ANALISIS EFEKTIVITAS FIREWALL DINAMIS BERBASIS LINUX TERHADAP SERANGAN SIBER BERDASARKAN DATA LOG SERVER UNIVERSITAS PALANGKA RAYA

DISUSUN OLEH:

ARICO SETIAWAN

223020503125

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2026

HALAMAN PENGESAHAN

ANALISIS EFEKTIVITAS *FIREWALL* DINAMIS BERBASIS LINUX TERHADAP SERANGAN SIBER BERDASARKAN DATA *LOG SERVER* UNIVERSITAS PALANGKA RAYA

SKRIPSI

Sebagai salah satu syarat untuk menyelesaikan Program Strata 1 pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

Oleh:

ARICO SETIAWAN

223020503125

Telah dipertahankan didepan tim penguji, pada:

Hari/Tanggal	: Rabu, 22 Juli 2026
Waktu	: 09:00-10:30 WIB
Tempat	: Ruang Ujian TI 2

1. AGUS SEHATMAN SARAGIH, S.T., M.Eng. **NIP. 198S08182012121003**	: .....(Ketua Penguji)
2. SEPTIAN GEGES, S.Kom., M.Kom. **NIP. 199109172020121004**	: .....(Pembimbing Utama/Pertama)
3. NAHUMI NUGRAHANINGSIH, S.T., M.T., Ph.D. **NIP. 197910092008012016**	: .....(Pembimbing Pendamping/Kedua)
4. LIYANDO HERMAWAN HASIBUAN, S.T., M.Kom. **NIP. 199607222024061001**	: .....(Anggota)

Mengetahui:
Jurusan / Program Studi Teknik Informatika
Fakultas Teknik Universitas Palangka Raya
Ketua Jurusan,

ARIESTA LESTARI, S.Kom., M.Cs., Ph.D.
NIP. 198003222005012004

HALAMAN PERNYATAAN

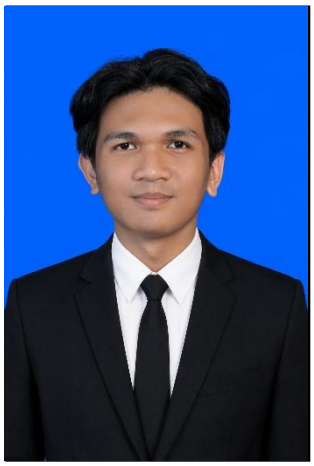
Dengan ini saya menyatakan dengan sebenar-benarnya bahwa dalam skripsi ini tidak terdapat Karya Ilmiah yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, serta tidak terdapat Karya Ilmiah atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam Skripsi ini dan disebutkan dalan Tinjauan Pustaka.

Palangka Raya, 23 - Juli - 2026

Arico Setiawan

223020503125

RIWAYAT PENYUSUN



Nama	: Arico Setiawan
NIM	: 223020503125
Fakultas	: Teknik
Program Studi	: Teknik Informatika
Jenjang	: Strata-1 (S-1)
Jenis Kelamin	: Laki-laki
Tempat, Tanggal Lahir	: Palangka Raya, 1 Desember 2004
Agama	: Islam
Status Dalam Keluarga	: Anak Kandung
Anak Ke	: Ke-2
Alamat	: Jalan Sidomulyo RT 003/RW 002, Tangkiling
No Telepon	: 085194121110

Data Orang Tua

Nama Ayah	: Eko Setio Santoso
Pekerjaan Ayah	: Serabutan
Nama Ibu	: Isnawati
Pekerjaan Ibu	: Buruh Tani
Alamat Orang Tua	: Jalan Sidomulyo RT 003/RW 002, Tangkiling
No Telepon	: 081256471808

Riwayat Pendidikan

Sekolah Dasar	: SDN 1 Panarung, Palangka Raya
Sekolah Menengah Pertama	: SMPN 1 Palangka Raya
Sekolah Menengah Kejuruan	: SMKN 8 Palangka Raya

HALAMAN PERSEMBAHAN

Puji dan syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga langkah yang panjang ini akhirnya mencapai garis akhir. Dengan kerendahan hati, karya sederhana ini saya persembahkan kepada:

Keluarga Tercinta Ibu Isnawati dan Bapak Eko Setio Santoso, terima kasih atas doa, kasih sayang, dan dukungan yang tak pernah putus. Terima kasih telah menoleransi setiap kesalahan, menghargai keputusan saya, dan selalu mengusahakan yang terbaik. Untuk kakak saya, Nico Setiawan, terima kasih atas bantuan nyata dan laptop yang menjadi senjata utama saya menyelesaikan skripsi ini.

Dosen Pembimbing Bapak Septian Geges, S.Kom., M.Kom. dan Ibu Nahumi Nugrahaningsih, S.T., M.T., Ph.D. Terima kasih atas segala arahan, kesabaran, dan bimbingannya. Saya sangat menyadari kelulusan dan selesainya skripsi yang masih memiliki banyak kekurangan ini, sebagian besar adalah berkat bantuan dan dedikasi Bapak dan Ibu.

Sahabat Seperjuangan Hariz Kurniawan, terima kasih sudah menjadi tempat curhat paling andal, teman bertukar pikiran, dan *partner* bercanda yang membuat beban serta stres selama perkuliahan terasa jauh lebih ringan.

Catatan untuk Diri Sendiri *"Skripsi yang baik adalah skripsi yang selesai."* Saya sangat memahami bahwa karya ini masih jauh dari kata sempurna. Namun, menaklukkan kerumitan skripsi ini adalah sebuah perjalanan yang berharga. Terima kasih, Arico, sudah berjuang, tidak menyerah, dan menyelesaikannya hingga akhir.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan Skripsi yang berjudul "Analisis Efektivitas Firewall Dinamis Berbasis Linux (Studi Kasus: Log Server Universitas Palangka Raya)" dengan baik.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Septian Geges, S.Kom., M.Kom. selaku Dosen Pembimbing I dan Ibu Nahumi Nugrahaningsih, S.T., M.T., Ph.D. selaku Dosen Pembimbing II, yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan memberi masukan selama proses pengerjaan Skripsi ini. Penulis juga mengucapkan terima kasih kepada Bapak/Ibu dosen penguji atas pengujian, saran, dan perbaikan yang diberikan. Selain itu, penulis mengucapkan terima kasih kepada pihak pengelola jaringan Universitas Palangka Raya yang telah mengizinkan penulis untuk menggunakan data log peladen sebagai objek pada penelitian ini.

Penulis menyadari bahwa laporan Skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan pendapat, kritik, dan saran yang membangun terhadap laporan ini. Dengan selesainya laporan Skripsi ini, penulis berharap hasilnya dapat memberikan manfaat serta wawasan baru bagi para pembaca, khususnya di bidang keamanan jaringan komputer.

Palangka Raya, 23 – Juli - 2026

ARICO SETIAWAN

223020503125

ANALISIS EFEKTIVITAS FIREWALL DINAMIS BERBASIS LINUX TERHADAP SERANGAN SIBER BERDASARKAN DATA LOG SERVER UNIVERSITAS PALANGKA RAYA

Arico Setiawan (223020503125)

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya

Kampus Tunjung Nyaho Jalan Yos Sudarso, Palangka Raya 73112

Email : aricosetiawan@mhs.eng.upr.ac.id

ABSTRAK

Infrastruktur peladen Universitas Palangka Raya (UPR) mengalami kerentanan pasca-migrasi sistem, yang dibuktikan dengan lonjakan anomali hingga lebih dari satu juta galat pada log peladen sehingga memicu kelumpuhan soket PHP-FPM. Pendekatan keamanan statis konvensional terbukti kurang efektif dalam membedakan antara serangan siber otomatis (*bot*) dan lalu lintas sah mahasiswa yang masif (*Flash Crowd*). Oleh karena itu, penelitian ini bertujuan untuk menganalisis efektivitas penerapan *firewall* dinamis berbasis Linux guna melindungi infrastruktur portal akademik dari agresi siber secara seketika (*real-time*).

