



**IMPLEMENTASI *ABSTRACTION LAYER* UNTUK
PROPAGASI DAFTAR BLOKIR DINAMIS PADA
INFRASTRUKTUR *FIREWALL MULTI-VENDOR***

SKRIPSI

**MUHAMMAD ABIAN ABDI PRATAMA
2207421039**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026**



**IMPLEMENTASI *ABSTRACTION LAYER* UNTUK
PROPAGASI DAFTAR BLOKIR DINAMIS PADA
INFRASTRUKTUR *FIREWALL MULTI-VENDOR***

SKRIPSI

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

**MUHAMMAD ABIAN ABDI PRATAMA
2207421039**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Abian Abdi Pratama

NIM : 2207421039

Jurusan/Program Studi : T. Informatika dan Komputer / T. Multimedia dan Jaringan

Judul skripsi : IMPLEMENTASI ABSTRACTION LAYER UNTUK
PROPAGASI DAFTAR BLOKIR DINAMIS PADA INFRASTRUKTUR
FIREWALL MULTI-VENDOR

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

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

Depok, 9 Juni, 2026
Yang membuat pernyataan



(Muhammad Abian Abdi Pratama)
NIM. 2207421039



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Muhammad Abian Abdi Pratama

NIM : 2207421039

Program Studi : TMJ

Judul Skripsi : IMPLEMENTASI ABSTRACTION LAYER UNTUK
PROPAGASI DAFTAR BLOKIR DINAMIS PADA INERASTRUKTUR
FIREWALL MULTI-VENDOR

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Senin Tanggal 22
Bulan Juni Tahun 2026 dan dinyatakan LULUS.

Disahkan oleh:

Pembimbing I : Ayu Rosyida Zain, S.ST., M.T

Penguji I : Dr. Defiana Arnaldy, S.Tp., M.Si.

Penguji II : Iik Muhamad Malik Matin, S.Kom., M.T.

Penguji III : Andika Riyadi Jasril, S.Pd., M.Pd.T

Mengetahui:

Jurusan Teknik Informatika dan Komputer

Ketua



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

NIP. 197908032003122003



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya, sehingga skripsi yang berjudul **"Implementasi *Abstraction Layer* untuk Propagasi Daftar Blokir Dinamis pada Infrastruktur *Firewall Multi-Vendor*"** ini dapat diselesaikan dengan baik. Penulis menyadari bahwa tanpa pertolongan-Nya, proses panjang ini tidak akan dapat dilalui.

Skripsi ini tidak lepas dari peran banyak pihak yang turut mendukung perjalanan penulis. Untuk itu, penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT, atas segala kesehatan, kekuatan, dan kemudahan yang diberikan, tidak hanya selama penyusunan skripsi, tetapi sepanjang masa perkuliahan.
2. Ibu Ayu Rosyida Zain, S.ST., M.T. selaku dosen pembimbing, yang dengan sabar meluangkan waktu dan pikirannya untuk memberikan arahan serta masukan yang sangat berarti bagi penulis.
3. Ibu Dr. Prihatin Oktivasari, S.Si., M.Si., atas ilmu dan motivasi yang beliau berikan selama perkuliahan, yang banyak menginspirasi penulis dalam menyelesaikan studi ini.
4. Kedua orang tua penulis, yang tak pernah berhenti memberikan kasih sayang, doa, dan dukungan baik moral maupun material, serta menjadi alasan terbesar penulis untuk terus melangkah.
5. Pratama dan Christian, rekan seperjuangan yang bersama-sama melewati setiap tantangan, berdiskusi, dan saling menguatkan selama pengerjaan tugas akhir ini.
6. Salsabilla, Reishafa, Izzaturrachmi, Safina, dan Azzuri selaku rekan-rekan PBL, atas kebersamaan, semangat, dan kenangan yang ditorehkan selama masa perkuliahan.
7. Jasmine Naila Rahman, sahabat baik penulis, yang tawa, senyum, dan kehadirannya memberikan semangat dan dukungan di tengah padatnya proses penyusunan skripsi ini



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

8. Seluruh dosen, staf, dan pihak lain yang tidak dapat disebutkan satu per satu, kontribusi kalian tetap penulis rasakan dan syukuri.

Penulis menyadari skripsi ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat penulis harapkan. Semoga skripsi ini dapat bermanfaat bagi pembaca dan berkontribusi bagi pengembangan ilmu pengetahuan di bidang abstraksi.

Depok, 07 Juni 2026

Muhammad Abian Abdi Pratama





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Muhammad Abian Abdi Pratama

NIM : 2207421039

Jurusan/Program Studi : T.Informatika dan Komputer / T. Multimedia dan Jaringan
Demi pengembangan ilmu pengetahuan , menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

IMPLEMENTASI ABSTRACTION LAYER UNTUK PROPAGASI DAFTAR BLOKIR DINAMIS PADA INFRASTRUKTUR FIREWALL MULTI-VENDOR

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 07, Juni, 2026

Yang menyatakan



Muhammad Abian Abdi Pratama



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Implementasi *Abstraction Layer* Untuk Propagasi Daftar Blokir Dinamis Pada Infrastruktur *Firewall Multi-Vendor*

Abstrak

Penelitian ini mengatasi kendala manajemen manual dan masalah heterogenitas pada infrastruktur firewall multi-vendor (pfSense dan OPNsense) yang memicu keterlambatan propagasi aturan keamanan. Solusi yang diusulkan adalah perancangan *Abstraction Layer* berbasis pola desain *Factory* dan *Adapter* sebagai jembatan komunikasi vendor-agnostic. Sistem ini mengimplementasikan mekanisme eksekusi paralel menggunakan *thread pool* untuk meminimalkan latensi distribusi daftar blokir dinamis, serta mengamankan kredensial akses menggunakan enkripsi AES-128-CBC. Pengujian dilakukan secara empiris dengan membandingkan metode eksekusi paralel dan sekuensial pada berbagai variasi beban *payload IP* dan batasan *bandwidth*. Hasil pengujian menunjukkan metode paralel berhasil mereduksi latensi propagasi secara signifikan dengan kestabilan yang tinggi. Pada kondisi *bandwidth* terbatas (50 KBps) dengan beban 40 IP, metode paralel mampu mempertahankan *success rate* 100%, sedangkan metode sekuensial menurun hingga 90% akibat akumulasi *queueing delay* yang memicu *timeout*. Arsitektur ini terbukti efektif meningkatkan responsivitas dan keandalan mitigasi ancaman pada lingkungan jaringan terdistribusi.

Kata Kunci: Eksekusi Paralel, Firewall Multi-Vendor, Lapisan Abstraksi, Latensi Propagasi, OPNsense, pfSense.

POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME.....	ii
LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI	
SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	vi
Abstrak.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
BAB I	
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
1.4.1 Tujuan Penelitian.....	4
1.4.2 Manfaat Penelitian.....	4
1.5 Sistematika Penulisan.....	4
BAB II	
TINJAUAN PUSTAKA.....	6
2.1 Konsep Security Orchestration, Automation, and Response (SOAR).....	6
2.2 Firewall sebagai Kendali Akses Jaringan.....	6
2.3 Platform Firewall Open-Source.....	6
2.4. Arsitektur Integrasi Antarmuka Pemrograman Aplikasi (API).....	7
2.5 Pola Desain Perangkat Lunak (Software Design Patterns).....	7
2.6. Konsep Konkurensi dan Eksekusi Paralel.....	8
2.7. Parameter Pengujian Performa.....	8
2.7.1. Latensi Propagasi (Propagation Latency).....	8
2.7.2. Tingkat Keberhasilan (Success Rate).....	9
2.8 Keamanan Data at Rest.....	9
2.9. Tinjauan Penelitian Terdahulu.....	10
2.10 Tantangan Teknis Implementasi.....	12
BAB III	
METODE PENELITIAN.....	13
3.1 Rancangan Penelitian.....	13
3.1.1 Tahapan Penelitian.....	13
3.1.2 Deskripsi Objek Penelitian.....	14
3.2 Sumber Data.....	15
3.3 Teknik Pengumpulan Data.....	16



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB IV

HASIL DAN PEMBAHASAN..... 17

4.1 Analisis Kebutuhan.....	17
4.1.1 Kebutuhan Sistem.....	17
4.1.2 Kebutuhan Perangkat Keras (Hardware Requirements).....	19
4.1.3 Kebutuhan Perangkat Lunak (Software Requirements).....	20
4.2 Perancangan Sistem Hub Manajemen Firewall.....	22
4.2.1 Perancangan Arsitektur Pola Desain Factory & Adapter.....	23
4.2.2 Perancangan Alur Translasi Protokol.....	26
4.2.3 Perancangan Alur Eksekusi Paralel (Konkurensi).....	29
4.2.4 Perancangan Isolasi Kegagalan (Fault Isolation).....	30
4.2.5 Perancangan Keamanan Kredensial (Data at Rest & In-Memory)....	31
4.3 Implementasi Manajemen Hub Firewall.....	32
4.3.1 Implementasi Lingkungan Pengembangan.....	33
4.3.2 Implementasi Konfigurasi Deployment Berbasis Kontainer (Docker).	35
4.3.3 Implementasi Konfigurasi Database dan Skema ORM.....	39
4.3.4 Implementasi Kriptografi pada Lapisan Model Database.....	44
4.3.5 Implementasi Modul Factory dan Adapter.....	47
4.3.6 Implementasi Translasi Protokol dan Pembaruan Konfigurasi pada pfSense.....	49
4.3.7 Implementasi Translasi Protokol Berbasis Native REST pada OPNsense.....	54
4.3.8 Implementasi Eksekusi Paralel Berbasis Thread Pool untuk Distribusi Perintah Multi-Node.....	60
4.4 Pengujian.....	64
4.4.1 Deskripsi Pengujian.....	65
4.4.2 Prosedur Pengujian.....	67
4.4.3 Data Hasil Pengujian.....	72
4.4.4 Analisis Data / Evaluasi Pengujian.....	80

BAB V

PENUTUP..... 89

5.1 Simpulan.....	89
5.2 Saran.....	89

DAFTAR PUSTAKA..... 91

DAFTAR RIWAYAT HIDUP PENULIS..... 95

LAMPIRAN..... 96



DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	13
Gambar 3.2 Rancangan arsitektur sistem yang diusulkan.....	14
Gambar 4.1 Arsitektur Makro Hub Manajemen Firewall.....	22
Gambar 4.2 Class Diagram FirewallFactory.....	26
Gambar 4.3 Flowchart Alur Translasi Protokol.....	28
Gambar 4.4 Diagram Alur Eksekusi Paralel.....	30
Gambar 4.5 Isolasi Kegagalan.....	31
Gambar 4.6 Alur Enkripsi dan Dekripsi Kredensial.....	32
Gambar 4.7 Diagram Infrastruktur Lingkungan Pengembangan.....	34
Gambar 4.8 Konfigurasi Ketergantungan dan Inisialisasi Layanan Middleware..	36
Gambar 4.9 Implementasi Verifikasi Ketersediaan Variabel Konfigurasi.....	37
Gambar 4.10 Konfigurasi Persistensi Data dan Pemeriksaan Status Database.....	38
Gambar 4.11 Status Container Hub Manajemen Firewall pada lingkungan pengembangan.....	38
Gambar 4.12 Potongan Kode Model FirewallInstance dan Definisi Skema ORM... 41	
Gambar 4.13 Skema Tabel firewall_instances pada database Relasional.....	42
Gambar 4.14 Potongan Kode Model BlockedIP dan Definisi Skema ORM.....	43
Gambar 4.15 Skema Tabel blocked_ips pada Database Relasional.....	44
Gambar 4.16 Potongan Kode Modul Utilitas Kriptografi crypto.py.....	45
Gambar 4.17 Potongan Kode hybrid_property pada Model FirewallInstance.....	46
Gambar 4.18 Hasil Query Tabel firewall_instances yang Menampilkan Ciphertext Kredensial.....	46
Gambar 4.19 Flowchart Logika Instansiasi Klien pada FirewallFactory.....	48
Gambar 4.20 Implementasi Modul Factory pada factory.py.....	48
Gambar 4.21 Antarmuka Pendaftaran Instance Firewall.....	49
Gambar 4.22 Flowchart Alur Translasi Protokol pfSense.....	51
Gambar 4.23 Potongan Kode Metode _get_auth_headers() pada PfSenseClient..	52
Gambar 4.24 Potongan Kode Metode test_connection() pada PfSenseClient.....	52
Gambar 4.25 Potongan Kode Metode update_blocklist() pada PfSenseClient.....	54
Gambar 4.26 Flowchart Alur Translasi Protokol OPNsense.....	55
Gambar 4.27 Potongan Kode Metode _get_credentials() pada OPNsenseClient..	56
Gambar 4.28 Potongan Kode Metode _get_alias_uuid() pada OPNsenseClient..	57
Gambar 4.29 Potongan Kode Metode test_connection() pada OPNsenseClient..	58
Gambar 4.30 Potongan Kode Metode update_blocklist() pada OPNsenseClient..	59
Gambar 4.31 Flowchart Alur Eksekusi Paralel Distribusi Daftar Blokir.....	61
Gambar 4.32 Potongan Kode Fungsi push_blocklist_to_firewalls().....	62
Gambar 4.33 Potongan Kode Fungsi _update_single_firewall().....	63
Gambar 4.34 Potongan Kode Agregasi Hasil dan Penanganan Exception.....	64



