Line Interface (CLI)* dan *Graphical User Interface (GUI)* melalui *WinBox*, sebuah aplikasi yang memungkinkan konfigurasi perangkat dengan tampilan yang lebih intuitif. Dengan kombinasi perangkat keras yang bervariasi dan sistem operasi yang kaya fitur, *MikroTik* dapat digunakan untuk berbagai kebutuhan jaringan, termasuk jaringan nirkabel, jaringan kampus, dan jaringan perusahaan. Selain itu, kemampuannya dalam melakukan *Quality of Service (QoS)* menjadikannya pilihan yang ideal untuk pengelolaan lalu lintas data yang efisien.

Dari segi keamanan, *MikroTik* menawarkan berbagai fitur yang mendukung proteksi jaringan dari ancaman eksternal maupun internal. *Firewall* bawaan dalam *RouterOS* memungkinkan administrator jaringan untuk mengontrol lalu lintas masuk dan keluar dengan aturan yang dapat disesuaikan. *MikroTik* juga mendukung implementasi *Virtual Private Network (VPN)* dengan berbagai protokol seperti *PPTP*, *L2TP*, dan *OpenVPN*, yang memberikan solusi komunikasi yang aman bagi pengguna jarak jauh. Selain itu, fitur *Hotspot MikroTik* memungkinkan penyedia layanan *internet (ISP)* dan pengelola jaringan publik untuk menerapkan sistem autentikasi pengguna secara efisien.

Dalam lanskap perangkat jaringan global, *MikroTik* bersaing dengan merek-merek besar seperti *Cisco*, *Juniper*, dan *Ubiquiti*. Namun, *MikroTik* memiliki daya tarik tersendiri karena kombinasi harga yang lebih terjangkau dengan fitur yang tidak kalah canggih.

Keberadaan komunitas pengguna dan dokumentasi yang luas juga menjadi faktor penting dalam adopsi *MikroTik* di berbagai negara. Dengan terus berkembangnya kebutuhan akan jaringan yang lebih efisien dan terjangkau, *MikroTik* tetap menjadi salah satu pilihan utama bagi individu dan perusahaan yang ingin membangun serta mengelola jaringan secara efektif (Welda, 2023).

2.7.1 Firewall pada MikroTik

Firewall pada *MikroTik* adalah sistem keamanan jaringan yang berfungsi untuk memfilter lalu lintas data berdasarkan aturan yang telah ditentukan oleh *administrator* jaringan. *Firewall* ini bekerja pada lapisan *OSI* tertentu, seperti *Network* dan *Transport*, dengan fitur utama seperti *filter rules*, *NAT (Network Address Translation)*, dan *mangle*. *MikroTik RouterOS* menggunakan konsep *chain* dalam *firewall*, yang terdiri dari *input*, *output*, dan *forward*. Fitur ini memungkinkan *administrator* untuk mengontrol lalu lintas data yang masuk, keluar, atau melewati *router*, sehingga dapat diterapkan kebijakan keamanan yang ketat guna melindungi jaringan dari ancaman eksternal.

2.7.2 Blocking dengan MikroTik

MikroTik memiliki berbagai metode untuk melakukan pemblokiran lalu lintas jaringan yang tidak diinginkan, baik berdasarkan alamat *IP*, *domain*, protokol, maupun jenis *traffic* tertentu. Salah satu metode yang umum digunakan adalah dengan *firewall filter rules*, di mana aturan dibuat untuk memblokir koneksi ke atau dari alamat tertentu. Selain itu, *MikroTik* mendukung pemblokiran berbasis *layer 7 protocol (L7)* untuk mendeteksi dan memblokir lalu lintas yang sesuai dengan pola tertentu, seperti akses ke situs *web* berbahaya atau aplikasi tertentu. Metode lain yang sering digunakan adalah *DNS filtering*, di mana *domain* yang tidak diizinkan dapat diarahkan ke alamat *IP* palsu atau diblokir sepenuhnya menggunakan fitur *static DNS*.

2.7.3 Address List pada MikroTik

Fitur *Address List* di *MikroTik* memungkinkan *administrator* jaringan untuk mengelompokkan alamat *IP* dalam daftar tertentu yang kemudian dapat digunakan dalam aturan *firewall*. *Address List* sangat berguna dalam mengelola kebijakan keamanan jaringan secara dinamis, seperti memblokir atau mengizinkan akses ke sekumpulan alamat *IP* tanpa harus menambah aturan *firewall* secara manual. *Address list* dapat diisi secara otomatis

melalui skrip, atau dapat diatur agar alamat *IP* tertentu masuk ke dalam daftar berdasarkan perilaku *traffic*, seperti jumlah koneksi yang mencurigakan dari satu sumber *IP*. Dengan *address list*, pengelolaan keamanan jaringan menjadi lebih fleksibel dan efisien. Dalam *MikroTik*, *address-list* bisa dilihat, diakses dan dikonfigurasi di sub menu `/ip firewall address-list`.

2.7.4 Import File .rsc ke dalam Address List

MikroTik mendukung impor konfigurasi dalam bentuk *file .rsc* (*RouterOS Script Configuration*) yang berisi perintah-perintah untuk mengatur berbagai aspek *router*, termasuk *Address List*. Administrator dapat membuat *file .rsc* yang berisi daftar alamat *IP* atau *domain* yang harus dimasukkan ke dalam *Address List*, kemudian mengimpornya menggunakan perintah `import filename.rsc`. Cara ini sangat berguna untuk memperbarui daftar pemblokiran secara berkala tanpa harus memasukkan data secara manual. *File .rsc* juga memungkinkan sinkronisasi *Address List* antara beberapa *router MikroTik* dalam jaringan yang lebih luas, memastikan kebijakan keamanan tetap konsisten. Perintah `import .rsc file` menggunakan *sintax* `/import file-name="nama file .rsc"`

2.7.5 Traffic Flow pada MikroTik

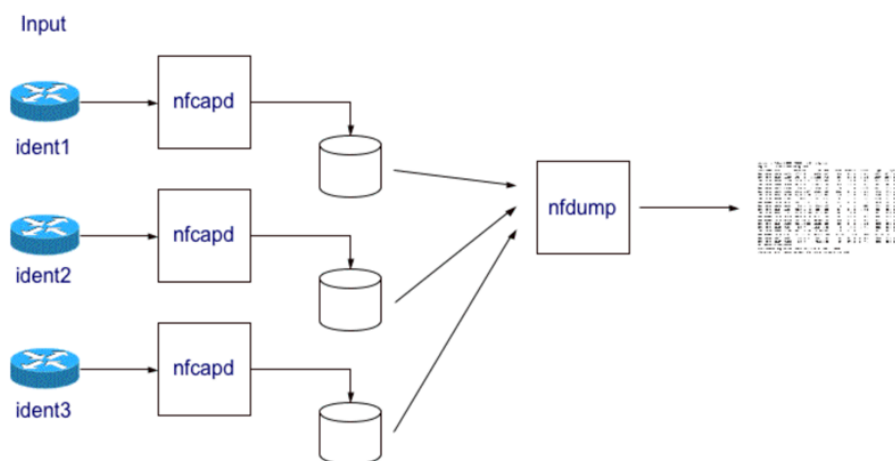
Traffic Flow adalah fitur *MikroTik* yang memungkinkan administrator untuk memantau dan menganalisis lalu lintas jaringan secara *real-time*. Dengan fitur ini, data mengenai sumber, tujuan, protokol, dan jumlah paket yang melewati *router* dapat dikumpulkan dan dikirim ke *server* analisis seperti *NetFlow Collector*. *Traffic Flow* berguna dalam mengidentifikasi pola penggunaan jaringan, mendeteksi anomali, dan mengoptimalkan kebijakan *firewall*. Selain itu, informasi yang diperoleh dari *Traffic Flow* dapat digunakan untuk memperbarui *Address List* secara otomatis berdasarkan perilaku *traffic*, seperti menambahkan alamat *IP* yang mencurigakan ke dalam daftar pemblokiran untuk meningkatkan keamanan jaringan. Aspek ini merupakan fitur penting dalam *MikroTik* yang dapat digunakan untuk meningkatkan keamanan dan efisiensi manajemen jaringan. Dengan kombinasi *firewall*, *address list*, dan *traffic monitoring*, *administrator* dapat menerapkan kebijakan keamanan yang lebih efektif dalam melindungi jaringan dari ancaman siber.

Berdasarkan tinjauan pustaka yang telah dipaparkan, penelitian ini mengusulkan kerangka konseptual yang berfokus pada tiga aspek utama. Pertama, *malicious domain* beroperasi dengan pola *traffic* yang berbeda dari *domain* legitimasi, memungkinkan

identifikasi karakteristik khusus yang dapat dideteksi. Kedua, *firewall* dengan kemampuan *filtering domain* berbahaya dapat mengidentifikasi dan memblokir *traffic* ini berdasarkan parameter yang telah didefinisikan, menciptakan mekanisme perlindungan yang efektif. Ketiga, pemblokiran ini berdampak signifikan pada efisiensi jaringan dengan mengurangi beberapa aspek kritis seperti volume data tidak perlu (*bytes*), jumlah paket tidak perlu (*packets*), penggunaan *bandwidth* (*bps*), serta penggunaan *resource* pemrosesan paket (*pps*), yang secara keseluruhan meningkatkan performa dan keamanan jaringan.

2.8 NFDUMP Open Source Tools

NFDUMP merupakan seperangkat *tools* yang dirancang khusus untuk mengumpulkan dan memproses data *NetFlow* melalui *command line interface*. *Tools* ini didistribusikan di bawah lisensi BSD dan merupakan bagian integral dari proyek *NfSen* yang memberikan solusi komprehensif untuk analisis *traffic* jaringan. *NFDUMP* memiliki karakteristik unik dalam pemisahan proses *collection* dari *processing*, dimana semua data yang dikumpulkan disimpan ke *disk* sebelum dilakukan analisis dengan struktur direktori yang diorganisir berdasarkan waktu. Arsitektur ini memungkinkan analisis data *NetFlow* dari masa lalu sekaligus *tracking* pola *traffic* yang menarik secara kontinyu, dimana batasan waktu ke masa lalu hanya dibatasi oleh ruang *disk* yang tersedia. Gambar 2.1 Menggambarkan contoh desain yang bertujuan untuk dapat menganalisis data netflow dari masa lalu sekaligus melacak pola lalu lintas yang menarik secara berkelanjutan.



Gambar 2.1 Prinsip operasi *NFDUMP*

Semua data disimpan ke *disk* sebelum dianalisis. Hal ini memisahkan proses penyimpanan dan analisis data. Data diorganisir berdasarkan waktu. Setiap *n* menit -

biasanya 5 menit - *nfcapd* melakukan rotasi dan mengganti nama *file output* dengan *cap* waktu *nfcapd.YYYYMMddhhmm* dari interval tersebut, misalnya *nfcapd.202307110845* berisi data dari 11 Juli 2023 pukul 08:45 dan seterusnya. Berdasarkan interval waktu 5 menit, ini menghasilkan 288 *file* per hari.

Analisis data dapat dilakukan untuk satu *file*, atau dengan menggabungkan beberapa *file* untuk satu kali proses. *Output* berupa teks *ASCII* atau data *biner*, ketika disimpan ke dalam *file*, siap untuk diproses lagi dengan *tools* yang sama.

Dalam pennggunaannya memungkinkan memiliki beberapa sumber netflow - misal 'router1' 'router2' dan seterusnya. Data diorganisir sebagai berikut:

```
/flow_base_dir/router1  
/flow_base_dir/router2
```

yang berarti router1 dan router2 adalah subdirektori dari *flow_base_di*. Untuk setiap sumber netflow, harus memulai dengan proses *nfcapd*:

```
nfcapd -w -D -l /flow_base_dir/router1 -p 23456  
nfcapd -w -D -l /flow_base_dir/router2 -p 23457
```

2.8.1 Konfigurasi

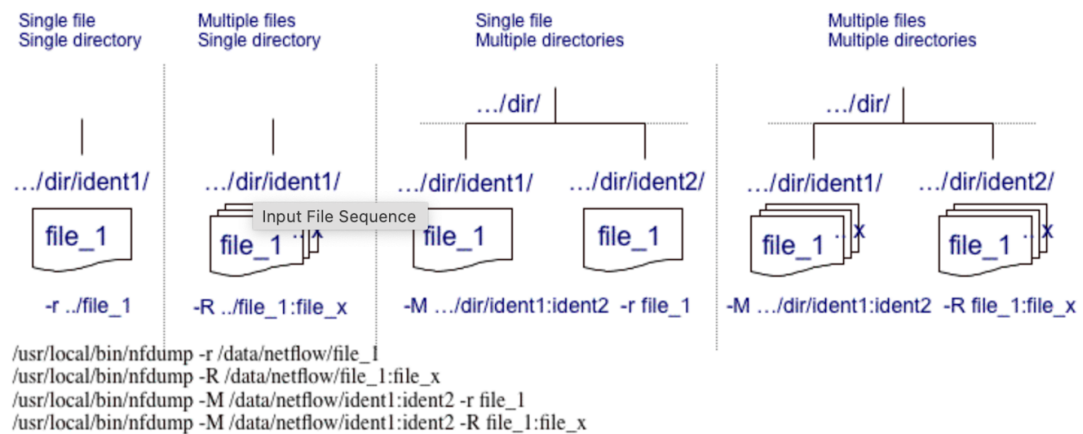
Konfigurasi pada router diperlukan untuk mengekspor *netflow*. Konfigurasi menyesuaikan dengan dokumentasi yang relevan untuk model yang digunakan. Contoh generik *MikroTik* untuk mengaktifkan *Netflow* pada sebuah *interface* bisa menggunakan *command* sebagai berikut:

```
/ip traffic-flow set enabled=yes interface=ether1  
/ip traffic-flow target add dst-address=x.x.x.x  
port=2055
```

Dimana *interface* merupakan *object interface* yang akan dibaca *trafficnya*, dan *dst-address x.x.x.x* merupakan *IP server collector* yang menjalankan *service collecting traffic* di *port 2055*

2.8.2 Netflow Processing

Disarankan untuk mengikuti halaman manual *nfdump* dalam operasi *system* untuk penjelasan rinci tentang semua opsi yang tersedia. Gambar 2.2 memberikan gambaran langkah-langkah pemrosesan *Netflow*, *Flow* bisa dibaca dibaca baik dari satu *file* atau dari rangkaian *file*.

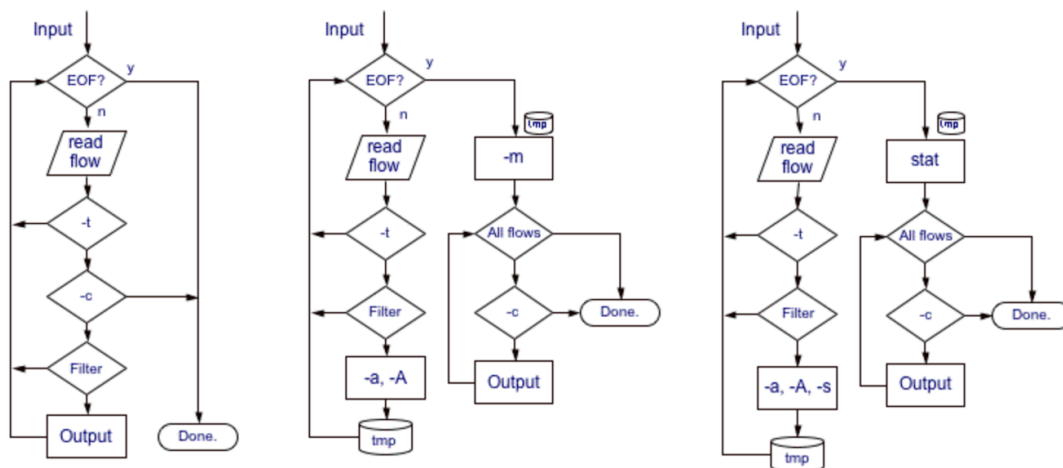


Gambar 2.2 Langkah pemrosesan *Netflow*

Sebagian atribut *command* yang bisa digunakan adalah sebagai berikut:

-r <single file>	Membaca <i>file</i> tunggal
-R </path/to/first-file:lastfile>	Membaca data berurutan dari /path/to/first-file sampai /path/to/last-file
-M /path/to/first-dir:next-dir:last-dir -r <single-file>	Membaca file berurutan dari beberapa directori:
-M /path/to/first-dir:next-dir:last-dir -R <first-file:last-file>	Urutan filenya: /path/to/first-dir/single-file /path/to/next-dir/single-file /path/to/last-dir/single-file atau /path/to/first-dir/first-file /path/to/first-dir/last-file /path/to/next-dir/first-file /path/to/next-dir/last-file /path/to/last-dir/first-file /path/to/last-dir/last-file
Khusus:	
-R </path/to/directory>	Membaca semua file dalam sebuah directory
-M /path/to/first-dir:next-dir:last-dir -R .	
-R </path/to/first-file>	Membaca semua file dalam sebuah directory yang dimulai dengan suatu nama file
-M /path/to/first-dir:next-dir:last-dir -R <first-file>	

Flow yang diproses dapat dicetak dalam *ASCII* ke *stdout* atau ditulis ke *file*. *File* biner dapat dibaca lagi oleh *nfdump* untuk pemrosesan lebih lanjut. Gambar 2.3 dibawah merupakan diagram di bawah ini menunjukkan pada titik mana berbagai opsi mempengaruhi pemrosesan *netflow* untuk daftar *flow* normal, daftar yang diurutkan berdasarkan tanggal, dan *output* statistik.



Gambar 2.3 Opsional output berdasarkan *command* pemrosesan

Format *output*: *nfdump* memiliki empat format output tetap: *raw*, *line*, *long*, dan *extended*. Namun pengguna dapat menentukan format output yang diinginkan menggunakan format *output* kustom *fmt:...* Format default adalah *line*, kecuali ditentukan lain.

Format *raw*: Format *raw* menampilkan setiap *record* dalam beberapa baris, dan mencetak informasi apa pun yang tersedia dalam *record*. *Record* yang dicetak tidak bergantung pada versi *netflow*, tetapi mungkin berisi *field* tambahan yang berbeda tergantung pada sumbernya.

2.8.3 Kompatibilitas dan Dukungan *Protocol*

NFDUMP memiliki kompatibilitas yang sangat luas dengan mendukung berbagai versi *NetFlow* termasuk *v1*, *v5*, *v7*, *v9*, dan *IPFIX* serta *limited set sFlow*. Sistem ini juga mendukung perangkat khusus seperti *CISCO ASA (NSEL)* dan *CISCO NAT (NEL)*, serta kompatibel dengan *IPv6*. Kemampuan kompatibilitas yang luas ini menjadikan *NFDUMP* sebagai solusi universal untuk analisis *traffic flow* dari berbagai *vendor* dan protokol jaringan yang berbeda. Dukungan protokol yang komprehensif memungkinkan implementasi *NFDUMP* dalam lingkungan jaringan yang heterogen tanpa memerlukan modifikasi signifikan pada infrastruktur yang ada. Semua tools mendukung *netflow v5*, *v7* dan *v9*, diantaranya:

1. *nfcapd* - *daemon* penangkap *netflow*.

Membaca data *netflow* dari jaringan dan menyimpan data ke dalam *file*. Secara otomatis merotasi *file* setiap *n* menit (biasanya setiap 5 menit). *nfcapd* membaca

flows netflow v5, v7 dan v9 secara transparan. Anda membutuhkan satu proses *nfcapd* untuk setiap aliran *netflow*.

2. *nfdump* - *dump netflow*.

Membaca data *netflow* dari *file* yang disimpan oleh *nfcapd*. Sintaksnya mirip dengan *tcpdump*. Menampilkan data *netflow* dan dapat membuat banyak statistik top N dari *flows*, alamat *IP*, *port*, dll yang diurutkan berdasarkan urutan yang Anda inginkan.

3. *nfprofile* - *profiler netflow*.

Membaca data *netflow* dari *file* yang disimpan oleh *nfcapd*. Memfilter data *netflow* sesuai dengan *filter sets* yang ditentukan (profil) dan menyimpan data yang telah difilter ke dalam *file* untuk penggunaan selanjutnya.

4. *nfreplay* - *replay netflow*.

Membaca data *netflow* dari *file* yang disimpan oleh *nfcapd* dan mengirimkannya melalui jaringan ke *host* lain.

5. *nfclean.pl* - pembersihan data lama.

Contoh script untuk membersihkan data lama. Anda dapat menjalankan script ini setiap jam atau lebih.

6. *ft2nfdump* - Membaca dan mengkonversi data *flow-tools*.

Membaca data *flow-tools* dari *file* atau dari *stdin* dalam rangkaian perintah *flow-tools* dan mengkonversi data ke dalam format *nfdump* untuk diproses oleh *nfdump*.

2.8.4 Fitur Analisis dan *Filtering*

NFDUMP dilengkapi dengan sistem filtering yang canggih dengan *syntax* yang mirip dengan *tcpdump*, memungkinkan *filtering* dan agregasi kompleks dari *flow records*. Sistem ini menyediakan berbagai format output yang dapat disesuaikan dengan kebutuhan analisis, mulai dari format dasar hingga format *extended* yang menampilkan informasi detail seperti *TCP flags*, *Type of Service (ToS)*, *packets per second (pps)*, *bits per second (bps)*, dan *bytes per packet (bpp)*. *Tools* ini juga mendukung *custom format output* menggunakan *tag-based formatting* yang memungkinkan pengguna untuk menentukan *field-field* spesifik yang ingin ditampilkan. Kemampuan agregasi data berdasarkan berbagai parameter seperti *source IP*, *destination IP*, *port*, *AS number*, dan *interface* memberikan fleksibilitas tinggi dalam analisis *traffic pattern*.

2.8.5 Kemampuan Statistik dan *Reporting*

NFDUMP menyediakan kemampuan statistik yang komprehensif dengan berbagai opsi analisis termasuk statistik tentang *record*, *IP address (source/destination)*, *port (source/destination)*, *AS number*, *interface*, dan protokol. Sistem *reporting* yang terintegrasi memungkinkan pembuatan *Top-N reports* berdasarkan berbagai parameter seperti *flows*, *packets*, *bytes*, *packets per second*, dan *bits per second*. Fitur *anonymizing flows* dengan menggunakan *key encryption* memungkinkan analisis data sensitif dengan tetap menjaga *privacy*. Kemampuan untuk menganalisis jutaan *flow records* dengan performa tinggi menjadikan *NFDUMP* *suitable* untuk implementasi dalam *environment enterprise* dengan volume *traffic* yang besar.

2.8.6 Performa dan Efisiensi

NFDUMP dioptimalkan untuk kecepatan dan efisiensi *processing* dengan kemampuan menganalisis ratusan ribu hingga jutaan *flow records* per detik. Berdasarkan dokumentasi resmi, sistem ini mampu memproses *flows* dengan kecepatan mencapai 488,001 *flows/second* pada sistem *metrics* dan 467,490 *flows/second* pada *wall time*, menunjukkan performa yang sangat baik dalam menghadapi volume data yang besar. Efisiensi *processing* ini dicapai melalui optimasi algoritma *filtering* dan struktur data yang memungkinkan akses cepat terhadap informasi *flow records*. Kemampuan untuk menangani file data berukuran besar dengan konsumsi *memory* yang minimal menjadikan *NFDUMP* sebagai solusi yang *scalable* untuk analisis *traffic* jaringan dalam skala *enterprise*.