Metodologi penelitian yang digunakan adalah *Action Research* yang mencakup tahapan *diagnosing, planning, taking,* dan *evaluating*. Sistem dibangun menggunakan arsitektur pertahanan berlapis (*Defense in Depth*) yang mengintegrasikan *Nginx Reverse Proxy*, *Uncomplicated Firewall* (UFW), dan *Fail2ban* sebagai *Intrusion Prevention System* (IPS). Penentuan parameter keamanan didasarkan pada metode *Statistical Anomaly Detection* menggunakan aturan 2-Sigma dan *Behavioral Profiling* terhadap data log historis peladen. Pengujian kinerja sistem dievaluasi melalui simulasi serangan *brute force*, pemindaian porta (*port scanning*), dan *HTTP Flood* (DoS) di lingkungan virtual yang terisolasi.

Hasil penelitian menunjukkan bahwa integrasi arsitektur *firewall* dinamis terbukti sangat efektif. Berdasarkan pengujian kinerja, sistem mencapai nilai *Detection Rate* (DR) sebesar 100% dengan rata-rata latensi pemblokiran (*Response Time*) di bawah 1 detik. Penerapan teknik *Behavioral Profiling* dengan penyesuaian parameter *maxretry* berhasil meniadakan pemblokiran salah sasaran (*False Positive*) pada lalu lintas mahasiswa yang sah. Selain itu, integrasi peringatan dini menggunakan Telegram Bot API dan dasbor berbasis *Smart Sync* (AJAX) terbukti berhasil meningkatkan visibilitas administrator dalam mengelola respons insiden secara efisien.

Kata kunci: Firewall Dinamis, Fail2ban, Log Server, *Behavioral Profiling*

ANALYSIS OF THE EFFECTIVENESS OF LINUX-BASED DYNAMIC FIREWALL AGAINST CYBER ATTACKS BASED ON SERVER LOG DATA FROM PALANGKA RAYA UNIVERSITY

Arico Setiawan (223020503125)

Department of Informatics Engineering, Faculty of Engineering, Palangka Raya University

Tunjung Nyaho Campus Jalan Yos Sudarso, Palangka Raya 73112

Email: aricosetiawan@mhs.eng.upr.ac.id

ABSTRACT

The server infrastructure of Palangka Raya University (UPR) experienced vulnerabilities post-system migration, evidenced by a surge of anomalies reaching over one million errors in the server logs, triggering the paralysis of the PHP-FPM socket. Conventional static security approaches proved ineffective in distinguishing between automated cyber-attacks (bots) and massive legitimate student traffic (Flash Crowd). Therefore, this research aims to analyze the effectiveness of implementing a Linux-based dynamic firewall to protect the academic portal infrastructure from cyber aggression in real-time.

The research methodology used is Action Research, which includes the stages of diagnosing, planning, taking, and evaluating. The system is built using a Defense in Depth architecture that integrates Nginx Reverse Proxy, Uncomplicated Firewall (UFW), and Fail2ban as an Intrusion Prevention System (IPS). The determination of security parameters is based on the Statistical Anomaly Detection method using the 2-Sigma rule and Behavioral Profiling on historical server log data. System performance was evaluated through simulated brute force attacks, port scanning, and HTTP Floods (DoS) in an isolated virtual environment.

The research results indicate that the integration of the dynamic firewall architecture is proven to be highly effective. Based on performance testing, the system achieved a Detection Rate (DR) of 100% with an average blocking latency (Response Time) of less than 1 second. The implementation of the Behavioral Profiling technique with maxretry parameter adjustments successfully eliminated False Positives on legitimate student traffic. Furthermore, the integration of an early warning system using the Telegram Bot API and a Smart Sync (AJAX) based dashboard successfully enhanced administrator visibility in managing incident responses efficiently.

Keywords: Dynamic Firewall, Fail2ban, Server Logs, Behavioral Profiling

DAFTAR ISI

HALAMAN PENGESAHAN……ii

HALAMAN PERNYATAAN.....iii

RIWAYAT PENYUSUN iv

HALAMAN PERSEMBAHAN.....v

KATA PENGANTAR.....vi

ABSTRAK.....vii

ABSTRACT.....viii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Batasan Masalah 3

1.4 Tujuan 3

1.5 Manfaat 4

1.6 Sistematika Penulisan.....4

1.7 Jadwal Skripsi 6

BAB II LANDASAN TEORI 7

2.1 Tinjauan Pustaka 7

2.2 Teori-teori Pendukung.....10

2.2.1 Keamanan Jaringan 10

2.2.2 Firewall Linux (iptables) 10

2.2.3 Protokol SSH dan Keamanannya 11

2.2.4 Jenis Serangan Siber pada Log Server.....11

2.2.5 Intrusion Detection System (IDS) 11

2.2.6 Analisis Log dan Audit Sistem 12

2.2.7 Parameter Efektivitas Keamanan 12
2.2.8 Uncomplicated Firewall (UFW) 13
2.2.9 Web Server dan Reverse Proxy (Nginx) 13
2.2.10 Notifikasi Real-time dan Telegram Bot API 13
2.2.11 Statistical Anomaly Detection dan Aturan Sigma 14
2.2.12 *Network Address Translation* (NAT) dan *Flash Crowd* 14
BAB III METODOLOGI PENELITIAN 16
3.1 Alur Penelitian 16
3.1.1 Diagram Alir Penelitian 16
3.2 Alat dan Bahan 18
3.2.1 Perangkat Keras (Hardware) 18
3.2.2 Perangkat Lunak (Software) 19
3.3 Bahan Penelitian (Datasets) 20
3.4 Metode Pengumpulan Data 20
3.5 Perancangan Sistem dan Implementasi Konfigurasi 21
3.5.1 Perancangan Arsitektur Web dan Reverse Proxy 21
3.5.2 Konfigurasi Dinding Api (Uncomplicated Firewall / UFW) 21
3.5.3 Konfigurasi Intrusion Prevention System (Fail2ban) 22
3.5.4 Implementasi Alarm Real-time Berbasis Telegram API 22
3.5.5 Analisis Kebutuhan Fungsional (Use Case Diagram) 23
3.5.6 Pemodelan Alur Sistem (*Activity Diagram*) 24
3.5.7 Perancangan Layanan UPR Security Dashboard 26
3.6 Skenario Pengujian Serangan 28
3.7 Teknik Analisis Data dan Evaluasi 28
3.7.1 Pemrosesan Awal Data Log (Data Pre-processing) 29
3.7.2 Analisis Forensik Jaringan 29