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.35 Grafik Perbandingan Rata-rata Latensi Propagasi antara Metode Paralel dan Sekuensial.....	81
Gambar 4.36 Grafik Perbandingan Rata-rata Penggunaan CPU antara Metode Paralel dan Sekuensial.....	83
Gambar 4.37 Grafik Perbandingan Rata-rata Penggunaan RAM antara Metode Paralel dan Sekuensial.....	84
Gambar 4.38 Grafik Perbandingan Latensi Metode Paralel pada Kondisi Bandwidth Terbatas.....	85
Gambar 4.39 Grafik Perbandingan Latensi Metode Sekuensial pada Kondisi Bandwidth Terbatas.....	86
Gambar 4.40 Perbandingan <i>Success Rate</i> Paralel dan Sekuensial di <i>Bandwidth</i> 50KBps.....	87





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 2.1. Tinjauan Penelitian Terdahulu.....	10
Tabel 2.2. Perbandingan Mekanisme API pfSense dan OPNsense.....	12
Tabel 4.1 Kebutuhan Perangkat Keras.....	19
Tabel 4.2 Kebutuhan Perangkat Lunak.....	21
Tabel 4.3 Perbandingan Parameter Komunikasi API Vendor.....	27
Tabel 4.4 Skema Tabel firewall_instances.....	41
Tabel 4.5 Skema Tabel blocked_ips.....	43
Tabel 4.6 Skenario Pengujian Fitur.....	69
Tabel 4.7 Parameter Pengujian Kinerja.....	71
Tabel 4.8 Parameter Pengujian Kondisi Jaringan.....	72
Tabel 4.9 Hasil Pengujian Fitur.....	73
Tabel 4.10 Hasil Pengujian Kinerja Metode Paralel.....	75
Tabel 4.11 Hasil Pengujian Kinerja Metode Sekuensial.....	76
Tabel 4.12 Hasil Penggunaan Sumber Daya Metode Paralel.....	77
Tabel 4.13 Hasil Penggunaan Sumber Daya Metode Sekuensial.....	77
Tabel 4.14 Hasil Pengujian Kondisi 50 KBps.....	78
Tabel 4.15 Hasil Pengujian Kondisi 100 KBps.....	79
Tabel 4.16 Hasil Pengujian Kondisi 150 KBps.....	80
Tabel 4.17 Hasil Pengujian Kondisi 200 KBps.....	80

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Keamanan siber modern menghadapi tantangan serius akibat keterbatasan respons manual. Laporan industri terbaru mengungkapkan bahwa elemen manusia menjadi titik lemah dominan, dengan 26% insiden disebabkan oleh kesalahan langsung (IBM, 2025) dan keterlibatan manusia tercatat dalam 60% total pelanggaran keamanan (Verizon, 2025). Data ini menegaskan bahwa mekanisme pertahanan konvensional tidak lagi memadai untuk menangani volume dan kompleksitas ancaman yang terus meningkat secara efisien.

Kesalahan manusia (*human error*) tersebut seringkali dipicu oleh beban kerja yang tidak rasional. Keterbatasan tersebut terlihat nyata pada operasional *Security Operations Centre* (SOC), di mana volume peringatan yang tinggi memicu kelelahan analis (*alert fatigue*). Studi komprehensif oleh Tariq et al. (2025) mengonfirmasi bahwa 51% tim SOC merasa kewalahan, 63% praktisi mengalami *burnout*, dan lebih dari 25% waktu analis tersita oleh *false positive*. Untuk mengatasi inefisiensi ini, otomatisasi melalui *Security Orchestration, Automation, and Response* (SOAR) menjadi kebutuhan fungsional, dengan *firewall* sebagai komponen penegakan kebijakan utamanya (Gartner, n.d.).

Meskipun otomasi respons menawarkan efisiensi, implementasinya terhambat oleh heterogenitas perangkat pada lingkungan *multi-vendor*, di mana perbedaan arsitektur dan API antar *firewall* menghalangi orkestrasi seragam (Lin, 2024). Kendala ini terlihat jelas pada pengelolaan *firewall* pfSense dan OPNsense yang memiliki mekanisme teknis berbeda (OPNsense.org, n.d; Hendrickson, n.d.). Permasalahan tersebut tercermin secara nyata pada kondisi operasional PT. XYZ, di mana pengelolaan manual terhadap kombinasi kedua perangkat ini tidak hanya memperlambat respons insiden tetapi juga meningkatkan risiko inkonsistensi konfigurasi di seluruh jaringan (Jorepalli dan Bairy, 2024).

Praktik mitigasi ancaman melalui daftar blokir statis terbukti tidak lagi efektif menghadapi serangan modern yang dinamis, sehingga menuntut pembaruan



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

aturan yang cepat dan berulang. Kompleksitas ini bertambah pada lingkungan terdistribusi, di mana eksekusi konfigurasi secara sekuensial menyebabkan keterlambatan propagasi kebijakan dan menciptakan ketidaksinkronan status perlindungan antar segmen jaringan. Keterlambatan ini tidak hanya memperpanjang waktu respons, tetapi juga membuka peluang eksploitasi pada titik jaringan yang belum menerima pembaruan aturan keamanan (Imoukhuede et al., 2025; Nelson et al., 2025).

Berdasarkan kondisi tersebut, diperlukan pendekatan yang melampaui metode konfigurasi manual maupun skrip otomatisasi sederhana. Penelitian ini menawarkan kebaruan melalui penerapan lapisan abstraksi (*abstraction layer*) berbasis pola desain *Factory* yang berfungsi sebagai jembatan komunikasi *vendor-agnostic* antara sistem orkestrasi terpusat dan perangkat *firewall multi-vendor*. Berbeda dengan pendekatan konvensional yang sering kali terkunci pada satu vendor (*vendor lock-in*), mekanisme ini memungkinkan instruksi mitigasi diterjemahkan secara otomatis ke dalam format protokol spesifik masing-masing perangkat (pfSense dan OPNsense) secara *seamless*. Dengan menempatkan kompleksitas teknis pada lapisan terpisah, proses pengelolaan kebijakan pemblokiran dapat dilakukan secara terkoordinasi dan konsisten dalam satu alur orkestrasi paralel.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, perumusan masalah dalam penelitian ini dirumuskan sebagai berikut:

1. Bagaimana merancang lapisan abstraksi yang menstandarisasi mekanisme komunikasi antara *firewall multi-vendor* yang memiliki perbedaan protokol autentikasi dan struktur *Application Programming Interface*?
2. Bagaimana mengimplementasikan mekanisme eksekusi paralel pada lapisan abstraksi untuk mendistribusikan daftar blokir dinamis pada infrastruktur *firewall multi-vendor*?
3. Bagaimana perbandingan tingkat latensi propagasi IP *blocklist* antara metode eksekusi paralel berbasis lapisan abstraksi dengan metode konfigurasi sekuensial tradisional?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Penelitian ini berangkat dari hipotesis bahwa penerapan arsitektur berbasis abstraksi dan eksekusi paralel mampu mereduksi latensi propagasi secara signifikan dibandingkan metode sekuensial, sehingga meningkatkan responsivitas sistem terhadap ancaman keamanan.

1.3 Batasan Masalah

Supaya penelitian ini tetap terarah dan fokus pada tujuan utama, penulis menetapkan batasan masalah sebagai berikut:

1. Platform Target: Implementasi lapisan abstraksi difokuskan pada dua platform *firewall open-source* yang merepresentasikan perbedaan model integrasi dan penggunaan aktual pada infrastruktur PT. XYZ, yaitu pfSense (menggunakan *package* API pihak ketiga) dan OPNsense (menggunakan API bawaan/*native*).
2. Lingkup Fitur: Fungsi orkestrasi dibatasi pada manajemen objek Alias (daftar alamat IP) untuk keperluan pemblokiran dan pembukaan blokir. Manajemen konfigurasi infrastruktur inti seperti NAT, VLAN, atau VPN berada di luar lingkup penelitian sejalan dengan kebijakan keamanan (*security policy*) dan batasan akses (*least privilege*) yang diterapkan pada lingkungan operasional PT. XYZ.
3. Fokus Pengembangan: Penelitian ini secara spesifik hanya mencakup pengembangan lapisan abstraksi *firewall* dan mekanisme propagasi data. Adapun modul deteksi anomali (Wazuh) dan logika otomatisasi (*Autoblock*) merupakan bagian dari sistem terintegrasi yang dievaluasi secara terpisah dalam lingkup skripsi kelompok ini.
4. Lingkungan Pengujian: Pengujian dilaksanakan dalam lingkungan pengembangan terkontrol pada infrastruktur *multi-node firewall* milik PT. XYZ, dengan jumlah *node* yang menyesuaikan kebijakan akses dan ketersediaan infrastruktur saat penelitian dilaksanakan.
5. Manajemen kunci enkripsi (*Master Key*) difokuskan pada pengamanan tingkat sistem operasi (*environment variable*) pada server lokal, dan tidak mencakup integrasi dengan layanan *Key Management System* (KMS) eksternal.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.4 Tujuan dan Manfaat

1.4.1 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun API berbasis pola desain *Factory* yang berfungsi sebagai jembatan komunikasi (*bridge*) antara sistem manajemen pusat dengan perangkat *multi-vendor*.
2. Mengimplementasikan mekanisme konkurensi (paralel) pada proses distribusi daftar blokir untuk meminimalkan latensi propagasi yang umumnya meningkat secara linier pada konfigurasi *multi-vendor*.
3. Menguji keefektifitas lapisan abstraksi dalam mewujudkan integrasi *multi-vendor* tanpa mengubah infrastruktur dasar jaringan yang telah ada.

1.4.2 Manfaat Penelitian

Penelitian ini diharapkan memberikan kontribusi signifikan bagi organisasi dan industri dengan menyediakan solusi orkestrasi keamanan yang bersifat *vendor-agnostic*, sehingga menghilangkan ketergantungan pada satu merek tertentu dan mempercepat waktu mitigasi serangan. Dari sisi akademis dan pengembangan teknologi, penelitian ini menawarkan referensi teknis mengenai penerapan pola desain perangkat lunak (*Factory* dan *Adapter*) dalam konteks otomasi jaringan, serta menyediakan pustaka kode modular yang dapat diperluas untuk mendukung integrasi vendor *firewall* baru di masa depan tanpa perlu merombak arsitektur inti sistem.

1.5 Sistematika Penulisan

Laporan skripsi ini disusun dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang masalah mengenai tantangan *multi-vendor* dan latensi pada manajemen *firewall*, perumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Bab ini memaparkan landasan teori yang menjadi fondasi penelitian, meliputi konsep *Security Orchestration, Automation, and Response* (SOAR), karakteristik *firewall* pfSense dan OPNsense, serta Pola Desain Perangkat Lunak (*Software*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Design Patterns) seperti *Factory* dan *Adapter*. Secara khusus, bab ini juga menyajikan sub-bab mengenai tinjauan penelitian terdahulu untuk memetakan posisi penelitian (*state of the art*) dan menganalisis kesenjangan penelitian (*research gap*) yang menjadi dasar kebaruan solusi yang ditawarkan.

BAB III METODE PENELITIAN

Bab ini menguraikan tahapan penelitian mulai dari analisis kebutuhan, perancangan arsitektur sistem menggunakan diagram alir (*flowchart*) dan diagram blok sistem, desain lapisan abstraksi, hingga skenario pengujian performa propagasi data.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi kode program, demonstrasi fungsionalitas modul integrasi *multi-vendor*, serta analisis data hasil pengujian latensi propagasi untuk memvalidasi efektivitas solusi yang ditawarkan.

BAB V PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran-saran untuk pengembangan sistem lebih lanjut di masa mendatang.

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Simpulan

Sistem *Abstraction Layer* terbukti efektif menstandarisasi komunikasi *firewall multi-vendor* (pfSense dan OPNsense) tanpa merombak infrastruktur dasar jaringan karena mengimplementasikan pola desain *Factory* dan *Adapter* yang mampu menerjemahkan perbedaan API secara otomatis. Implementasi sistem ini beroperasi dengan sangat efisien karena didukung oleh manajemen sumber daya yang ringan, hanya mengonsumsi RAM di kisaran 119–123 MB dan utilisasi CPU sebesar 12,9%–14,6%, sembari tetap menjaga keamanan kredensial melalui enkripsi AES-128-CBC.

Dari sisi performa, mekanisme eksekusi paralel terbukti sangat efektif mereduksi latensi propagasi daftar blokir karena instruksi didistribusikan secara serentak ke seluruh perangkat, sehingga memangkas akumulasi waktu antrean (*queueing delay*). Hal ini dibuktikan oleh catatan rata-rata latensi paralel yang sangat rendah di angka 4,90–6,61 detik pada distribusi beban 10–40 IP. Selain itu, arsitektur ini terbukti efektif menjaga keandalan sistem pada kondisi jaringan ekstrem (*bandwidth* 50 KBps, beban 40 IP) karena *thread* yang independen mampu mempertahankan *success rate* mutlak 100% dengan latensi 9,71 detik, mengungguli metode sekuensial yang rentan memicu *timeout* dengan lonjakan latensi hingga 19,93 detik dan penurunan *success rate* menjadi 90%.