2.8.7 Integrasi dengan Sistem *Monitoring*

NFDUMP dapat diintegrasikan dengan berbagai sistem *monitoring* dan *security tools* melalui format *output* yang beragam dan *interface command line* yang fleksibel. Integrasi dengan *NfSen* web *interface* memberikan kemudahan dalam visualisasi dan monitoring *real-time traffic patterns*. Kemampuan untuk membaca data dari *sequence of files* dan *multiple directories* memungkinkan integrasi dengan sistem *log management* dan *SIEM (Security Information and Event Management)* yang ada. Dukungan untuk *batch processing* dan *automated analysis* melalui *script integration* menjadikan *NFDUMP* sebagai komponen yang dapat diandalkan dalam infrastruktur *security monitoring* yang komprehensif.

BAB 3

Metodologi Penelitian

3.1 Literature Review

Filtering malicious domain merupakan salah satu metode efektif yang banyak digunakan untuk memitigasi ancaman dari *malicious internet traffic*. Teknik ini berfokus pada pemblokiran akses ke *domain* yang diketahui berbahaya, seperti *domain* yang digunakan untuk mendistribusikan *malware*, serangan *phishing*, atau pengoperasian *botnet*. Penelitian sebelumnya menunjukkan bahwa penggunaan daftar hitam (*blacklist*) *domain* berbahaya dapat secara signifikan mengurangi risiko serangan siber dengan memutuskan akses ke infrastruktur yang digunakan oleh pelaku ancaman. Pendekatan berbasis *blacklist* ini, meskipun efektif, memiliki keterbatasan dalam mendeteksi ancaman baru atau *domain* yang belum terdaftar, sehingga diperlukan integrasi dengan metode lain seperti analisis reputasi *domain* atau pemantauan *real-time*.

Desain sistem *filtering malicious domain* biasanya mencakup pengumpulan data *domain* dari sumber terpercaya, seperti *Threat Intelligence Feeds*, serta pembaruan daftar secara berkala untuk memastikan cakupan yang luas terhadap ancaman terbaru. Sistem ini sering kali diterapkan pada *firewall* atau perangkat *gateway* jaringan untuk memfilter lalu lintas sebelum mencapai pengguna akhir. Penelitian telah mengidentifikasi bahwa *filtering* berbasis *domain* tidak hanya memberikan perlindungan langsung terhadap ancaman yang diketahui, tetapi juga mampu meningkatkan efisiensi lalu lintas jaringan dengan mengurangi beban dari akses tidak sah. Selain itu, teknik ini dapat diintegrasikan dengan algoritma *machine learning* untuk mendeteksi *domain* yang menunjukkan pola perilaku mencurigakan meskipun belum terdaftar dalam *blacklist*.

Efektivitas *filtering malicious domain* juga bergantung pada dampaknya terhadap performa jaringan. Beberapa studi mencatat bahwa aktivasi fitur ini dapat menurunkan volume lalu lintas berbahaya secara signifikan, namun dapat sedikit meningkatkan latensi jika tidak diimplementasikan dengan baik. Analisis dampak menunjukkan bahwa *filtering* tidak hanya mencegah ancaman, tetapi juga memberikan manfaat sekunder berupa peningkatan efisiensi jalur *internet* dengan mengeliminasi lalu lintas yang tidak relevan. Oleh karena itu, metode ini menjadi bagian integral dari strategi keamanan jaringan, terutama dalam mendukung lingkungan yang memerlukan perlindungan berlapis untuk menghadapi ancaman yang terus berkembang.

3.2 Lokasi dan Waktu Penelitian

Penelitian ini dilakukan di salah satu *Internet Service Provider (ISP)* yang beroperasi di Banda Aceh, yang merupakan salah satu penyedia layanan *internet* utama di wilayah tersebut. Pemilihan *ISP* ini didasarkan pada infrastruktur jaringan yang mendukung implementasi fitur keamanan, termasuk *firewall* dengan kemampuan *filtering malicious domain*, serta akses ke data lalu lintas jaringan yang relevan untuk penelitian ini. Lingkungan penelitian yang tersedia di *ISP* ini memungkinkan pengumpulan data yang representatif terkait lalu lintas jaringan, termasuk *domain* berbahaya dan pola ancaman yang terjadi.

Penelitian berlangsung selama tiga bulan, mulai dari Januari hingga Maret 2023. Rentang waktu ini dipilih untuk memastikan analisis yang komprehensif terhadap variasi lalu lintas jaringan dalam periode tertentu. Selama waktu tersebut, data *log firewall* dan lalu lintas jaringan dianalisis secara berkala untuk mengevaluasi efektivitas fitur *filtering malicious domain* yang diterapkan oleh *ISP*. Aktivitas penelitian melibatkan pengamatan langsung terhadap operasional sistem keamanan jaringan, analisis *log* secara mendalam, serta diskusi dengan tim teknis *ISP* guna memastikan keakuratan data yang digunakan dalam penelitian.

3.3 Identifikasi Sistem

Tahap pertama dalam penelitian ini adalah identifikasi terhadap sistem dan infrastruktur *ISP* yang menjadi objek penelitian. Identifikasi meliputi beberapa aspek berikut:

a. Arah fokus riset dan teknologi *ISP*

ISP yang dipilih sebagai objek penelitian harus memiliki fokus riset yang relevan dengan topik keamanan siber. *ISP* ini belum menerapkan sistem *firewall* yang secara khusus dirancang untuk melindungi pelanggan dari ancaman siber.

b. Skala Operasional dan Infrastruktur

ISP yang menjadi objek penelitian memiliki skala operasional yang luas, didukung oleh infrastruktur *Fiber Optic* dan *Wireless*, serta menggunakan *multi-homing Border Gateway Protocol (BGP)* untuk koneksi *upstream*. Infrastruktur ini dinilai relevan untuk mendukung skala penelitian yang direncanakan.

c. Perijinan akses data

Pemilihan *ISP* juga mempertimbangkan kemampuan untuk memberikan akses data atau *log* jaringan secara legal dan sesuai dengan regulasi privasi. Dalam hal ini, peneliti merupakan anggota tim konsultan yang telah mendapatkan izin resmi untuk melakukan penelitian di jaringan *ISP* tersebut.

Selain identifikasi sistem, dilakukan juga survei fisik untuk memahami perangkat keras yang digunakan di *ISP*. Hasil survei menunjukkan bahwa *ISP* menggunakan:

- *Router MikroTik.*
- *Server dengan sistem operasi Linux.*
- *Optical Line Terminal (OLT).*
- *Optical Network Terminal (ONT).*
- *Wireless Point-to-Point.*
- *Wireless Point-to-Multi-Point.*

Perangkat-perangkat ini memiliki antarmuka (*interface*) berupa *Ethernet* dan *fiber optic Small Form-Factor Pluggable (SFP)*, yang nantinya relevan dengan desain dan kompatibilitas sistem yang akan dirancang.

Keberagaman perangkat yang digunakan oleh *ISP* ini menjadi faktor penting dalam perancangan sistem, terutama dalam aspek kompatibilitas dan integrasi. Dengan berbagai jenis *type* perangkat yang menggunakan antarmuka seperti *Ethernet* dan *fiber optic SFP*, sistem yang akan dikembangkan harus mampu berinteraksi dengan infrastruktur yang ada secara optimal. Salah satu aspek krusial dalam perancangan ini adalah implementasi mekanisme keamanan, khususnya dalam pengelolaan daftar hitam (*blacklist*) untuk *malicious domain*.

Penggunaan *blacklist* untuk memblokir *domain* berbahaya (*malicious domains*) dalam *firewall* tetap sangat relevan hingga saat ini sebagai lapisan pertahanan utama dalam keamanan jaringan. Dengan semakin banyaknya ancaman siber seperti *phishing*, *malware*, dan *command-and-control (C2) server*, *blacklist* memungkinkan organisasi untuk secara proaktif memblokir akses ke *domain* yang telah diketahui berbahaya. Meskipun metode ini memiliki keterbatasan dalam menghadapi ancaman baru (*zero-day threats*), integrasi *blacklist* dengan *threat intelligence* yang terus diperbarui membantu meningkatkan efektivitasnya. Studi oleh (Cozza *et al.*, 2020), (Marques *et al.*, 2021b), (Pitafi *et al.*, 2023) menunjukkan bahwa penggunaan *blacklist* yang terintegrasi dengan solusi keamanan lainnya, seperti *DNS filtering* dan *Intrusion Prevention Systems (IPS)*, mampu mengurangi risiko serangan secara signifikan. Pendekatan ini tetap menjadi bagian integral dari strategi keamanan berlapis (*layered security*).

Tahap berikutnya adalah pengumpulan data *malicious domain* dari berbagai sumber terpercaya, seperti *feed intelijen* ancaman, *database* publik, dan hasil penelusuran forensik. Sumber data yang digunakan mencakup *domain* yang diketahui terlibat dalam aktivitas berbahaya, baik yang baru terdeteksi maupun yang sudah lama diidentifikasi. Salah satu

referensi utama yang digunakan adalah data dari *CERT Polska*, tim tanggap insiden pertama di Polandia yang berada di bawah naungan *NASK (Scientific and Academic Computer Network)*, sebuah lembaga penelitian ilmiah. Informasi lengkap dapat diakses melalui situs web resmi mereka, yaitu <https://cert.pl>. *CERT Polska* memfokuskan penelitiannya pada pengumpulan, verifikasi, dan validasi data *malicious domain* dari berbagai sumber di seluruh dunia. *Domain* yang diklasifikasikan sebagai berbahaya biasanya terlibat dalam aktivitas seperti:

- ☐ Pengendalian *botnet*
- ☐ Investasi palsu daring
- ☐ Lampiran email palsu
- ☐ Permintaan transfer dana cepat
- ☐ Toko daring palsu
- ☐ *Phishing* pada gim daring
- ☐ Situs judi daring, dan sebagainya.

Data *malicious domain* yang dikumpulkan oleh *CERT Polska* tersedia dalam format yang kompatibel untuk digunakan pada berbagai platform perangkat jaringan, termasuk *router*. Format ini memungkinkan data tersebut diintegrasikan sebagai bagian dari atribut *firewall* untuk meningkatkan keamanan jaringan. Informasi ini dapat diakses secara terbuka, diupdate setiap 5 menit, dan digunakan oleh siapa saja yang membutuhkan, sehingga menjadi salah satu sumber data yang penting dalam penelitian ini.

3.4 Analisis Kebutuhan Sistem

Berdasarkan identifikasi sebelumnya, dilakukan analisis kebutuhan sistem untuk merancang dan mengembangkan mekanisme *filtering* berbasis *address list* dengan memanfaatkan data *domain* yang telah dikumpulkan. Sistem ini dirancang untuk menggunakan *address list* sebagai parameter deteksi lalu lintas jaringan, yang kemudian akan dimasukkan ke dalam kategori *malicious domain*. Fungsi utama *router* adalah memblokir permintaan menuju *domain* berbahaya yang telah terdaftar, mencatat *IP source* dan *IP destination*, serta memonitor *traffic flow* yang kemudian disimpan dalam mesin kolektor.

Dari studi literatur pada *ISP*, jenis *traffic flow* berdasarkan parameter *source address (src-addr)* dan *destination address (dst-addr)* dapat dikategorikan sebagai berikut:

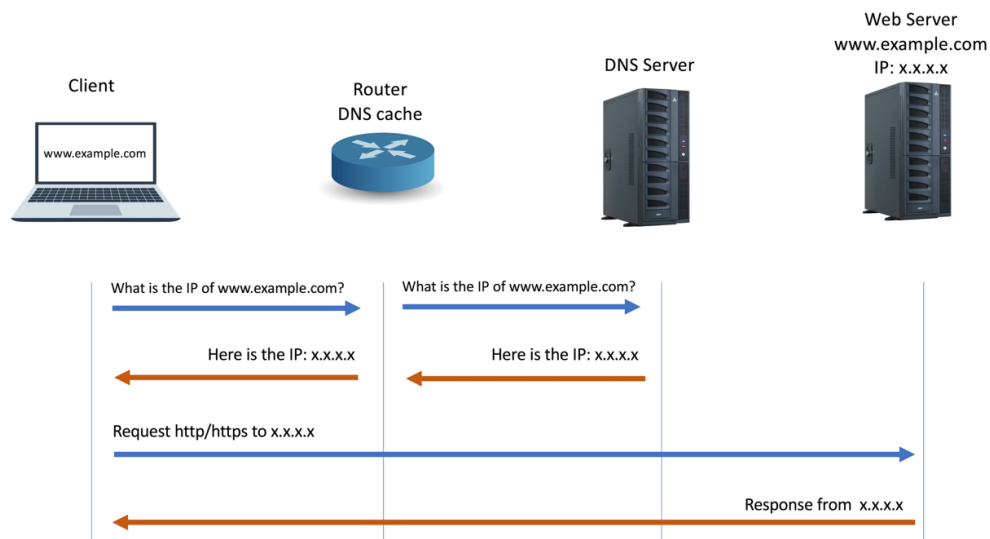
- ☐ *Traffic* dari *Client* ke *Router* (Upload – Request *Client* to *Router*)
- ☐ *Traffic* dari *Router* ke *Client* (Download – Respons *Router* to *Client*)

- *Traffic* dari *Router* ke *Internet* (Upload – *Request Router to Internet*)
- *Traffic* dari *Internet* ke *Router* (Download – *Respons from Internet to Router*)
- *Traffic* dari *Client* ke *Internet* (Upload – *Request Client to Internet*)
- *Traffic* dari *Internet* ke *Client* (Download – *Respons from Internet to Client*)

Selain itu, lalu lintas komunikasi pada *Domain Name System (DNS)* terbagi menjadi dua kategori:

- *DNS Request*: Permintaan dari *Client* ke *DNS Server*.
- *DNS Response*: Balasan dari *DNS Server* ke *Client*.

Gambar 3.1 Mengilustrasikan proses resolusi *DNS* dan akses *web* dalam jaringan computer. Sebagai contoh dimulai dengan klien yang ingin mengakses *www.example.com*. Proses komunikasi berlangsung dalam empat tahap utama: pertama, klien mengirimkan permintaan "Berapa *IP* dari *www.example.com*?" ke *router* dengan *DNS cache*; kedua, *router* meneruskan permintaan ke *server DNS* jika informasi tidak tersedia dalam *cache*; ketiga, setelah *IP* (*x.x.x.x*) diperoleh oleh *router*, akan disimpan dalam *cache* dan diteruskan kembali ke klien, klien mengirimkan permintaan *HTTP/HTTPS* langsung ke alamat *IP* tersebut yang merupakan *web server* tempat *www.example.com* berada; dan akhirnya, *web server* dengan *IP* *x.x.x.x* memberikan respons balik ke klien, menyelesaikan proses komunikasi yang mendasari hampir semua interaksi *browsing web* atau *domain* pada jaringan *modern*.



Gambar 3.1 *Client traffic communication domain request*.

Karakteristik lalu lintas pada *ISP* dapat diamati menggunakan fitur *torch* dan *sniffing* di *router*. Jika tidak ada permintaan (*request* atau *upload*), maka tidak akan ada lalu lintas *download* atau respons. Mekanisme komunikasi antara *Client* dan alamat *domain* di *internet* dapat dijelaskan sebagai berikut:

- *Client* mengirimkan permintaan *IP* dari sebuah nama *domain* ke *Router* (*Router* berfungsi sebagai *DNS Cache*).
- *Router* meneruskan permintaan *IP domain* tersebut ke *DNS Server*.
- *DNS Server* memberikan balasan berupa informasi *IP Domain* kepada *Router*.
- *Router* mengirimkan balasan informasi *IP Domain* tersebut ke *Client*.
- *Client* menghubungi *IP Domain Server* sesuai informasi yang diterima dari *Router* (*DNS Cache*).
- *Domain Server* mengirimkan respons ke *Client*.

Berdasarkan mekanisme komunikasi ini, *firewall* dirancang untuk melakukan *drop traffic* permintaan *Client* menuju *IP* yang terdaftar dalam *blacklist malicious domain*. File berisi *address list malicious domain* dalam format *.rsc* diimpor ke dalam *Router* dan diintegrasikan melalui menu *address-list*. *Firewall* filter diterapkan pada lalu lintas langsung dari *Client* ke *IP Domain* yang teridentifikasi sebagai berbahaya. Lalu lintas yang dikumpulkan adalah *traffic* pada interface capture di *router* yang mengarah ke jaringan *Client*. *Router* mengirimkan semua data lalu lintas yang ditangkap ke *server* kolektor menggunakan protokol *TCP* pada *port* 2556. Penelitian juga memanfaatkan fitur *Traffic Flow* pada *Router MikroTik*, yang memungkinkan pembacaan lalu lintas jaringan yang melewati antarmuka *router*. Data yang diperoleh dari pembacaan ini dapat diteruskan ke *server* berbasis Linux sebagai kolektor. Aplikasi *open-source NFDUMP* digunakan pada *server* kolektor, yang berfungsi untuk:

- Menampung hasil tangkapan dari *Traffic Flow* melalui protokol *TCP* pada *port* 2556.
- Menganalisis data lalu lintas yang terkumpul.

Dengan integrasi ini, analisis terhadap *traffic flow* dapat dilakukan secara lebih mendalam untuk mengidentifikasi pola dan aktivitas berbahaya, sehingga sistem dapat lebih proaktif dalam melindungi jaringan dari ancaman *malicious domain*.

3.5 Perancangan dan Implementasi

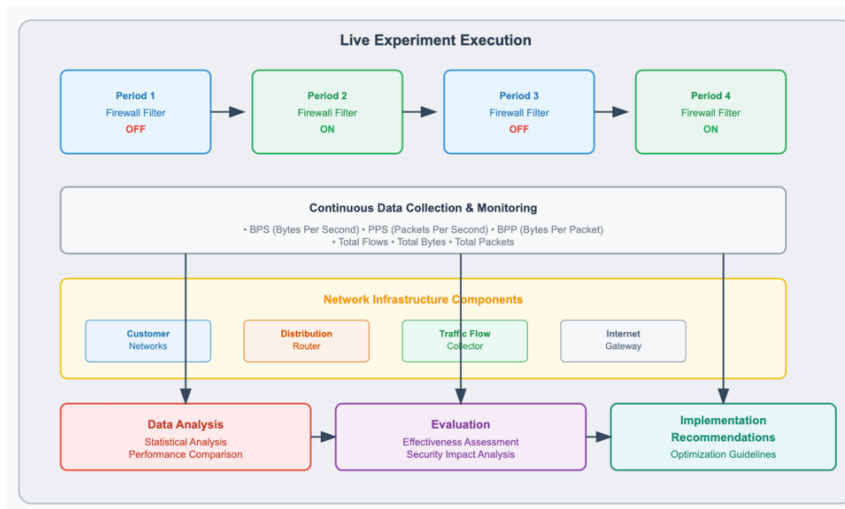
Tahap ini bertujuan untuk membangun prototipe sistem *filtering* dan menguji efektivitasnya baik dalam lingkungan simulasi maupun jaringan nyata. Pengujian dilakukan untuk

memastikan sistem dapat mengenali *domain* berbahaya yang terdaftar dalam *blacklist* secara akurat, serta untuk mengevaluasi kinerja sistem berdasarkan beberapa parameter:

- *True Positive Rate (TPR)* – Ketika *firewall* memblokir *traffic* yang sebenarnya berbahaya dan berpotensi merusak.
- *True Negatif Rate (TNR)* - Ketika *firewall* tidak memblokir *traffic* yang sebenarnya aman dan tidak berbahaya.
- *False Positive Rate (FPR)* – Ketika *firewall* memblokir *traffic* yang sebenarnya aman dan tidak berbahaya.
- *False Negatif Rate (FNR)*: Ketika *firewall* tidak memblokir *traffic* yang sebenarnya berbahaya dan berpotensi merusak.
- Jumlah *Traffic* yang *tercapture* – Volume data lalu lintas yang berhasil direkam.
- Fungsi *Firewall* – Kemampuan *firewall* dalam memblokir akses menuju *domain* berbahaya.

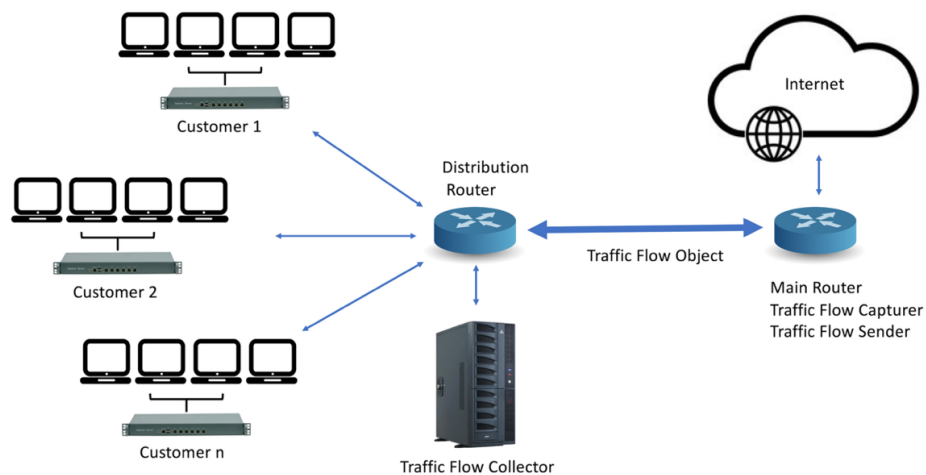
Dalam konteks firewall filter, benchmark utamanya adalah untuk meminimalkan False Positif dan False Negatif, serta memaksimalkan True Positif dan True Negatif.

Gambar 3.2 Memberikan gambaran tahapan dalam *live experiment* yang dilakukan untuk menguji efektivitas *firewall* filter dalam infrastruktur jaringan. Eksperimen dilakukan dalam 4 periode dengan konfigurasi bergantian antara *firewall filter ON* dan *OFF*, dimulai dari Periode 1 (*OFF*), Periode 2 (*ON*), Periode 3 (*OFF*), dan Periode 4 (*ON*). Selama eksperimen berlangsung, sistem melakukan pengumpulan data kontinyu dan pemantauan terhadap berbagai metrik kinerja seperti *BPS* (*Bytes Per Second*), *PPS* (*Packets Per Second*), *BPP* (*Bytes Per Packet*), serta total *flows*, *bytes*, dan *packets* pada seluruh komponen infrastruktur jaringan yang terdiri dari *Customer Networks*, *Distribution Router*, *Traffic Flow Collector*, dan *Internet Gateway*. Data yang terkumpul kemudian diolah melalui tahap analisis data dengan perbandingan kinerja statistik, dilanjutkan dengan evaluasi untuk menilai efektivitas dan dampak keamanan, dan akhirnya menghasilkan rekomendasi implementasi berupa panduan optimalisasi sistem *firewall* berdasarkan hasil eksperimen tersebut.



Gambar 3.2 Tahapan dalam *live experiment*.