3.7.3 Pengujian Performa Keamanan (Performance Testing) 30
3.7.4 Metrik Evaluasi Efektivitas 30
3.7.5 Analisis Hasil dan Penarikan Kesimpulan 31
3.7.6 Intelijen Ancaman dan API Geolokasi 31
BAB IV HASIL DAN PEMBAHASAN 32
4.1 Analisis Karakteristik dan Struktur Log Server 32
4.1.1 Anatomi Nginx Access Log (access.log) 32
4.1.2 Anatomi Nginx Error Log (error.log) 34
4.1.3 Visualisasi Distribusi Log dan Dampak Migrasi Sistem 35
4.1.4 Analisis Statistik Anomali Log 36
4.1.5 Analisis *Behavioral Profiling* dan Penyesuaian Ambang Batas (*Maxretry*) 38
4.2 Hasil Implementasi Sistem 39
4.3.1 Integrasi Intelijen Ancaman Berbasis API Geolokasi 45
4.3.2 Implementasi Eksekusi Aksi Jarak Jauh (*Ban, Unban*, dan *Whitelist*) 47
4.3.3 Pembaruan Data Asinkron Berbasis AJAX 47
4.3.4 Perancangan Integrasi Sistem Peringatan Dini (Telegram Bot API) 48
4.3.5 Hasil Implementasi Antarmuka Sistem 52
4.4 Hasil Pengujian Keamanan Sistem (*Black-Box Testing*) 54
4.4 Pembahasan Hasil Parameter 58
4.5 Hasil Pengujian dan Analisis Kinerja Sistem 58
4.5.1 Hasil Pengujian Latensi Layanan SSH 59
4.5.2 Hasil Pengujian Latensi Layanan Nginx 60
4.5.3 Pembahasan dan Perbandingan Performa 61
4.6 Simulasi Serangan dan Analisis Komparatif 62
4.6.1 Simulasi Brute Force SSH (Otomatis) 62
4.6.2 Simulasi HTTP Flood (Nginx) 63

4.6.3 Analisis Perbandingan dengan Dataset Eksternal 63
4.6.4 Karakteristik IP Mencurigakan.....66
4.6.5 Perbandingan Kinerja Sebelum dan Sesudah Implementasi 66
BAB V KESIMPULAN.....71
5.1 Kesimpulan 71
5.2 Saran 72
DAFTAR PUSTAKA.....74
LAMPIRAN.....76

DAFTAR TABEL

Tabel 1.1 Matriks Jadwal Kegiatan Skripsi 6

Tabel 2.1 Perbandingan Penelitian Terdahulu.....9

Tabel 3.1 Daftar Perangkat Keras……...18

Tabel 3.2 Daftar Perangkat Lunak.....19

Tabel 4.1 Anatomi dan Parameter Nginx Access Log.....32

Tabel 4.2 Anatomi dan Parameter Nginx Error Log 34

Tabel 4.3 Skenario dan Hasil Pengujian Keamanan Jaringan 55

Tabel 4.4 Hasil Pengujian Latensi Deteksi Serangan SSH 59

Tabel 4.5 Hasil Pengujian Latensi Deteksi Serangan Nginx.....60

Tabel 4.6 Perbandingan Parameter Serangan: Simulasi Mandiri vs Dataset Eksternal 64

Tabel 4.7 Komparasi Kinerja Sistem Sebelum dan Sesudah Implementasi 66

DAFTAR GAMBAR

Gambar 3.1 Diagram Use Case UPR Security Dashboard 23

Gambar 3.2 *Activity Diagram* UPR *Security Dashboard* 25

Gambar 4.1 Distribusi Volume Log Galat per Bulan.....35

Gambar 4.2 Bukti Aktivitas Refresh Halaman Tertinggi 38

Gambar 4.3 Arsitektur Topologi Jaringan Universitas Palangka Raya.....40

Gambar 4.4 Status Konfigurasi Uncomplicated Firewall (UFW).....43

Gambar 4.5 Status Layanan Fail2ban dan Daftar Jail.....43

Gambar 4.6 Konfigurasi Antarmuka Jaringan Peladen (Target) 44

Gambar 4.7 Tampilan Halaman Autentikasi Dasbor Web 52

Gambar 4.8 Tampilan Halaman Utama Dasbor Keamanan.....53

Gambar 4.9 Tampilan Halaman Pemantauan Riwayat Log.....53

Gambar 4.10 Tampilan Peringatan Dini dan Tombol Aksi pada Telegram.....54

Gambar 4.11 Simulasi Serangan HTTP Flood menggunakan ApacheBench.....57

Gambar 4.12 Hasil Pemindaian Port menggunakan Nmap dari Mesin.....57

Gambar 4.13 Pencatatan Waktu Pemicu Serangan pada auth.log.....59

Gambar 4.14 Pencatatan Waktu Eksekusi Pemblokiran pada fail2ban.log.....60

Gambar 4.15 Pencatatan Log Akses Nginx (Pemicu/Trigger Serangan) 61

Gambar 4.16 Simulasi HTTP Flood Sebelum Implementasi (Sistem Tanpa Proteksi) 68

Gambar 4 17 Simulasi Akses Ilegal (Brute Force) Sebelum Implementasi 69

Gambar 4 18 Simulasi HTTP Flood Sesudah Implementasi (Sistem Terlindungi) 69

Gambar 4 19 Simulasi Akses Ilegal (Brute Force) Sesudah Implementasi (Sistem Terlindungi) 70

DAFTAR PUSTAKA

Al Amien, J. (2020). Implementasi Keamanan Jaringan Dengan Iptables Sebagai Firewall Menggunakan Metode Port Knocking. *Jurnal Fasilkom*, 10(2), 159-165.

Ariyadi, T., Pohan, M. R., Hadi, M. K., & Widodo, A. A. (2023). Implementasi Firewall Pada Protokol SSH Linux Ubuntu Menggunakan Iptables. *Seminar Riset Mahasiswa - Computer & Electrical (SERIMA-CE)*, 1(1), 170-175.

Az Zahra, D. R., Ilham, F. P., Ramdhani, H. N., & Setiawan, A. (2024). Penerapan dan Pengujian Keamanan SSH Pada Server Linux menggunakan Hydra. *Journal of Internet and Software Engineering*, 1(3), 1-10. https://doi.org/10.47134/pjise.v1i3.2627

Badan Siber dan Sandi Negara. (2025). *Peraturan Badan Siber dan Sandi Negara Nomor 5 Tahun 2025 tentang Peta Jalan Pelindungan Infrastruktur Informasi Vital Sektor Administrasi Pemerintahan Tahun 2025-2029*.

Batistuta, A. D., Hendrawan, A. H., & Ritzkal. (2024). Analisis Keamanan Jaringan Server Terhadap Serangan Dictionary Menggunakan Tools Fail2ban Dengan Notifikasi Telegram. *INFOTECH Journal*, 10(1), 64-73. https://doi.org/10.31949/infotech.v10i1.8730

Muse, B. A., & Abebe, S. L. (2020). Application Layer DDOS Attack Detection in the Presence of Flash Crowd. *Journal of EEA*, 38, 75-91.

Muzakki, Tahir, M., Khoironi, Romadhoni, A. B., & Ridwan, M. (2025). Simulasi dan Evaluasi Sistem IPS/IDS Berbasis Honeypot dan Firewall pada Server Ubuntu. *Jurnal Restikom: Riset Teknik Informatika dan Komputer*, 7(1), 106-110.

Romanov, M. A., Badalin, A. O., Rybalka, D. V., Borisov, V. V., & Novikova, D. D. (2025). Методы повышения безопасности при администрировании

серверов Linux [Methods for improving security when administering Linux servers]. *Программные системы и вычислительные методы [Software Systems and Computational Methods]*, (4), 143-157. https://doi.org/10.7256/2454-0714.2025.4.77322

Suryadin, A., Latif, K. A., Widyawati, L., Azhar, R., & Azwar, M. (2025). Evaluasi Penerapan Pertahanan Proaktif Waf Dan Hids Terhadap Eksploitasi Kerentanan Aplikasi Web. *Seminar Nasional Corisindo*, 491-500.