5.2 Saran

Untuk pengembangan dan penyempurnaan sistem Hub Manajemen *Firewall* ini lebih lanjut di masa mendatang, penulis menyarankan beberapa poin pengembangan sebagai berikut:

1. Ekspansi vendor: Karena arsitektur sistem sudah mengikuti *Open-Closed Principle*, pengembangan selanjutnya dapat difokuskan pada penambahan *Adapter* untuk vendor *firewall* komersial lain seperti Fortinet, Palo Alto Networks, atau Cisco ASA tanpa perlu mengubah logika inti sistem.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2. Manajemen kredensial: Sistem enkripsi saat ini masih mengandalkan *environment variable server* lokal untuk menyimpan kunci *master*. Untuk deployment skala produksi, disarankan mengintegrasikan solusi *Key Management System* eksternal seperti HashiCorp Vault, AWS KMS, atau Azure Key Vault guna pengelolaan siklus hidup kunci yang lebih aman.
3. Konfigurasi jaringan Docker: Konfigurasi *network_mode: host* yang digunakan saat ini sudah memadai untuk pengembangan lokal, namun pada lingkungan produksi sebaiknya diganti dengan *custom bridge network* yang dikombinasikan dengan aturan *iptables* yang ketat untuk meminimalkan *attack surface* kontainer.
4. Perluasan fitur orkestrasi: Penelitian ini membatasi cakupan pada manajemen Alias untuk pemblokiran IP. Pengembangan selanjutnya dapat memperluas fitur ke manajemen aturan yang lebih kompleks seperti konfigurasi *Port Forwarding*, segmentasi VLAN, atau manajemen profil VPN.

**POLITEKNIK
NEGERI
JAKARTA**

DAFTAR PUSTAKA

- Ahmed, D. dan Amin, R.H. 2022. Multi-layered firewall to mitigate the impact of Distributed Denial of Service on a network. *Passer journal of basic and applied sciences*, 4(Special issue), pp.51–62. doi:<https://doi.org/10.24271/psr.2022.160803>.
- Atri, P. 2024. Enhancing Big Data Security through Comprehensive Data Protection Measures: A Focus on Securing Data at Rest and In-Transit. *International journal of computing and engineering*, 5(4), pp.44–55. doi:<https://doi.org/10.47941/ijce.1920>.
- Chao, C.-S. 2023. Developing a Feasible Firewall System with Parallel Rule Allocation Optimization for High Service Availability under Large-Scale Network Attacks. *IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp.56–60. doi:<https://doi.org/10.1109/ecice59523.2023.10383082>.
- Costa, M.A., Costa, Y.M., Almeida, Y.O., Cardoso, F.J. dan Gomes, R.L. 2024. Connection Management Using Automated Firewall Based on Threat Intelligence. *LANC '24: Proceedings of the 2024 Latin America Networking Conference*, pp.32–37. doi:<https://doi.org/10.1145/3685323.3685331>.
- Damanik, H.A. dan Anggraeni, M. 2025. Sistem Deteksi dan Pencegahan System Attack pada Infrastruktur Jaringan Menggunakan Realtime Honeypot dan Automatic Iptables. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 12(6). doi:<https://doi.org/10.25126/jtiik.2025126>.
- Gartner. n.d.. Definition of Security Orchestration, Automation and Response (SOAR) - Gartner Information Technology Glossary. : <https://www.gartner.com/en/information-technology/glossary/security-orchestration-automation-response-soar> [9 Januari 2026].
- Hendrickson, J. (n.d.). Home - pfSense REST API Guide. *Pfrest.org*. : <https://pfrest.org/> [8 Januari 2026].
- Hugo, V. dan Proust, M. 2022. Integrating Firewalls with SIEM and SOAR Platforms for Automated Threat Response. *International Journal of Trend in Scientific Research and Development*, [online] 6(3). : <https://www.ijtsrd.com/computer-science/other/49651/integrating-firewalls-wi>



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

th-siem-and-soar-platforms-for-automated-threat-response/victor-hugo [15 Januari 2026].

IBM. 2025. *Cost of a Data Breach Report 2025: The AI Oversight Gap*. [online] IBM. : <https://www.ibm.com/reports/data-breach> [8 Januari 2026].

Imoukhuede, A.B., Mahmoud, A.H., Sheltami, T.R. dan Barnawi, A.Y. 2025. Optimization of network device *hardening* in a *multivendor* environment. *Scientific Reports*, 15(1). doi:<https://doi.org/10.1038/s41598-025-97894-4>.

Ivanov, I.G., Hristov, G.V. dan Stoykova, V.D. 2021. Algorithms for *optimizing* packet *propagation* latency in *software-defined networks*. *IOP Conference Series Materials Science and Engineering*, 1031(1), pp.012072–012072. doi:<https://doi.org/10.1088/1757-899x/1031/1/012072>.

Jorepalli, S.K.R. dan Bairy, V. 2024. *International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Leveraging Network Automation with Python, Terraform, and Ansible to Enhance Security and Operational Efficiency in Large- Scale Networks*. [online] ResearchGate. https://www.researchgate.net/publication/390209265_Leveraging_Network_Automation_with_Python_Terraform_and_Ansible_to_Enhance_Security_and_Operational_Efficiency_in_Large-Scale_Networks [13 Januari 2026].

Kim, T.N., Tri, T.N., Nguyen, L.T. dan Truong, D.T. 2022. A COMBINATION OF THE *INTRUSION DETECTION SYSTEM* AND THE *OPEN-SOURCE FIREWALL* USING *PYTHON LANGUAGE*. *International journal of Computer Networks & Communications*, 14(1), pp.59–69. doi:<https://doi.org/10.5121/ijcnc.2022.14104>.

Kreculj, D., Dihovični, Đ., Ratković Kovačević, N., Gaborov, M. dan Zajeganović, M. 2023. pfSense Router and *Firewall Software*. *Proceedings of the International Scientific Conference - Sinteza 2023*. [online] doi:<https://doi.org/10.15308/sinteza-2023-132-137>.

Lin, H. 2024. *Journal of Telecommunication System and Management Commentary Challenges and Solutions in Multi-site Networking*. doi:<https://doi.org/10.37421/2167-0919.2024.13.417>.

Nelson, A., Rekhi, S., Souppaya, M. dan Scarfone, K. 2025. *Incident response recommendations and considerations for cybersecurity risk management : NIST Special Publication 800*. doi:<https://doi.org/10.6028/nist.sp.800-61r3>.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

OPNsense. 2025. White paper: A *full feature overview* of OPNsense® including *high-end features such as high availability, traffic shaping, intrusion detection and easy OpenVPN client setup*. [online] : <https://opnsense.org/files/OPNsense-Whitepaper-features.pdf#:~:text=The%20complete%204th%20Edition%20of,Online%20Documentation> [15 Januari 2026].

Opnsense.org. (n.d.). *API Reference — OPNsense documentation*. [online] : <https://docs.opnsense.org/development/api.html> [8 Januari 2026].

Patel, H. 2024. A research paper on *software design patterns*. *World Journal of Advanced Engineering Technology and Sciences*, [online] 13(1), pp.803–813. doi:<https://doi.org/10.30574/wjaets.2024.13.1.0499>.

Sari, R., Cakraningrat, M.S. dan Zain, A.R. 2023. Implementasi Enkripsi Dekripsi Paket Data pada Rancang Bangun Smart Home Menggunakan Protokol MQTT. *MULTINETICS: Jurnal Multimedia Networking Informatics*, 8(2), pp.168–176. doi:<https://doi.org/10.32722/multinetics.v8i2.4110>.

Sholihan, A.W.N., Mukti, A.R., Suryayusra, S. dan Dasmen, R.N. 2023. Implementation of Network Security and Anticipating Attackers Using pfSense Firewall. *CESS (Journal of Computer Engineering, System and Science)*, [online] 8(1), pp.175–183. doi:<https://doi.org/10.24114/cess.v8i1.42377>.

Swanzy, P.N., Abukari, A.M. dan Ansong, E.D. 2024. Data Security Framework for Protecting Data in Transit and Data at Rest in the Cloud. *Current journal of applied science and technology*, 43(6), pp.61–77. doi:<https://doi.org/10.9734/cjast/2024/v43i64387>.

Tariq, S., Chhetri, M.B., Nepal, S. dan Paris, C. 2025. *Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities*. *ACM Computing Surveys*, 57(9). doi:<https://doi.org/10.1145/3723158>.

Togay, C., Kasif, A., Catal, C. dan Tekinerdogan, B. 2021. A Firewall Policy Anomaly Detection Framework for Reliable Network Security. *IEEE Transactions on Reliability*, 71(1), pp.1–9. doi:<https://doi.org/10.1109/tr.2021.3089511>.

Verizon. 2025. *2025 Data Breach Investigations Report*. [online] Verizon Business. t:



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

<https://www.verizon.com/business/resources/reports/2025-dbir-data-breach-investigations-report.pdf>.

Yuzhen, Y., Xiaoliang, F. dan Wei, Z. 2025. *Automation of Abnormal IP Blocking in Security Systems Using OCR-Driven Web Interaction and Real-Time Alert Integration. Journal of Cyber Security and Mobility.* doi:<https://doi.org/10.13052/jcsm2245-1439.1449>.



DAFTAR RIWAYAT HIDUP PENULIS



Muhammad Abian Abdi Pratama, lahir di Depok, 18 Juni 2004. Lulus dari SMK Wira Buana 2, Citayam, Bogor dari jurusan Teknik Komputer dan Jaringan. Sekarang sedang melanjutkan pendidikan di Politeknik Negeri Jakarta, Jurusan Teknik Informatika dan Komputer, dengan Program Studi Teknik Multimedia dan Jaringan

© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

LAMPIRAN 1: Kode Sumber Antarmuka Dasar (*base_client.py*)

```

"""
Mendefinisikan antarmuka dasar (kontrak) untuk semua klien firewall.
"""
from abc import ABC, abstractmethod
from typing import Set, Tuple
from flask import current_app

from app.models.firewall import FirewallInstance

class BaseFirewallClient(ABC):
    """
    Kelas dasar abstrak untuk klien firewall.
    Setiap klien konkret (mis., OPNsenseClient) HARUS mengimplementasikan
    metode-metode yang didekorasi dengan @abstractmethod.
    """

    def __init__(self, instance: FirewallInstance):
        self.instance = instance
        self.logger = current_app.logger

    @abstractmethod
    def test_connection(self) -> Tuple[bool, str]:
        """
        Menguji konektivitas dan validitas konfigurasi ke instance firewall.
        Harus mengembalikan tuple: (sukses: bool, pesan: str).
        """
        pass

    @abstractmethod
    def update_blocklist(self, ip_set: Set[str]) -> Tuple[bool, str]:
        """
        Memperbarui alias blocklist di firewall dengan set IP yang diberikan.
        Harus mengembalikan tuple: (sukses: bool, pesan: str).
        """
        pass

```

LAMPIRAN 2: Kode Sumber Pabrik Klien Dinamis (*factory.py*)

```

"""
Factory untuk membuat instance klien firewall yang sesuai.

Pola desain Factory ini memusatkan logika pembuatan objek, memisahkan
kode service layer dari pengetahuan tentang kelas klien mana yang harus dibuat.
"""

from app.models.firewall import FirewallInstance, FirewallType
from .base_client import BaseFirewallClient
from .opnsense_client import OPNsenseClient
from .pfsense_client import PfSenseClient

def get_firewall_client(instance: FirewallInstance) -> BaseFirewallClient:
    """
    Pabrik yang membangun dan mengembalikan klien firewall yang tepat
    berdasarkan jenis instance.

    Args:
        instance: Objek model FirewallInstance.

    Returns:
        Sebuah instance dari kelas klien yang sesuai (mis., OPNsenseClient).
    """

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

Raises:
    NotImplementedError: Jika jenis firewall tidak didukung.
"""
if instance.firewall_type == FirewallType.OPNSENSE:
    return OPNSenseClient(instance)
elif instance.firewall_type == FirewallType.PFSENSE:
    return PfSenseClient(instance)
else:
    # Ini penting untuk masa depan; jika kita menambahkan jenis baru ke Enum
    # tetapi lupa menambahkannya di sini, aplikasi akan gagal dengan cepat
    # dan jelas.
    raise NotImplementedError(
        f"Firewall type '{instance.firewall_type}' is not supported by the
factory.")
)