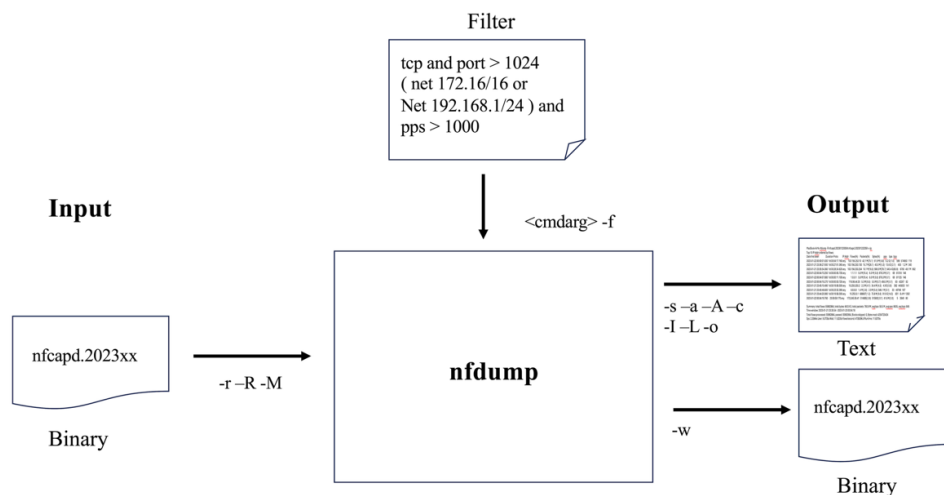
Gambar 3.3 Mengilustrasikan arsitektur sistem pengumpulan data lalu lintas jaringan (*NetFlow*) yang terdiri dari beberapa komponen strategis: Jaringan pelanggan (*Customer 1, 2, hingga n*) terhubung ke *Distribution Router* yang menjadi titik agregasi lalu lintas, sementara *Main Router* berfungsi sebagai *Traffic Flow Capturer* dan *Traffic Flow Sender* yang menghubungkan jaringan lokal dengan *Internet*. Data lalu lintas jaringan (*Traffic Flow Object*) dikumpulkan dari *interface* antara kedua *router* dan dikirimkan ke *Traffic Flow Collector* yang berperan sebagai *server* penyimpanan dan pemrosesan data. Sistem ini memungkinkan pemantauan dan analisis komprehensif terhadap pola lalu lintas jaringan dari berbagai segmen pengguna, memberikan visibilitas untuk keperluan keamanan, optimasi kinerja, dan forensik digital dari *file-file nfcapd* yang didapat.



Gambar 3.3 Prototipe diagram penelitian.

NFDUMP, sebuah perangkat lunak *open-source*, digunakan pada *server Traffic Flow Collector* sebagai kolektor data lalu lintas jaringan. *Command line* dalam *nfdump* bisa digunakan untuk memfilter keluaran dari *flow file record* dengan format *nfcapd* (Jurkiewicz, 2022; Jurkiewicz *et al.*, 2021). Lalu lintas yang dikirimkan oleh *router MikroTik* akan disimpan dalam format *file* biner dengan penamaan *nscapd.YYYYYMMDDhhmm*, yang dihasilkan setiap lima menit.

Gambar 3.4 memberikan ilustrasi alur pemrosesan data jaringan menggunakan utilitas *nfdump*. *File binary input* bernama *nfcapd.2023xxxx* merupakan *file-file* capture jaringan dari DATA. *File-file* tersebut diproses menggunakan suatu parameter untuk membaca data. Proses *nfdump* kemudian menerapkan *filter* spesifik untuk mengidentifikasi koneksi *TCP*. *Output* dari pemrosesan dapat berupa dua format: format teks yang menghasilkan laporan terstruktur, atau format *binary* yang menghasilkan *file binary* baru dengan nama serupa *nfcapd.2023xx*. Proses ini mencerminkan metodologi analisis lalu lintas jaringan yang sistematis untuk penelitian keamanan jaringan atau forensik digital. Dengan pendekatan ini, sistem mampu merekam dan menganalisis data *traffic flow* secara efisien, yang memungkinkan pengambilan keputusan berdasarkan pola lalu lintas yang diamati. Hasil analisis tidak hanya mendukung pengujian prototipe, tetapi juga menjadi dasar untuk evaluasi efektivitas sistem dalam mendeteksi dan memitigasi ancaman *malicious domain*.



Gambar 3.4 NFDUMP sebagai *software* kolektor dan *analyzer*.

Pada bagian ini, serangkaian eksperimen dilakukan untuk mensimulasikan mekanisme sesuai dengan alur kerja yang direncanakan secara parsial. Langkah pertama meliputi pengambilan, modifikasi, dan pengujian *file address-list*, yang bertujuan untuk

memastikan bahwa *router* dapat menerima format skrip *file address-list* dan mengenali alamat *IP* dari *domain* berbahaya yang terdaftar dalam *database*. Langkah kedua menekankan pada penerapan dan verifikasi fungsi pemblokiran terhadap daftar alamat *IP* yang terdeteksi sebagai *malicious domain*, memastikan bahwa *router* beroperasi dengan efektif.

Pada langkah ketiga, *router* dikonfigurasi sebagai sistem *capturer*, yang dapat membaca lalu lintas (*traffic*) yang mengalir pada jalur yang dideteksi. *Router* juga berfungsi sebagai sender untuk mengirimkan hasil tangkapan data ke *collector*, di mana data tersebut akan disimpan. Desain akhir yang telah disimulasikan diimplementasikan pada mesin produksi *ISP*, dengan *blacklist* diintegrasikan menggunakan fitur *address-list* pada *Router Traffic Flow Capturer & Sender*.

Untuk memastikan validitas statistik dari temuan kami, kami mengumpulkan data selama empat periode waktu yang berbeda, bergantian antara kondisi *firewall* AKTIF dan *firewall* NONAKTIF, dengan setiap periode berlangsung selama sembilan hari. Pendekatan ini memungkinkan kami untuk mempertimbangkan variasi temporal dalam lalu lintas jaringan dan perilaku pengguna. Kami menghitung interval kepercayaan untuk semua metrik kunci (95% CI) dan melakukan pengujian signifikansi statistik ($p < 0,05$) untuk memvalidasi bahwa perbedaan yang diamati antara kondisi *firewall* bukanlah karena variasi acak (Motulsky *et al.*, 2025). Faktor-faktor perancu potensial, seperti perubahan dalam perilaku pengguna atau kondisi jaringan eksternal, dipantau sepanjang periode penelitian.

Penting untuk memastikan bahwa *router ISP* dapat mengenali alamat *IP* dari *domain-domain* yang terdaftar dalam *address-list*. Selanjutnya, pengaturan *firewall* dilakukan untuk drop semua paket permintaan klien yang destination *address*nya tercantum dalam *blacklist address*, serta mencatat alamat *IP* dari jaringan lokal (*Local Area Network/LAN*) yang melakukan permintaan, dalam sebuah *log*.

Proses *traffic capture* dilakukan secara kontinu selama 24 jam setiap harinya selama dua minggu di setiap tahap percobaan. Tahap pertama menangkap lalu lintas saat *firewall* dalam keadaan aktif (*ON*), sedangkan tahap kedua mencatat lalu lintas saat *firewall* dinonaktifkan (*OFF*). Tahap ketiga kembali melakukan penangkapan lalu lintas dengan *firewall* aktif (*ON*), diikuti oleh tahap keempat yang mencakup penangkapan *traffic* saat *firewall* dinonaktifkan (*OFF*). Pengamatan dilakukan secara berkala untuk memastikan bahwa data disimpan di *collector* sesuai dengan yang diharapkan. Mengingat volume data yang dihasilkan, perhatian khusus diberikan untuk memastikan ruang *hard disk drive (HDD)* mencukupi kapasitas untuk menampung data.

3.6 Analisis dan Evaluasi

Tahapan ini bertujuan untuk menganalisis hasil pengujian dalam rangka menilai efektivitas metode yang telah dikembangkan. Analisis dilakukan dengan membandingkan hasil *traffic* jaringan saat *firewall* diaktifkan dan saat *firewall* dinonaktifkan. Perbandingan ini bertujuan untuk mengukur peningkatan perlindungan jaringan yang dicapai oleh sistem. Selama proses ini, semua alamat *IP* yang terblokir dan yang melakukan permintaan ke *malicious domain* dicatat secara rinci. Hasil evaluasi mencakup:

- Jumlah *domain* yang berhasil dideteksi dan diblokir.
- Tingkat keberhasilan sistem dalam mengidentifikasi *domain* berbahaya tanpa memengaruhi performa jaringan.
- Efektivitas *firewall* dalam mencegah akses ke *domain* berbahaya berdasarkan jumlah permintaan yang berhasil dihentikan.

3.7 Rekomendasi Implementasi

Berdasarkan hasil penelitian, rekomendasi praktis disusun untuk diimplementasikan dalam sistem keamanan jaringan. Rekomendasi ini dirancang untuk mendukung organisasi dalam menentukan strategi *firewall* yang sesuai dengan prinsip sederhana (*simplicity*) tanpa mengorbankan efektivitas (*complexity*). Tujuannya adalah memberikan panduan untuk memilih strategi *firewall* yang tepat berdasarkan kebutuhan spesifik, seperti:

- Memanfaatkan daftar alamat *blacklist* berbasis *domain* untuk mencegah akses ke *domain* berbahaya.
- Mengintegrasikan sistem *filtering* ke dalam infrastruktur yang ada tanpa memerlukan investasi besar.
- Menyeimbangkan efisiensi dan kompleksitas sistem agar dapat diadaptasi dengan berbagai skenario lingkungan jaringan.

BAB 4

Hasil dan Pembahasan

Penelitian yang mencakup perencanaan, simulasi, dan implementasi menunjukkan bahwa sistem berjalan dengan baik tanpa mengganggu operasional *ISP*. Selama proses penelitian, mulai dari awal hingga akhir, data-data pendukung yang dihasilkan cukup memadai untuk dilakukan analisis lebih lanjut. Dari perspektif forensika digital, implementasi sistem *firewall* ini menghasilkan berbagai artefak digital yang dapat dianalisis untuk memahami pola serangan dan efektivitas sistem. Proses *logging* dan *monitoring* yang terintegrasi dalam sistem memungkinkan pelacakan komprehensif terhadap aktivitas jaringan, mulai dari upaya akses ke domain berbahaya hingga respons sistem terhadap ancaman. Artefak-artefak digital bisa ditemukan dalam *log firewall*, rekaman *DNS query* yang bersumber dari jaringan lokal, *metadata* koneksi jaringan yang bisa ditemukan dalam data traffic flow, dan jejak digital dari berbagai upaya serangan yang berhasil diblokir. Analisis forensik terhadap artefak-artefak ini memberikan wawasan mendalam tentang karakteristik ancaman, pola perilaku penyerang, dan tingkat keberhasilan sistem pertahanan yang telah diimplementasikan.

Desain penelitian ini memberikan solusi ideal bagi *ISP* kecil dengan keterbatasan sumber daya. Pendekatan ini secara ekonomi tidak membutuhkan investasi tambahan karena memanfaatkan infrastruktur yang sudah ada, seperti *Distribution Router*, *Main Router*, dan *Traffic Flow Collector*. Sistem *firewall* dapat diaktifkan hanya dengan menyesuaikan konfigurasi perangkat yang ada, sementara data analisis keamanan diperoleh dari artefak *traffic flow* yang otomatis tersedia dalam sistem. Solusi ini sangat cocok untuk *ISP* dengan modal terbatas, pendapatan kecil, dan sumber daya minim karena tidak memerlukan perangkat keras baru, lisensi mahal, atau tenaga ahli tambahan. Dengan memanfaatkan infrastruktur dan *firewall* yang sudah tersedia, *ISP* dapat mengimplementasikan sistem monitoring dan keamanan yang efektif tanpa menambah beban biaya operasional.

4.1 Malicious Domain Address List

Malicious domains merupakan salah satu sumber daya utama yang diperlukan oleh penyerang untuk menjalankan serangan melalui internet (Alhayan *et al.*, 2025). Deteksi malicious domain adalah salah satu pendekatan paling efektif yang diterapkan dalam mendeteksi *Advanced Persistent Threat (APT)*, ancaman dan tersembunyi dalam jaringan (Almarzooqi *et al.*, 2022; Palaniappan *et al.*, 2020b). *Malicious DNS* merujuk pada nama

domain yang digunakan untuk aktivitas jahat atau untuk tujuan analisis nama domain memberikan wawasan kepada para ahli keamanan untuk mengidentifikasi *Command and Control (C&C)* yang merupakan infrastruktur penting dalam serangan siber. Aktor jahat dapat mengeksploitasi kerentanan ini untuk mendapatkan data sensitif, meluncurkan serangan *distributed denial-of-service (DDoS)*, dan bahkan mengambil alih infrastruktur kritis (Tariq *et al.*, 2023a). Dengan demikian, malicious domain menjadi ancaman serius dalam ekosistem keamanan siber karena berfungsi sebagai gerbang utama untuk berbagai jenis serangan yang dapat merugikan organisasi dan individu.

Data *malicious domain* dari *CERT.PL* yang aktif mulai tahun 1996 memiliki keunggulan signifikan dibandingkan data serupa yang tersedia di Indonesia karena karakteristik kualitas, cakupan, dan infrastruktur yang lebih matang. *CERT.PL* diakui sebagai "*high quality phishing domain list*" yang digunakan oleh *mobile ISPs* Polandia (Cremer *et al.*, 2022a), dengan *Warning List* yang telah beroperasi selama hampir lima tahun dan berhasil menghentikan jutaan percobaan koneksi ke domain berbahaya, menjadikannya sebagai "*most effective tool in the fight against phishing websites*" (Ahsan *et al.*, 2022). *Database CERT.PL* mencakup hampir 42,000 domain berbahaya yang ditambahkan sepanjang tahun, dengan 33,000 domain ditambahkan pada tahun 2021 (Tariq *et al.*, 2023b), menunjukkan proses kurasi yang intensif dan berkelanjutan. Sistem mereka menyediakan data dalam berbagai format (*CSV, JSON, TXT, XML, AdBlock, hosts, MikroTik RSC, RPZ*) dengan pembaruan harian hingga menunjukkan infrastruktur teknologi yang *robust*. Sementara itu, Indonesia masih menghadapi tantangan serius dalam *cybersecurity* dengan 130 serangan *ransomware* unik, 4,046 serangan *phishing*, dan masuk ranking global dalam jumlah data *breaches* pada tahun 2023 (Socradar, 2024), yang mengindikasikan bahwa infrastruktur *threat intelligence* domestik masih dalam tahap pengembangan dibandingkan dengan standar internasional seperti *CERT.PL*.

Banyaknya format *address list* yang disediakan *CERT.PL* memberikan peluang penerapan dalam berbagai platform *router*. Penerapan dalam *router MikroTik*, format daftar alamat yang dipilih adalah *blacklist* dengan ekstensi *file .rsc*. Berdasarkan analisis terhadap berbagai format *file* yang tersedia, ditemukan bahwa *file .rsc* memiliki isi yang minimal, sementara *file .csv* berisi data yang jauh lebih lengkap, sehingga diperlukan konversi dari *file .csv* menjadi *.rsc*

Hingga akhir tahun 2022, *file .csv* mencatat total 84.765 *malicious domain*. Mengingat keterbatasan isi *file .rsc*, dilakukan proses modifikasi dan konversi data dari

format *.csv* menjadi *.rsc* untuk memenuhi kebutuhan implementasi *firewall* di *router MikroTik*. Proses konversi ini memastikan bahwa seluruh daftar *domain* berbahaya dapat diintegrasikan ke dalam sistem dengan format yang kompatibel tanpa kehilangan data penting (Stock & De Carlo, 2023; Yu *et al.*, 2021). Dengan melakukan hal tersebut, integritas informasi tetap terjaga, dan *firewall* lebih siap untuk melindungi dari ancaman-ancaman ini (Karpowicz, 2021; Van Der Toorn *et al.*, 2022).

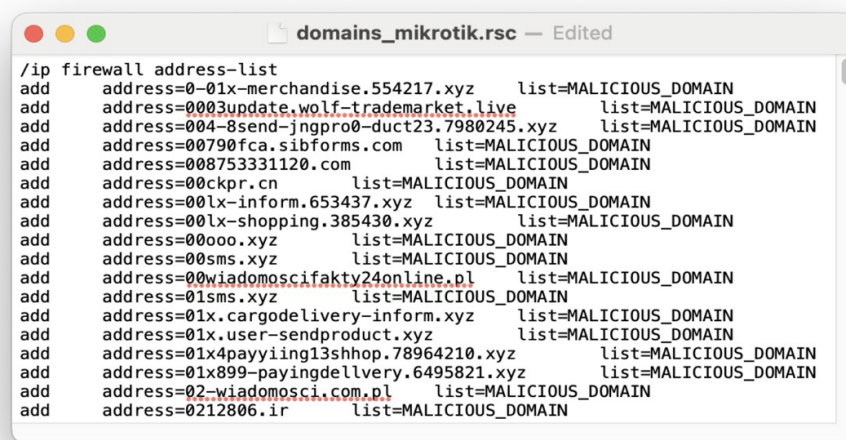
Tabel 4.1 Menampilkan 10 entri terakhir dari total 84.764 *domain* berbahaya yang tercatat dalam database *cert.pl*, dimana semua *domain* ini ditambahkan pada tanggal 31 Desember 2022 dalam rentang waktu sekitar 14 jam dari pukul 07:38 hingga 21:55. *Domain-domain* tersebut menunjukkan berbagai pola serangan siber yang umum, termasuk *typosquatting* seperti "*netflspl.com*" yang menyamar sebagai *Netflix*, "*dicsord-snows.com*" yang meniru *Discord*, serta beberapa *domain* dengan struktur *phishing* yang mencurigakan seperti "*profisthebitsera-po*" dengan berbagai *subdomain* dan "*inpost-pl-hid226ks.antenaj3.xyz*" yang tampaknya menargetkan layanan pos Polandia. Keberagaman ekstensi *domain* (*.com*, *.xyz*, *.dev*) dan penggunaan nama yang menyesatkan seperti "*auth-web-apps.com*" dan "*my-diya-inc.com*" mengindikasikan upaya sistematis untuk menipu pengguna melalui berbagai vektor serangan, mulai dari *social engineering* hingga *credential harvesting*, yang semuanya berhasil diidentifikasi dan dikategorikan sebagai ancaman oleh sistem *monitoring cert.pl* pada akhir tahun 2022.

Tabel 4. 1 Daftar 10 Baris Terakhir dari 84765 *Malicious Domain Address List File .csv*

No	Domain Address	Date of Entry
84756	balender3d.com	2022-12-31T07:38:55
84757	profisthebitsera-po.fababeok.com	2022-12-31T07:54:22
84758	inpost-pl-hid226ks.antenaj3.xyz	2022-12-31T13:00:19
84759	netflspl.com	2022-12-31T13:45:02
84760	profisthebitsera-po.betulaup.com	2022-12-31T15:33:59
84761	auth-web-apps.com	2022-12-31T15:38:22
84762	zz3v1x.webwave.dev	2022-12-31T15:47:46
84763	dicsord-snows.com	2022-12-31T17:59:27
84764	my-diya-inc.com	2022-12-31T20:14:59
84765	profisthebitsera-po.biferyal.com	2022-12-31T21:55:51

Gambar 4.1 Menunjukkan sebuah *file script MikroTik* dengan ekstensi *.rsc* yang berisi bagian konfigurasi *firewall address-list* untuk memblokir *domain-domain* berbahaya. *Script* ini terdiri dari serangkaian perintah "*add*" yang menambahkan alamat *IP* berdasarkan nama *domain* ke dalam *address-list* bernama "*MALICIOUS_DOMAIN*", dimana setiap entri

merepresentasikan situs *web* yang berpotensi membahayakan seperti situs *phishing*, *malware*, atau penipuan *online* (terlihat dari nama *domain* seperti merchandise palsu, layanan pengiriman palsu, dan *platform* pembayaran mencurigakan). *File* ini merupakan hasil konversi dari data *CSV* yang diperoleh dari *cert.pl*. *File csv* diformat ulang menjadi *script MikroTik* agar dapat diimport langsung ke *router* dan secara otomatis *MikroTik* akan mengenali *IP address* dari *domain* yang terdaftar dalam menu `/ip firewall address-list`. Bila ada perubahan *IP* dalam *DNS zone domain*, maka *IP* dalam *address list* juga akan berubah secara otomatis.



Gambar 4.1 Hasil konversi *address list .rsc* dari *.csv file*

4.1.1 Analisis Forensik Artefak Domain Berbahaya

Dari perspektif forensika digital, artefak-artefak yang ditemukan dalam *database cert.pl* menunjukkan pola serangan yang dapat dikategorikan berdasarkan teknik dan target yang digunakan. Analisis forensik terhadap 84.765 domain berbahaya mengungkap beberapa kategori artefak penting:

1. Artefak *Typosquatting*: Domain seperti "*netflspl.com*" dan "*dicsord-snows.com*" menunjukkan pola substitusi karakter yang sistematis, dimana penyerang menggunakan teknik *homoglyph attack* dan *character substitution* untuk menyamarkan *domain legitimate*. Jejak digital ini mengindikasikan upaya *social engineering* yang terstruktur dengan target *platform* populer.
2. Artefak *Subdomain Phishing*: Domain dengan struktur "*profisthebitsera-po*" yang muncul dengan berbagai *subdomain* (*fababeok.com*, *betulaup.com*, *biferyal.com*)

menunjukkan pola *campaign phishing* yang terkoordinasi. Artefak ini mengungkap bahwa penyerang menggunakan teknik *domain generation algorithm (DGA)* untuk menghindari deteksi dan mempertahankan persistensi serangan.

3. Artefak Temporal: Analisis *timestamp* dari 10 domain terakhir menunjukkan aktivitas yang intens pada tanggal 31 Desember 2022, dengan interval waktu yang bervariasi dari 16 menit hingga 6 jam. Pola temporal ini mengindikasikan kemungkinan *automated campaign* atau koordinasi manual dalam *deployment domain* berbahaya.
4. Artefak Geolokasi dan *Targeting*: Domain seperti "*inpost-pl-hid226ks.antenaj3.xyz*" menunjukkan *targeting* geografis spesifik (Polandia), sementara "*auth-web-apps.com*" mengindikasikan *targeting* berbasis layanan. Artefak ini mengungkap strategi penyerang yang menggunakan *localization* dan *service-specific targeting* untuk meningkatkan efektivitas serangan.

4.1.2 Jejak Digital dan Metadata Serangan

Proses konversi dari format `.csv` ke `.rsc` menghasilkan artefak forensik tambahan berupa metadata konversi yang mencakup *timestamp* pemrosesan, integritas data, dan *mapping* antara *format file*. Artefak-artefak ini penting untuk *audit trail* dan memastikan *chain of custody* dalam konteks forensika digital. Selain itu, implementasi *address-list* di MikroTik menghasilkan *log* forensik yang mencatat setiap upaya resolusi *DNS* terhadap *domain* yang terdaftar dalam *blacklist*, memberikan jejak digital yang dapat dianalisis untuk memahami pola serangan *real-time*. Di dalam *router*, artefak didapatkan di dalam beberapa menu berikut:

```
/ip firewall address-list
/ip dns cache
/file
```

4.2 Malicious Domain IP Traffic

Pada awal pengumpulan data, dilakukan evaluasi *traffic request* dari alamat sumber (*source address*) *client* menuju ke *malicious domain*, serta *inbound traffic* dari *malicious domain* menuju alamat lokal *client* (*local address*). Selama dua minggu pertama, dilakukan pengamatan terhadap *traffic* tersebut. Berdasarkan *log* selama penelitian, yang tercatat dalam daftar *firewall*, terdapat 1.090 alamat *IP client* yang melakukan *request* ke *malicious domain*

dan 1.919 IP terdeteksi sebagai *malicious domain*. IP-IP ini tersimpan dalam *file* bernama *REQUEST_MALICIOUS-DOMAIN* di menu */file*.