Nakamura, C., Toda, M., Sakamoto, M., Suzumura, N., & Toda, H. (2026). Extending the Three-Sigma Limit for Non-Gaussian Data Using Extreme Value Theory. *IEEE Access*, 14, 18222-18234. https://doi.org/10.1109/ACCESS.2026.3654980[cite: 14]

LAMPIRAN

Skrip Konfigurasi Aksi Fail2ban (/etc/fail2ban/action.d/telegram.conf)

```
[Definition]
actionban = /usr/bin/python3 /usr/local/bin/send_ban_notif.py
<ip> <name>

actionunban = curl -s -X POST
"https://api.telegram.org/bot<token>/sendMessage" -d
chat_id=<chat_id> -d text="✅ Alert: IP <ip> telah dilepas dari
penjara <name>."

[Init]
token = 8613892626:AAFc2T6s51BAxiIrF89NE_0mVs6ensYhKWM
chat_id = 8138521116
```

Skrip Pengirim Notifikasi Ban (/usr/local/bin/send_ban_notif.py)

```
import sys
import json
import os
import urllib.request
import urllib.parse

TOKEN = '8613892626:AAFc2T6s51BAxiIrF89NE_0mVs6ensYhKWM'
ADMIN_CHAT_ID = '8138521116'
SAFE_IPS_FILE = '/etc/fail2ban/safe_ips.json'

def send_telegram_notif(ip, jail):
    safe_ips = {}
    if os.path.exists(SAFE_IPS_FILE):
        try:
            with open(SAFE_IPS_FILE, 'r') as f:
                safe_ips = json.load(f)
        except Exception:
            pass

    peringatan_tambahan = ""
    if ip in safe_ips:
        keterangan = safe_ips[ip]
        peringatan_tambahan = f"\n\n⚠️ PERINGATAN KRITIKAL:\nIP
ini
        terdaftar sebagai PENGGUNA SAH!\nKeterangan:
        {keterangan}\nKemungkinan ini adalah False Positive."

    pesan = f"🚨 Fail2ban Alert!\n\nJail: {jail}\nIP: {ip} baru
saja
    diblokir.{peringatan_tambahan}"
```

```
    keyboard = {
        "inline_keyboard": [
            [{"text": "✅ Unban IP Ini", "callback_data":
            f"unban_{ip}"}],
            [{"text": "⭐ Tandai Pengguna Sah",
"callback_data":
            f"marksafe_{ip}"}]
        ]
    }
    reply_markup = json.dumps(keyboard)

    url = f"https://api.telegram.org/bot{TOKEN}/sendMessage"
    data = urllib.parse.urlencode({
        'chat_id': ADMIN_CHAT_ID,
        'text': pesan,
        'reply_markup': reply_markup
    }).encode('utf-8')

    try:
        req = urllib.request.Request(url, data=data)
        urllib.request.urlopen(req)
    except Exception as e:
        print(f"Gagal kirim notif: {e}")

if __name__ == '__main__':
    if len(sys.argv) > 2:
        send_telegram_notif(sys.argv[1], sys.argv[2])
```

Skrip Agen Pendengar Bot Telegram (/usr/local/bin/bot_unban.py)

```
import urllib.request
import urllib.parse
import json
import time
import subprocess
import os

TOKEN = '8613892626:AAFc2T6s51BAxiIrF89NE_0mVs6ensYhKWM'
ADMIN_CHAT_ID = '8138521116'
SAFE_IPS_FILE = '/etc/fail2ban/safe_ips.json'

def load_safe_ips():
    if os.path.exists(SAFE_IPS_FILE):
        try:
            with open(SAFE_IPS_FILE, 'r') as f:
                return json.load(f)
        except Exception:
            return {}
    return {}

def save_safe_ips(data):
    with open(SAFE_IPS_FILE, 'w') as f:
        json.dump(data, f, indent=4)

def unban_ip(ip, jail="sshd"):
```

```
    cmd = ["fail2ban-client", "set", jail, "unbanip", ip]
    res = subprocess.run(cmd, capture_output=True, text=True)
    return res.returncode == 0

def manual_ban(ip, jail="sshd"):
    cmd = ["fail2ban-client", "set", jail, "banip", ip]
    res = subprocess.run(cmd, capture_output=True, text=True)
    return res.returncode == 0

def get_all_banned_ips():
    try:
        res = subprocess.run(["fail2ban-client", "status"],
        capture_output=True, text=True)
        line = [l for l in res.stdout.split("\n") if "Jail
list" in l][0]
        jails = [j.strip() for j in line.split(":")[-
1].split(",")]

        banned_data = []
        for jail in jails:
            status = subprocess.run(["fail2ban-client",
"status", jail],
            capture_output=True, text=True)
            for s_line in status.stdout.split("\n"):
                if "Banned IP list" in s_line:
                    ips = s_line.split(":")[-1].strip().split()
                    for ip in ips:
                        banned_data.append({"ip": ip, "jail":
jail})
        return banned_data
    except Exception:
        return []

def send_msg(chat_id, text, reply_markup=None):
    url = f"https://api.telegram.org/bot{TOKEN}/sendMessage"
    payload = {'chat_id': chat_id, 'text': text, 'parse_mode':
    'Markdown'}
    if reply_markup:
        payload['reply_markup'] = reply_markup
    data = urllib.parse.urlencode(payload).encode('utf-8')
    try:
        urllib.request.urlopen(urllib.request.Request(url,
data=data))
    except Exception: pass

def edit_msg(chat_id, message_id, text):
    url =
f"https://api.telegram.org/bot{TOKEN}/editMessageText"
    payload = {
        'chat_id': chat_id,
        'message_id': message_id,
        'text': f"✅ *Riwayat Aksi:*\n{text}",
        'parse_mode': 'Markdown'
    }
    data = urllib.parse.urlencode(payload).encode('utf-8')
    try:
```

```
        urllib.request.urlopen(urllib.request.Request(url,
data=data))
    except Exception: pass

def main():
    offset = None
    menu_keyboard = json.dumps({
        "keyboard": [
            [{"text": "📋 Daftar IP Sah"}, {"text": "🚫 Daftar
IP Terblokir"}],
            [{"text": "➕ Tambah Manual"}]
        ],
        "resize_keyboard": True
    })

    while True:
        try:
            url =
f"https://api.telegram.org/bot{TOKEN}/getUpdates?timeout=100"
            if offset: url += f"&offset={offset}"
            req = urllib.request.urlopen(url)
            updates = json.loads(req.read())

            for item in updates.get("result", []):
                offset = item["update_id"] + 1

                if "message" in item and "text" in
item["message"]:
                    msg = item["message"]
                    chat_id = str(msg["chat"]["id"])
                    text = msg["text"]

                    if chat_id == ADMIN_CHAT_ID:
                        if text in ["/start", "/menu"]:
                            send_msg(chat_id, "🛡️ *Panel
Kontrol Fail2ban
                            Aktif*\nSilakan gunakan menu di
bawah.",
                            menu_keyboard)

                        elif text == "📋 Daftar IP Sah":
                            safe_ips = load_safe_ips()
                            if not safe_ips:
                                send_msg(chat_id, "📂 Daftar IP
Sah masih
                                kosong.", menu_keyboard)
                            else:
                                inline_kb = [[{"text": f"❌
Hapus {ip}",
                                "callback_data":
f"delsafe_{ip}"}] for ip
                                in safe_ips]
                                markup =
json.dumps({"inline_keyboard":
                                inline_kb})
```

```
                                send_msg(chat_id, "📂 *Daftar
IP Sah:*",
                                markup)

                        elif text == "🚫 Daftar IP Terblokir":
                            banned_list = get_all_banned_ips()
                            if not banned_list:
                                send_msg(chat_id, "✅ Tidak ada
IP yang
                                sedang terblokir saat ini.",
                                menu_keyboard)
                            else:
                                inline_kb = [[{"text": f"🔓
Unban
                                {b['ip']} ({b['jail']})",
                                "callback_data":
f"unban_{b['ip']}_{b['jail']}"}] for b in
                                banned_list]
                                markup =
json.dumps({"inline_keyboard":
                                inline_kb})
                                send_msg(chat_id, "🚫 *IP yang
Sedang
                                Terblokir:*\nKlik tombol untuk
                                membebaskan:", markup)

                        elif text == "➕ Tambah Manual":
                            send_msg(chat_id, "Ketik perintah
dengan
                            format:\n`TAMBAH <IP>
                            <KETERANGAN>`\n\nContoh:\n`TAMBAH
192.168.1.5
                            Laptop_Arico`", menu_keyboard)

                        elif text.startswith("TAMBAH "):
                            parts = text.split(" ", 2)
                            if len(parts) >= 3:
                                ip_add, ket = parts[1],
parts[2]
                                safe_ips = load_safe_ips()
                                safe_ips[ip_add] = ket
                                save_safe_ips(safe_ips)
                                send_msg(chat_id, f"✅ IP
`{ip_add}`
                                berhasil ditambahkan sebagai:
{ket}",
                                menu_keyboard)

                if "callback_query" in item:
                    cb = item["callback_query"]
                    cb_id = cb["id"]
                    data = cb.get("data", "")
                    c_id = cb["message"]["chat"]["id"]
                    m_id = cb["message"]["message_id"]
```

```
                    if str(cb["from"]["id"]) == ADMIN_CHAT_ID:
                        result_text = ""

                        if data.startswith("unban_"):
                            parts = data.split("_")
                            ip, jail = parts[1], parts[2] if
len(parts) >
                            2 else "sshd"
                            success = unban_ip(ip, jail)
                            result_text = f"IP {ip} telah
dibebaskan dari
                            jail {jail}." if success else
f"Gagal
                            membebaskan IP {ip}."

                        elif data.startswith("delsafe_"):
                            ip = data.split("_")[1]
                            safe_ips = load_safe_ips()
                            if ip in safe_ips:
                                del safe_ips[ip]
                                save_safe_ips(safe_ips)
                                result_text = f"IP {ip} telah
dihapus
                                dari daftar aman."

                        elif data.startswith("manualban_"):
                            ip = data.split("_")[1]
                            success = manual_ban(ip)
                            result_text = f"IP {ip} telah
diblokir secara
                            manual oleh Admin." if success else
f"Gagal
                            memblokir IP {ip}."

                        elif data.startswith("marksafe_"):
                            ip = data.split("_")[1]
                            safe_ips = load_safe_ips()
                            safe_ips[ip] = "Ditandai via Tombol
Notif"
                            save_safe_ips(safe_ips)
                            result_text = f"IP {ip} berhasil
dimasukkan
                            ke daftar aman."

                        if result_text:

urllib.request.urlopen(f"https://api.teleg

ram.org/bot{TOKEN}/answerCallbackQuery

?callback_query_id={cb_id}&text={urllib.

parse.quote(result_text)}&show_alert=true")
                            edit_msg(c_id, m_id, result_text)
```

```
        except Exception: pass
        time.sleep(2)

if __name__ == '__main__':
    main()
```