```

LAMPIRAN 3: Kode Sumber Klien OPNSense (opnsense_client.py)

```

"""
Implementasi klien firewall spesifik untuk OPNSense.
Menggunakan endpoint legacy /api/firewall/alias/*.
"""
import requests
from typing import Set, Tuple
from flask import current_app

from .base_client import BaseFirewallClient

class OPNSenseClient(BaseFirewallClient):
    """Klien untuk berinteraksi dengan OPNSense Firewall API (model Key/Secret)."""

    def _get_credentials(self) -> Tuple[requests.auth.AuthBase | None, str]:
        """Mendapatkan kredensial Basic Auth (Key/Secret) untuk OPNSense."""
        key = current_app.config["ENCRYPTION_KEY"]
        creds = self.instance.get_opnsense_credentials(key)
        if not creds:
            msg = (
                f"Could not decrypt OPNSense API credentials for
'{self.instance.name}'"
            )
            self.logger.error(msg)
            return None, msg

        api_key, api_secret = creds
        return requests.auth.HTTPBasicAuth(api_key, api_secret), ""

    def _get_alias_uuid(self) -> Tuple[str | None, str]:
        """Mendapatkan UUID dari alias yang dikonfigurasi dari OPNSense API."""
        auth, error_msg = self._get_credentials()
        if not auth:
            return None, error_msg

        try:
            search_url = f"{self.instance.url}/api/firewall/alias/get"
            response = requests.get(search_url, auth=auth, verify=False, timeout=10)

            if response.status_code == 200 and "failed" in response.text:
                msg = "Authentication failed. Please check API Key and Secret."
                self.logger.warning(f"{self.instance.name}: {msg}")
                return None, msg

            response.raise_for_status()
            data = response.json()

            aliases = data.get("alias", {}).get("aliases", {}).get("alias", {})

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

for uuid, alias_details in aliases.items():
    if alias_details.get("name") == self.instance.alias_name:
        return uuid, "Alias found."

    msg = f"Connection successful, but alias '{self.instance.alias_name}' was
not found on the firewall."
    self.logger.warning(f"{self.instance.name}: {msg}")
    return None, f"{msg} Please create it on OPNsense first."

except requests.exceptions.HTTPError as e:
    if e.response.status_code == 404:
        return (
            None,
            f"Endpoint not found ({e.response.status_code}). Check URL and
OPNsense API version.",
        )
    return None, f"HTTP Error: {e}"
except requests.exceptions.RequestException as e:
    msg = (
        f"Connection failed. Check the URL and network connectivity. Error:
{e}"
    )
    self.logger.error(f"{self.instance.name}: {msg}")
    return None, msg
except Exception as e:
    msg = f"An unknown error occurred: {e}"
    self.logger.error(f"{self.instance.name}: {msg}", exc_info=True)
    return None, msg

def test_connection(self) -> Tuple[bool, str]:
    """Menguji koneksi dengan mencoba mengambil UUID alias."""
    self.logger.info(f"Testing OPNsense connection for '{self.instance.name}'")
    uuid, error_message = self._get_alias_uuid()
    if uuid:
        return (
            True,
            f"Successfully connected and found alias
'{self.instance.alias_name}'.",
        )
    else:
        return False, error_message

def update_blocklist(self, ip_set: Set[str]) -> Tuple[bool, str]:
    """Memperbarui alias blocklist di instance OPNsense ini."""
    self.logger.info(
        f"Starting blocklist update for OPNsense instance '{self.instance.name}'"
    )

    auth, error_msg = self._get_credentials()
    if not auth:
        return False, error_msg

    alias_uuid, error_msg = self._get_alias_uuid()
    if not alias_uuid:
        return False, f"Cannot update blocklist: {error_msg}"

    content = "\n".join(sorted(list(ip_set))) if ip_set else ""
    api_url = f"{self.instance.url}/api/firewall/alias/setItem/{alias_uuid}"
    payload = {
        "alias": {
            "content": content,
            "enabled": "1",
            "name": self.instance.alias_name,
            "type": "host",
            "description": "Blocked IPs from Firewall Management Hub",
        }
    }

    try:
        response = requests.post(

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        api_url, auth=auth, json=payload, verify=False, timeout=15
    )
    response.raise_for_status()
    if response.json().get("result") != "saved":
        return False, f"API reported failure: {response.text}"

    reconfigure_url = f"{self.instance.url}/api/firewall/alias/reconfigure"
    reconfigure_response = requests.post(
        reconfigure_url, auth=auth, json={}, verify=False, timeout=20
    )
    reconfigure_response.raise_for_status()

    if reconfigure_response.json().get("status") == "ok":
        return True, "Firewall updated successfully."
    else:
        return False, f"Reconfigure command failed: {reconfigure_response.text}"

except requests.exceptions.RequestException as e:
    return False, f"Network error: {e}"

```

LAMPIRAN 4: Kode Sumber Klien pfSense (*pfsense_client.py*)

```

"""
Implementasi klien firewall spesifik untuk pfSense V2 REST API.
Disesuaikan untuk otentikasi X-API-Key dan pembaruan batch.
"""

import requests
from typing import Set, Tuple, Dict, Any, List
from flask import current_app

from .base_client import BaseFirewallClient

class PfSenseClient(BaseFirewallClient):
    """Klien untuk berinteraksi dengan pfSense V2 REST API."""

    def _get_auth_headers(self) -> Tuple[Dict[str, str] | None, str]:
        """Membuat header otentikasi X-API-Key."""
        key = current_app.config["ENCRYPTION_KEY"]

        # --- FIX 4: Pengecekan Kredensial yang Lebih Granular ---
        # 1. Cek apakah field terenkripsi di DB adalah None.
        if not self.instance._encrypted_pfsense_api_key:
            msg = f"PfSense API Key for '{self.instance.name}' is missing in the database."
            self.logger.error(msg)
            return None, msg

        # 2. Coba dekripsi. Ini akan menguji apakah kunci master benar.
        api_key = self.instance.get_pfsense_api_key(key)

        if not api_key:
            # Jika field ada tapi dekripsi gagal.
            msg = f"Could not decrypt API Key for pfSense instance '{self.instance.name}'. Check the application's ENCRYPTION_KEY."
            self.logger.critical(msg)
            # Gunakan CRITICAL untuk kegagalan enkripsi/dekripsi
            return None, msg

        # -----

        return {"X-API-Key": api_key, "Content-Type": "application/json"}, ""

    def _get_all_aliases(self) -> Tuple[List[Dict[str, Any]] | None, str]:
        """Helper untuk mengambil SEMUA alias dari firewall."""
        headers, error_msg = self._get_auth_headers()

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

if not headers:
    return None, error_msg

api_url = f"{self.instance.url}/api/v2/firewall/aliases"
try:
    response = requests.get(api_url, headers=headers, verify=False, timeout=15)
    response.raise_for_status()
    data = response.json()
    if data.get("code") == 200 and "data" in data:
        return data["data"], "Successfully fetched all aliases."
    else:
        return None, data.get("message", "Failed to fetch aliases.")
except requests.exceptions.RequestException as e:
    return None, f"Connection error while fetching aliases: {e}"

# === IMPLEMENTASI YANG HILANG DITAMBAHKAN KEMBALI ===
def test_connection(self) -> Tuple[bool, str]:
    """Menguji koneksi dengan mencari alias target di antara semua alias."""
    self.logger.info(f"Testing pfSense connection for '{self.instance.name}'")
    all_aliases, message = self._get_all_aliases()
    if all_aliases is None:
        # Jika _get_all_aliases gagal, itu sudah mengembalikan pesan kesalahan yang
        relevan.
        return False, message

    target_name = self.instance.alias_name
    if any(alias.get("name") == target_name for alias in all_aliases):
        return True, f"Successfully connected and found alias '{target_name}'."

    return False, f"Connected, but alias '{target_name}' was not found."

def update_blocklist(self, ip_set: Set[str]) -> Tuple[bool, str]:
    """Memperbarui alias dengan mengambil, memodifikasi, dan mengirim kembali semua
    alias."""
    self.logger.info(f"Starting blocklist update for '{self.instance.name}'")
    all_aliases, message = self._get_all_aliases()
    if all_aliases is None:
        return False, f"Cannot update: {message}"

    target_name = self.instance.alias_name
    if not any(alias.get("name") == target_name for alias in all_aliases):
        return False, f"Cannot update: Alias '{target_name}' not found on
        firewall."

    for alias in all_aliases:
        if alias.get("name") == target_name:
            alias["address"] = sorted(list(ip_set))
            break

    headers, error_msg = self._get_auth_headers()
    if not headers:
        return False, error_msg

    api_url = f"{self.instance.url}/api/v2/firewall/aliases"
    payload = all_aliases

    try:
        response = requests.put(
            api_url, headers=headers, json=payload, verify=False, timeout=30
        )
        response.raise_for_status()
        data = response.json()

        if data.get("code") == 200 and data.get("status") == "ok":
            apply_url = f"{self.instance.url}/api/v2/firewall/apply"
            apply_response = requests.post(
                apply_url, headers=headers, json={}, verify=False, timeout=30
            )
            apply_response.raise_for_status()
            if apply_response.json().get("code") == 200:

```



```

        return True, "pfSense blocklist updated and changes applied."
    else:
        return (
            False,
            f"Update sent, but failed to apply changes:
{apply_response.text}",
        )
    else:
        return False, data.get(
            "message", f"API returned non-OK status: {response.text}"
        )
except requests.exceptions.RequestException as e:
    self.logger.error(f"API Error during pfSense update: {e}", exc_info=True)
    return False, f"API Error during update: {e}"

```

LAMPIRAN 5: Berkas Konfigurasi Kontainerisasi (docker-compose.yml)

```

version: '3.8'

services:
  postgres:
    image: postgres:15
    container_name: db_postgres_storage
    environment:
      POSTGRES_DB: orm_db
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: <SECURE_POSTGRES_PASSWORD> # Diubah demi keamanan (NDA)
    ports:
      - "5432:5432"
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U $$POSTGRES_USER -d $$POSTGRES_DB -h
localhost"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_period: 10s
    volumes:
      - postgres_data:/var/lib/postgresql/data
    restart: unless-stopped

  redis:
    image: redis:7-alpine
    container_name: db_redis_cache
    ports:
      - "6379:6379"
    volumes:
      - redis_data:/data
    restart: unless-stopped

  clickhouse:
    image: clickhouse/clickhouse-server:latest
    container_name: db_clickhouse_analytics
    environment:
      - CLICKHOUSE_USER=default
      - CLICKHOUSE_PASSWORD=<SECURE_CLICKHOUSE_PASSWORD> # Diubah demi keamanan (NDA)
      - CLICKHOUSE_DB=default
    ports:
      - "8123:8123"
      - "9000:9000"
    volumes:
      - clickhouse_data:/var/lib/clickhouse/
    ulimits:
      nproc: 65535
      nofile:
        soft: 262144
        hard: 262144
    restart: unless-stopped

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
middleware-app:
  build: .
  container_name: orchestrator_web_app
  restart: unless-stopped
  env_file: .env
  ports:
    - "5002:5001" # Host 5002 -> Container 5001
  depends_on:
    postgres:
      condition: service_healthy
    redis:
      condition: service_started
    clickhouse:
      condition: service_started
  environment:
    DATABASE_URL:
postgresql://postgres:<SECURE_POSTGRES_PASSWORD>@postgres:5432/orm_db
    REDIS_URL: redis://redis:6379/0
    CLICKHOUSE_HOST: clickhouse
  command: >
  sh -c "
    flask db upgrade && \
    python3 scripts/make_dev_cert.py && \
    gunicorn --bind 0.0.0.0:5001 --certfile instance/ssl/devcert.crt --keyfile
instance/ssl/devcert.key run:app
"

middleware-scheduler:
  build: .
  container_name: orchestrator_scheduler_service
  restart: unless-stopped
  env_file: .env
  depends_on:
    postgres:
      condition: service_healthy
    redis:
      condition: service_started
    clickhouse:
      condition: service_started
  command: python scheduler.py
  environment:
    DATABASE_URL:
postgresql://postgres:<SECURE_POSTGRES_PASSWORD>@postgres:5432/orm_db
    REDIS_URL: redis://redis:6379/0
    CLICKHOUSE_HOST: clickhouse

volumes:
  postgres_data:
  redis_data:
  clickhouse_data:
```

LAMPIRAN 6: Skrip Pengujian Otomatis Kinerja (benchmark.py)

```
import time
import csv
import random
import statistics
import logging
import threading
import psutil
import os
import argparse
from datetime import datetime

from app import create_app
from app.models.firewall import FirewallInstance
```



- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
from app.firewall.factory import get_firewall_client
from app.ip_blocking.firewall_service import push_blocklist_to_firewalls