Tabel 4.2 menunjukkan 10 entri terakhir dari total 1090 alamat IP lokal yang terdeteksi melakukan *request* ke *domain* berbahaya, dimana semua entri ini tercatat dalam *file log address-list* bernama "*REQUEST_MALICIOUS-DOMAIN*" pada menu */file router MikroTik* dengan *flag "D"* yang menandakan bahwa entri tersebut bersifat dinamis. Data ini merekam aktivitas mencurigakan dari berbagai perangkat dalam jaringan lokal dengan rentang IP 192.168.x.x dan satu IP dari *subnet* 10.116.216.22, yang terjadi dalam rentang waktu sekitar 9 jam mulai dari tanggal 24 Januari 2023 pukul 23:41:51 hingga 25 Januari 2023 pukul 09:15:16. Semakin banyak IP tercatat dalam *list* ini, menandakan kinerja yang baik dari firewall. Pencatatan ini menunjukkan bahwa sistem keamanan jaringan berhasil mendeteksi dan mendokumentasikan upaya akses ke situs-situs berbahaya oleh perangkat-perangkat *internal*, yang dapat digunakan sebagai dasar untuk melakukan tindakan keamanan lebih lanjut seperti investigasi perangkat yang terinfeksi *malware* atau edukasi kepada pengguna terkait keamanan siber.

Tabel 4. 2 Daftar 10 Baris Terakhir dari 1090 IP Client Request ke Malicious Domain

#jan /25/2023 9:15:53 by RouterOS 6.49.6	
#software id = PVT6-1QVA	
#	
Flags	: X - disabled, D - dynamic
1081	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.15.81
	creation-time=jan/24/2023 23:41:51
1082	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.13.88
	creation-time=jan/25/2023 00:16:19
1083	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.26.110
	creation-time=jan/25/2023 01:38:35
1084	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.26.115
	creation-time=jan/25/2023 04:29:27
1085	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.27.240
	creation-time=jan/25/2023 07:27:24
1086	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.26.121
	creation-time=jan/25/2023 07:58:44
1087	D list=REQUEST_MALICIOUS-DOMAIN address=10.116.216.22
	creation-time=jan/25/2023 08:02:06

1088	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.15.104
	creation-time=jan/25/2023 08:47:22
1089	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.26.47
	creation-time=jan/25/2023 09:02:04
1090	D list=REQUEST_MALICIOUS-DOMAIN address=192.168.15.109
	creation-time=jan/25/2023 09:15:16

Tabel 4.3 Menampilkan 10 entri terakhir dari total 1919 alamat *IP* eksternal yang teridentifikasi sebagai sumber *traffic* berbahaya dan tercatat dalam file *log address-list* bernama "*INBOUND_MALICIOUS-DOMAIN*" pada menu */file* di *router MikroTik* dengan *flag* dinamis. Data menunjukkan berbagai alamat *IP* publik dari rentang yang berbeda, termasuk *IP* dari *Amazon Web Services* (18.67.111.x dan 13.x.x.x) serta *Cloudflare* (104.21.65.107), yang mencoba melakukan koneksi masuk ke jaringan lokal dalam periode waktu sekitar 9 jam dari tanggal 25 Januari 2023 pukul 00:14:13 hingga 09:11:45. Semakin banyak *IP* tercatat dalam *list* ini, menandakan konerja yang baik dan efektif dari firewall. Pencatatan ini mengindikasikan adanya upaya serangan atau aktivitas mencurigakan dari *server-server* eksternal yang mungkin digunakan untuk *hosting malware*, *phishing*, atau layanan berbahaya lainnya, dimana sistem keamanan jaringan berhasil mendeteksi dan mendokumentasikan setiap upaya koneksi masuk tersebut untuk keperluan *monitoring* dan analisis keamanan lebih lanjut.

Tabel 4. 3 Daftar 10 Baris Terakhir dari 1919 *IP Malicious Domain*

# jan /25/2023 9:16:55 by RouterOS 6.49.6	
# software id = PVT6-1QVA	
#	
Flags	: X - disabled, D - dynamic
1910	D list=INBOUND_MALICIOUS-DOMAIN address=18.67.111.7
	creation-time=jan/25/2023 00:14:13
1911	D list=INBOUND_MALICIOUS-DOMAIN address=18.67.111.35
	creation-time=jan/25/2023 00:24:45
1912	D list=INBOUND_MALICIOUS-DOMAIN address=18.67.111.77
	creation-time=jan/25/2023 00:31:22
1913	D list=INBOUND_MALICIOUS-DOMAIN address=18.67.111.100
	creation-time=jan/25/2023 00:31:22

1914	D list=INBOUND_MALICIOUS-DOMAIN address=13.227.74.103
	creation-time=jan/25/2023 00:47:53
1915	D list=INBOUND_MALICIOUS-DOMAIN address=104.21.65.107
	creation-time=jan/25/2023 01:11:42
1916	D list=INBOUND_MALICIOUS-DOMAIN address=13.225.95.77
	creation-time=jan/25/2023 01:24:56
1917	D list=INBOUND_MALICIOUS-DOMAIN address=13.225.95.127
	creation-time=jan/25/2023 03:26:07
1918	D list=INBOUND_MALICIOUS-DOMAIN address=13.33.88.12
	creation-time=jan/25/2023 04:03:03
1919	D list=INBOUND_MALICIOUS-DOMAIN address=13.227.74.66
	creation-time=jan/25/2023 09:11:45

4.3 Analisis Forensik Sistem *Firewall* dan Deteksi Ancaman

Implementasi sistem *firewall* menghasilkan berbagai artefak forensik yang dapat dianalisis untuk evaluasi efektivitas dan investigasi insiden keamanan. Sistem *logging* yang terintegrasi dalam *router MikroTik* menghasilkan artefak digital berupa *connection logs*, *DNS query logs*, dan *firewall rule execution logs* yang menjadi bukti digital penting dalam proses forensik.

1. Artefak *Connection Logs*: Setiap upaya koneksi ke domain yang terdaftar dalam *blacklist* menghasilkan *log entry* yang mencatat *source IP*, *destination domain*, *timestamp*, dan *action* yang diambil sistem. Artefak ini memberikan jejak digital lengkap tentang upaya akses ke situs berbahaya dan memungkinkan analisis pola perilaku pengguna serta identifikasi potensi *compromise*.
2. Artefak *DNS Query* Forensik: Sistem mencatat setiap *query DNS* yang diarahkan ke domain berbahaya, termasuk *metadata* seperti *query type*, *response code*, dan *resolution time*. Artefak ini penting untuk memahami bagaimana *malware* atau *script* berbahaya mencoba melakukan *command and control communication* atau *data exfiltration*.

Analisis forensik terhadap artefak-artefak ini memungkinkan evaluasi efektivitas sistem melalui pemantauan jejak digital yang dihasilkan. Jejak digital yang tercatat juga memungkinkan rekonstruksi *timeline* serangan dan identifikasi sumber ancaman untuk keperluan investigasi lanjutan, serta menjadi dasar untuk pengembangan strategi mitigasi yang lebih efektif.

4.4 Validasi Ancaman Berdasarkan Karakteristik *Domain* Berbahaya

Analisis forensik terhadap 84.765 domain berbahaya dalam *database cert.pl* menunjukkan bukti konkret bahwa *malicious domain* merupakan ancaman nyata yang memerlukan mitigasi serius. Beberapa indikator ancaman yang dapat diidentifikasi dari data penelitian:

1. Tingkat Aktivitas Tinggi: Penambahan 10 *domain* berbahaya dalam rentang waktu 14 jam (31 Desember 2022) menunjukkan aktivitas penciptaan domain berbahaya yang intensif. Pola temporal ini mengindikasikan bahwa ancaman *malicious domain* bukan merupakan kejadian sporadis, melainkan upaya sistematis dan berkelanjutan yang memerlukan respons proaktif.
2. Diversifikasi Teknik Serangan: Analisis terhadap 10 domain terakhir mengungkap variasi teknik yang digunakan penyerang, mulai dari *typosquatting* ("*netflspl.com*" menyamar sebagai *Netflix*, "*discord-snows.com*" meniru *Discord*) hingga subdomain *phishing* dengan pola "*profisthebitsera-po*" yang muncul di berbagai *host*. Diversifikasi ini menunjukkan tingkat sofistikasi ancaman yang tinggi dan kemampuan adaptasi penyerang terhadap mekanisme deteksi.
3. Targeting Geografis dan Layanan Spesifik: Domain seperti "*inpost-pl-hid226ks.antenaj3.xyz*" yang menargetkan layanan pos Polandia dan "*auth-web-apps.com*" yang menyasar layanan otentikasi menunjukkan bahwa penyerang melakukan *reconnaissance* dan *targeting* yang spesifik. Hal ini mengindikasikan ancaman yang terorganisir dengan tujuan yang jelas.
4. Persistensi dan Redundansi: Kemunculan domain "*profisthebitsera-po*" di tiga *subdomain* berbeda (*fababeok.com*, *betulaup.com*, *biferyal.com*) dalam satu hari menunjukkan upaya penyerang untuk mempertahankan persistensi serangan meskipun sebagian domain telah diblokir. Strategi redundansi ini mengonfirmasi bahwa *malicious domain* merupakan ancaman yang *persistent* dan memerlukan pendekatan komprehensif.
5. Kompleksitas Identifikasi: Penggunaan ekstensi *domain* yang beragam (*.com*, *.xyz*, *.dev*) dan nama yang tampak legitim seperti "*auth-web-apps.com*" dan "*my-diya-inc.com*" menunjukkan upaya penyerang untuk menghindari deteksi manual. Kompleksitas ini membuktikan bahwa identifikasi *malicious domain* memerlukan sistem otomatis dan *database* yang terus diperbarui.
6. Volume Ancaman: Total 84.765 domain berbahaya yang tercatat hingga akhir 2022 menunjukkan skala ancaman yang sangat besar. Volume ini mengindikasikan bahwa

tanpa sistem mitigasi yang efektif, organisasi akan terpapar risiko yang signifikan dari berbagai vektor serangan.

Analisis forensik terhadap karakteristik ini memungkinkan evaluasi efektivitas sistem melalui pemantauan jejak digital yang dihasilkan. Jejak digital yang tercatat juga memungkinkan rekonstruksi *timeline* serangan dan identifikasi sumber ancaman untuk keperluan investigasi lanjutan, serta menjadi dasar untuk pengembangan strategi mitigasi yang lebih efektif berdasarkan pemahaman mendalam tentang *nature* ancaman yang dihadapi. Dalam konteks ini tujuan utama adalah untuk meminimalkan False Positif dan False Negatif, serta memaksimalkan True Positif dan True Negatif.

Berbagai usaha untuk meningkatkan kesadaran terhadap *social engineering* telah dilakukan melalui pendekatan yang komprehensif dan sistematis. Penelitian sistematis mengenai metode pelatihan *cybersecurity* telah dikembangkan untuk merancang dan menguji program pelatihan berbasis bukti bagi karyawan (Syafitri *et al.*, 2022) dimana program pelatihan kesadaran keamanan bertujuan untuk meningkatkan kesadaran keamanan dalam organisasi dan harus menjadi program berkelanjutan karena pelatihan cenderung terlupakan seiring waktu. Upaya-upaya ini mencakup pelatihan kesadaran keamanan yang dilakukan secara berkala melalui berbagai mekanisme seperti pembaruan *email*, pesan SMS, distribusi materi di meja kerja, petugas lantai kantor, *hackathon*, dan bahkan program *bug bounty* untuk melibatkan staf dalam proses keamanan (Aldawood & Skinner, 2019a). Selain itu, telah dilakukan tinjauan literatur sistematis untuk meningkatkan kesadaran keamanan informasi karyawan di organisasi swasta dan publik guna menjaga kerahasiaan, integritas, dan ketersediaan aset sistem informasi sensitif (Neetu Bansla *et al.*, 2019). Namun, tantangan yang dihadapi menunjukkan bahwa meskipun organisasi menganggap perlindungan *cybersecurity* sebagai prioritas utama, pengetahuan *cybersecurity* sumber daya manusia masih sangat rendah, terutama dalam hal kesadaran terhadap *social engineering* (Siddiqi *et al.*, 2022a), sehingga memerlukan pendekatan yang lebih efektif dan berkelanjutan dalam implementasi program pelatihan dan kesadaran keamanan.

Meskipun kesadaran manusia terhadap ancaman *malicious domain* terus mengalami peningkatan, perlindungan menggunakan *firewall* masih menjadi kebutuhan yang sangat krusial dalam arsitektur keamanan siber modern. Hal ini disebabkan oleh kondisi psikologis manusia yang secara inheren masih rentan terhadap serangan *social engineering*, sebagaimana dijelaskan dalam penelitian interdisipliner yang menyatakan bahwa *social engineering* didefinisikan sebagai tindakan menggunakan manipulasi dan penipuan untuk mendapatkan akses ke informasi rahasia (Aldawood & Skinner, 2019b). Faktor psikologis

manusia menjadi titik kelemahan utama karena *social engineering* sebagai proses yang mengeksploitasi psikologi manusia daripada metode *hacking* yang canggih (Siddiqi *et al.*, 2022a), dimana penyerang memanfaatkan aspek emosional, motivasi, dan perilaku alami manusia dalam situasi profesional. Kondisi ini diperkuat oleh temuan yang menunjukkan bahwa orang mudah terjebak dalam trik *social engineering* berdasarkan konteks psikologis, sehingga meskipun individu telah memiliki pengetahuan tentang ancaman domain berbahaya, mereka tetap dapat dimanipulasi melalui teknik-teknik psikologis yang kompleks. Oleh karena itu, implementasi *firewall* sebagai lapisan perlindungan teknis tetap diperlukan untuk mengkompensasi kerentanan psikologis manusia yang sulit dihilangkan sepenuhnya, menciptakan sistem keamanan berlapis yang menggabungkan aspek teknologi dan *awareness* manusia.

4.5 Traffic Flow Data Analysis

Tahapan pengumpulan data menggunakan *server collector*, diperoleh data sebesar 199,21 GB yang tersimpan dalam 12.351 *file*. Setiap artefak *file* mencatat *traffic capture* selama 5 menit. Dari seluruh data, diambil empat kelompok sampel untuk dianalisis, masing-masing kelompok merepresentasikan periode selama sembilan hari:

- ☐ Periode pertama: *firewall* filter dalam kondisi *OFF*.
- ☐ Periode kedua: *firewall* filter dalam kondisi *ON*.
- ☐ Periode ketiga: *firewall* filter dalam kondisi *OFF*.
- ☐ Periode keempat: *firewall* filter dalam kondisi *ON*.

Gambar 4.2 Folder DATA_PENELITIAN yang ditampilkan pada gambar merupakan repositori penelitian yang substansial dengan ukuran 199,21 GB, terdiri dari 12.351 item. Gambar menunjukkan bahwa *dataset* penelitian ini merupakan koleksi *file* yang komprehensif dan terorganisir yang tidak dikonfigurasi sebagai *folder* bersama atau terkunci, sehingga memungkinkan akses penuh terhadap konten penelitian yang disimpan di dalamnya.

Dataset file nfcapd di atas merupakan kumpulan artefak digital yang sangat berharga untuk analisis forensika jaringan. Setiap *file* yang merekam *traffic capture* selama 5 menit menghasilkan artefak forensik berupa *network flow records* yang dapat dianalisis untuk memahami pola komunikasi, mengidentifikasi anomali keamanan, dan melakukan investigasi insiden. Metodologi pengumpulan data dengan empat periode berbeda (*firewall*

ON/OFF secara bergantian) menciptakan artefak komparatif yang memungkinkan analisis *differential* forensik untuk mengevaluasi efektivitas sistem keamanan.

Dari perspektif forensika digital, struktur data yang terorganisir ini memungkinkan implementasi *chain of custody* yang proper, dimana setiap artefak dapat dilacak integritasnya dari waktu pengumpulan hingga analisis. Format *nfcapd* yang digunakan merupakan standar industri untuk penyimpanan *network flow data*, memastikan bahwa artefak digital yang dihasilkan dapat diverifikasi dan direproduksi untuk keperluan investigasi forensik.



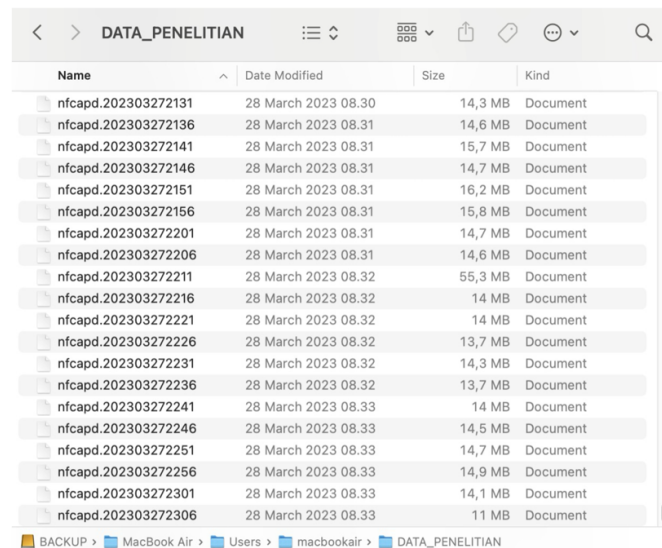
Gambar 4.2 Data yang dikumpulkan dari *server* kolektor.

Folder DATA_PENELITIAN pada Gambar 4.3 menunjukkan kumpulan dokumen penelitian berformat *nfcapd* dengan penamaan sistematis menggunakan penomoran sekuensial (20230327XXXX). Semua *file* dibuat otomatis dalam rentang waktu penelitian dengan ukuran bervariasi. Dokumen-dokumen tersebut merupakan data penelitian berbasis perekaman jaringan (*network flow capture data*) tersimpan secara terorganisir dalam hierarki penyimpanan, menandakan koleksi data penelitian yang konsisten dengan format penomoran sekuensial dan diambil dalam waktu yang relatif berdekatan. *File* dalam *collector* disimpan dalam format biner dengan nama *nfcapd.YYYYMMDDhhmm*, dan kapasitas tiap *file* bervariasi. Struktur yang terorganisir ini memungkinkan peneliti atau administrator jaringan untuk menganalisis data lalu lintas jaringan yang ditangkap secara detail, memantau pola lalu lintas dan berpotensi mengidentifikasi anomali atau insiden keamanan (Jacob *et al.*, 2022). Contoh perintah analisis untuk menampilkan *Total Flows*, *Total Bytes*, *Total Packets*, *Avg bps*, *Avg pps*, dan *Avg bpp* pada 14 Januari 2023 adalah sebagai berikut:

```
nfdump -R nfcapd.202301140000:nfcapd.202301142355 -s ip
```

Artefak digital yang tersimpan dalam format *nfcapd.YYYYMMDDhhmm* mengandung *metadata* temporal yang kritis untuk analisis forensika jaringan. Setiap *file* berfungsi sebagai *timestamp container* yang menyimpan jejak digital aktivitas jaringan pada periode waktu spesifik. Variasi ukuran *file* yang terekam mencerminkan fluktuasi aktivitas jaringan dan dapat menjadi indikator awal adanya anomali atau insiden keamanan.

Dari sudut pandang forensika digital, konsistensi penamaan *file* dengan format sekuensial menciptakan *audit trail* yang dapat diandalkan untuk rekonstruksi *timeline* kejadian. Artefak ini memungkinkan *investigator* untuk melakukan *temporal correlation analysis*, dimana aktivitas mencurigakan dapat dikorelasikan dengan *event* lain yang terjadi dalam rentang waktu yang sama, baik dari log sistem maupun dari sumber artefak digital lainnya.



Name	Date Modified	Size	Kind
nfcapd.202303272131	28 March 2023 08.30	14,3 MB	Document
nfcapd.202303272136	28 March 2023 08.31	14,6 MB	Document
nfcapd.202303272141	28 March 2023 08.31	15,7 MB	Document
nfcapd.202303272146	28 March 2023 08.31	14,7 MB	Document
nfcapd.202303272151	28 March 2023 08.31	16,2 MB	Document
nfcapd.202303272156	28 March 2023 08.31	15,8 MB	Document
nfcapd.202303272201	28 March 2023 08.31	14,7 MB	Document
nfcapd.202303272206	28 March 2023 08.31	14,6 MB	Document
nfcapd.202303272211	28 March 2023 08.32	55,3 MB	Document
nfcapd.202303272216	28 March 2023 08.32	14 MB	Document
nfcapd.202303272221	28 March 2023 08.32	14 MB	Document
nfcapd.202303272226	28 March 2023 08.32	13,7 MB	Document
nfcapd.202303272231	28 March 2023 08.32	14,3 MB	Document
nfcapd.202303272236	28 March 2023 08.32	13,7 MB	Document
nfcapd.202303272241	28 March 2023 08.33	14 MB	Document
nfcapd.202303272246	28 March 2023 08.33	14,5 MB	Document
nfcapd.202303272251	28 March 2023 08.33	14,7 MB	Document
nfcapd.202303272256	28 March 2023 08.33	14,9 MB	Document
nfcapd.202303272301	28 March 2023 08.33	14,1 MB	Document
nfcapd.202303272306	28 March 2023 08.33	11 MB	Document

Gambar 4.3 Data *file nfcapd* dari *server* kolektor.

Pada dasarnya nalisis data traffic flow dilakukan menggunakan berbagai alternatif selain NFDUMP, yang masing-masing memiliki kelebihan dan karakteristik spesifik. Alternatif open-source yang tersedia antara lain *pmacct* untuk *collecting* dan *accounting traffic flow*, *Wireshark* (Luca Deri, n.d.) untuk *deep packet inspection* dan analisis protokol secara detail, *tcpdump* untuk *command-line packet capture* dan *basic analysis*, serta *ntopng* untuk *real-time monitoring* dan *web-based interface*. Beberapa *tools* lainnya seperti *Zeek* (formerly *Bro*) unggul untuk *network forensics* dan *security-oriented analytics* (Cremer et al., 2022b), sementara *flow-tools*, *cflowd*, *flowd*, *FlowScan*, *Autofocus*, dan *Fluxoscope* dapat digunakan untuk *NetFlow* dan *sFlow processing* (Sanog & Bhutan, n.d.). Untuk solusi

yang lebih komprehensif, terdapat *Capsa Network Analyzer* yang menawarkan *real-time monitoring*, *network performance optimization*, dan *security analysis features* (Colasoft, 2017). Kunci pemilihan tool yang tepat adalah memahami bahwa setiap tool memiliki kekuatan masing-masing - beberapa *excel* dalam *real-time alerting*, beberapa dalam *deep packet inspection* (Cremer *et al.*, 2022a), sehingga pemilihan harus disesuaikan dengan kebutuhan spesifik analisis dan infrastruktur yang tersedia.