Kode Antarmuka Utama dan JavaScript (index.html)

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>UPR Network Dashboard</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/
    bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.datatables.net/1.13.6/css/
    dataTables.bootstrap5.min.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
    icons@1.11.1/font/bootstrap-icons.css">
    <style>
        .card-stat { transition: transform 0.2s; }
        .live-dot { height: 10px; width: 10px; background-
color: #28a745;
        border-radius: 50%; display: inline-block; animation:
blink 1s
        infinite; }
        @keyframes blink { 0% { opacity: 1; } 50% { opacity:
0.3; } 100%
        { opacity: 1; } }
    </style>
</head>
<body class="bg-light">

    <nav class="navbar navbar-dark bg-dark mb-4 shadow-sm">
        <div class="container">
            <span class="navbar-brand fw-bold">
                <i class="bi bi-shield-lock-fill text-
primary"></i> UPR
                Security Dashboard
                <small class="ms-3 text-muted" style="font-
size:
                0.6em;"><span class="live-dot"></span> LIVE
MODE</small>
            </span>
            <div>
                <a href="/logs" class="btn btn-outline-info
btn-sm me-
                2"><i class="bi bi-journal-text"></i> Log
Server</a>
                <a href="/logout" class="btn btn-outline-danger
btn-
```

```
                sm"><i class="bi bi-box-arrow-right"></i>
Logout
                ({{ user }})</a>
            </div>
        </div>
    </nav>

    <div class="container">

        {% with messages =
get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{ category }}
alert-
                    dismissible fade show shadow-sm"
role="alert">
                        {{ message }}
                        <button type="button" class="btn-close"
data-bs-
                        dismiss="alert" aria-
label="Close"></button>
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}

        <div class="row mb-4">
            <div class="col-md-4 mb-3">
                <div class="card bg-primary text-white shadow
card-stat">
                    <div class="card-body d-flex align-items-
center">
                        <div class="display-5 me-3"><i
class="bi bi-
                        person-x-fill"></i></div>
                        <div>
                            <h6 class="card-title mb-0
small">Penyerang
                            Unik</h6>
                            <h2 class="fw-bold mb-0" id="stat-
total">{{
                            total }} IP</h2>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-4 mb-3">
                <div class="card bg-warning text-dark shadow
card-stat">
                    <div class="card-body d-flex align-items-
center">
                        <div class="display-5 me-3"><i
class="bi bi-
                        exclamation-octagon-fill"></i></div>
                        <div>
```

```
                            <h6 class="card-title mb-0
small">Penjara
                            Teraktif</h6>
                            <h2 class="fw-bold mb-0" id="stat-
sibuk">{{
                            sibuk }}</h2>
                        </div>
                    </div>
                </div>
            </div>
            <div class="col-md-4 mb-3">
                <div class="card id-status-card text-white
shadow card-
                stat {% if status == 'Aktif' %}bg-success{%
else %}bg-
                danger{% endif %}" id="status-card">
                    <div class="card-body d-flex align-items-
center">
                        <div class="display-5 me-3"><i
class="bi bi-cpu-
                        fill"></i></div>
                        <div>
                            <h6 class="card-title mb-0
small">Status
                            Fail2ban</h6>
                            <h2 class="fw-bold mb-0" id="stat-
status">{{
                            status }}</h2>
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-lg-5">

                <div class="card shadow mb-4 border-success">
                    <div class="card-header bg-success text-
white fw-
                    bold"><i class="bi bi-shield-check"></i>
                    Whitelist</div>
                    <div class="card-body">
                        <form action="/whitelist/add"
method="POST"
                        class="mb-3">
                            <div class="row g-2">
                                <div class="col-md-5">
                                    <input type="text"
                                    name="whitelist_ip"
class="form-
                                    control border-success"
                                    placeholder="IP Address..."
required>
                                </div>
                                <div class="col-md-5">
```

```
                                    <input type="text"
                                    name="whitelist_note"
class="form-
                                    control border-success"

placeholder="Nama/Keterangan..."
                                    required>
                                </div>
                                <div class="col-md-2">
                                    <button type="submit"
class="btn btn-
                                    success w-100 fw-
bold">Add</button>
                                </div>
                            </div>
                        </form>