CPU_CORE_COUNT = psutil.cpu_count(logical=True)
ITERATIONS = 30
SLEEP_BETWEEN = 1

def parse_args():
    parser = argparse.ArgumentParser(
        description="Benchmark skrip untuk Firewall Hub"
    )
    parser.add_argument(
        "--method",
        choices=["parallel", "sequential"],
        required=True,
        help="Metode eksekusi: parallel atau sequential",
    )
    parser.add_argument(
        "--ip-count",
        type=int,
        choices=[10, 20, 30, 40],
        required=True,
        help="Jumlah IP dalam payload: 10, 20, 30, atau 40",
    )
    parser.add_argument(
        "--bandwidth",
        type=str,
        required=True,
        help="Kondisi bandwidth saat ini (label saja, misal: 50KBps, 100KBps, unlimited)",
    )
    return parser.parse_args()

def setup_logger(method, ip_count, bandwidth):
    log_file = f"benchmark_error_{method}_{ip_count}ip_{bandwidth}.log"
    logger = logging.getLogger(f"benchmark_{method}_{ip_count}_{bandwidth}")
    logger.setLevel(logging.ERROR)
    handler = logging.FileHandler(log_file)
    handler.setFormatter(logging.Formatter("%(asctime)s - %(levelname)s - %(message)s"))
    logger.addHandler(handler)
    return logger

def generate_dummy_ips(count):
    ips = set()
    while len(ips) < count:
        ips.add(f"192.168.{random.randint(0, 255)}.{random.randint(1, 254)}")
    return ips

def sample_resources(stop_event, cpu_samples, ram_samples, interval=0.1):
    """
    Worker thread untuk sampling CPU dan RAM secara periodik.
    CPU disimpan sebagai nilai raw per-core (bisa melebihi 100% jika multi-core).
    Konversi ke core dilakukan saat pelaporan.
    """
    process = psutil.Process(os.getpid())
    process.cpu_percent(interval=None) # Inisialisasi, nilai pertama selalu 0.0

    while not stop_event.is_set():
        raw_cpu = process.cpu_percent(interval=None) # Raw, tanpa normalisasi
        ram_mb = process.memory_info().rss / (1024 * 1024)
        cpu_samples.append(raw_cpu)
        ram_samples.append(ram_mb)
        time.sleep(interval)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def format_cpu(cpu_percent):
    """Helper untuk format tampilan CPU: persentase + ekuivalen core."""
    cores = cpu_percent / 100
    return f"{cpu_percent:.1f}% ({cores:.2f} core)"

def run_sequential(instances, ip_set):
    """
    Eksekusi sekuensial: kirim ke setiap firewall satu per satu.
    Mengembalikan format yang sama dengan push_blocklist_to_firewalls()
    yaitu list of (name, success, message).
    """
    results = []
    for instance in instances:
        try:
            client = get_firewall_client(instance)
            success, message = client.update_blocklist(ip_set)
            results.append((instance.name, success, message))
        except Exception as e:
            results.append((instance.name, False, f"Error: {e}"))
    return results

def run_benchmark(method, ip_count, bandwidth, logger):
    benchmark_start = time.monotonic() # Catat waktu mulai keseluruhan

    app = create_app(
        log_filename=f"benchmark_{method}_{ip_count}ip_{bandwidth}_app.log"
    )
    with app.app_context():

        # Pre-flight check
        active_instances = FirewallInstance.query.filter_by(is_active=True).all()
        if not active_instances:
            print("Error: Tidak ada firewall aktif yang ditemukan.")
            return

        print("Menjalankan pre-flight connection test...")
        for instance in active_instances:
            client = get_firewall_client(instance)
            success, msg = client.test_connection()
            if not success:
                print(f"Error: Pre-flight check gagal untuk {instance.name}: {msg}")
                return

        print(
            f"Pre-flight checks berhasil. Mulai benchmark {method.upper()} | "
            f"{ip_count} IP | Bandwidth: {bandwidth} | CPU Cores: {CPU_CORE_COUNT}\n"
        )

        latencies = []
        success_count = 0
        csv_data = []

        all_peak_cpu = []
        all_avg_cpu = []
        all_peak_ram = []
        all_avg_ram = []

        for i in range(1, ITERATIONS + 1):
            ips = generate_dummy_ips(ip_count)

            # Setup resource sampler
            stop_event = threading.Event()
            cpu_samples = []
            ram_samples = []
            sampler_thread = threading.Thread(
                target=sample_resources,
                args=(stop_event, cpu_samples, ram_samples),
                daemon=True,
            )
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Jalankan eksekusi sesuai metode
sampler_thread.start()
start_time = time.monotonic()

if method == "parallel":
    results = push_blocklist_to_firewalls(ips)
else:
    results = run_sequential(active_instances, ips)

end_time = time.monotonic()
stop_event.set()
sampler_thread.join()

latency = end_time - start_time
all_success = all(res[1] for res in results)

# Hitung resource iterasi ini
peak_cpu = max(cpu_samples) if cpu_samples else 0.0
avg_cpu = statistics.mean(cpu_samples) if cpu_samples else 0.0
peak_ram = max(ram_samples) if ram_samples else 0.0
avg_ram = statistics.mean(ram_samples) if ram_samples else 0.0

all_peak_cpu.append(peak_cpu)
all_avg_cpu.append(avg_cpu)
all_peak_ram.append(peak_ram)
all_avg_ram.append(avg_ram)

latencies.append(latency)
if all_success:
    success_count += 1
    status_str = "Sukses"
else:
    status_str = "Gagal"
    error_msgs = [f"{r[0]}: {r[2]}" for r in results if not r[1]]
    logger.error(f"Iterasi {i} gagal. Detail: {error_msgs}")

csv_data.append(
    [
        i,
        f"{latency:.3f}",
        status_str,
        format_cpu(peak_cpu),
        format_cpu(avg_cpu),
        f"{peak_ram:.1f}",
        f"{avg_ram:.1f}",
    ]
)

print(
    f"[{i:02d}/30] Latensi: {latency:.3f}s | "
    f"CPU Peak: {format_cpu(peak_cpu)} Avg: {format_cpu(avg_cpu)} | "
    f"RAM Peak: {peak_ram:.1f}MB Avg: {avg_ram:.1f}MB | "
    f"Status: {status_str}"
)

time.sleep(SLEEP_BETWEEN)

# Ringkasan statistik
mean_lat = statistics.mean(latencies)
min_lat = min(latencies)
max_lat = max(latencies)
std_dev = statistics.stdev(latencies) if len(latencies) > 1 else 0.0
success_rate = (success_count / ITERATIONS) * 100

overall_peak_cpu = max(all_peak_cpu)
overall_avg_cpu = statistics.mean(all_avg_cpu)
overall_peak_ram = max(all_peak_ram)
overall_avg_ram = statistics.mean(all_avg_ram)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Hitung total waktu benchmark
total_duration = time.monotonic() - benchmark_start
total_minutes = int(total_duration // 60)
total_seconds = total_duration % 60

# Simpan ke CSV
timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
filename = f"result_{method}_{ip_count}ip_{bandwidth}_{timestamp}.csv"

with open(filename, "w", newline="") as f:
    writer = csv.writer(f)

    # Header info
    writer.writerow(["Metode", method.upper()])
    writer.writerow(["Jumlah IP", ip_count])
    writer.writerow(["Kondisi Bandwidth", bandwidth])
    writer.writerow(["Jumlah Iterasi", ITERATIONS])
    writer.writerow(["CPU Core Count", CPU_CORE_COUNT])
    writer.writerow([])

    # Data per iterasi
    writer.writerow(
        [
            "Iterasi",
            "Latensi (s)",
            "Status",
            "CPU Peak",
            "CPU Avg",
            "RAM Peak (MB)",
            "RAM Avg (MB)",
        ]
    )
    writer.writerows(csv_data)
    writer.writerow([])

    # Ringkasan latensi
    writer.writerow(["--- RINGKASAN LATENSI ---", "", "", "", "", "", ""])
    writer.writerow(["Rata-rata (s)", f"{mean_lat:.3f}", "", "", "", "", ""])
    writer.writerow(["Minimum (s)", f"{min_lat:.3f}", "", "", "", "", ""])
    writer.writerow(["Maksimum (s)", f"{max_lat:.3f}", "", "", "", "", ""])
    writer.writerow(["Std. Deviasi (s)", f"{std_dev:.3f}", "", "", "", "", ""])
    writer.writerow(
        ["Success Rate", f"{success_rate:.0f}%", "", "", "", "", ""]
    )
    writer.writerow([])

    # Ringkasan resource
    writer.writerow(["--- RINGKASAN RESOURCE ---", "", "", "", "", "", ""])
    writer.writerow(
        [
            "CPU Peak Keseluruhan",
            format_cpu(overall_peak_cpu),
            "",
            "",
            "",
            "",
            "",
        ]
    )
    writer.writerow(
        ["CPU Avg Keseluruhan", format_cpu(overall_avg_cpu), "", "", "", "", ""]
    )
    writer.writerow(
        [
            "RAM Peak Keseluruhan (MB)",
            f"{overall_peak_ram:.1f}",
            "",
            "",
            "",
            "",
        ]
    )
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        """,
        """,
    ]
)
writer.writerow(
    [
        "RAM Avg Keseluruhan (MB)",
        f"{overall_avg_ram:.1f}",
        "",
        "",
        "",
        "",
        "",
        "",
    ]
)
writer.writerow([])

# Total waktu
writer.writerow(["--- TOTAL WAKTU ---", "", "", "", "", "", "", ""])
writer.writerow(
    [
        "Total Waktu Benchmark",
        f"{total_minutes}m {total_seconds:.1f}s",
        "",
        "",
        "",
        "",
        "",
        "",
    ]
)

print(f"\nBenchmark selesai. Hasil disimpan di {filename}")
print(
    f"Latensi - Rata-rata: {mean_lat:.3f}s | Min: {min_lat:.3f}s | "
    f"Max: {max_lat:.3f}s | StdDev: {std_dev:.3f}s"
)
print(f"Success Rate: {success_rate:.0f}%")
print(
    f"CPU - Peak: {format_cpu(overall_peak_cpu)} | Avg: "
    f"{format_cpu(overall_avg_cpu)}"
)
print(f"RAM - Peak: {overall_peak_ram:.1f}MB | Avg: {overall_avg_ram:.1f}MB")
print(f"Total Waktu Benchmark: {total_minutes}m {total_seconds:.1f}s")