Pemilihan *NFDUMP* didasarkan pada kelebihan signifikan dibandingkan *tools* analisis *traffic flow* lainnya yang membuatnya menjadi pilihan unggul untuk analisis *NetFlow* dalam skala *enterprise*. Sistem ini dioptimalkan untuk kecepatan dan efisiensi, memungkinkan *filtering* dan agregasi kompleks dari *flow records* dengan *syntax* yang mirip *tcpdump*, serta *tools* ini dioptimalkan untuk kecepatan *filtering* yang efisien. *NFDUMP* memiliki keunggulan dalam kompatibilitas yang luas dengan mendukung berbagai versi *NetFlow* (v1, v5, v7, v9) dan *IPFIX* serta *limited set sflow*, termasuk dukungan untuk perangkat *CISCO ASA (NSEL)* dan *CISCO NAT (NEL)*, dan *fully IPv6 compatible NFDUMP*. Keunggulan arsitektur *NFDUMP* terletak pada pemisahan proses *collection* dari *processing*, dimana semua data yang dikumpulkan disimpan ke *disk* sebelum analisis, dengan data yang diorganisir dalam struktur direktori berbasis waktu. Desain ini memungkinkan analisis data *NetFlow* dari masa lalu sekaligus *tracking* pola *traffic* yang menarik secara kontinyu, dimana batasan waktu ke masa lalu hanya dibatasi oleh ruang *disk* yang tersedia (Palaniappan *et al.*, 2020a). Kombinasi kecepatan *processing*, fleksibilitas *filtering*, kompatibilitas *protocol* yang luas, dan kemampuan *historical analysis* menjadikan *NFDUMP* superior dibandingkan *tools* alternatif yang umumnya memiliki keterbatasan dalam satu atau lebih aspek tersebut.

4.5.1 Artefak Flow Records dan Analisis Behavioral

Eksekusi perintah *nfdump* terhadap artefak digital *nfcapd* menghasilkan *metadata* forensik yang komprehensif dalam bentuk *flow statistics*. Setiap *flow record* yang dihasilkan merupakan artefak digital yang mengandung informasi komunikasi jaringan, termasuk *source/destination IP*, *port*, *protocol*, *timestamp*, dan *volume data*. Artefak ini sangat *valuable* untuk forensika jaringan karena memungkinkan *reconstructed network communication patterns* tanpa perlu menyimpan *full packet capture* yang memerlukan *storage* yang jauh lebih besar.

Dari perspektif investigasi forensik, *flow records* berfungsi sebagai digital *fingerprint* dari aktivitas jaringan yang dapat digunakan untuk mengidentifikasi

communication patterns, detect data exfiltration attempts, dan menganalisis lateral movement dalam jaringan. Kombinasi *multiple flow records* dari periode berbeda (*firewall ON/OFF*) menciptakan *comparative forensic dataset* yang memungkinkan *identification of blocked vs. allowed traffic patterns*.

Output dari *nfdump* yang menampilkan metrik seperti *total flows*, *total bytes*, *total packets*, *average bps*, *average pps*, dan *average bpp* memberikan gambaran statistik lalu lintas jaringan yang dikumpulkan dari *NetFlow records*. *Total flows* menunjukkan jumlah aliran jaringan individual yang tercatat, di mana satu *flow* merepresentasikan komunikasi dari satu sumber ke satu tujuan dengan protokol tertentu dalam periode waktu tertentu. *Total bytes* dan *total packets* masing-masing menggambarkan jumlah keseluruhan data (dalam *byte*) dan jumlah paket yang dikirim selama periode tersebut. *Average bps* (*bits per second*) menghitung rata-rata kecepatan *transfer* data, *average pps* (*packets per second*) mencerminkan rata-rata jumlah paket yang ditransmisikan per detik, dan *average bpp* (*bytes per packet*) menunjukkan rata-rata ukuran tiap paket. Statistik ini sangat berguna untuk menganalisis performa jaringan, mengidentifikasi anomali, atau mendeteksi potensi serangan.

4.5.2 Interpretasi Forensik Network Flow Metrics

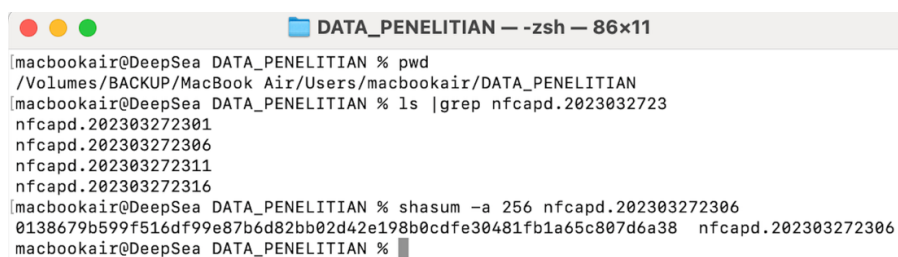
Analisis forensik terhadap artefak *network flow metrics* mengungkap beberapa indikator penting untuk investigasi keamanan jaringan1.

1. *Flow Density Analysis*: Perbandingan *total flows* antara periode *firewall ON* dan *OFF* memberikan indikator forensik tentang volume komunikasi yang diblokir. Penurunan signifikan dalam *flow count* saat *firewall* aktif mengindikasikan efektivitas *blocking mechanism* terhadap *malicious traffic*.
2. *Bandwidth Utilization Patterns*: Analisis *average bps* dan *total bytes flow* periode yang berbeda dapat mengungkap artefak digital berupa *bandwidth consumption anomalies*. *Spike* atau penurunan drastis dalam penggunaan *bandwidth* dapat mengindikasikan aktivitas *malicious* seperti *DDoS attacks* atau *data exfiltration attempts*.
3. *Packet Size Distribution*: *Average bpp* (*bytes per packet*) berfungsi sebagai *behavioral fingerprint* yang dapat mengidentifikasi jenis *traffic*. *Malicious traffic* seringkali memiliki *packet size patterns* yang berbeda dari *normal user traffic*, menjadikan metrik ini berguna untuk analisis forensik.

4. *Temporal Correlation*: Korelasi antara *flow metrics* dengan *timeline* dalam implementasi *firewall rules* memberikan bukti forensik tentang hubungan sebab-akibat antara kontrol keamanan dan perilaku jaringan. Artefak ini sangat penting untuk membuktikan efektivitas tindakan keamanan yang telah diimplementasikan.

4.5.3 *Chain of Custody* dan Integritas Artefak

Untuk memastikan validitas forensik dari artefak digital yang dikumpulkan, implementasi *proper chain of custody* menjadi sangat penting. Setiap *file nfcapd* harus di-hash menggunakan algoritma kriptografi (*MD5, SHA-256*) untuk memastikan integritas data selama proses analisis. Dokumentasi *timestamp, file size, dan checksum values* membentuk *forensic audit trail* yang dapat diverifikasi di pengadilan atau audit keamanan. Artefak digital yang dihasilkan dari penelitian ini tidak hanya memiliki nilai penelitian, tetapi juga dapat berfungsi sebagai dasar untuk investigasi forensik selanjutnya. Pola aliran jaringan yang terekam dapat dijadikan *signatures* untuk mengidentifikasi pola serangan sejenis atau untuk memvalidasi efektivitas kontrol keamanan yang diterapkan di lingkungan yang berbeda. Gambar 4.4 contoh cara melihat kode *hash* dari salah satu file data dengan nama file *nfcapd.202303272306* yang tersimpan dalam *host MacOS*. Kode *hash file* ini harus sama dengan kode *hash file* yang sama yang tersimpan dalam *server* kolektor. Validasi ini untuk memastikan data yang dianalisa dalam *MacOS* adalah data yang sama dengan data dalam *Server Kolektor*.



```
DATA_PENELITIAN — -zsh — 86x11
macbookair@DeepSea DATA_PENELITIAN % pwd
/Volumes/BACKUP/MacBook Air/Users/macbookair/DATA_PENELITIAN
macbookair@DeepSea DATA_PENELITIAN % ls |grep nfcapd.2023032723
nfcapd.202303272301
nfcapd.202303272306
nfcapd.202303272311
nfcapd.202303272316
macbookair@DeepSea DATA_PENELITIAN % shasum -a 256 nfcapd.202303272306
0138679b599f516df99e87b6d82bb02d42e198b0cdf30481fb1a65c807d6a38  nfcapd.202303272306
macbookair@DeepSea DATA_PENELITIAN %
```

Gambar 4.4 Pengecekan kode *hash file* data

Tabel 4.4 Menunjukkan analisis *traffic* jaringan selama periode 9 hari (14-22 Januari 2023) ketika *firewall filter* dalam kondisi tidak aktif, dengan rata-rata harian mencapai sekitar 61,4 juta *flows*, volume data berkisar antara 643,9-955,2 GB, dan jumlah paket antara 0,761-1,1 miliar paket per hari. Data menunjukkan fluktuasi *traffic* yang cukup signifikan,

dimana puncak tertinggi terjadi pada tanggal 16 Januari dengan 955,2 GB data dan *bandwidth* rata-rata 86,7 Mbps, sementara *traffic* terendah tercatat pada 22 Januari dengan hanya 643,9 GB. Konsistensi ukuran paket rata-rata (*avg bpp*) yang stabil di kisaran 841-855 *bytes* menunjukkan karakteristik *traffic* yang relatif homogen, namun variasi pada parameter lain seperti *flows* dan *bandwidth* mengindikasikan adanya pola aktivitas jaringan yang berubah-ubah selama periode observasi tanpa adanya *filtering* terhadap *domain* berbahaya.

Tabel 4. 4 Data Harian Periode I Selama 9 Hari - *FILTER OFF*

No	PERIODE I - FILTER OFF 11 Jan - 25 Jan '23	DAILY Total Flows	DAILY Total Bytes (G)	DAILY Total Packets (G)	DAILY Avg bps (M)	DAILY Avg pps	DAILY Avg bpp
1	14-Jan	60716624	859,000	1,000	78,000	11426,000	853,000
2	15-Jan	57732688	683,900	0,803	62,100	9114,000	851,000
3	16-Jan	67616336	955,200	1,100	86,700	12762,000	846,000
4	17-Jan	66751328	797,100	0,948	72,300	10747,000	841,000
5	18-Jan	60318768	838,800	0,986	55,900	8213,000	850,000
6	19-Jan	63935200	939,600	1,100	85,300	12670,000	841,000
7	20-Jan	61572992	862,400	1,000	78,200	11431,000	855,000
8	21-Jan	64138704	872,000	1,000	78,900	11687,000	844,000
9	22-Jan	55982864	643,900	0,761	58,500	8635,000	846,000

Tabel 4.5 Menampilkan analisis *traffic* jaringan selama periode 9 hari (3-11 Februari 2023) ketika *firewall filter* dalam kondisi aktif, dengan rata-rata harian mencapai sekitar 75,9 juta *flows*, volume data berkisar antara 707,9-890,7 GB, dan jumlah paket antara 1,1-1,3 miliar paket per hari. Dibandingkan dengan periode tanpa *filter*, terlihat peningkatan signifikan pada jumlah *flows* (naik sekitar 23,6%) dan jumlah paket (naik sekitar 20%), namun dengan penurunan drastis pada ukuran paket rata-rata (*avg bpp*) menjadi 647-736 *bytes* dari sebelumnya 841-855 *bytes*, yang mengindikasikan adanya perubahan karakteristik *traffic* akibat implementasi *filtering*. Data menunjukkan bahwa meskipun *filter* aktif memblokir akses ke *domain* berbahaya, aktivitas jaringan justru meningkat dalam hal volume transaksi dengan paket-paket yang berukuran lebih kecil, kemungkinan karena adanya *retry attempts*, fragmentasi paket, atau peningkatan aktivitas *legitimate traffic* yang sebelumnya terganggu oleh *traffic malicious*.

Tabel 4. 5 Data Harian Periode II Selama 9 Hari - *FILTER ON*

	PERIODE II - FILTER ON 3 Feb - 12 Feb '23	DAILY Total Flows	DAILY Total Bytes (G)	DAILY Total Packets (G)	DAILY Avg bps (M)	DAILY Avg pps	DAILY Avg bpp
--	---	----------------------	--------------------------	----------------------------	----------------------	------------------	------------------

10	03-Feb	74730912	773,500	1,200	70,200	13556,000	647,000
11	04-Feb	80766400	890,700	1,200	80,800	13715,000	736,000
12	05-Feb	76764672	763,800	1,100	69,400	12142,000	714,000
13	06-Feb	82242384	860,900	1,200	78,100	13639,000	716,000
14	07-Feb	77254992	707,900	1,100	64,200	12135,000	661,000
15	08-Feb	79376176	837,200	1,200	75,900	13695,000	693,000
16	09-Feb	79823323	876,100	1,300	79,500	14438,000	688,000
17	10-Feb	72858844	843,500	1,200	76,900	14046,000	684,000
18	11-Feb	59037903	855,700	1,300	77,500	14451,000	670,000

Tabel 4.6 Menunjukkan analisis *traffic* jaringan selama periode 9 hari (23 Februari - 3 Maret 2023) ketika *firewall* filter kembali dinonaktifkan, dengan rata-rata harian mencapai sekitar 59,8 juta *flows*, volume data berkisar antara 568,6-962,4 GB, dan jumlah paket antara 1,2-1,5 miliar paket per hari. Data menunjukkan tren penurunan yang konsisten dari awal hingga akhir periode, dimana *traffic* tertinggi terjadi pada 28 Februari dengan 962,4 GB dan terendah pada 3 Maret dengan hanya 568,6 GB, sementara ukuran paket rata-rata (*avg bpp*) juga mengalami penurunan drastis dari 655 *bytes* menjadi 491 *bytes*. Dibandingkan dengan periode sebelumnya ketika *filter* aktif, terlihat bahwa meskipun jumlah *flows* menurun sekitar 21%, jumlah paket per hari justru meningkat signifikan (rata-rata 1,36 miliar vs 1,18 miliar), namun dengan ukuran paket yang semakin mengecil, mengindikasikan kemungkinan adanya fragmentasi *traffic* yang lebih intensif atau peningkatan aktivitas *malicious* yang menggunakan paket-paket kecil untuk menghindari deteksi.

Tabel 4. 6 Data Harian Periode III Selama 9 Hari - *FILTER OFF*

	PERIODE III - <i>FILTER OFF</i> 22 Feb - 6 Mar '23	DAILY Total <i>Flows</i>	DAILY Total <i>Bytes (G)</i>	DAILY Total <i>Packets (G)</i>	DAILY Avg <i>bps (M)</i>	DAILY Avg pps	DAILY Avg bpp
19	23-Feb	60277152	933,000	1,400	84,600	16148,000	655,000
20	24-Feb	58886320	904,100	1,500	82,000	16602,000	617,000
21	25-Feb	57878688	900,100	1,500	81,700	16671,000	612,000
22	26-Feb	58573056	793,200	1,300	71,900	15167,000	592,000
23	27-Feb	60715168	882,200	1,400	80,000	15403,000	649,000
24	28-Feb	64275680	962,400	1,400	87,300	16391,000	666,000
25	01-Mar	65456144	906,000	1,400	82,200	15580,000	659,000
26	02-Mar	57725264	754,200	1,300	68,400	14753,000	579,000
27	03-Mar	54834154	568,600	1,200	51,600	13113,000	491,000

Tabel 4.7. Menampilkan analisis *traffic* jaringan selama periode 9 hari (6-16 Maret 2023) ketika filter keamanan kembali diaktifkan, dengan rata-rata harian mencapai sekitar 49,2 juta *flows*, volume data berkisar antara 428,3-601,7 GB, dan jumlah paket antara 0,784-

1,1 miliar paket per hari. Dibandingkan dengan periode sebelumnya tanpa *filter*, terlihat penurunan signifikan pada semua parameter *traffic* dengan *flows* turun sekitar 17,7%, volume data turun rata-rata 35%, dan jumlah paket turun sekitar 30%, sementara ukuran paket rata-rata (*avg bpp*) tetap dalam rentang rendah 532-590 *bytes* yang konsisten dengan pola fragmentasi. Data menunjukkan efektivitas *filter* dalam mengurangi *overall traffic* volume secara substansial, dimana *bandwidth* rata-rata turun dari kisaran 51,6-87,3 Mbps menjadi 38,9-54,6 Mbps, mengindikasikan bahwa implementasi *filtering* berhasil memblokir sebagian besar *traffic* berbahaya dan menormalkan pola komunikasi jaringan menjadi lebih efisien dan aman.

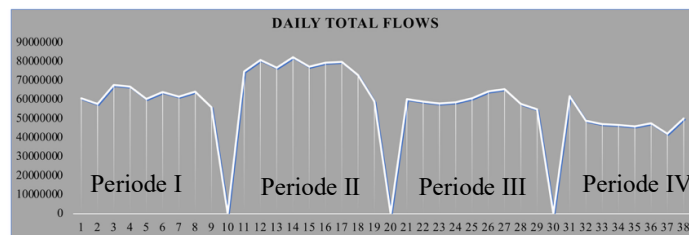
Tabel 4. 7 Data Harian Periode IV Selama 9 Hari - *FILTER ON*

	PERIODE IV - FILTER ON 6 Mar - 17 Mar '23	DAILY Total Flows	DAILY Total Bytes (G)	DAILY Total Packets (G)	DAILY Avg bps (M)	DAILY Avg pps	DAILY Avg bpp
28	06-Mar	61738320	587,000	1,100	53,300	12526,000	532,000
29	09-Mar	48862832	574,100	1,000	52,100	11842,000	550,000
30	10-Mar	47089168	567,700	0,980	51,600	11132,000	579,000
31	11-Mar	46614128	601,700	1,000	54,600	11555,000	590,000
32	12-Mar	45930624	481,700	0,859	43,700	9746,000	560,000
33	13-Mar	47556448	486,800	0,902	44,200	10228,000	539,000
34	14-Mar	42094480	453,400	0,784	41,100	8889,000	578,000
35	15-Mar	50017408	428,300	0,799	38,900	9067,000	536,000
36	16-Mar	52654560	503,300	0,931	45,700	10562,000	540,000

Tabel di atas menunjukkan adanya fluktuasi pada *Total Flows*, *Total Bytes*, dan parameter lainnya berdasarkan kondisi *firewall filter*. Analisis lebih lanjut dilakukan untuk melihat kecenderungan perubahan antara *firewall* dalam kondisi *OFF* dan *ON*, yang disajikan dalam visualisasi grafis berikut ini.

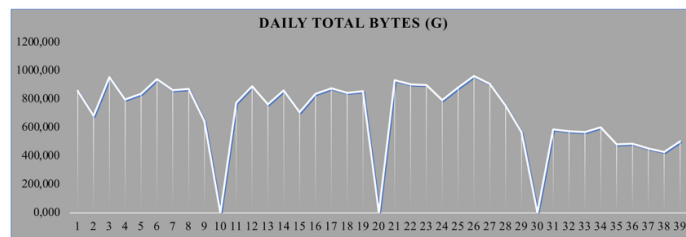
Gambar 4.5 Menunjukkan grafik "*DAILY TOTAL FLOWS*" yang memvisualisasikan volume lalu lintas data selama 39 periode. Terlihat pola yang jelas di mana terdapat empat periode utama. Penurunan drastis ke nilai 0 pada titik 10, 20, dan 30 merupakan *cut off* data. Periode 1 (titik 1-9) dan periode 3, titik 21-29 menunjukkan kondisi *firewall filter OFF* dengan jumlah *flows* sekitar 55-65 juta. Sementara pada periode 2, titik 11-19 dan periode 4, titik 31-38 saat *firewall filter ON*, terlihat perbedaan signifikan - periode 2 mencapai puncak tertinggi dengan *flows* mencapai 75-80 juta, sedangkan periode 4 justru memiliki nilai terendah sekitar 45-50 juta. Analisis perbandingan menunjukkan bahwa ketika *firewall* diaktifkan pada periode 2 terjadi peningkatan *flows* sekitar 25% dibandingkan periode 1,

kemungkinan karena *firewall* menghasilkan lebih banyak *flow connection tracking*. Namun pada perbandingan periode 3 dan 4, terjadi fenomena berbeda di mana pengaktifan *firewall* justru menurunkan total *flows* sekitar 20%. Secara global, grafik menunjukkan tren penurunan *flows* dari waktu ke waktu, dengan pola yang menunjukkan *firewall* memiliki dampak berbeda pada kondisi jaringan yang berbeda atau *traffic patterns* (Vake et al., 2025), menandakan adanya optimasi konfigurasi atau perubahan dalam karakteristik *traffic* yang mempengaruhi perilaku *firewall*.



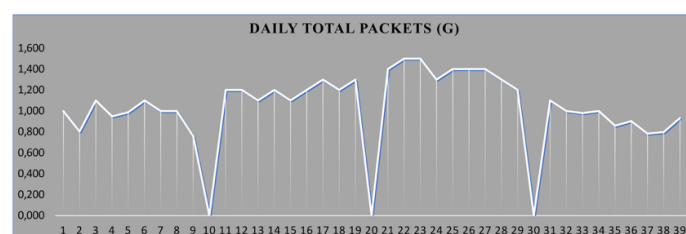
Gambar 4.5 Grafik *daily total flows*.

Gambar 4.6 Menampilkan volume data dalam *gigabyte* selama 38 periode dengan pola pembagian yang serupa dengan grafik sebelumnya. Penurunan tajam ke titik 0 pada periode 10, 20, dan 30 merupakan *cut off* data. Periode 1, titik 1-9 dan periode 3, titik 21-29 menunjukkan kondisi *firewall* filter *OFF* dengan volume data berkisar antara 800,000 hingga 950,000 GB. Sementara periode 2, titik 11-19 dan periode 4, titik 31-39 saat *firewall* filter *ON* menunjukkan perbedaan signifikan, periode 2 relatif stabil sekitar 800,000 GB, sedangkan periode 4 mengalami penurunan drastis hingga 400,000-600,000 GB. Perbandingan periode 1 dengan 2 menunjukkan bahwa pengaktifan *firewall* menyebabkan sedikit penurunan total *bytes* sekitar 5-10%, berbeda dengan efeknya terhadap *flows* yang justru meningkat. Perbandingan periode 3 dengan 4 memperlihatkan dampak dramatis di mana pengaktifan *firewall* menurunkan volume data hingga 40%. Secara global, grafik menunjukkan bahwa *firewall* semakin efektif dalam mengendalikan volume data seiring waktu, dengan pengaruh terbesar terlihat pada periode terakhir. Hal ini mengindikasikan bahwa meskipun jumlah *flows* pada grafik pertama bervariasi, efisiensi *firewall* dalam memfilter *traffic* berdasarkan volume data meningkat signifikan, kemungkinan akibat optimasi *rules filtering*, penyesuaian terhadap *traffic patterns*, atau perubahan kebijakan keamanan yang mengutamakan pembatasan volume data dibandingkan jumlah koneksi.



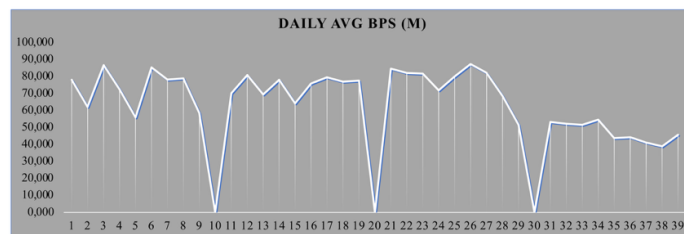
Gambar 4.6 Grafik *daily total bytes*.