                        <div class="table-responsive"
style="max-height:
                        200px; overflow-y: auto;">
                            <table class="table table-sm table-
hover
                            align-middle border">
                                <thead class="table-light"
                                style="position: sticky; top:
0;">
                                    <tr>
                                        <th>IP
Dikecualikan</th>
                                        <th>Keterangan</th>
                                        <th class="text-
end">Aksi</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    {% for item in
whitelist_ips %}
                                    <tr>
                                        <td><strong
class="text-
                                        success">{{ item.ip
}}</strong>
                                        </td>
                                        <td><small class="text-
muted">{{
                                        item.note
}}</small></td>
                                        <td class="text-end"><a
href="{{
                                        url_for('remove_whitelist',
                                        ip=item.ip) }}"
class="btn btn-
                                        outline-danger btn-sm
py-
                                        0">Hapus</a></td>
```

```
                                    </tr>
                                    {% else %}
                                    <tr>
                                        <td colspan="3"
class="text-
                                        center text-muted
small">Belum
                                        ada IP di
Whitelist</td>
                                    </tr>
                                    {% endfor %}
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>

                <div class="card shadow mb-4 border-danger">
                    <div class="card-header bg-danger text-
white fw-
                    bold"><i class="bi bi-hammer"></i> Blokir
                    Manual</div>
                    <div class="card-body">
                        <form action="/ban" method="POST">
                            <div class="mb-2">
                                <input type="text"
name="ip_address"
                                class="form-control border-
danger"
                                placeholder="Ketik IP
Penyerang..."
                                required>
                            </div>
                            <div class="mb-2">
                                <select name="jail_name"
class="form-
                                select border-danger" required>
                                    <option value="" disabled
selected>--
                                    Pilih Penjara Target --
</option>
                                    <option value="sshd">sshd
(Brute
                                    Force)</option>
                                    <option value="ufw-
portscan">ufw-
                                    portscan (Port
Scanning)</option>
                                    <option value="nginx-502-
upr">nginx-
                                    502-upr (Serangan
Web)</option>
                                </select>
                            </div>
                            <button type="submit" class="btn
btn-danger
```

```
                            w-100 fw-bold">Eksekusi
Blokir!</button>
                        </form>
                    </div>
                </div>

            </div>

            <div class="col-lg-7">
                <div class="card shadow mb-5">
                    <div class="card-header bg-dark text-white
fw-
                    bold"><i class="bi bi-broadcast"></i>
Daftar IP
                    Terblokir</div>
                    <div class="card-body">
                        <table id="tabelLog" class="table
table-hover
                        align-middle border">
                            <thead class="table-light">
                                <tr>
                                    <th>IP Target</th>
                                    <th>Asal Negara</th>
                                    <th>Status Penjara</th>
                                    <th>Aksi</th>
                                </tr>
                            </thead>
                            <tbody id="table-body">
                                {% for item in banned_ips %}
                                <tr>
                                    <td><strong class="text-
danger">{{
                                    item.ip }}</strong></td>
                                    <td><i class="bi bi-geo-
alt-fill
                                    text-secondary"></i> {{
item.country
                                    }}</td>
                                    <td><span class="badge bg-
warning
                                    text-dark">{{ item.jail
                                    }}</span></td>
                                    <td><a href="{{
url_for('unban',
                                    jail=item.jail, ip=item.ip)
}}"
                                    class="btn btn-success btn-
sm fw-bold
                                    py-0">Bebaskan</a></td>
                                </tr>
                                {% endfor %}
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
```

```
        </div>
    </div>

    <script src="https://code.jquery.com/jquery-
3.7.0.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist
    /js/bootstrap.bundle.min.js"></script>
    <script
src="https://cdn.datatables.net/1.13.6/js/jquery.dataTable
    s.min.js"></script>
    <script
src="https://cdn.datatables.net/1.13.6/js/dataTables
    .bootstrap5.min.js"></script>

    <script>
        $(document).ready(function() {
            // 1. Fungsi Inisialisasi DataTables
            function initTable() {
                return $('#tabelLog').DataTable({
                    "language": { "url":
"//cdn.datatables.net/plug-
            ins/1.13.6/i18n/id.json" },
                    "pageLength": 5,
                    "lengthMenu": [5, 10, 25, 50]
                });
            }

            let table = initTable();

            // 2. Failsafe: Fungsi Manual Menutup Pop-up Alert
            $(document).on('click', '.btn-close', function() {
                $(this).closest('.alert').fadeOut(300,
function() {
                    $(this).remove();
                });
            });

            // 3. Fungsi Real-time Update (Statistik & Tabel)
            function updateData() {
                let cacheBuster = new Date().getTime();

                // A. Update Statistik Atas
                fetch('/api/stats?_=' + cacheBuster)
                    .then(response => response.json())
                    .then(data => {
                        $('#stat-total').text(data.total + "
IP");
                        $('#stat-sibuk').text(data.sibuk);
                        $('#stat-status').text(data.status);
                    })
                    .catch(error => console.error('Error
mengambil data
                    statistik:', error));

                // B. Update Tabel Daftar IP Terblokir
```

```
                fetch(window.location.pathname + '?_=' +
cacheBuster)
                    .then(response => response.text())
                    .then(html => {
                        let parser = new DOMParser();
                        let doc = parser.parseFromString(html,
                        'text/html');
                        let newTableBody =
doc.querySelector('#table-
                        body').innerHTML;
                        let oldTableBody =
                        document.getElementById('table-
body').innerHTML;

                        // Perbarui hanya jika ada perubahan
teks HTML
                        agar layar tidak berkedip
                        if (newTableBody.trim() !==
oldTableBody.trim())
                        {
                            let currentPage = table.page();
                            let currentSearch = table.search();

                            table.destroy();
                            document.getElementById('table-
                            body').innerHTML = newTableBody;
                            table = initTable();

                            table.search(currentSearch);

table.page(currentPage).draw('page');
                        }
                    })
                    .catch(error => console.error('Error
mengambil data
                    tabel:', error));
            }

            // Eksekusi pembaruan setiap 3 detik
            setInterval(updateData, 3000);
        });
    </script>
</body>
</html>
```

Kode Antarmuka Autentikasi (login.html)

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Login - UPR Security Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/
    dist/css/bootstrap.min.css" rel="stylesheet">
</head>
```

```
<body class="bg-light d-flex align-items-center justify-
content-center" style="height: 100vh;">
    <div class="card shadow p-4" style="width: 350px;">
        <div class="text-center mb-4">
            <h4>🛡️ UPR Security</h4>
            <span class="text-muted">Panel Admin
Fail2ban</span>
        </div>

        {% with messages = get_flashed_messages() %}
          {% if messages %}
            {% for message in messages %}
              <div class="alert alert-danger p-2 text-center"
              role="alert">
                {{ message }}
              </div>
            {% endfor %}
          {% endif %}
        {% endwith %}

        <form method="POST">
            <div class="mb-3">
                <label class="form-label">Username</label>
                <input type="text" name="username" class="form-
control"
                placeholder="Masukkan username" required>
            </div>
            <div class="mb-4">
                <label class="form-label">Password</label>
                <input type="password" name="password"
class="form-
                control" placeholder="Masukkan password"
required>
            </div>
            <button type="submit" class="btn btn-primary w-
            100">Masuk</button>
        </form>
    </div>
</body>
</html>
```

Kode Antarmuka Pemantauan Log (logs.html)

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Log Viewer - UPR Security Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/
    dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
    icons@1.11.1/font/bootstrap-icons.css">
    <link rel="stylesheet"
href="https://cdn.datatables.net/1.13.6/css/
```

```
    dataTables.bootstrap5.min.css">
    <link rel="stylesheet"
href="https://cdn.datatables.net/buttons/2.4.2
    /css/buttons.bootstrap5.min.css">
    <style>
        .dt-buttons { margin-bottom: 15px; }
        .btn-success { background-color: #198754 !important;
border:
        none; }
        .btn-danger { background-color: #dc3545 !important;
border: none;
        }
        /* Tambahan styling agar kotak filter di bawah tabel
terlihat
        rapi dan proporsional */
        tfoot input, tfoot select { width: 100%; box-sizing:
border-box;
        font-size: 0.85rem; }
    </style>
</head>
<body class="bg-light">