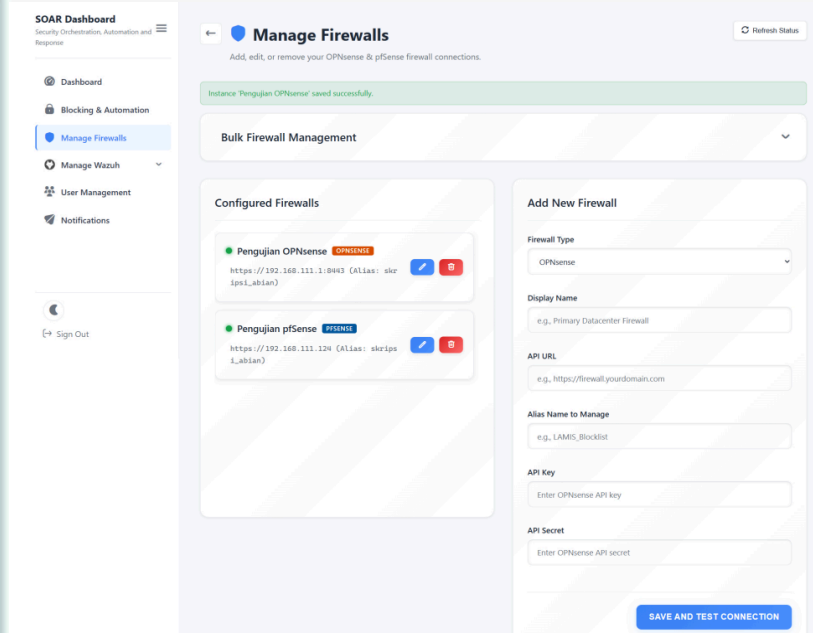
if __name__ == "__main__":
    args = parse_args()
    logger = setup_logger(args.method, args.ip_count, args.bandwidth)
    run_benchmark(args.method, args.ip_count, args.bandwidth, logger)
```



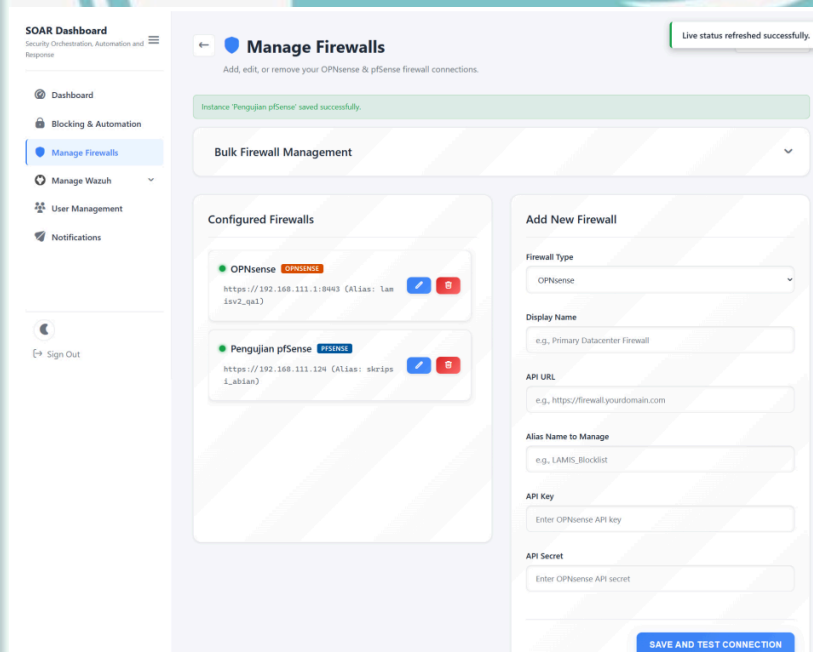
LAMPIRAN 7: Hasil Pengujian Fungsional

A. Tangkapan Layar Verifikasi Autentikasi API OPNsense dan pfSense(Uji FT-01)

a. Autentikasi OPNsense



b. Autentikasi pfSense



B. Tangkapan Layar Sinkronisasi Pemblokiran IP (Uji FT-02)



a. Block IP di OPNsense

b. Block IP di pfSense

c. Notifikasi Blok di UI

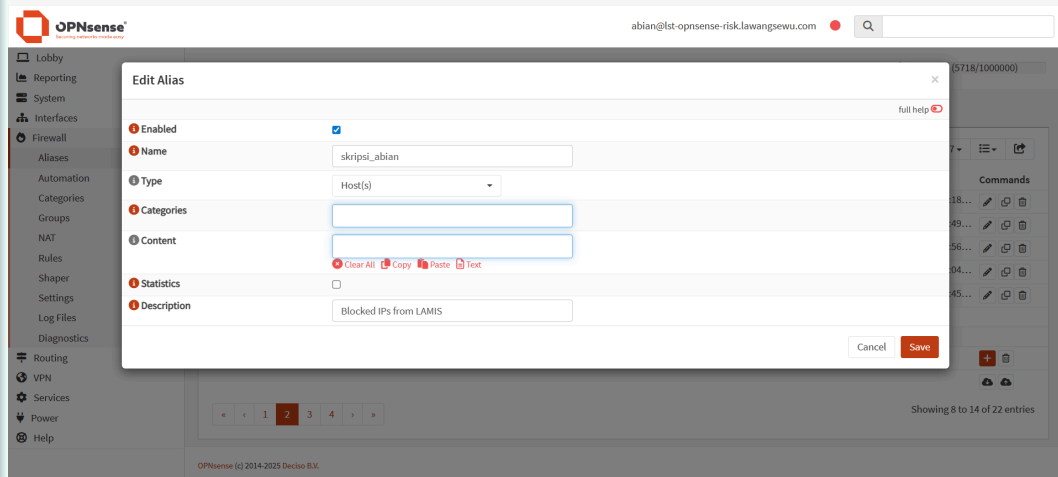


Hak Cipta :

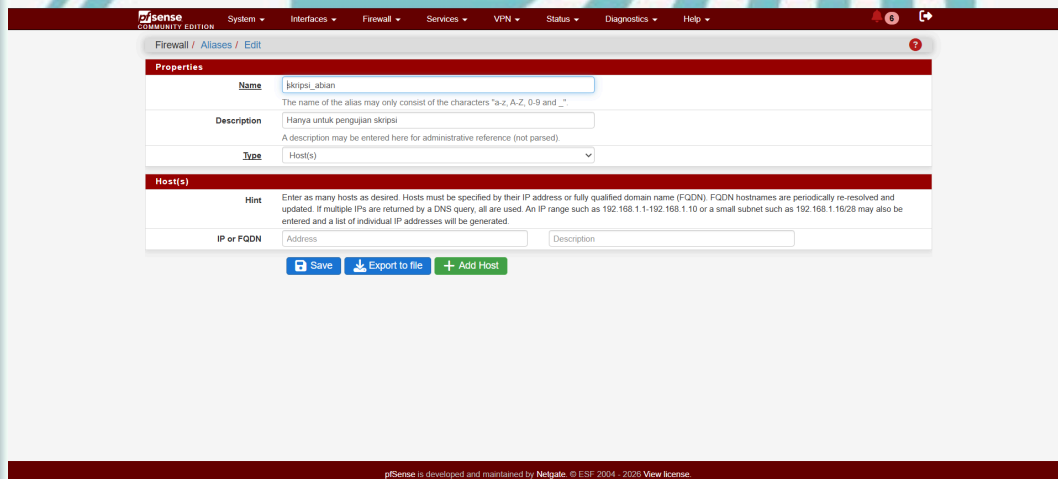
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

C. Tangkapan Layar Sinkronisasi Pembukaan Blokir / Unblock (Uji FT-03)

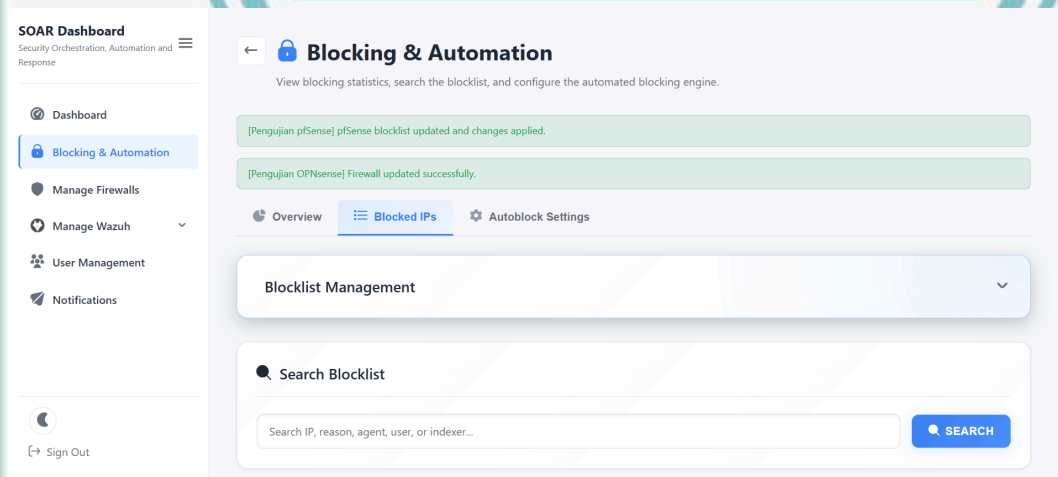
a. Unblock IP di OPNsense



b. Unblock IP di pfSense



c. Notifikasi Unblock IP di UI





D. Log Penanganan Eksepsi dan Isolasi Kegagalan / Fault Tolerance (Uji FT-04)

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
ProtocolError: ('Connection aborted.',
RemoteDisconnected('Remote end closed connection
without response'))
```

During handling of the above exception, another exception occurred:

```
+----- Traceback (most recent call last) -----+
```

```
C:\...\app\firewall\pfsense_ |
client.py:98 in update_blocklist
```

```
95     payload = all_aliases
96
97     try:
> 98         response = requests.put(
99             api_url, headers=heade
100         )
101         response.raise_for_status(
```

```
C:\...\venv\Lib\site-package |
s\requests\api.py:130 in put
```

```
127     :rtype: requests.Response
128     """
129
> 130     return request("put", url, data=da
131
132
133     def patch(url, data=None, **kwargs):
```

```
C:\...\venv\Lib\site-package |
s\requests\api.py:59 in request
```

```
56     # avoid leaving sockets open which
57     # cases, and look like a memory le
58     with sessions.Session() as session
> 59     return session.request(method=
60
61
62     def get(url, params=None, **kwargs):
```

```
C:\...\venv\Lib\site-package |
s\requests\sessions.py:589 in request
```

```
586         "allow_redirects": allow_r
587     }
588     send_kwargs.update(settings)
> 589     resp = self.send(prepare, **send_
590
591     return resp
592
```

```
C:\...\venv\Lib\site-package |
s\requests\sessions.py:703 in send
```

```
700     start = preferred_clock()
701
702     # Send the request
> 703     r = adapter.send(request, **kw
704
705     # Total elapsed time of the re
706     elapsed = preferred_clock() -
```

```
C:\...\venv\Lib\site-package |
s\requests\adapters.py:659 in send
```



Hak Cipta:

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

656         )
657
658         except (ProtocolError, OSError)
659             raise ConnectionError(err,
660
661         except MaxRetryError as e:
662             if isinstance(e.reason, Co

```

ConnectionError: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))

[2026-06-04 21:36:58] INFO app: Starting *blocklist* update for OPNsense instance 'Penguian OPNsense'

[2026-06-04 21:37:18] ERROR app: Penguian OPNsense: Connection failed. Check the URL and network connectivity. Error: HTTPConnectionPool(host='192.168.111.1', port=8443): Read timed out.

[2026-06-04 21:37:19] ERROR benchmark_sequential_40_50KBps: Iterasi 16 gagal. Detail: ["Penguian pfSense: API Error during update: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))", "Penguian OPNsense: Cannot update *blocklist*: Connection failed. Check the URL and network connectivity. Error: HTTPConnectionPool(host='192.168.111.1', port=8443): Read timed out."]

[16/30] Latensi: 42.422s | CPU Peak: 117.6% (1.18 core) Avg: 5.0% (0.05 core) | RAM Peak: 119.9MB Avg: 119.0MB | Status: Gagal

[2026-06-04 21:37:20] INFO app: Starting *blocklist* update for 'Penguian pfSense'

E. Bukti Keamanan Data At Rest (Uji FT-05)