Gambar 4.7 Menampilkan volume paket dalam gigapaket selama 39 periode dengan pembagian yang konsisten dengan grafik-grafik sebelumnya. Penurunan ke titik 0 pada periode 10, 20, dan 30 merupakan *cut off* data. Periode 1, titik 1-9 dan periode 3, titik 21-29 menunjukkan kondisi *firewall* filter *OFF* mati dengan perbedaan signifikan, periode 1 memiliki volume sekitar 0,8-1,1 G paket, sedangkan periode 3 menunjukkan peningkatan substansial mencapai 1,2-1,5 G paket. Sementara itu, periode, 2 titik 11-19 dan periode 4, titik 31-39 saat *firewall filter ON* juga menunjukkan perbedaan yang jelas, periode 2 sekitar 1,1-1,3 G paket, sedangkan periode 4 turun ke 0,8-1,0 G paket. Perbandingan periode 1 dengan 2 menunjukkan bahwa pengaktifan *firewall* meningkatkan jumlah paket sekitar 20%. Analisis periode 3 dengan 4 memperlihatkan penurunan drastis sekitar 30-40% jumlah paket saat *firewall* diaktifkan, konsisten dengan penurunan pada volume data. Secara global, grafik menunjukkan pola menarik di mana terjadi peningkatan jumlah paket yang signifikan pada periode 3 (*firewall OFF*) dibandingkan periode lainnya, dan kemudian penurunan tajam di periode 4 (*firewall ON*). Ini mengindikasikan evolusi efektivitas *firewall* dalam mengendalikan *traffic* jaringan, di mana pada implementasi terakhir *firewall* berhasil mengurangi tidak hanya volume data tetapi juga jumlah paket secara signifikan, menunjukkan kemungkinan adanya perbaikan algoritma *filtering*, aturan yang lebih spesifik untuk *traffic* kontrolnya, atau respons adaptif terhadap perubahan karakteristik serangan *traffic* yang mencoba melewati *firewall*.



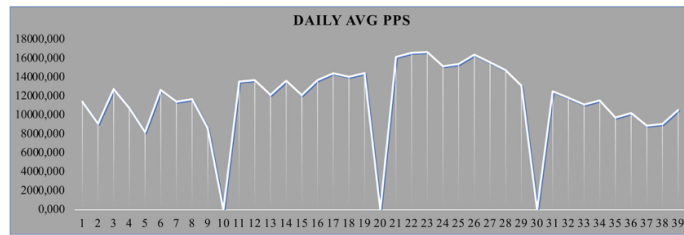
Gambar 4.7 Grafik *daily total packets*.

Gambar 4.8 Menunjukkan *traffic* jaringan selama 39 hari, terlihat pola yang signifikan terkait dengan status *firewall* filter. Pada periode 1 dan 3 ketika *firewall* filter dalam kondisi *OFF*, *traffic bandwidth* menunjukkan nilai yang relatif tinggi dan berfluktuasi, dengan beberapa puncak mencapai sekitar 80.000-90.000 *BPS*, menandakan bahwa semua jenis *traffic* dapat melewati jaringan tanpa pembatasan atau *filtering*. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* dalam kondisi *ON*, terjadi penurunan drastis dalam *traffic bandwidth* hingga mendekati nilai 10.000-20.000 *BPS*, menunjukkan bahwa *firewall* secara efektif memblokir atau membatasi *traffic* tertentu sesuai dengan *rule* yang dikonfigurasi. Pola ini mengindikasikan bahwa *firewall filter* berfungsi dengan baik dalam mengontrol dan mengurangi beban *traffic* jaringan.



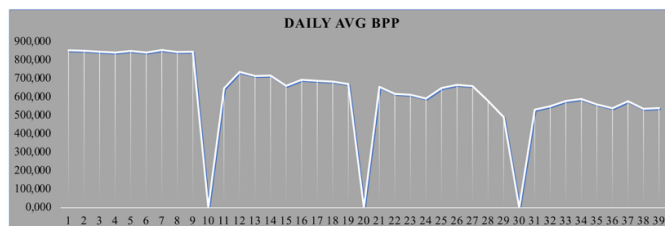
Gambar 4.8 Grafik *daily average BPS*.

Gambar 4.9 Mencakup pengamatan selama 39 hari, terlihat korelasi yang kuat antara status *firewall filter* dengan volume paket yang diproses dalam jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, jumlah paket per detik menunjukkan nilai yang sangat tinggi dengan puncak mencapai 16.000-18.000 *PPS*, mengindikasikan bahwa seluruh paket data dapat melewati jaringan tanpa adanya proses *filtering* atau inspeksi yang menghambat aliran *traffic*. Namun pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), terjadi penurunan signifikan dalam jumlah *PPS* hingga berada pada kisaran 2.000-4.000 *PPS*, menunjukkan bahwa *firewall* secara efektif melakukan proses inspeksi dan *filtering* paket, sehingga hanya paket yang memenuhi kriteria keamanan yang diizinkan untuk lewat. Pola ini mengkonfirmasi efektivitas *firewall* dalam mengendalikan *traffic* berbasis jumlah paket, dimana aktivasi *filter* tidak hanya mengurangi *bandwidth* tetapi juga secara drastis menurunkan volume paket yang diproses, yang dapat mengindikasikan keberhasilan dalam memblokir *traffic* anomali atau serangan yang biasanya ditandai dengan volume paket yang tinggi.



Gambar 4.9 Grafik *daily average PPS*.

Pada Gambar 4.10 Transisi dari Periode 1 ke Periode 2 (*firewall OFF* ke *ON*), terjadi penurunan signifikan nilai *BPP* dari level tinggi sekitar 800.000-900.000 *bytes per packet* menjadi level yang lebih rendah dan stabil di kisaran 600.000-700.000 *bytes per packet*. Pola serupa namun dengan magnitude yang lebih kecil terjadi pada transisi dari Periode 3 ke Periode 4, dimana aktivasi *firewall* kembali menyebabkan penurunan *BPP* dari sekitar 600.000-700.000 *bytes per packet* menjadi 500.000-600.000 *bytes per packet*. Konsistensi penurunan ini mengindikasikan bahwa *firewall filter* memiliki dampak langsung dan terukur dalam mengurangi ukuran rata-rata paket data yang melewati jaringan, kemungkinan karena *firewall* memfilter atau memblokir *traffic* dengan *payload* besar, melakukan *deep packet inspection* yang mempengaruhi fragmentasi paket, atau membatasi jenis aplikasi tertentu yang menggunakan paket berukuran besar, sehingga secara keseluruhan menghasilkan efisiensi transmisi data dengan ukuran paket yang lebih optimal ketika *firewall* dalam kondisi aktif.



Gambar 4.10 Grafik *daily average BPP*.

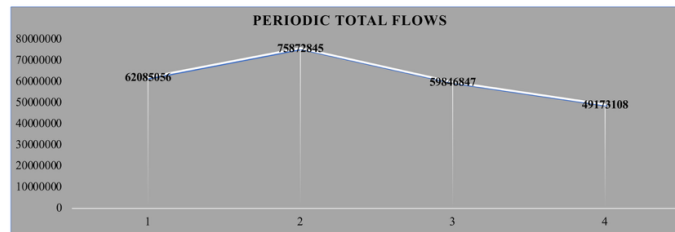
Tabel 4.11 Menyajikan ringkasan komparatif *traffic* jaringan dalam empat periode pengamatan yang menunjukkan dampak implementasi *filter* keamanan terhadap karakteristik *traffic* secara keseluruhan. Data memperlihatkan pola yang konsisten dimana aktivasi *filter* cenderung meningkatkan jumlah *flows* (dari rata-rata 62,1 juta menjadi 75,9 juta pada periode awal, namun menurun drastis menjadi 49,2 juta pada periode akhir) sementara secara signifikan mengurangi ukuran paket rata-rata dari 847 *bytes* tanpa *filter*

menjadi 690-556 *bytes* dengan *filter* aktif. Menariknya, periode tanpa *filter* menunjukkan evolusi *traffic* yang mengkhawatirkan dimana jumlah paket meningkat dari 0,967 miliar menjadi 1,378 miliar per hari dengan ukuran paket yang semakin mengecil (dari 847 menjadi 613 *bytes*), mengindikasikan kemungkinan peningkatan aktivitas *malicious* atau fragmentasi *traffic*, sementara implementasi *filter* pada periode terakhir berhasil mengurangi semua parameter *traffic* secara substansial, menunjukkan efektivitas sistem *filtering* dalam menormalkan dan mengamankan komunikasi jaringan.

Tabel 4. 8 Data Periodik - *FILTER OFF* dan *ON*

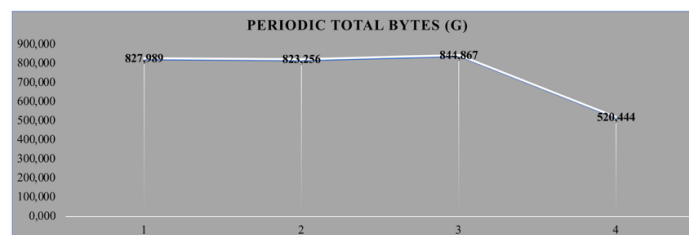
	PERIODIC Total Flows	PERIODIC Total Bytes (G)	PERIODIC Total Packets (G)	PERIODIC Avg bps (M)	PERIODIC Avg pps	PERIODIC vg bpp
<i>FILTER OFF</i>	62085056	827,989	0,967	72,878	10742,778	847,444
<i>FILTER ON</i>	75872845	823,256	1,200	74,722	13535,222	689,889
<i>FILTER OFF</i>	59846847	844,867	1,378	76,633	15536,444	613,333
<i>FILTER ON</i>	49173108	520,444	0,928	47,244	10616,333	556,000

Gambar 4.10 Berdasarkan grafik *Periodic Total Flows* yang menampilkan data agregat untuk 4 periode, terlihat dampak yang signifikan dari pengaturan *firewall filter* terhadap total aliran data dalam jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, total *flows* mencapai nilai yang sangat tinggi yaitu 62.085.056 dan 59.846.847 *flows respectively*, menunjukkan volume aktivitas jaringan yang masif dimana seluruh jenis koneksi dan sesi komunikasi dapat terbentuk tanpa pembatasan, termasuk kemungkinan adanya *traffic* anomali, serangan, atau aktivitas scanning yang menghasilkan banyak *flow connections*. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), terjadi penurunan drastis dalam *total flows* menjadi 75.872.845 pada periode 2 dan turun tajam hingga 49.173.108 pada periode 4, mengindikasikan kemampuan *firewall* untuk secara efektif memblokir pembentukan koneksi yang tidak diinginkan dan mengendalikan jumlah sesi aktif dalam jaringan. Penurunan bertahap dari periode 2 ke periode 4 saat *firewall* tetap aktif menunjukkan adanya *learning curve* atau penyesuaian konfigurasi yang semakin ketat, dimana *firewall* semakin efisien dalam mengidentifikasi dan memblokir *pattern traffic* yang mencurigakan, sehingga menghasilkan lingkungan jaringan yang lebih aman dengan jumlah *flow connections* yang terkontrol dan *legitimate*.



Gambar 4.11 Grafik fluktuasi *periodic total flows*.

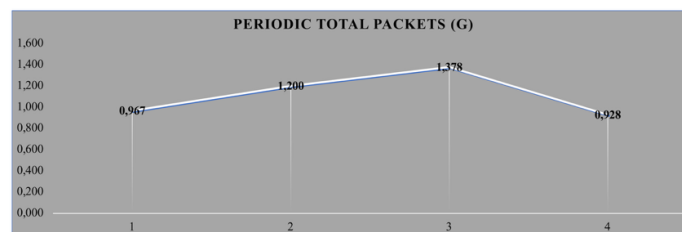
Gambar 4.12 Menunjukkan konsumsi *bandwidth* dalam satuan *Gigabytes* untuk 4 periode, terlihat pola yang konsisten dengan efektivitas *firewall filter* dalam mengendalikan volume data yang melewati jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, total *bytes* mencapai nilai yang tinggi yaitu 827,989 GB dan 844,867 GB, mengindikasikan bahwa semua jenis *traffic* data dapat mengalir bebas melalui jaringan tanpa pembatasan, termasuk kemungkinan adanya *transfer* data yang tidak perlu, aktivitas *malicious*, atau *bandwidth abuse* yang mengkonsumsi kapasitas jaringan secara berlebihan. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), meskipun periode 2 masih menunjukkan nilai yang relatif tinggi (823,256 GB), terjadi penurunan signifikan pada periode 4 hingga mencapai 520,444 GB, menunjukkan bahwa *firewall* secara bertahap menjadi lebih efektif dalam memblokir *traffic* yang tidak diinginkan dan mengoptimalkan penggunaan *bandwidth*. Penurunan drastis pada periode 4 mengindikasikan bahwa *firewall filter* tidak hanya berhasil mengurangi jumlah koneksi dan paket, tetapi juga secara substansial mengurangi konsumsi *bandwidth* keseluruhan, sehingga menciptakan efisiensi jaringan yang signifikan dengan tetap mempertahankan *legitimate traffic* untuk operasional bisnis yang normal.



Gambar 4.12 Grafik fluktuasi *periodic total bytes*.

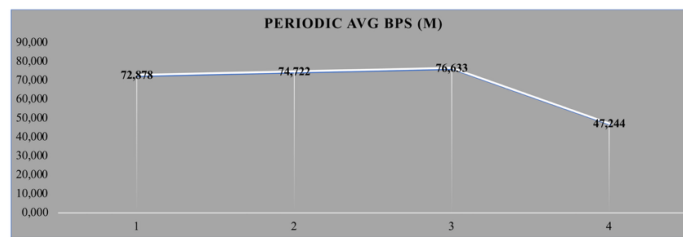
Gambar 4.13 Menampilkan jumlah paket dalam satuan *Gigapackets* untuk 4 periode, terlihat pola yang menarik terkait dengan dampak *firewall filter* terhadap volume paket yang diproses dalam jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, total *packets* menunjukkan nilai 0,967 dan 1,378 *Gigapackets*, dimana periode 3 mencapai puncak tertinggi, mengindikasikan adanya lonjakan aktivitas jaringan yang signifikan

dengan volume paket yang sangat besar, kemungkinan termasuk *traffic anomaly* atau aktivitas lain yang menghasilkan banyak paket berukuran kecil. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), meskipun periode 2 menunjukkan peningkatan menjadi 1,200 *Gigapackets* (mungkin karena initial *spike* saat *firewall* mulai aktif), periode 4 menunjukkan penurunan drastis hingga 0,928 *Gigapackets*, yang merupakan nilai terendah di seluruh periode pengamatan. Hal ini mengkonfirmasi efektivitas *firewall filter* dalam mengendalikan volume paket, dimana setelah periode adaptasi dan *fine-tuning*, *firewall* berhasil secara konsisten memblokir *traffic* berbasis paket yang tidak diinginkan, sehingga menghasilkan lingkungan jaringan yang lebih stabil dengan volume paket yang terkontrol dan didominasi oleh *legitimate traffic*.



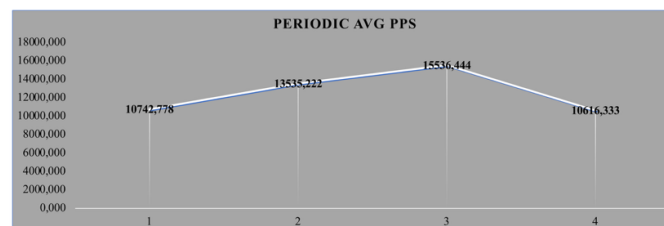
Gambar 4.13 Grafik fluktuasi *periodic total packets*.

Gambar 4.14 Menunjukkan rata-rata *bandwidth* dalam satuan *Mega BPS* untuk 4 periode, terlihat pola yang konsisten dengan efektivitas *firewall filter* dalam mengoptimalkan penggunaan *bandwidth* jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, rata-rata *BPS* menunjukkan nilai yang tinggi yaitu 72,878 *MBPS* dan 76,633 *MBPS*, dengan periode 3 mencapai puncak tertinggi, mengindikasikan bahwa seluruh *traffic* dapat mengkonsumsi *bandwidth* secara bebas tanpa pembatasan, termasuk kemungkinan adanya *bandwidth abuse*, transfer data yang tidak efisien, atau aktivitas *malicious* yang menghabiskan kapasitas jaringan secara berlebihan. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), meskipun periode 2 masih menunjukkan nilai yang relatif tinggi (74,722 *MBPS*), terjadi penurunan drastis pada periode 4 hingga mencapai 47,244 *MBPS*, yang merupakan sekitar 38% lebih rendah dibandingkan periode sebelumnya. Penurunan signifikan ini menunjukkan bahwa *firewall filter* tidak hanya berhasil mengurangi jumlah koneksi dan paket, tetapi juga secara efektif mengoptimalkan penggunaan *bandwidth* dengan memblokir *traffic* yang tidak produktif, sehingga menciptakan efisiensi jaringan yang substansial sambil tetap mempertahankan kualitas layanan untuk aplikasi bisnis yang *legitimate*.



Gambar 4.14 Grafik fluktuasi *periodic total average BPS*.

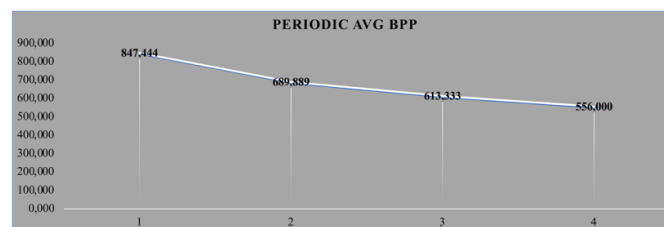
Gambar 4.15 Menampilkan rata-rata paket per detik untuk 4 periode, terlihat pola yang sangat jelas terkait dengan dampak *firewall filter* terhadap volume paket yang diproses dalam jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, rata-rata *PPS* menunjukkan nilai 10.742,778 dan mencapai puncak tertinggi pada periode 3 dengan 15.536,444 *PPS*, mengindikasikan adanya lonjakan aktivitas jaringan yang signifikan dengan volume paket yang sangat tinggi, kemungkinan mencakup *traffic* anomali, serangan berbasis *flooding*, atau aktivitas *scanning* yang menghasilkan *burst* paket dalam jumlah besar. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (*ON*), meskipun periode 2 menunjukkan peningkatan menjadi 13.535,222 *PPS* (mungkin karena initial *response* atau adaptasi sistem), periode 4 menunjukkan penurunan drastis hingga 10.616,333 *PPS*, yang hampir setara dengan periode 1 namun dalam kondisi terkontrol. Hal ini mengkonfirmasi kemampuan *firewall filter* dalam mengendalikan *rate* paket secara efektif, dimana setelah melalui fase pembelajaran dan optimasi, *firewall* berhasil menstabilkan volume *PPS* pada level yang aman dan terkendali, sehingga melindungi jaringan dari serangan berbasis volume paket tinggi sambil tetap mempertahankan *throughput* yang memadai untuk operasional normal.



Gambar 4.15 Grafik fluktuasi *periodic total average PPS*.

Gambar 4.16 Menunjukkan rata-rata *bytes per packet* untuk 4 periode, terlihat pola penurunan yang konsisten dan menarik terkait dengan dampak *firewall filter* terhadap karakteristik ukuran paket dalam jaringan. Pada periode 1 dan 3 ketika *firewall filter* dalam kondisi *OFF*, rata-rata *BPP* menunjukkan nilai yang tinggi yaitu 847,444 *bytes* dan 613,333

bytes per paket, dengan periode 1 mencapai nilai tertinggi, mengindikasikan bahwa *traffic* yang melewati jaringan didominasi oleh paket-paket berukuran besar yang mungkin termasuk *legitimate data transfer*, *file sharing*, atau aplikasi yang memerlukan *payload* besar. Sebaliknya, pada periode 2 dan 4 ketika *firewall filter* diaktifkan (ON), terjadi penurunan progresif dalam ukuran rata-rata paket menjadi 689,889 bytes pada periode 2 dan turun lebih jauh hingga 556,000 bytes pada periode 4. Pola penurunan BPP yang berkelanjutan ini menunjukkan bahwa *firewall filter* secara selektif cenderung memblokir atau membatasi paket-paket berukuran besar yang mungkin dianggap mencurigakan atau tidak sesuai dengan *policy* keamanan, sehingga menghasilkan *traffic* yang didominasi oleh paket berukuran lebih kecil namun lebih aman dan terkontrol, menciptakan profil jaringan yang lebih efisien dan *secure*.



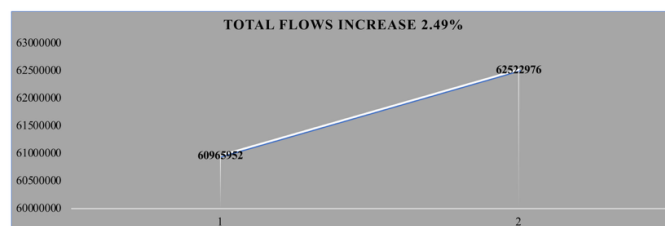
Gambar 4.16 Grafik fluktuasi *periodic average BPP*.

Tabel 4.9 Analisis komparatif keseluruhan menunjukkan bahwa implementasi *filter* keamanan menghasilkan dampak yang paradoks namun menguntungkan terhadap karakteristik *traffic* jaringan. Meskipun terjadi peningkatan kecil pada jumlah *flows* sebesar 2,49% (dari 60,97 juta menjadi 62,52 juta), *filter* berhasil menurunkan secara signifikan volume data sebesar 24,50%, jumlah paket sebesar 10,15%, *bandwidth* rata-rata sebesar 22,58%, dan ukuran paket rata-rata sebesar 17,25%. Penurunan substansial pada volume bytes dan *bandwidth* mengindikasikan bahwa *filter* efektif memblokir *traffic* berbahaya yang cenderung mengonsumsi *bandwidth* besar, sementara sedikit peningkatan pada *flows* dengan penurunan ukuran paket menunjukkan adanya fragmentasi atau *retry attempts* dari koneksi yang diblokir. Secara keseluruhan, data ini membuktikan efektivitas sistem *filtering* dalam mengoptimalkan penggunaan *bandwidth* dan mengurangi beban jaringan sambil tetap mempertahankan konektivitas *legitimate*, dengan *trade-off* berupa sedikit peningkatan *overhead* administratif dalam bentuk *flows* tambahan.

Tabel 4. 9 Total Data – Persentase Kenaikan/Penurunan Saat FILTER OFF dan ON

	TOTAL Flows	TOTAL Bytes (G)	TOTAL Packets (G)	TOTAL Avg bps (M)	TOTAL Avg pps	TOTAL Avg bps
<i>FILTER OFF</i>	60965952	836,428	1,172	74,756	13139,611	730,389
<i>FILTER ON</i>	62522976	671,850	1,064	60,983	12075,778	622,944
INCREASE/DECREASE In PERCENT	2,49%	-24,50%	-10,15%	-22,58%	-8,81%	-17,25%

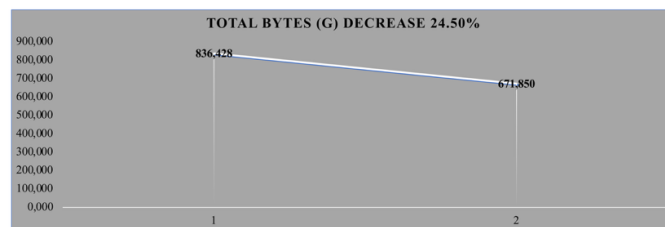
Gambar 4.17 Menunjukkan perbandingan sebelum dan sesudah implementasi *firewall filter*, terlihat hasil yang mengejutkan dimana terjadi peningkatan *total flows* sebesar 2,49% dari 60.965.952 menjadi 62.522.976 *flows*. Peningkatan ini tampak kontradiktif dengan ekspektasi umum bahwa *firewall* seharusnya mengurangi jumlah koneksi, namun fenomena ini dapat dijelaskan melalui beberapa faktor teknis yang kompleks. Pertama, aktivasi *firewall filter* mungkin memicu mekanisme *retry* dan *reconnection* dari aplikasi dan sistem yang mengalami *blocking*, sehingga menghasilkan lebih banyak *attempt connections* dalam upaya untuk mempertahankan konektivitas. Kedua, *firewall* yang dikonfigurasi dengan *rule* yang terlalu permisif atau tidak optimal dapat menyebabkan fragmentasi *traffic* atau pembentukan *multiple short-lived connections* sebagai pengganti *single long-lived connections* yang sebelumnya diblokir. Ketiga, implementasi *firewall* mungkin mengaktifkan *logging* dan monitoring *flows* yang sebelumnya tidak terhitung, sehingga memberikan visibilitas yang lebih akurat terhadap *actual network activity*. Meskipun terjadi peningkatan jumlah *flows*, hal ini tidak selalu negatif karena *firewall* masih dapat memberikan *benefit* dalam hal *filtering malicious content*, kontrol *bandwidth*, dan peningkatan *security posture*, namun memerlukan *fine-tuning* konfigurasi untuk mengoptimalkan rasio antara *security* dan *network efficiency*.