    <nav class="navbar navbar-dark bg-dark mb-4 shadow-sm">
        <div class="container">
            <span class="navbar-brand fw-bold"><i class="bi bi-
shield-
            lock-fill text-primary"></i> UPR Security
Dashboard</span>
            <div>
                <a href="/" class="btn btn-outline-light btn-sm
me-2"><i
                class="bi bi-house-door"></i> Kembali ke
Home</a>
                <a href="/logout" class="btn btn-outline-danger
btn-
                sm"><i class="bi bi-box-arrow-right"></i>
Logout ({{ user
                }})</a>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="card shadow mb-5">
            <div class="card-header bg-success text-white d-
flex justify-
            content-between align-items-center">
                <strong><i class="bi bi-journal-text"></i>
Catatan Sistem
                Forensik (Fail2ban Logs)</strong>
                <a href="/download/log" class="btn btn-light
btn-sm fw-
                bold shadow-sm">
                    <i class="bi bi-file-earmark-arrow-down
text-
                    success"></i> Download Raw Data (.log)
```

```
                </a>
            </div>
            <div class="card-body">
                <table id="tabelSistemLog" class="table table-
hover
                align-middle table-sm border">
                    <thead class="table-light">
                        <tr>
                            <th>Waktu Eksekusi</th>
                            <th>IP Address</th>
                            <th>Asal Negara</th>
                            <th>Modul Penjara</th>
                            <th>Detail Aktivitas</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for log in logs %}
                        <tr>
                            <td class="text-nowrap fw-bold">{{
                            log.datetime }}</td>
                            <td><strong class="text-danger">{{
log.ip
                            }}</strong></td>
                            <td><small><i class="bi bi-geo-alt-
fill text-
                            muted"></i> {{ log.country
}}</small></td>
                            <td><span class="badge bg-
secondary">{{
                            log.jail }}</span></td>
                            <td class="text-muted small">{{
log.message
                            }}</td>
                        </tr>
                        {% endfor %}
                    </tbody>
                    <tfoot class="table-light">
                        <tr>
                            <th><input type="text" class="form-
control
                            form-control-sm" placeholder="??
Cari
                            Waktu..."></th>
                            <th><input type="text" class="form-
control
                            form-control-sm border-danger"
                            placeholder="?? Cari IP..."></th>
                            <th><input type="text" class="form-
control
                            form-control-sm" placeholder="??
Cari
                            Negara..."></th>
                            <th><input type="text" class="form-
control
                            form-control-sm border-warning"
```

```
                                placeholder="?? Cari
Modul/PID..."></th>
                                <th><input type="text" class="form-
control
                                form-control-sm" placeholder="??
Cari
                                Detail/Jail..."></th>
                            </tr>
                        </tfoot>
                    </table>
                </div>
            </div>
        </div>

        <script src="https://code.jquery.com/jquery-
3.7.0.min.js"></script>
        <script src="https://cdn.datatables.net/1.13.6/js/jquery.
        dataTables.min.js"></script>
        <script
src="https://cdn.datatables.net/1.13.6/js/dataTables.
        bootstrap5.min.js"></script>
        <script src="https://cdn.datatables.net/buttons/2.4.2/
        js/dataTables.buttons.min.js"></script>
        <script
src="https://cdn.datatables.net/buttons/2.4.2/js/buttons.
        bootstrap5.min.js"></script>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.10.1/
        jszip.min.js"></script>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/
        pdfmake.min.js"></script>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/
        vfs_fonts.js"></script>
        <script src="https://cdn.datatables.net/buttons/2.4.2/
        js/buttons.html5.min.js"></script>

        <script>
            $(document).ready(function() {
                $('#tabelSistemLog').DataTable({
                    "language": { "url":
"//cdn.datatables.net/plug-
                    ins/1.13.6/i18n/id.json" },
                    "order": [[ 0, "desc" ]],
                    "dom": 'Bfrtip',
                    "buttons": [
                        {
                            extend: 'excelHtml5',
                            className: 'btn btn-success fw-bold
btn-sm
                        shadow-sm me-2',
                            text: '<i class="bi bi-file-earmark-
excel"></i>
                        Export Excel Filter'
                        },
```

```
                    {
                        extend: 'pdfHtml5',
                        className: 'btn btn-danger fw-bold btn-
sm shadow-
                        sm',
                        text: '<i class="bi bi-file-earmark-
pdf"></i>
                        Export PDF Filter',
                        orientation: 'landscape', // Agar muat
5 kolom di
                        PDF
                        pageSize: 'A4'
                    }
                ],
                // FUNGSI INTI UNTUK MENGAKTIFKAN FILTER DI
TFOOT
                initComplete: function () {
                    this.api().columns().every(function () {
                        let column = this;
                        let footer = $(column.footer());
                        let input = footer.find('input');

                        // Jika kolom tersebut memiliki input
text
                        if (input.length) {
                            input.on('keyup change clear',
function () {
                                if (column.search() !==
this.value) {

column.search(this.value).draw();
                                }
                            });
                        }
                    });
                }
            });
        });
    </script>
</body>
</html>
```

Kode Backend Dashboard Flask (app.py)

```
from flask import Flask, render_template, request, redirect,
session, flash, url_for, send_file, jsonify
import subprocess
import os
import requests
import re

app = Flask(__name__)
app.secret_key = 'UPR_Informatika_Arico_2026'
```

```
ADMIN_USERS = {"arico": "admin123", "dosen_ahli":
"rahasia_jaringan"}
WHITELIST_FILE = "whitelist.txt"

# Cache untuk menyimpan data negara agar web tidak lambat
karena terus-menerus request ke API
geo_cache = {}

def get_country(ip):
    """Mendeteksi negara asal IP menggunakan ip-api.com"""
    # Deteksi IP Privat/Lokal (VirtualBox/LAN)
    if ip.startswith("192.168.") or ip.startswith("10.") or i
    p.startswith("127."):
        return "Lokal (LAN)"

    if ip in geo_cache:
        return geo_cache[ip]

    try:
        response = requests.get(f"http://ip-
        api.com/json/{ip}?fields=status,country", timeout=2)
        data = response.json()
        if data.get('status') == 'success':
            country = data.get('country')
            geo_cache[ip] = country
            return country
        return "Unknown"
    except:
        return "Unknown"

def get_system_info():
    """Mengambil data IP terblokir, daftar penjara aktif, dan
negaranya"""
    all_banned = []
    active_jails = []
    try:
        res_jails = subprocess.run(["sudo", "fail2ban-client",
"status"], capture_output=True, text=True)
        if "Jail list:" in res_jails.stdout:
            jail_line = res_jails.stdout.split("Jail
list:")[1].strip()
            active_jails = [j.strip() for j in
jail_line.split(",")]

            for jail in active_jails:
                res_status = subprocess.run(["sudo", "fail2ban-
client", "status", jail], capture_output=True, text=True)
                for line in res_status.stdout.split('\n'):
                    if "Banned IP list" in line:
                        ips = line.split(':')[-
1].strip().split()
                        for ip in ips:
                            negara = get_country(ip)
                            all_banned.append({"ip": ip,
"jail": jail,
                            "country": negara})
```