```
benchmark_parallel_10.py U | firewall_instances X | benchmark_parallel_20.py U | benchmark_parallel_30.py U | benchmark_parallel_40.py U | benchmark_parallel_40.py U | benchmark_sequential_10.py U | benchmark_sequential_20.py U | ▶ 🔍 ⌵ ⌶ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵ ⌵
```

LAMPIRAN 8: Hasil Pengujian Non Fungsional

A. Pengujian Tanpa Batas Bandwidth

a. Data Pengujian Metode Paralel 10 IP

Metode					Paralel	
Jumlah IP					10	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	3.75	Sukses	113.6% (1.14 core)	15.0% (0.15 core)	120.2	119.1
2	3.297	Sukses	99.5% (0.99 core)	15.9% (0.16 core)	120.2	119.8
3	3.875	Sukses	99.4% (0.99 core)	16.1% (0.16 core)	120.5	119.5
4	9.735	Sukses	100.3% (1.00 core)	6.5% (0.07 core)	120.9	119
5	5.359	Sukses	91.4% (0.91 core)	10.0% (0.10 core)	120.1	119.4



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

6	4.109	Sukses	87.5% (0.88 core)	14.5% (0.14 core)	120.5	119
7	5.531	Sukses	99.4% (0.99 core)	10.5% (0.10 core)	120.5	119.1
8	4.828	Sukses	100.0% (1.00 core)	14.6% (0.15 core)	120.6	119.4
9	4.203	Sukses	100.3% (1.00 core)	15.1% (0.15 core)	120.9	119.4
10	4.422	Sukses	113.6% (1.14 core)	14.3% (0.14 core)	120.7	119.2
11	5	Sukses	100.3% (1.00 core)	12.4% (0.12 core)	120.3	119.1
12	3.86	Sukses	100.3% (1.00 core)	17.7% (0.18 core)	120.8	119.8
13	3.171	Sukses	100.3% (1.00 core)	17.5% (0.17 core)	120.6	119.3
14	3.422	Sukses	100.3% (1.00 core)	19.4% (0.19 core)	120.9	119.6
15	3.359	Sukses	100.4% (1.00 core)	20.3% (0.20 core)	120.9	119.5
16	7.25	Sukses	116.4% (1.16 core)	8.7% (0.09 core)	122.6	120.3
17	4.157	Sukses	99.7% (1.00 core)	14.7% (0.15 core)	121.2	120.3
18	4.797	Sukses	87.5% (0.88 core)	12.7% (0.13 core)	121.2	120.1
19	7.328	Sukses	100.8% (1.01 core)	7.7% (0.08 core)	121.3	120.1
20	6.359	Sukses	116.4% (1.16 core)	9.9% (0.10 core)	122.2	120.7
21	4.469	Sukses	114.7% (1.15 core)	15.2% (0.15 core)	121.5	119.7
22	3.734	Sukses	100.3% (1.00 core)	15.3% (0.15 core)	121.6	120.6
23	5.547	Sukses	100.0% (1.00 core)	10.3% (0.10 core)	121.5	120.3
24	4.484	Sukses	99.4% (0.99 core)	12.4% (0.12 core)	120.9	120.2
25	4.313	Sukses	113.6% (1.14 core)	14.2% (0.14 core)	123.5	122.6
26	3.438	Sukses	100.0% (1.00 core)	16.8% (0.17 core)	121.9	121.2
27	8.094	Sukses	100.0% (1.00 core)	9.7% (0.10 core)	120.9	119.9
28	5.156	Sukses	113.6% (1.14 core)	13.5% (0.14 core)	121.7	121
29	4.656	Sukses	114.7% (1.15 core)	14.7% (0.15 core)	121.4	121.2
30	5.343	Sukses	99.7% (1.00 core)	11.1% (0.11 core)	121.1	119.7
--- RINGKASAN LATENSI ---						
Rata-rata (s)		POLITEKNIK				4.902
Minimum (s)		NEGERI				3.171
Maksimum (s)		JAKARTA				9.735
Std. Deviasi (s)						1.533
Success Rate						100%
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan					116.4% (1.16 core)	
CPU Avg Keseluruhan					13.6% (0.14 core)	
RAM Peak Keseluruhan (MB)					123.5	
RAM Avg Keseluruhan (MB)					119.9	
--- TOTAL WAKTU ---						
Total Waktu Benchmark					3m 1.4s	

b. Data Pengujian Metode Paralel 20 IP

Metode					Paralel	
Jumlah IP					20	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	3.75	Sukses	100.8% (1.01 core)	15.1% (0.15 core)	119.8	118.8



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2	7.766	Sukses	100.3% (1.00 core)	8.5% (0.09 core)	120.1	117.7
3	5.468	Sukses	100.8% (1.01 core)	12.2% (0.12 core)	119.6	118.1
4	6.781	Sukses	99.4% (0.99 core)	9.3% (0.09 core)	122	119.8
5	4.485	Sukses	100.0% (1.00 core)	16.9% (0.17 core)	120.3	117.8
6	5.906	Sukses	100.3% (1.00 core)	11.1% (0.11 core)	120.2	118.6
7	4.813	Sukses	116.4% (1.16 core)	14.1% (0.14 core)	120.6	118.9
8	8.078	Sukses	114.7% (1.15 core)	7.9% (0.08 core)	119.5	117.9
9	4.579	Sukses	100.8% (1.01 core)	15.7% (0.16 core)	120.6	119.6
10	3.094	Sukses	100.3% (1.00 core)	19.0% (0.19 core)	120.5	119.2
11	4.672	Sukses	100.3% (1.00 core)	12.7% (0.13 core)	120.3	119.3
12	5.563	Sukses	86.0% (0.86 core)	11.6% (0.12 core)	121.1	119.8
13	3.25	Sukses	100.0% (1.00 core)	17.4% (0.17 core)	120.5	119.9
14	6.032	Sukses	99.4% (0.99 core)	10.7% (0.11 core)	120.6	120.4
15	4.843	Sukses	86.0% (0.86 core)	11.9% (0.12 core)	120.9	119.8
16	4.984	Sukses	113.6% (1.14 core)	12.9% (0.13 core)	120.7	120.3
17	3.891	Sukses	100.3% (1.00 core)	15.9% (0.16 core)	120.5	119.6
18	3.062	Sukses	111.6% (1.12 core)	19.6% (0.20 core)	121.1	119.3
19	8.406	Sukses	100.3% (1.00 core)	7.6% (0.08 core)	120.9	120.4
20	3.547	Sukses	100.3% (1.00 core)	16.7% (0.17 core)	121.2	120.9
21	5.234	Sukses	116.4% (1.16 core)	21.5% (0.21 core)	121.5	120.2
22	4.063	Sukses	100.0% (1.00 core)	18.6% (0.19 core)	121.5	121
23	5.265	Sukses	114.7% (1.15 core)	18.1% (0.18 core)	121.7	120.3
24	15.438	Sukses	116.4% (1.16 core)	7.0% (0.07 core)	121.8	120.4
25	5.735	Sukses	100.3% (1.00 core)	14.5% (0.15 core)	121.9	121
26	5.812	Sukses	100.3% (1.00 core)	12.8% (0.13 core)	121.9	120.7
27	9	Sukses	87.5% (0.88 core)	6.0% (0.06 core)	121.9	120.3
28	6.171	Sukses	100.0% (1.00 core)	7.5% (0.07 core)	121.9	120.4
29	3.406	Sukses	112.5% (1.12 core)	17.8% (0.18 core)	121.9	121.5
30	3.063	Sukses	116.4% (1.16 core)	19.1% (0.19 core)	121.6	121.3
--- RINGKASAN LATENSI ---						
Rata-rata (s)					5.539	
Minimum (s)					3.062	
Maksimum (s)					15.438	
Std. Deviasi (s)					2.466	
Success Rate					100%	
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan					116.4% (1.16 core)	
CPU Avg Keseluruhan					13.7% (0.14 core)	
RAM Peak Keseluruhan (MB)					122	
RAM Avg Keseluruhan (MB)					119.8	
--- TOTAL WAKTU ---						
Total Waktu Benchmark					3m 20.9s	



c. Data Pengujian Metode Paralel 30 IP

Metode					Paralel	
Jumlah IP					30	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	3.61	Sukses	100.8% (1.01 core)	16.9% (0.17 core)	119.4	118.5
2	8.704	Sukses	116.4% (1.16 core)	7.9% (0.08 core)	119.4	118.2
3	5.797	Sukses	100.3% (1.00 core)	12.1% (0.12 core)	120.8	119.3
4	4.656	Sukses	99.4% (0.99 core)	15.1% (0.15 core)	120.2	118.7
5	5.203	Sukses	99.4% (0.99 core)	12.1% (0.12 core)	120.1	118.8
6	4.453	Sukses	100.0% (1.00 core)	13.7% (0.14 core)	120.5	119.3
7	3.969	Sukses	114.7% (1.15 core)	17.8% (0.18 core)	120.2	119.4
8	4.407	Sukses	100.3% (1.00 core)	15.8% (0.16 core)	121.7	121.1
9	7.672	Sukses	116.4% (1.16 core)	8.3% (0.08 core)	121.1	119.4
10	3.937	Sukses	100.3% (1.00 core)	16.5% (0.16 core)	120.1	119.4
11	5.547	Sukses	100.3% (1.00 core)	11.7% (0.12 core)	120.9	119.1
12	3.328	Sukses	116.4% (1.16 core)	18.2% (0.18 core)	120.8	120.4
13	8.235	Sukses	100.3% (1.00 core)	6.9% (0.07 core)	122	119.6
14	9.406	Sukses	113.6% (1.14 core)	7.9% (0.08 core)	120.8	119.3
15	5.078	Sukses	100.3% (1.00 core)	11.8% (0.12 core)	121.2	120.6
16	3.641	Sukses	100.3% (1.00 core)	14.0% (0.14 core)	121.2	120.6
17	5.812	Sukses	100.3% (1.00 core)	11.5% (0.12 core)	122.9	121.1
18	3.938	Sukses	100.3% (1.00 core)	16.0% (0.16 core)	121.9	121.5
19	6.672	Sukses	100.3% (1.00 core)	11.3% (0.11 core)	122.3	121.3
20	3.672	Sukses	100.0% (1.00 core)	17.2% (0.17 core)	121.5	120.4
21	5.188	Sukses	100.3% (1.00 core)	13.5% (0.14 core)	122.4	119.8
22	5.125	Sukses	100.2% (1.00 core)	13.0% (0.13 core)	119	117.9
23	5.328	Sukses	100.0% (1.00 core)	12.2% (0.12 core)	119.1	117.9
24	6.625	Sukses	99.7% (1.00 core)	9.6% (0.10 core)	118.8	117.6
25	4.25	Sukses	100.0% (1.00 core)	15.1% (0.15 core)	118.4	117.8
26	5.516	Sukses	100.8% (1.01 core)	12.2% (0.12 core)	118.5	117.5
27	4.125	Sukses	99.4% (0.99 core)	15.0% (0.15 core)	118.5	117.8
28	5.735	Sukses	100.3% (1.00 core)	11.7% (0.12 core)	120.1	119.3
29	7.954	Sukses	100.3% (1.00 core)	8.0% (0.08 core)	119.4	119.1
30	4.75	Sukses	99.7% (1.00 core)	12.6% (0.13 core)	118.9	118.4
--- RINGKASAN LATENSI ---						
Rata-rata (s)					5.411	
Minimum (s)					3.328	
Maksimum (s)					9.406	
Std. Deviasi (s)					1.622	

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Success Rate	100%
--- RINGKASAN RESOURCE ---	
CPU Peak Keseluruhan	116.4% (1.16 core)
CPU Avg Keseluruhan	12.9% (0.13 core)
RAM Peak Keseluruhan (MB)	122.9
RAM Avg Keseluruhan (MB)	119.3
--- TOTAL WAKTU ---	
Total Waktu Benchmark	3m 18.0s

d. Data Pengujian Metode Paralel 40 IP

Metode					Paralel	
Jumlah IP					40	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	11.828	Sukses	117.6% (1.18 core)	7.5% (0.08 core)	121.9	119.5
2	5.875	Sukses	116.4% (1.16 core)	21.8% (0.22 core)	120.9	119.8
3	4.422	Sukses	108.0% (1.08 core)	26.4% (0.26 core)	120.7	119.5
4	5.218	Sukses	113.6% (1.14 core)	18.4% (0.18 core)	121	120
5	10.25	Sukses	116.4% (1.16 core)	11.2% (0.11 core)	120.8	119.4
6	5.094	Sukses	100.3% (1.00 core)	11.0% (0.11 core)	120.6	119.1
7	3.265	Sukses	99.4% (0.99 core)	19.5% (0.19 core)	120.5	119.5
8	8.844	Sukses	113.6% (1.14 core)	7.1% (0.07 core)	121.6	120.3
9	6.687	Sukses	100.0% (1.00 core)	9.0% (0.09 core)	120.7	119.7
10	9.063	Sukses	116.4% (1.16 core)	7.2% (0.07 core)	120.8	119.3
11	4.203	Sukses	87.5% (0.88 core)	12.7% (0.13 core)	120.6	120.2
12	3.672	Sukses	100.3% (1.00 core)	15.1% (0.15 core)	120.6	119.8
13	9.281	Sukses	116.4% (1.16 core)	7.0% (0.07 core)	121.3	120.4
14	5.468	Sukses	100.8% (1.01 core)	11.5% (0.12 core)	121.4	120.6
15	3.391	Sukses	86.0% (0.86 core)	15.7% (0.16 core)	121.4	121
16	3.125	Sukses	99.4% (0.99 core)	18.3% (0.18 core)	121.7	120.5
17	5.891	Sukses	116.4% (1.16 core)	10.7% (0.11 core)	122.6	121.3
18	7.89	Sukses	116.4% (1.16 core)	7.9% (0.08 core)	123.3	121.6
19	4.031	Sukses	99.7% (1.00 core)	13.1% (0.13 core)	121.7	120.8
20	11.265	Sukses	116.4% (1.16 core)	8.9% (0.09 core)	121.7	120.3
21	7.312	Sukses	114.7% (1.15 core)	11.0% (0.11 core)	121.8	119.9
22	8.297	Sukses	100.8% (1.01 core)	10.1% (0.10 core)	122.4	120.8
23	5.86	Sukses	117.6% (1.18 core)	23.5% (0.24 core)	122.2	121.7
24	7.547	Sukses	125.0% (1.25 core)	17.0% (0.17 core)	122.1	121.2
25	3.907	Sukses	117.6% (1.18 core)	15.7% (0.16 core)	122.1	121.5
26	4.969	Sukses	113.6% (1.14 core)	27.9% (0.28 core)	122	121.3
27	3.875	Sukses	116.4% (1.16 core)	27.8% (0.28 core)	122.3	122



28	12.547	Sukses	109.5% (1.09 core)	9.5% (0.09 core)	122.1	121.2
29	7.203	Sukses	113.6% (1.14 core)	20.1% (0.20 core)	122.5	121.6
30	8.172	Sukses	100.3% (1.00 core)	15.0% (0.15 core)	122.5	122
--- RINGKASAN LATENSI ---						
Rata-rata (s)					6.615	
Minimum (s)					3.125	
Maksimum (s)					12.547	
Std. Deviasi (s)					2.681	
Success Rate					100%	
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan					125.0% (1.25 core)	
CPU Avg Keseluruhan					14.6% (0.15 core)	
RAM Peak Keseluruhan (MB)					123.3	
RAM Avg Keseluruhan (MB)					120.5	
--- TOTAL WAKTU ---						