Gambar 4.17 Kenaikan *total flows* 2.49%.

Gambar 4.18 Menunjukkan perbandingan konsumsi *bandwidth* sebelum dan sesudah implementasi *firewall filter*, terlihat penurunan yang sangat signifikan sebesar 24,50% dari 836,428 GB menjadi 671,850 GB, mengindikasikan keberhasilan *firewall* dalam mengoptimalkan penggunaan *bandwidth* jaringan secara substansial. Penurunan hampir

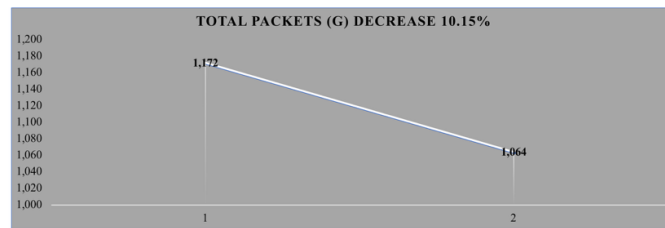
seperempat dari total konsumsi *bytes* ini menunjukkan bahwa *firewall filter* secara efektif memblokir *traffic* yang tidak produktif, tidak diinginkan, atau berpotensi berbahaya yang sebelumnya mengkonsumsi *bandwidth* secara berlebihan. Hal ini dapat mencakup *blocking* terhadap aktivitas yang menghabiskan *bandwidth* dalam jumlah besar. Pengurangan 164,578 GB ini tidak hanya memberikan penghematan biaya operasional yang signifikan, tetapi juga meningkatkan kualitas layanan untuk aplikasi bisnis yang *legitimate* dengan menyediakan lebih banyak *bandwidth* yang tersedia untuk *traffic* produktif. Hasil ini mengkonfirmasi bahwa meskipun terjadi peningkatan jumlah *flows* sebesar 2,49%, *firewall* berhasil mengurangi volume data yang ditransfer secara drastis, menunjukkan bahwa sistem lebih efisien dalam memproses banyak koneksi kecil yang *legitimate* dibandingkan dengan mengizinkan sedikit koneksi besar yang berpotensi tidak produktif atau berbahaya.



Gambar 4.18 Penurunan *total bytes* 24.50%.

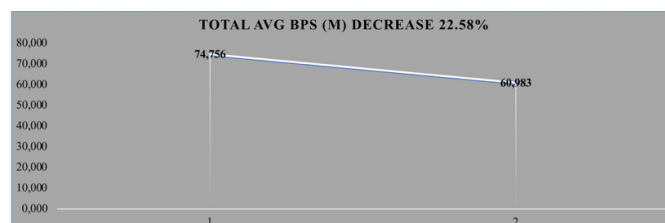
Gambar 4.19 Menunjukkan perbandingan jumlah paket sebelum dan sesudah implementasi *firewall filter*, terlihat hasil yang positif dimana terjadi penurunan sebesar 10,15% dari 1,172 *Gigapackets* menjadi 1,064 *Gigapackets*. Penurunan ini mengindikasikan bahwa *firewall filter* berhasil memblokir sejumlah signifikan paket yang tidak diinginkan, kemungkinan termasuk *traffic* dari serangan berbasis volume tinggi seperti *DDoS attacks*, *network scanning*, *port probing*, atau *spam packets* yang biasanya ditandai dengan *burst* paket dalam jumlah besar namun berukuran kecil. Hasil ini menunjukkan efektivitas *firewall* dalam melakukan *packet-level filtering*, dimana meskipun *total flows* mengalami peningkatan 2,49%, *firewall* mampu mengurangi *actual packet volume* secara keseluruhan, mengindikasikan bahwa peningkatan *flows* tersebut mungkin berasal dari *legitimate connections* yang memiliki karakteristik paket lebih efisien. Kombinasi penurunan 10,15% pada *total packets* dan penurunan 24,50% pada *total bytes* menunjukkan bahwa *firewall* tidak hanya mengurangi volume paket, tetapi juga secara khusus menargetkan paket-paket yang mengkonsumsi *bandwidth* tinggi, sehingga menghasilkan optimalisasi jaringan yang

komprehensif dengan mengurangi *network congestion* dan meningkatkan *overall network efficiency*.



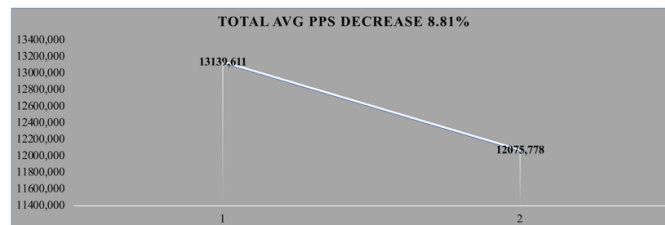
Gambar 4.19 Penurunan *total packets* 10.15%.

Gambar 4.20 Menunjukkan perbandingan rata-rata *bandwidth* sebelum dan sesudah implementasi *firewall filter*, terlihat hasil yang sangat positif dimana terjadi penurunan signifikan sebesar 22,58% dari 74,756 MBPS menjadi 60,983 MBPS. Penurunan yang substansial ini mengindikasikan bahwa *firewall filter* berhasil secara efektif mengoptimalkan penggunaan *bandwidth* dengan memblokir *traffic* yang mengkonsumsi kapasitas jaringan secara berlebihan, kemungkinan termasuk *streaming* media yang tidak *authorized*, *file transfer* yang tidak produktif, *P2P activities*, atau bahkan *traffic malicious* seperti data *exfiltration* dan *bandwidth-intensive attacks*. Hasil ini menunjukkan konsistensi dengan penurunan *total bytes* sebesar 24,50%, dimana *firewall* tidak hanya mengurangi volume data keseluruhan tetapi juga berhasil menurunkan rata-rata konsumsi *bandwidth* per unit waktu, mengindikasikan efektivitas *filtering* yang berkelanjutan sepanjang periode operasional. Penurunan 22,58% dalam *average BPS* memberikan manfaat operasional yang signifikan berupa peningkatan *quality of service* untuk aplikasi bisnis kritikal, pengurangan *latency* jaringan, dan tersedianya *bandwidth reserve* yang dapat dimanfaatkan untuk pertumbuhan *legitimate traffic* atau implementasi aplikasi baru, sambil tetap mempertahankan *connectivity* yang diperlukan untuk operasional normal dengan performa yang lebih optimal dan stabil.



Gambar 4.20 Penurunan *total average BPS* 22.58%.

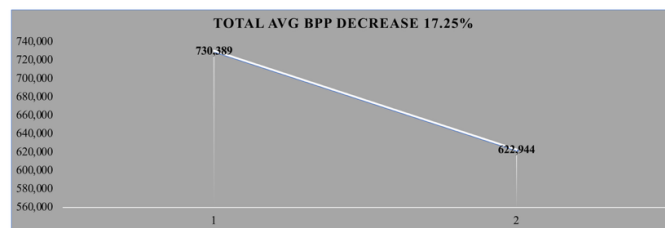
Gambar 4.21 Menunjukkan perbandingan rata-rata paket per detik sebelum dan sesudah implementasi *firewall filter*, terlihat hasil yang positif dimana terjadi penurunan sebesar 8,81% dari 13.139.611 *PPS* menjadi 12.075.778 *PPS*. Penurunan ini mengindikasikan bahwa *firewall filter* berhasil mengendalikan *rate* paket yang melewati jaringan dengan memblokir *traffic* berbasis volume tinggi, kemungkinan termasuk serangan *DDoS*, *network flooding*, *automated scanning tools*, atau *bot activities* yang biasanya ditandai dengan *burst* paket dalam jumlah besar namun berukuran kecil. Hasil ini menunjukkan efektivitas *firewall* dalam melakukan *rate limiting* dan *packet filtering*, dimana meskipun penurunan *PPS* (8,81%) relatif lebih kecil dibandingkan dengan penurunan *BPS* (22,58%) dan *total bytes* (24,50%), hal ini mengkonfirmasi bahwa *firewall* secara selektif menargetkan paket-paket yang mengkonsumsi *bandwidth* tinggi daripada sekadar memblokir semua jenis paket secara indiskriminatif. Kombinasi penurunan 8,81% pada *average PPS* dengan penurunan signifikan pada *metrics* lainnya menunjukkan bahwa *firewall* berhasil mengoptimalkan karakteristik *traffic* dengan mengurangi *packet overhead* sambil mempertahankan *legitimate communications*, sehingga menghasilkan *network efficiency* yang lebih baik dengan mengurangi *processing load* pada *network devices* dan meningkatkan *overall network stability*.



Gambar 4.21 Penurunan *total average PPS* 8.81%.

Gambar 4.22 Menunjukkan perbandingan rata-rata *bytes per packet* sebelum dan sesudah implementasi *firewall filter*, terlihat hasil yang signifikan dimana terjadi penurunan sebesar 17,25% dari 730,389 *bytes* menjadi 622,944 *bytes* per paket. Penurunan ini mengindikasikan bahwa *firewall filter* secara selektif memblokir atau membatasi paket-paket berukuran besar yang mungkin dianggap mencurigakan atau tidak sesuai dengan *security policy*, kemungkinan termasuk *large file transfers* yang tidak *authorized*, media streaming dengan *payload* besar, atau bahkan data *exfiltration attempts* yang menggunakan paket berukuran maksimal untuk mengoptimalkan *transfer rate*. Hasil ini menunjukkan karakteristik *filtering* yang *sophisticated*, dimana *firewall* tidak hanya melakukan *blocking* berdasarkan volume atau *frequency*, tetapi juga mempertimbangkan ukuran paket sebagai

indikator potensi ancaman atau aktivitas yang tidak diinginkan. Kombinasi penurunan 17,25% pada *average BPP* dengan penurunan yang lebih besar pada *total bytes* (24,50%) dan *average BPS* (22,58%) namun penurunan yang lebih kecil pada *average PPS* (8,81%) mengkonfirmasi bahwa *firewall* berhasil mengoptimalkan *traffic profile* dengan mengurangi *payload size per packet* sambil tetap mempertahankan *reasonable packet flow rate*, sehingga menghasilkan *network efficiency* yang lebih baik dengan karakteristik *traffic* yang lebih aman dan terkontrol.



Gambar 4.22 Penurunan *total average BPP* 17.25%.

Dalam upaya mengidentifikasi *domain malicious*, penelitian ini mengembangkan pendekatan multidimensional yang mencakup berbagai parameter kritis. Salah satu metode utama adalah menggunakan daftar *blacklist* yang komprehensif, dimana hingga akhir tahun 2022, telah teridentifikasi sejumlah 84.765 *domain* berbahaya. Proses identifikasi ini tidak hanya mengandalkan daftar statis, melainkan juga melakukan analisis mendalam terhadap pola *traffic* yang mencurigakan. Karakteristik utama dalam menentukan sebuah *domain* sebagai *malicious* meliputi beberapa indikator penting. Pertama, peneliti melakukan evaluasi terhadap jumlah *request* yang dilakukan oleh *client* menuju *domain* yang dicurigai. Dalam penelitian ini, tercatat 1.090 alamat *IP client* yang melakukan *request* ke *domain* berbahaya. Selanjutnya, dilakukan analisis volume *traffic inbound* dari *domain* tersebut, dengan tercatatnya 1.919 alamat *IP inbound* dari *domain malicious*. Metode ini diperkaya dengan analisis mendalam terhadap *traffic capture*, yang memungkinkan identifikasi pola *transfer* data yang tidak normal atau mencurigakan.

Berkaitan dengan implementasi *firewall* untuk *ISP*, penelitian ini merumuskan strategi yang mempertimbangkan tiga aspek kunci: kompatibilitas perangkat, sumber daya manusia, dan investasi minimal. Kompatibilitas perangkat dicapai melalui penggunaan *router MikroTik* dengan format *file .rsc* yang efisien, yang memungkinkan konversi data dari format *.csv* ke format yang sesuai dengan kebutuhan *router*. Pertimbangan sumber daya manusia difokuskan pada penggunaan sistem *firewall* yang mudah dikonfigurasi dan

memiliki kemampuan konversi data yang efektif, serta konfigurasi *RouterOS* yang tidak kompleks.

Analisis dampak penerapan *firewall* terhadap *traffic flow* menghasilkan temuan menarik. Meskipun terjadi sedikit peningkatan total *flows* (+2,49%), penelitian menunjukkan penurunan signifikan dalam beberapa parameter kunci. *Total bytes* mengalami penurunan sebesar 24,50%, yang mengindikasikan keberhasilan *firewall* dalam membatasi transfer data berbahaya. Penurunan total *packets* sebesar 10,15% menunjukkan kemampuan sistem dalam mencegah paket tidak aman. Dampak lebih lanjut terlihat pada penurunan *bandwidth* sebesar 22,58% dan rata-rata paket per detik sebesar 8,81%.

Penelitian menegaskan bahwa sistem *firewall* yang dikembangkan berhasil mengidentifikasi dan membatasi *domain malicious* dengan gangguan minimal terhadap operasional jaringan *ISP*. Untuk memastikan efektivitas berkelanjutan, penelitian merekomendasikan beberapa strategi lanjutan. Pertama, pembaruan berkala terhadap daftar *domain malicious*. Kedua, menjaga keseimbangan antara keamanan jaringan dan performa sistem. Terakhir, melakukan pengujian berkala untuk mengoptimalkan konfigurasi *firewall*, memastikan sistem tetap adaptif terhadap ancaman keamanan yang selalu berkembang.

Pendekatan komprehensif ini tidak hanya memberikan solusi teknis dalam menghadapi ancaman *domain malicious*, tetapi juga menawarkan kerangka kerja yang fleksibel dan efisien bagi infrastruktur jaringan *ISP*. Desain *firewall* yang dikembangkan dirancang dengan fokus pada aspek praktis, terutama dalam kemudahan implementasi dan konfigurasi yang dapat diintegrasikan secara *seamless* dengan infrastruktur jaringan *ISP* yang sudah ada. Arsitektur modular yang digunakan memungkinkan *deployment* bertahap tanpa mengganggu operasional jaringan, sementara *interface* konfigurasi yang intuitif meminimalkan kompleksitas teknis dan mengurangi waktu *setup* secara signifikan. Kemudahan integrasi ini tercermin dari kompatibilitas sistem dengan berbagai vendor perangkat jaringan dan protokol standar industri, sehingga *ISP* tidak perlu melakukan perubahan signifikan pada topologi jaringan eksisting mereka. Aspek praktis diukur melalui kemudahan *deployment* yang dapat dilakukan dalam waktu singkat, tingkat kompleksitas konfigurasi yang rendah, dan kebutuhan training minimal bagi administrator jaringan.

Dari segi ekonomis, solusi ini menawarkan penghematan *Capital Expenditure (CAPEX)* – Pengeluaran Modal dan pengurangan *Operational Expenditure (OPEX)* – Pengeluaran Operasional yang substansial dibandingkan dengan implementasi *multiple security appliances* terpisah. *ISP* tidak perlu mengeluarkan modal dalam pengadaan infrastruktur baru dan tidak ada *annual fee* yang dikeluarkan tiap tahun seperti halnya lisensi

pada *Commercial Enterprise Firewall*. Efektivitas sistem divalidasi melalui tingkat deteksi *malicious domains* yang tinggi, *false positive rate* yang rendah, dan *response time* yang optimal untuk *domain filtering*. Efisiensi operasional tercermin dalam penggunaan *resource* sistem yang minimal, *memory footprint* yang rendah untuk *handling concurrent sessions* dalam jumlah besar, dan kemampuan mempertahankan *network throughput* yang mendekati *baseline performance*. Dengan memanfaatkan teknologi *filtering* dan analisis data yang mendalam, penelitian ini membuka peluang baru dalam manajemen keamanan jaringan yang proaktif dan berkelanjutan. Konsep praktis yang ditawarkan tidak hanya mengacu pada kemudahan implementasi dan konfigurasi, tetapi juga mencakup aspek *maintenance* yang minimal, *scalability* yang responsif terhadap pertumbuhan jaringan, dan *cost-effectiveness* dalam hal *OPEX* maupun *CAPEX* bagi operator ISP.

Benchmark utama sistem difokuskan pada optimalisasi *confusion matrix* dengan target meminimalkan *False Positive* dan *False Negative*, serta memaksimalkan *True Positive* dan *True Negative*. Minimalisasi *False Positive* memastikan *domain legitimate* tidak salah terblokir sehingga tidak mengganggu pengalaman pengguna, sedangkan minimalisasi *False Negative* memastikan tidak ada *domain malicious* yang lolos dari sistem deteksi. Sebaliknya, maksimalisasi *True Positive* mengoptimalkan kemampuan sistem dalam mengidentifikasi dan memblokir *domain malicious* secara akurat, sementara maksimalisasi *True Negative* memastikan *domain legitimate* dapat diakses tanpa hambatan. Keseimbangan optimal keempat parameter ini menjadi indikator kunci performa sistem dalam memberikan perlindungan maksimal tanpa mengorbankan aksesibilitas layanan yang *legitimate*.

Disamping pendekatan teknis perlu melakukan pendekatan non teknis terutama kesadaran terhadap ancaman keamanan siber. Meskipun kesadaran manusia terhadap ancaman *malicious domain* terus mengalami peningkatan, perlindungan menggunakan *firewall* masih menjadi kebutuhan yang sangat krusial dalam arsitektur keamanan siber modern. Hal ini disebabkan oleh kondisi psikologis manusia yang secara inheren masih rentan terhadap serangan *social engineering*, sebagaimana dijelaskan dalam penelitian interdisipliner yang menyatakan bahwa *social engineering* didefinisikan sebagai tindakan menggunakan manipulasi dan penipuan untuk mendapatkan akses ke informasi rahasia (Washo, 2021). Faktor psikologis manusia menjadi titik kelemahan utama karena *social engineering* sebagai proses yang mengeksploitasi psikologi manusia daripada metode *hacking* yang canggih (Siddiqi et al., 2022b) dimana penyerang memanfaatkan aspek emosional, motivasi, dan perilaku alami manusia dalam situasi profesional. Kondisi ini

diperkuat oleh temuan yang menunjukkan bahwa orang mudah terjebak dalam trik *social engineering* berdasarkan konteks psikologis (Gold, 2010) sehingga meskipun individu telah memiliki pengetahuan tentang ancaman domain berbahaya, mereka tetap dapat dimanipulasi melalui teknik-teknik psikologis yang *sophisticated*. Oleh karena itu, implementasi *firewall* sebagai lapisan perlindungan teknis tetap diperlukan untuk mengkompensasi kerentanan psikologis manusia yang sulit dihilangkan sepenuhnya, menciptakan sistem keamanan berlapis yang menggabungkan aspek teknologi dan *awareness* manusia.

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini membuktikan bahwa pendekatan *malicious domain filtering* melalui layanan *perimeter firewall* merupakan strategi fundamental yang memberikan perlindungan proaktif terhadap ancaman siber yang terus berkembang. Metode identifikasi *domain* berbahaya yang dikembangkan melampaui pendekatan konvensional dengan mengintegrasikan empat parameter kritis: analisis riwayat aktivitas domain, validasi reputasi keamanan melalui *database* internasional, pelaporan pengguna, dan analisis korelasi antardomain. Pendekatan multidimensional ini terbukti efektif dalam memberikan manajemen keamanan jaringan yang dinamis dan komprehensif.

Artefak yang didapatkan dari forensika jaringan ditemukan di beberapa tempat. Pertama di *address-list*, di dalam *address-list dinamic Router MikroTik* didapatkan data berupa *IP address public* yang digunakan oleh *malicious domain*. Apabila ada perubahan *IP domain* yang dilakukan dalam *DNS zone*, akan mengubah *IP* yang ada ada dalam *address list* secara otomatis. Kedua, alamat *domain* bisa didapatkan dalam *DNS Cache MikroTik*. Ketiga, *file log* bernama *REQUEST_MALICIOUS-DOMAIN.txt* yang *digenerate* dari *dynamic address list*. Isi *file* ini berupa kumpulan *IP* lokal yang melakukan request ke *malicious domain*. Keempat, *file log* bernama *INBOUND_MALICIOUS-DOMAIN.txt* yang *digenerate* dari *dynamic address list*. Isi *file* ini berupa kumpulan *IP public malicious domain* yang terblokir oleh *firewall*. Kelima, *file data binary* yang ada di folder penyimpanan di *server* kolektor bernama *nfcapd.YYYYMMDDhhmm*. *File* ini bisa diekstrak menggunakan atribut *tools NFDUMP* untuk mendapatkan informasi *source IP*, *destination IP*, protokol, ukuran paket, *bandwith*, *traffic flow*, *bytes packet*, *packet per second*, *byte per second*.

Penelitian membuktikan bahwa *Malicious Domain Filtering* merupakan solusi *firewall* yang efektif dan sesuai untuk infrastruktur *ISP*. Metode ini menunjukkan keunggulan signifikan dalam empat aspek utama. Pertama, efisiensi infrastruktur tercapai melalui pemanfaatan perangkat yang sudah tersedia seperti *router MikroTik* dan *server Linux*. Kedua, pendekatan ini mampu meminimalkan investasi sampai dengan Rp. 0 dengan tidak memerlukan pengadaan perangkat dan sistem baru. Kondisi ini sangat sesuai untuk *ISP* di Indonesia tergolong Usaha Mikro dan Kecil yang memiliki modal terbatas dan pendapatan

yang tidak besar, jumlah karyawan, dan sumber daya yang terbatas. Ketiga, metode ini memiliki adaptabilitas tinggi terhadap sumber daya manusia, kompatibel dengan kemampuan teknis personel *ISP* yang ada. Terakhir, fleksibilitas konfigurasi memungkinkan penyesuaian yang mudah sesuai kebutuhan spesifik jaringan.

Analisis kuantitatif penelitian mengungkapkan penerapan *firewall* menunjukkan dampak efektivitas yang signifikan dalam mengamankan jaringan. Tercatat 1.090 alamat *IP client* yang pernah mengakses *domain* berbahaya, sementara 1.919 *domain* berbahaya berhasil diblokir. Perubahan metrik jaringan menunjukkan hasil yang mengesankan: peningkatan total aliran sebesar 2,49%, penurunan *total byte* mencapai 24,5%, pengurangan total paket 10,5%, penurunan rata-rata *bandwidth* 22,58%, dan berkurangnya paket per detik sebesar 8,81%. Implementasi *malicious domain filtering* memiliki implikasi yang jauh melampaui sekadar perlindungan teknis. Metode ini secara komprehensif meningkatkan keamanan infrastruktur digital, melindungi pengguna dari ancaman siber, mengoptimalkan efisiensi lalu lintas jaringan, dan signifikan mengurangi risiko pelanggaran data.