```
        return all_banned, active_jails
    except Exception as e:
        print(f"Error tarik data: {e}")
        return [], []

def get_system_logs():
    """Membaca log, memisahkan IP, dan mencari negaranya untuk
diekspor ke Excel/PDF"""
    logs = []
    try:
        result = subprocess.run(["sudo", "tail", "-n", "200",
        "/var/log/fail2ban.log"], capture_output=True,
text=True)
        for line in result.stdout.strip().split('\n'):
            if line:
                parts = line.split()
                if len(parts) >= 5:
                    date_time = f"{parts[0]}
{parts[1].split(',')[0]}"
                    jail_or_info = parts[3]
                    message = " ".join(parts[4:])

                    # Ekstrak IP dari pesan log menggunakan
Regex
                    ip_match = re.search(r'\b(?:[0-
9]{1,3}\.){3}[0-
                    9]{1,3}\b', message)
                    extracted_ip = ip_match.group() if ip_match
else "-"
                    negara = get_country(extracted_ip) if
extracted_ip !=
                    "-" else "-"

                    logs.append({
                        "datetime": date_time,
                        "ip": extracted_ip,
                        "country": negara,
                        "jail": jail_or_info,
                        "message": message
                    })
        return logs[::-1]
    except Exception as e:
        print(f"Error baca log: {e}")
        return []

# === FUNGSI WHITELIST ===
def get_whitelist():
    """Membaca daftar IP whitelist dan catatannya dari file"""
    if not os.path.exists(WHITELIST_FILE):
        return []
    whitelist = []
    with open(WHITELIST_FILE, "r") as f:
        for line in f:
            line = line.strip()
            if line:
```

```
                # Format baru: IP|Keterangan
                if "|" in line:
                    ip, note = line.split("|", 1)
                    whitelist.append({"ip": ip.strip(), "note":
                    note.strip()})
                else:
                    # Kompatibilitas untuk file lama yang hanya
berisi IP
                    whitelist.append({"ip": line.strip(),
"note": "-"})
    return whitelist

def save_whitelist(whitelist_data):
    """Menyimpan daftar IP dan catatan whitelist ke file"""
    with open(WHITELIST_FILE, "w") as f:
        for item in whitelist_data:
            f.write(f"{item['ip']}|{item['note']}\n")

def sync_whitelist_to_fail2ban():
    """Menerapkan IP whitelist ke semua penjara (jails)
Fail2ban yang aktif"""
    whitelist = get_whitelist()
    if whitelist:
        _, active_jails = get_system_info()
        for jail in active_jails:
            for item in whitelist:
                # Ambil hanya 'ip'-nya saja untuk dikirim ke
Fail2ban
                subprocess.run(["sudo", "fail2ban-client",
"set", jail, "addignoreip", item['ip']], capture_output=True)
# ========================

@app.route('/')
def home():
    if 'user' not in session: return redirect(url_for('login'))

    # Sinkronisasi whitelist setiap kali halaman home dimuat
agar Fail2ban tidak lupa
    sync_whitelist_to_fail2ban()

    banned_data, jails = get_system_info()
    unique_ips = set(item['ip'] for item in banned_data)

    jail_counts = {}
    for item in banned_data:
        jail_counts[item['jail']] =
jail_counts.get(item['jail'], 0) + 1
    most_active_jail = max(jail_counts, key=jail_counts.get) if
    jail_counts else "-"

    check_service = subprocess.run(["systemctl", "is-active",
    "fail2ban"], capture_output=True, text=True)
    status_layanan = "Aktif" if "active" in
check_service.stdout else
    "Nonaktif"
```

```
    whitelist_ips = get_whitelist()

    return render_template('index.html',
banned_ips=banned_data,
    total=len(unique_ips),
                           sibuk=most_active_jail,
status=status_layanan,
                           user=session['user'],
                           whitelist_ips=whitelist_ips)

@app.route('/logs')
def view_logs():
    if 'user' not in session: return redirect(url_for('login'))
    log_data = get_system_logs()
    return render_template('logs.html', logs=log_data,
user=session['user'])

@app.route('/api/stats')
def api_stats():
    if 'user' not in session: return jsonify({"error":
"unauthorized"}),
    401
    banned_data, _ = get_system_info()
    unique_ips = set(item['ip'] for item in banned_data)

    jail_counts = {}
    for item in banned_data:
        jail_counts[item['jail']] =
jail_counts.get(item['jail'], 0) + 1
    most_active_jail = max(jail_counts, key=jail_counts.get) if
    jail_counts else "-"

    check_service = subprocess.run(["systemctl", "is-active",
    "fail2ban"], capture_output=True, text=True)
    status_layanan = "Aktif" if "active" in
check_service.stdout else
    "Nonaktif"

    return jsonify({
        "total": len(unique_ips), "sibuk": most_active_jail,
        "status": status_layanan, "banned_ips": banned_data
    })

@app.route('/ban', methods=['POST'])
def manual_ban():
    if 'user' not in session: return redirect(url_for('login'))
    ip = request.form.get('ip_address')
    jail = request.form.get('jail_name')
    subprocess.run(["sudo", "fail2ban-client", "set", jail,
"banip", ip])
    flash(f"IP {ip} berhasil dijebloskan ke {jail}!", "danger")
    return redirect(url_for('home'))

@app.route('/unban/<jail>/<ip>')
def unban(jail, ip):
    if 'user' not in session: return redirect(url_for('login'))
```

```
    subprocess.run(["sudo", "fail2ban-client", "set", jail,
"unbanip",
    ip])
    flash(f"IP {ip} sudah dibebaskan dari {jail}!", "success")
    return redirect(url_for('home'))

# === ROUTE WHITELIST ===
@app.route('/whitelist/add', methods=['POST'])
def add_whitelist():
    if 'user' not in session: return redirect(url_for('login'))
    ip = request.form.get('whitelist_ip')
    note = request.form.get('whitelist_note', '-') # Default '-
' jika tidak ada catatan

    if ip:
        whitelist = get_whitelist()
        # Cek apakah IP sudah ada di dalam list
        if not any(item['ip'] == ip for item in whitelist):
            whitelist.append({"ip": ip, "note": note})
            save_whitelist(whitelist)
            flash(f"IP {ip} ({note}) berhasil dimasukkan ke
Whitelist!", "success")
        else:
            flash(f"IP {ip} sudah berada di Whitelist.",
"warning")
    return redirect(url_for('home'))

@app.route('/whitelist/remove/<ip>')
def remove_whitelist(ip):
    if 'user' not in session: return redirect(url_for('login'))
    whitelist = get_whitelist()

    # Saring ulang daftar whitelist, buang item yang IP-nya
cocok
    new_whitelist = [item for item in whitelist if item['ip']
!= ip]

    # Jika jumlahnya berkurang, berarti IP berhasil dihapus
    if len(new_whitelist) < len(whitelist):
        save_whitelist(new_whitelist)

        # Hapus pengecualian dari semua penjara aktif
        _, active_jails = get_system_info()
        for jail in active_jails:
            subprocess.run(["sudo", "fail2ban-client", "set",
jail,
            "delignoreip", ip])

        flash(f"IP {ip} telah dihapus dari Whitelist dan
kembali
        diawasi.", "success")
    return redirect(url_for('home'))
# ======================

@app.route('/download/log')
def download_log():
```

```
    if 'user' not in session: return redirect(url_for('login'))
    return send_file("/var/log/fail2ban.log",
as_attachment=True)

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        user = request.form.get('username')
        pw = request.form.get('password')
        if user in ADMIN_USERS and ADMIN_USERS[user] == pw:
            session['user'] = user
            return redirect(url_for('home'))
        flash("Username atau Password salah!", "danger")

    return render_template('login.html')

@app.route('/logout')
def logout():
    session.pop('user', None)
    return redirect(url_for('login'))

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```