Total Waktu Benchmark					3m 51.6s	

e. Data Pengujian Metode Sekuensial 10 IP

Metode		Sekuensial				
Jumlah IP		10				
Kondisi Bandwidth		Unlimited				
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	3.719	Sukses	100.0% (1.00 core)	18.3% (0.18 core)	118	117.4
2	6.562	Sukses	99.4% (0.99 core)	10.5% (0.10 core)	118	117.1
3	7.188	Sukses	99.7% (1.00 core)	7.7% (0.08 core)	118	117.2
4	3.594	Sukses	117.6% (1.18 core)	21.2% (0.21 core)	117.9	117.4
5	5.578	Sukses	87.5% (0.88 core)	13.4% (0.13 core)	118.1	117.5
6	5.75	Sukses	71.0% (0.71 core)	11.3% (0.11 core)	118.2	117.6
7	7.266	Sukses	117.6% (1.18 core)	8.9% (0.09 core)	118.1	117
8	6.703	Sukses	100.3% (1.00 core)	10.5% (0.10 core)	118.2	117.3
9	5.75	Sukses	86.0% (0.86 core)	14.0% (0.14 core)	118.6	118.3
10	6.235	Sukses	100.3% (1.00 core)	10.8% (0.11 core)	118.7	118
11	6.688	Sukses	99.7% (1.00 core)	9.1% (0.09 core)	118.3	118.2
12	7.157	Sukses	99.4% (0.99 core)	8.3% (0.08 core)	118.3	118.2
13	4.766	Sukses	116.4% (1.16 core)	14.2% (0.14 core)	118.7	117.9
14	5.578	Sukses	100.8% (1.01 core)	12.4% (0.12 core)	118.2	117.7
15	7.938	Sukses	100.8% (1.01 core)	8.6% (0.09 core)	118.5	118.4
16	8.312	Sukses	116.4% (1.16 core)	11.7% (0.12 core)	118.8	118.7
17	6.032	Sukses	100.3% (1.00 core)	10.7% (0.11 core)	119	118.9
18	11.438	Sukses	86.0% (0.86 core)	5.7% (0.06 core)	119	117.6
19	6.829	Sukses	100.3% (1.00 core)	9.3% (0.09 core)	118.4	118.1

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

20	5.141	Sukses	99.7% (1.00 core)	12.3% (0.12 core)	118.7	118.5
21	5.234	Sukses	86.0% (0.86 core)	10.1% (0.10 core)	118.9	118.8
22	5.703	Sukses	85.2% (0.85 core)	11.4% (0.11 core)	118.9	118
23	5.312	Sukses	86.0% (0.86 core)	11.7% (0.12 core)	118.6	118.3
24	9.015	Sukses	100.3% (1.00 core)	7.5% (0.08 core)	118.7	118.7
25	6.187	Sukses	71.7% (0.72 core)	10.7% (0.11 core)	118.8	118.7
26	5.344	Sukses	85.2% (0.85 core)	11.1% (0.11 core)	118.8	117.8
27	4.453	Sukses	99.7% (1.00 core)	15.7% (0.16 core)	118.3	118.2
28	9.187	Sukses	99.7% (1.00 core)	7.6% (0.08 core)	118.5	118.4
29	6.172	Sukses	86.0% (0.86 core)	11.6% (0.12 core)	118.7	118.6
30	6.281	Sukses	87.5% (0.88 core)	9.7% (0.10 core)	118.7	117.9
--- RINGKASAN LATENSI ---						
Rata-rata (s)						6.37
Minimum (s)						3.594
Maksimum (s)						11.438
Std. Deviasi (s)						1.64
Success Rate						100%
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan						117.6% (1.18 core)
CPU Avg Keseluruhan						11.2% (0.11 core)
RAM Peak Keseluruhan (MB)						119
RAM Avg Keseluruhan (MB)						118
--- TOTAL WAKTU ---						
Total Waktu Benchmark						3m 49.5s

f. Data Pengujian Metode Sekuensial 20 IP

Metode					Sekuensial	
Jumlah IP					20	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	5.531	Sukses	99.7% (1.00 core)	10.5% (0.10 core)	117.7	117.1
2	9.328	Sukses	114.7% (1.15 core)	14.5% (0.14 core)	117.5	117.1
3	7.891	Sukses	114.7% (1.15 core)	15.8% (0.16 core)	117.6	116.7
4	9.468	Sukses	113.6% (1.14 core)	13.4% (0.13 core)	117.6	116.9
5	8.547	Sukses	100.8% (1.01 core)	13.5% (0.13 core)	117.8	117
6	14.797	Sukses	100.8% (1.01 core)	9.0% (0.09 core)	117.8	117.1
7	7.688	Sukses	116.4% (1.16 core)	17.7% (0.18 core)	117.7	116.9
8	4.75	Sukses	100.8% (1.01 core)	21.4% (0.21 core)	117.4	116.9
9	5.578	Sukses	83.1% (0.83 core)	11.1% (0.11 core)	118	117.4
10	7.578	Sukses	99.7% (1.00 core)	7.5% (0.08 core)	118	117.3
11	5.812	Sukses	100.8% (1.01 core)	9.9% (0.10 core)	117.7	117.6



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

12	6.578	Sukses	85.2% (0.85 core)	10.1% (0.10 core)	117.7	117.5
13	7.766	Sukses	75.0% (0.75 core)	7.9% (0.08 core)	118	117.7
14	14.891	Sukses	100.3% (1.00 core)	3.7% (0.04 core)	118	117.8
15	4.25	Sukses	99.7% (1.00 core)	12.3% (0.12 core)	118	117.8
16	3.469	Sukses	99.7% (1.00 core)	17.2% (0.17 core)	118.1	117.5
17	4.109	Sukses	100.3% (1.00 core)	16.1% (0.16 core)	118	117.8
18	8.172	Sukses	99.7% (1.00 core)	7.0% (0.07 core)	118.2	117.7
19	4	Sukses	99.7% (1.00 core)	14.5% (0.15 core)	118.1	118
20	6.688	Sukses	99.7% (1.00 core)	10.6% (0.11 core)	118.1	117.4
21	7.312	Sukses	100.0% (1.00 core)	8.7% (0.09 core)	118.1	117.9
22	6.578	Sukses	84.0% (0.84 core)	8.2% (0.08 core)	118.1	117.7
23	12.406	Sukses	99.7% (1.00 core)	5.1% (0.05 core)	118.2	117.8
24	4.719	Sukses	75.0% (0.75 core)	13.2% (0.13 core)	118.2	117.7
25	5.453	Sukses	100.8% (1.01 core)	10.4% (0.10 core)	118.2	117.9
26	13.75	Sukses	86.0% (0.86 core)	5.2% (0.05 core)	118.2	117.9
27	4.891	Sukses	86.0% (0.86 core)	11.8% (0.12 core)	118.2	117.8
28	7.437	Sukses	100.3% (1.00 core)	10.9% (0.11 core)	118.3	118.2
29	6.062	Sukses	114.7% (1.15 core)	21.3% (0.21 core)	118.4	118.4
30	10.813	Sukses	100.3% (1.00 core)	10.5% (0.11 core)	118.5	118.4
--- RINGKASAN LATENSI ---						
Rata-rata (s)					7.544	
Minimum (s)					3.469	
Maksimum (s)					14.891	
Std. Deviasi (s)					3.122	
Success Rate					100%	
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan					116.4% (1.16 core)	
CPU Avg Keseluruhan					11.6% (0.12 core)	
RAM Peak Keseluruhan (MB)					118.5	
RAM Avg Keseluruhan (MB)					117.6	
--- TOTAL WAKTU ---						
Total Waktu Benchmark					4m 20.9s	

g. Data Pengujian Metode Sekuensial 30 IP

Metode					Sekuensial	
Jumlah IP					30	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	6.672	Sukses	86.0% (0.86 core)	9.2% (0.09 core)	117.8	117.2
2	5.968	Sukses	116.4% (1.16 core)	13.9% (0.14 core)	117.8	117
3	8.422	Sukses	116.4% (1.16 core)	10.8% (0.11 core)	117.8	117.2



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4	11.828	Sukses	116.4% (1.16 core)	8.7% (0.09 core)	117.8	117
5	7.281	Sukses	100.3% (1.00 core)	11.1% (0.11 core)	117.8	117.1
6	5.609	Sukses	99.7% (1.00 core)	14.3% (0.14 core)	117.9	117
7	15.922	Sukses	99.4% (0.99 core)	5.2% (0.05 core)	117.8	116.8
8	6.265	Sukses	113.6% (1.14 core)	15.2% (0.15 core)	117.6	115.4
9	6.25	Sukses	86.0% (0.86 core)	9.8% (0.10 core)	115.3	114.6
10	6.047	Sukses	99.4% (0.99 core)	10.8% (0.11 core)	115.1	114.4
11	11.812	Sukses	116.4% (1.16 core)	6.9% (0.07 core)	115.5	114.5
12	8.875	Sukses	116.4% (1.16 core)	10.1% (0.10 core)	115.6	114.8
13	6.281	Sukses	116.4% (1.16 core)	15.5% (0.15 core)	115.4	114.9
14	13.766	Sukses	113.6% (1.14 core)	5.7% (0.06 core)	115.8	114.6
15	4.593	Sukses	99.7% (1.00 core)	17.2% (0.17 core)	115.6	114.5
16	4.468	Sukses	116.4% (1.16 core)	19.0% (0.19 core)	115.8	115.2
17	7.828	Sukses	100.3% (1.00 core)	10.8% (0.11 core)	115.4	114.5
18	10.594	Sukses	100.3% (1.00 core)	8.0% (0.08 core)	115.4	114.4
19	3.938	Sukses	100.3% (1.00 core)	21.3% (0.21 core)	115.9	115.5
20	8.656	Sukses	99.7% (1.00 core)	10.5% (0.10 core)	115.5	114.9
21	4.859	Sukses	100.8% (1.01 core)	17.6% (0.18 core)	115.5	114.7
22	7.64	Sukses	100.0% (1.00 core)	10.3% (0.10 core)	115.4	114.6
23	8.859	Sukses	116.4% (1.16 core)	9.7% (0.10 core)	115.7	115.1
24	5.266	Sukses	86.0% (0.86 core)	12.1% (0.12 core)	115.7	114.8
25	4.172	Sukses	117.6% (1.18 core)	22.3% (0.22 core)	115.7	115.1
26	9.093	Sukses	100.3% (1.00 core)	11.0% (0.11 core)	115.4	114.9
27	8.421	Sukses	83.1% (0.83 core)	7.7% (0.08 core)	115.9	115
28	4.016	Sukses	87.5% (0.88 core)	17.4% (0.17 core)	115.7	115.3
29	11.047	Sukses	100.3% (1.00 core)	11.1% (0.11 core)	115.4	114.9
30	10.953	Sukses	100.3% (1.00 core)	10.5% (0.11 core)	115.8	115.4
--- RINGKASAN LATENSI ---						
Rata-rata (s)					7.847	
Minimum (s)					3.938	
Maksimum (s)					15.922	
Std. Deviasi (s)					3.03	
Success Rate					100%	
--- RINGKASAN RESOURCE ---						
CPU Peak Keseluruhan					117.6% (1.18 core)	
CPU Avg Keseluruhan					12.1% (0.12 core)	
RAM Peak Keseluruhan (MB)					117.9	
RAM Avg Keseluruhan (MB)					115.4	
--- TOTAL WAKTU ---						
Total Waktu Benchmark					4m 30.1s	



h. Data Pengujian Metode Sekuensial 40 IP

Metode					Paralel	
Jumlah IP					10	
Kondisi Bandwidth					Unlimited	
Iterasi	Latensi (s)	Status	CPU Peak	CPU Avg	RAM Peak (MB)	RAM Avg (MB)
1	5.875	Sukses	116.4% (1.16 core)	12.3% (0.12 core)	117.5	116.6
2	4.094	Sukses	117.6% (1.18 core)	18.7% (0.19 core)	117.5	116.9
3	17.39	Sukses	83.1% (0.83 core)	4.0% (0.04 core)	117.5	116.4
4	5.281	Sukses	86.0% (0.86 core)	9.5% (0.09 core)	117.5	117.3
5	12.812	Sukses	85.2% (0.85 core)	4.9% (0.05 core)	117.8	116.9
6	7.516	Sukses	99.7% (1.00 core)	7.9% (0.08 core)	117.6	116.7
7	7	Sukses	86.0% (0.86 core)	9.3% (0.09 core)	117.5	116.7
8	3.735	Sukses	99.7% (1.00 core)	16.1% (0.16 core)	117.5	116.9
9	4.766	Sukses	100.3% (1.00 core)	13.9% (0.14 core)	117.9	117.1
10	9.391	Sukses	100.3% (1.00 core)	6.3% (0.06 core)	117.7	116.8
11	4.828	Sukses	85.2% (0.85 core)	13.3% (0.13 core)	117.7	117.1
12	5.859	Sukses	117.6% (1.18 core)	11.1% (0.11 core)	117.5	116.9
13	15.25	Sukses	86.0% (0.86 core)	3.8% (0.04 core)	118	117.2
14	4.922	Sukses	100.3% (1.00 core)	13.8% (0.14 core)	117.9	117.3
15	5.938	Sukses	87.5% (0.88 core)	12.1% (0.12 core)	118	116.9
16	3.718	Sukses	86.0% (0.86 core)	16.2% (0.16 core)	118.1	117.5
17	4.203	Sukses	100.3% (1.00 core)	16.1% (0.16 core)	118.1	117.5
18	9.657	Sukses	100.3% (1.00 core)	6.5% (0.07 core)	118.2	117.1
19	5.047	Sukses	86.0% (0.86 core)	12.1% (0.12 core)	118	117.1
20	4.953	Sukses	86.0% (0.86 core)	12.5% (0.13 core)	117.8	117.2
21	4.312	Sukses	100.3% (1.00 core)	15.6% (0.16 core)	118.2	117.3
22	7.422	Sukses	99.7% (1.00 core)	9.5% (0.10 core)	118.2	117.4
23	5.922	Sukses	85.2% (0.85 core)	10.0% (0.10 core)	118.5	117.7
24	11.032	Sukses	116.4% (1.16 core)	5.9% (0.06 core)	118.5	117.4
25	7.235	Sukses	99.4% (0.99 core)	8.0% (0.08 core)	118	117
26	3.64	Sukses	100.3% (1.00 core)	18.0% (0.18 core)	118	117.3
27	9.656	Sukses	99.4% (0.99 core)	6.8% (0.07 core)	118.1	117.1
28	7.657	Sukses	116.4% (1.16 core)	9.1% (0.09 core)	117.8	117.2
29	8.234	Sukses	86.0% (0.86 core)	8.2% (0.08 core)	117.8	117.1
30	4.234	Sukses	75.0% (0.75 core)	13.5% (0.14 core)	118.1	117.1
--- RINGKASAN LATENSI ---						
Rata-rata (s)					7.053	
Minimum (s)					3.64	
Maksimum (s)					17.39	
Std. Deviasi (s)					3.434	
Success Rate					100%	

- Hak Cipta :**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
 2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

--- RINGKASAN RESOURCE ---	
CPU Peak Keseluruhan	117.6% (1.18 core)
CPU Avg Keseluruhan	10.8% (0.11 core)
RAM Peak Keseluruhan (MB)	118.5
RAM Avg Keseluruhan (MB)	117.1
--- TOTAL WAKTU ---	
Total Waktu Benchmark	4m 4.1s

Dokumentasi data mentah selengkapnya untuk seluruh skenario uji tersedia secara publik pada:

<https://drive.google.com/drive/folders/1X48lfXNwbob81wCyp9rrLjMbMuBr-60a>