5.2 Saran

Berdasarkan kesimpulan penelitian, beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan implementasi yang optimal adalah sebagai berikut:

1. Peningkatan dan pemeliharaan daftar *blacklist*

Black-list malicious domain yang dijadikan sumber terupdate tiap 5 menit, sementara *black-list* yang digunakan untuk simulasi penelitian masih bersifat statis, belum disinkronisasikan secara realtime dengan *black-list* sumber, penting dilakukan sinkronisasi secara otomatis agar *black list* selalu *up to date*.

2. Data parameter analisis masih mengandalkan data fluktuasi global dari *Traffic Flow*, penelitian lebih lanjut bisa diperdalam dengan menganalisa secara detail berdasarkan *IP-IP* dan protokol yang berjalan di dalam pertukaran data.
3. Data diambil dari *interface router* yang sama akan tetapi dalam rentang waktu yang berbeda. Hal ini bisa menghasilkan data yang berbeda karena aktivitas pelanggan yang tidak sama di waktu yang berbeda. Stabilitas koneksi di waktu yang berbeda juga bisa berpengaruh pada data yang dihasilkan. Data lebih *valid* apabila pengambilan data dilakukan menggunakan metode yang sama dalam waktu yang sama atau minimal di waktu yang berdekatan.
4. Pengujian prototipe masih dilakukan di satu *ISP*, disarankan untuk melakukan pengujian di *ISP* berbeda di wilayah regional yang berbeda. Pengguna *internet* di

regional berbeda, kemungkinan memiliki karakter dan kebiasaan yang berbeda, hal ini juga kemungkinan akan berpengaruh terhadap data yang didapatkan.

5. Memungkinkan penggunaan *analysis tools* lain sebagai pembanding sehingga bisa mendapatkan perbandingan hasil untuk melakukan *validasi*.

Dengan menerapkan saran-saran ini, diharapkan efektivitas *filtering malicious domain* dapat terus ditingkatkan, sehingga jaringan dan pengguna tetap aman di tengah lanskap ancaman siber yang dinamis.

Daftar Pustaka

- Ahammad, S. H., Kale, S. D., Upadhye, G. D., Pande, S. D., Babu, E. V., Dhumane, A. V., & Bahadur, M. D. K. J. (2022). Phishing URL detection using machine learning methods. *Advances in Engineering Software*, 173. <https://doi.org/10.1016/j.advengsoft.2022.103288>
- Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F. (2022). Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. In *Journal of Cybersecurity and Privacy* (Vol. 2, Issue 3, pp. 527–555). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/jcp2030027>
- Aldawood, H., & Skinner, G. (2019a). Reviewing cyber security social engineering training and awareness programs-pitfalls and ongoing issues. In *Future Internet* (Vol. 11, Issue 3). MDPI AG. <https://doi.org/10.3390/fi11030073>
- Aldawood, H., & Skinner, G. (2019b). Reviewing cyber security social engineering training and awareness programs-pitfalls and ongoing issues. In *Future Internet* (Vol. 11, Issue 3). MDPI AG. <https://doi.org/10.3390/fi11030073>
- Alhayan, F., Alshuhail, A., Ismail, A. O. A., Alrusaini, O., Alahmari, S., Yahya, A. E., Abdullah, M., & Al Zanin, S. (2025). Artificial intelligence-driven cybersecurity: enhancing malicious domain detection using attention-based deep learning model with optimization algorithms. *Scientific Reports*, 15(1). <https://doi.org/10.1038/s41598-025-99420-y>
- Aljabri, M., & Mohammad, R. M. A. (2023). Click fraud detection for online advertising using machine learning. *Egyptian Informatics Journal*, 24(2), 341–350. <https://doi.org/10.1016/j.eij.2023.05.006>
- Almarzooqi, A., Mahmoud, J., Alzaabi, B., Ghebremichael, A., & Aldwairi, M. (2022). *Detecting Malicious Domains Using Statistical Internationalized Domain Name Features in Top Level Domains*. <http://arxiv.org/abs/2211.08020>
- APJII - ISP. (2023). *Survei Internet Service Provider Industri & Market Profile*.
- APJII - UMKM & Korporasi. (2023). *Survei penggunaan Internet oleh UMKM dan Korporasi*.
- APNIC. (2022). *APNIC Survey Report*.
- Babari, P., Hielscher, M., Edelsbrunner, P. A., Conti, M., Honegger, B. D., & Marinus, E. (2023). A literature review of children's and youth's conceptions of the internet. In

- International Journal of Child-Computer Interaction* (Vol. 37). Elsevier B.V.
<https://doi.org/10.1016/j.ijcci.2023.100595>
- Badan Pusat Statistik. (2022). *Statistik Telekomunikasi Indonesia*.
- Böck, L., Vasilomanolakis, E., Wolf, J. H., & Mühlhäuser, M. (2019). Autonomously detecting sensors in fully distributed botnets. *Computers and Security*, 83, 1–13.
<https://doi.org/10.1016/j.cose.2019.01.004>
- Ceron, J. M., Scholten, C., Pras, A., & Santanna, J. (2020, April 1). MikroTik Devices Landscape, Realistic Honeypots, and Automated Attack Classification. *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020*.
<https://doi.org/10.1109/NOMS47738.2020.9110336>
- Colasoft. (2017). *capsa-pro-whitepaper*. <https://www.colasoft.com/download/capsa-pro-whitepaper.pdf>
- Cozza, F., Guarino, A., Isernia, F., Malandrino, D., Rapuano, A., Schiavone, R., & Zaccagnino, R. (2020). Hybrid and lightweight detection of third party tracking: Design, implementation, and evaluation. *Computer Networks*, 167.
<https://doi.org/10.1016/j.comnet.2019.106993>
- Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022a). Cyber risk and cybersecurity: a systematic review of data availability. *Geneva Papers on Risk and Insurance: Issues and Practice*, 47(3), 698–736.
<https://doi.org/10.1057/s41288-022-00266-6>
- Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022b). Cyber risk and cybersecurity: a systematic review of data availability. *Geneva Papers on Risk and Insurance: Issues and Practice*, 47(3), 698–736.
<https://doi.org/10.1057/s41288-022-00266-6>
- Cucchiarelli, A., Morbidoni, C., Spalazzi, L., & Baldi, M. (2021). Algorithmically generated malicious domain names detection based on n-grams features. *Expert Systems with Applications*, 170. <https://doi.org/10.1016/j.eswa.2020.114551>
- Dabić, M., Maley, J. F., Švarc, J., & Poček, J. (2023). Future of digital work: Challenges for sustainable human resources management. *Journal of Innovation and Knowledge*, 8(2).
<https://doi.org/10.1016/j.jik.2023.100353>
- Dimitriadis, A., Ivezic, N., Kulvatunyou, B., & Mavridis, I. (2020). D4I - Digital forensics framework for reviewing and investigating cyber attacks. *Array*, 5, 100015.
<https://doi.org/10.1016/j.array.2019.100015>

- Dong, Z., Chen, X., Zhao, J., Zhao, S., & Wu, J. (2022). Malicious Domain Name Detection based on Knowledge Graph. *Proceedings of 2022 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control, SDPC 2022*, 251–256. <https://doi.org/10.1109/SDPC55702.2022.9915824>
- Emara, A. H., Elhamahmy, M. E., Elgazzar, M. M. A., & Emara, A.-H. M. (2019). A Proposed Approach for Management of Multiple Firewalls Using REST API Architecture Combination of Stylo-based Features and Frequency-based Features for Identifying the Author of Short Arabic Text View project A Proposed Approach for Management of Multip. *International Journal of P2P Network Trends and Technology (IJPTT)*, 9.
- Erola, A., Agrafiotis, I., Nurse, J. R. C., Axon, L., Goldsmith, M., & Creese, S. (2022). A system to calculate cyber-value-at-risk. *Computers and Security*, 113. <https://doi.org/10.1016/j.cose.2021.102545>
- Ghosh, M., Ghosh, D., Halder, R., & Chandra, J. (2023). Investigating the impact of structural and temporal behaviors in Ethereum phishing users detection. *Blockchain: Research and Applications*, 4(4). <https://doi.org/10.1016/j.bcra.2023.100153>
- Gold, S. (2010). Social engineering today: Psychology, strategies and tricks. *Network Security*, 2010(11), 11–14. [https://doi.org/10.1016/S1353-4858\(10\)70135-5](https://doi.org/10.1016/S1353-4858(10)70135-5)
- Guarascio, M., Cassavia, N., Pisani, F. S., & Manco, G. (2022). Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection. *Future Generation Computer Systems*, 135, 30–43. <https://doi.org/10.1016/j.future.2022.04.028>
- Gusev, A. (2022). Domestic private banking solutions can be quite successful as an effective protection against whaling-style cyber attacks which are used as a basis for more complex targeted phishing. *Procedia Computer Science*, 213(C), 391–399. <https://doi.org/10.1016/j.procs.2022.11.083>
- Hadjadj, T. E., Tebourbi, R., Bouhoula, A., & Ksantini, R. (2019). *Optimization of Parallel Firewalls Filtering Rules*.
- Han, S., & Yu, J. (2021). *Optimal feature selection for firewall log analysis using Machine learning and Hybrid Metaheuristic algorithms*. <https://doi.org/10.31224/osf.io/pm3hy>
- Hausken, K. (2020a). Cyber resilience in firms, organizations and societies. In *Internet of Things (Netherlands)* (Vol. 11). Elsevier B.V. <https://doi.org/10.1016/j.iot.2020.100204>

- Hausken, K. (2020b). Cyber resilience in firms, organizations and societies. In *Internet of Things (Netherlands)* (Vol. 11). Elsevier B.V. <https://doi.org/10.1016/j.iot.2020.100204>
- Ismail, H., Febiyanto, F., Kevin, & Moniaga, J. V. (2022). Methods to prevent privacy violations on the internet on the personal level in Indonesia. *Procedia Computer Science*, 216, 650–654. <https://doi.org/10.1016/j.procs.2022.12.180>
- Jacob, S., Qiao, Y., Ye, Y., & Lee, B. (2022). Anomalous distributed traffic: Detecting cyber security attacks amongst microservices using graph convolutional networks. *Computers and Security*, 118. <https://doi.org/10.1016/j.cose.2022.102728>
- Jaïdi, F. (2019). A Quantified Trust-Risk Assessment Approach for Enhancing Firewalls-Filtering Services Advanced Access Control View project Risk Based Access Control Management View project A Quantified Trust-Risk Assessment Approach for Enhancing Firewalls-Filtering Services. In *Journal of Information Assurance and Security* (Vol. 14).
- Jurkiewicz, P. (2022). flow-models: A framework for analysis and modeling of IP network flows. *SoftwareX*, 17. <https://doi.org/10.1016/j.softx.2021.100929>
- Jurkiewicz, P., Rzym, G., & Boryło, P. (2021). Flow length and size distributions in campus Internet traffic. *Computer Communications*, 167, 15–30. <https://doi.org/10.1016/j.comcom.2020.12.016>
- Karpowicz, M. P. (2021). Adaptive tuning of network traffic policing mechanisms for DDoS attack mitigation systems. *European Journal of Control*, 61, 101–118. <https://doi.org/10.1016/j.ejcon.2021.07.001>
- Karthikeyan, T., Govindarajan, M., & Vijayakumar, V. (2023). An effective fraud detection using competitive swarm optimization based deep neural network. *Measurement: Sensors*, 27. <https://doi.org/10.1016/j.measen.2023.100793>
- Lazarus, S., Whittaker, J. M., McGuire, M. R., & Platt, L. (2023). What do we know about online romance fraud studies? A systematic review of the empirical literature (2000 to 2021). *Journal of Economic Criminology*, 2, 100013. <https://doi.org/10.1016/j.jeconc.2023.100013>
- Li, X. J., Ma, M., & Arjun, N. (2019, November 1). An Encryption Algorithm to Prevent Domain Name System Cache Poisoning Attacks. *2019 29th International Telecommunication Networks and Applications Conference, ITNAC 2019*. <https://doi.org/10.1109/ITNAC46935.2019.9078019>

- Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*, 7, 8176–8186. <https://doi.org/10.1016/j.egyr.2021.08.126>
- Luca Deri. (n.d.). *Wireshark 101 Essential Skills for Network Analysis* ®. Retrieved August 1, 2025, from <https://luca.ntop.org/gr2022/Wireshark.pdf>
- Manasrah, A. M., Khmour, T., & Freehat, R. (2022). DGA-based botnets detection using DNS traffic mining. *Journal of King Saud University - Computer and Information Sciences*, 34(5), 2045–2061. <https://doi.org/10.1016/j.jksuci.2022.03.001>
- Marques, C., Malta, S., & Magalhães, J. P. (2021a). DNS dataset for malicious domains detection. *Data in Brief*, 38. <https://doi.org/10.1016/j.dib.2021.107342>
- Marques, C., Malta, S., & Magalhães, J. P. (2021b). DNS dataset for malicious domains detection. *Data in Brief*, 38. <https://doi.org/10.1016/j.dib.2021.107342>
- Meland, P. H., Nesheim, D. A., Bernsmed, K., & Sindre, G. (2022). Assessing cyber threats for storyless systems. *Journal of Information Security and Applications*, 64. <https://doi.org/10.1016/j.jisa.2021.103050>
- Miloslavskaya, N., Stodelov, D., Malakhov, M., & Marachyova, A. (2022). Security Architecture of Network Security Centers as Part of Modern Intranets. *Procedia Computer Science*, 213(C), 58–63. <https://doi.org/10.1016/j.procs.2022.11.038>
- Mishra, A., Alzoubi, Y. I., Anwar, M. J., & Gill, A. Q. (2022). Attributes impacting cybersecurity policy development: An evidence from seven nations. *Computers and Security*, 120. <https://doi.org/10.1016/j.cose.2022.102820>
- Mizerek, E., Wolf, L., & Moon, M. D. (2023). Identifying and Mitigating Fraud When Using Social Media for Research Recruitment. *Journal of Emergency Nursing*, 49(4), 530–533. <https://doi.org/10.1016/j.jen.2023.04.002>
- Mota, F. P. B., & Cilento, I. (2020). Competence for internet use: Integrating knowledge, skills, and attitudes. *Computers and Education Open*, 1, 100015. <https://doi.org/10.1016/j.caeo.2020.100015>
- Motulsky, H. J., Head, T., & Clarke, P. B. S. (2025). Analyzing lognormal data: A nonmathematical practical guide. In *Pharmacological Reviews* (Vol. 77, Issue 3). Elsevier Inc. <https://doi.org/10.1016/j.pharmr.2025.100049>
- Mvula, P. K., Branco, P., Jourdan, G. V., & Viktor, H. L. (2022). COVID-19 malicious domain names classification[Formula presented]. *Expert Systems with Applications*, 204. <https://doi.org/10.1016/j.eswa.2022.117553>

- Nanath, K., & Olney, L. (2023). An investigation of crowdsourcing methods in enhancing the machine learning approach for detecting online recruitment fraud. *International Journal of Information Management Data Insights*, 3(1). <https://doi.org/10.1016/j.jjimei.2023.100167>
- Neetu Bansla, Khusboo Jain, & Swati Kunwar. (2019). *Social Engineering -A Technique for Managing Human Behavior*. 5(1).
- Palaniappan, G., Sangeetha, S., Rajendran, B., Sanjay, Goyal, S., & Bindhumadhava, B. S. (2020a). Malicious Domain Detection Using Machine Learning on Domain Name Features, Host-Based Features and Web-Based Features. *Procedia Computer Science*, 171, 654–661. <https://doi.org/10.1016/j.procs.2020.04.071>
- Palaniappan, G., Sangeetha, S., Rajendran, B., Sanjay, Goyal, S., & Bindhumadhava, B. S. (2020b). Malicious Domain Detection Using Machine Learning on Domain Name Features, Host-Based Features and Web-Based Features. *Procedia Computer Science*, 171, 654–661. <https://doi.org/10.1016/j.procs.2020.04.071>
- Pitafi, S., Anwar, T., Widia, I. D. M., Yimwadsana, B., & Pitafi, S. (2023). Revolutionizing Perimeter Intrusion Detection: A Machine Learning-Driven Approach with Curated Dataset Generation for Enhanced Security. *IEEE Access*, 11, 106954–106966. <https://doi.org/10.1109/ACCESS.2023.3318600>
- Qu, J., Ma, X., & Liu, W. (2021). Querent-centric Domain Name System Modeling and Its Application in Passive Software Discovery. *Proceedings - 2021 8th International Conference on Dependable Systems and Their Applications, DSA 2021*, 152–160. <https://doi.org/10.1109/DSA52907.2021.00024>
- Sanog, & Bhutan. (n.d.). *Flow-tools Tutorial*.
- Shaengchart, Y., & Kraiwanit, T. (2023). Starlink satellite project impact on the Internet provider service in emerging economies. *Research in Globalization*, 6. <https://doi.org/10.1016/j.resglo.2023.100132>
- Shen, X., Zhang, X., & Chen, Y. (2022). Deep Learning Powered Adversarial Sample Attack Approach for Security Detection of DGA Domain Name in Cyber Physical Systems. *IEEE Wireless Communications*, 29(2), 16–21. <https://doi.org/10.1109/MWC.001.2100247>
- Siddiqi, M. A., Pak, W., & Siddiqi, M. A. (2022a). A Study on the Psychology of Social Engineering-Based Cyberattacks and Existing Countermeasures. In *Applied Sciences (Switzerland)* (Vol. 12, Issue 12). MDPI. <https://doi.org/10.3390/app12126042>

- Siddiqi, M. A., Pak, W., & Siddiqi, M. A. (2022b). A Study on the Psychology of Social Engineering-Based Cyberattacks and Existing Countermeasures. In *Applied Sciences (Switzerland)* (Vol. 12, Issue 12). MDPI. <https://doi.org/10.3390/app12126042>
- Socradar. (2024). *INDONESIA Threat Landscape Report*. <https://socradar.io/>
- Song, Y., Yang, L., & Li, L. (2024). A study on the impact mechanism of internet embedding on rural E-commerce entrepreneurship. *Research in International Business and Finance*, 68. <https://doi.org/10.1016/j.ribaf.2023.102196>
- Steinebach, M., Zenglein, S., & Brandl, K. (2021). Phishing detection on tor hidden services. *Forensic Science International: Digital Investigation*, 36, 301117. <https://doi.org/10.1016/j.fsidi.2021.301117>
- Stock, S. R., & De Carlo, F. (2023). Meta-data for absorption tomography measurements. *Tomography of Materials and Structures*, 3, 100015. <https://doi.org/10.1016/j.tmater.2023.100015>
- Suchacka, G., Cabri, A., Rovetta, S., & Masulli, F. (2021). Efficient on-the-fly Web bot detection. *Knowledge-Based Systems*, 223. <https://doi.org/10.1016/j.knosys.2021.107074>
- Sugasawa, S., Shibahashi, Y., Kunimune, H., Goromaru, H., & Tanimoto, S. (2023). *DNS-tunneling-detection Method by Monitoring DNS Subdomain Length for General Usage*. 121–122. <https://doi.org/10.1109/gcce56475.2022.10014255>
- Syafitri, W., Shukur, Z., Mokhtar, U. A., Sulaiman, R., & Ibrahim, M. A. (2022). Social Engineering Attacks Prevention: A Systematic Literature Review. *IEEE Access*, 10, 39325–39343. <https://doi.org/10.1109/ACCESS.2022.3162594>
- Takefuji, Y. (2023). Case report on enormous economic losses caused by fraud from Japan to the world. *Journal of Economic Criminology*, 1, 100003. <https://doi.org/10.1016/j.jeconc.2023.100003>
- Tariq, U., Ahmed, I., Bashir, A. K., & Shaukat, K. (2023a). A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. In *Sensors* (Vol. 23, Issue 8). MDPI. <https://doi.org/10.3390/s23084117>
- Tariq, U., Ahmed, I., Bashir, A. K., & Shaukat, K. (2023b). A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. In *Sensors* (Vol. 23, Issue 8). MDPI. <https://doi.org/10.3390/s23084117>
- Vake, D., Vičić, J., & Tošić, A. (2025). Hive: A secure, scalable framework for distributed Ollama inference. *SoftwareX*, 30. <https://doi.org/10.1016/j.softx.2025.102183>

- Van Der Toorn, O., Müller, M., Dickinson, S., Hesselman, C., Sperotto, A., & Van Rijswijk-Deij, R. (2022). Addressing the challenges of modern DNS a comprehensive tutorial. In *Computer Science Review* (Vol. 45). Elsevier Ireland Ltd. <https://doi.org/10.1016/j.cosrev.2022.100469>
- Varga, S., Brynielsson, J., & Franke, U. (2021). Cyber-threat perception and risk management in the Swedish financial sector. *Computers and Security*, 105. <https://doi.org/10.1016/j.cose.2021.102239>
- Washo, A. H. (2021). An interdisciplinary view of social engineering: A call to action for research. *Computers in Human Behavior Reports*, 4. <https://doi.org/10.1016/j.chbr.2021.100126>
- Welda. (2023). *Penentuan Router MikroTik Berdasarkan Harga dan Kualitas Menggunakan Metode Simple Additive Weighting (SAW)*. <https://doi.org/10.59688/bufnets>
- Yaqoob, I., Salah, K., Jayaraman, R., & Omar, M. (2023). Metaverse applications in smart cities: Enabling technologies, opportunities, challenges, and future directions. In *Internet of Things (Netherlands)* (Vol. 23). Elsevier B.V. <https://doi.org/10.1016/j.iot.2023.100884>
- Yoshida, K., Fujiwara, K., Sato, A., & Sannomiya, S. (2020). Cardinality Analysis to Classify Malicious Domain Names. *Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020*, 826–832. <https://doi.org/10.1109/COMPSAC48688.2020.0-161>
- Yu, L., Dong, J., Chen, L., Li, M., Xu, B., Li, Z., Qiao, L., Liu, L., Zhao, B., & Zhang, C. (2021). PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection. *Computer Networks*, 194. <https://doi.org/10.1016/j.comnet.2021.108117>
- Yuryna Connolly, A., & Borrion, H. (2022). Reducing Ransomware Crime: Analysis of Victims' Payment Decisions. *Computers and Security*, 119. <https://doi.org/10.1016/j.cose.2022.102760>
- Zeng, M., Du, J., Zhu, X., & Deng, X. (2023). Does internet use drive rural household savings? Evidence from 7825 farmer households in rural China. *Finance Research Letters*, 57. <https://doi.org/10.1016/j.frl.2023.104275>
- Zeng, Y. (2022). AI Empowers Security Threats and Strategies for Cyber Attacks. *Procedia Computer Science*, 208, 170–175. <https://doi.org/10.1016/j.procs.2022.10.025>

- Zhao, C., Zhang, Y., Zang, T., Chen, Y., & Wang, Y. (2020). *A Multi-feature-based Approach to Malicious Domain Name Identification from DNS Traffic*. 1–5. <https://doi.org/10.1109/ict49546.2020.9239506>
- Zheng, Y., Li, Z., Xu, X., & Zhao, Q. (2022). Dynamic defenses in cyber security: Techniques, methods and challenges. *Digital Communications and Networks*, 8(4), 422–435. <https://doi.org/10.1016/j.dcan.2021.07.006>