RANCANG BANGUN NETWORK TRAFFIC MONITORING DAN DDOS DETECTION MENGGUNAKAN FASTNETMON DAN GRAFANA PADA IOT SERVER

SKRIPSI

ARIF SETIAWAN 1907421003

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2023

RANCANG BANGUN NETWORK TRAFFIC MONITORING DAN DDOS DETECTION MENGGUNAKAN FASTNETMON DAN GRAFANA PADA IOT SERVER

SKRIPSI

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

ARIF SETIAWAN

1907421003

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2023

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini :

Nama : Arif Setiawan

NIM : 1907421003

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

Judul skripsi : Rancang Bangun Network Traffic Monitoring Dan DDoS Detection Menggunakan Fastnetmon Dan Grafana Pada Iot Server

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, 12 Juli 2023

Yang membuat pernyataan

(Arif Setiawan)

NIM.1907421003

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Arif Setiawan

NIM : 1907421003

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

Judul skripsi : Rancang Bangun Network Traffic Monitoring Dan DDoS Detection Menggunakan Fastnetmon Dan Grafana Pada Iot Server

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari **Jum'at**, Tanggal **28**, Bulan **Juli**, Tahun **2023** dan dinyatakan **LULUS**.

Disahkan oleh

		Tanda Tangan
Pembimbing I	: Ayu Rosyida Zain, S.ST., M.T	
Penguji I	: Maria Agustin, S.Kom., M.Kom	
Penguji II	: Indra Hermawan, S.kom., M.kom	
Penguji III	: Ariawan Andi Suhandana, S.Kom, M.T.I	

Mengetahui :

Jurusan Teknik Informatika dan Komputer

Ketua

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

NIP.197908032003122003

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian dan penyusunan laporan skripsi ini dengan judul " Rancang Bangun Network Traffic Monitoring Dan DDoS Detection Menggunakan Fastnetmon Dan Grafana Pada Iot Server " tepat pada waktunya. Laporan skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana di Program Studi "Teknik Multimedia dan Jaringan" di "Politeknik Negeri Jakarta". Dalam penyusunan laporan ini, penulis sadar bahwa selesainya laporan skripsi ini berkat dukungan dan bimbingan dari berbagai pihak, baik bersifat moril dan materi. Oleh karena itu penulis mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa yang selalu memberikan hikmat dan rahmatnya dalam menyelesaikan Tugas Akhir.
2. Orang tua dan keluarga serta sahabat penulis yang telah memberikan bantuan dukungan moral dan material.
3. Ibu Ayu Rosyida Zain, S.ST., M.T., selaku dosen pembimbing yang telah mengarahkan dalam menyelesaikan penelitian ini.
4. Rekan seperjuangan program studi Teknik Multimedia dan Jaringan yang telah membantu, mendukung, dan menemani hingga selesainya penelitian.
5. Seluruh jajaran Dosen dan Staf Teknik Informatika dan Komputer Politeknik Negeri Jakarta.
6. Teman berjuang yang selalu mau direpotkan, Nurul Inayah.

Akhir kata, penulis menyadari bahwa laporan skripsi ini tidak lepas dari kekurangan dan keterbatasan. Oleh karena itu, penulis sangat mengharapkan kritik, saran, dan masukan yang membangun guna perbaikan di masa mendatang. Semoga laporan skripsi ini dapat bermanfaat bagi semua pihak yang membutuhkannya.

Depok, 12 Juli 2023

Arif Setiawan

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Arif Setiawan

NIM : 1907421003

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

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

RANCANG BANGUN NETWORK TRAFFIC MONITORING DAN DDOS DETECTION MENGGUNAKAN FASTNETMON DAN GRAFANA PADA IOT SERVER

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, 12 Juli 2023

(Arif Setiawan)

NIM.1907421003

Rancang Bangun Network Traffic Monitoring dan DDoS Detection Menggunakan Fastnetmon Dan Grafana Pada Iot Server

Abstrak

Penggunaan IoT yang semakin meluas di berbagai sektor membuat kejahatan pada teknologi ini ikut meningkat. Salah satunya adalah serangan DDoS yang menyerang server IoT dengan tujuan membuat server tidak dapat diakses. Dalam hal ini, diperlukan upaya untuk mengamankan server IoT dari serangan DDoS. Selain itu, penting juga untuk memiliki sistem monitoring dan deteksi yang kuat agar serangan DDoS dapat dideteksi dan ditangani secepat mungkin. Oleh karena itu, maka dilakukanlah rancang bangun network traffic monitoring dan DDoS detection. Sistem ini memiliki tujuan untuk melakukan monitoring traffic yang masuk dan keluar dari server IoT dan memberikan informasi jika terjadi serangan DDoS. Sistem ini terdiri dari beberapa perangkat lunak yang diintegrasikan, seperti Netflow untuk merekam dan mengirim data lalu lintas, FastNetMon untuk menganalisis data dan mendeteksi serangan DDoS, InfluxDB untuk menyimpan data dari secara real-time, Grafana untuk memvisualisasikan data secara real-time, dan Bot telegram untuk pemberi notifikasi apabila terjadi serangan DDoS. Penelitian ini menghasilkan rata-rata waktu untuk mendeteksi dan mengirimkan notifikasi jika terjadi serangan DDoS berbasis flooding adalah 1 dan 3 detik, sedangkan untuk serangan DDoS berbasis Resource adalah 15,8 detik dan 19,5 detik. Penggunaan resource usage pada Server Monitoring juga cukup stabil untuk mendeteksi serangan DDoS. Dimana rata-rata penggunaan CPU adalah 7% dan rata-rata penggunaan Memory adalah sebesar 31,3%.

Kata Kunci: *Bot Telegram, DDoS, Fastnetmon, Grafana, IoT*

DAFTAR ISI

SURAT PERNYATAAN BEBAS PLAGIARISME iii

LEMBAR PENGESAHAN iv

KATA PENGANTAR v

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS vi

DAFTAR ISI viii

DAFTAR TABEL xi

DAFTAR GAMBAR xii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan dan Manfaat 3
1.5 Sistematika Penulisan 4

BAB II TINJAUAN PUSTAKA 5

2.1. Penelitian Sejenis 5
2.2. Monitoring jaringan 7
2.3. Internet of Things (IoT) 8
2.4. VMWare Workstation Pro 9
2.5. Debian Bullseye 9
2.6. DDoS 9
2.7. Network-Based IDS (NIDS) 11
2.8. FastNetMon 12
2.9. NetFlow/IP Traffic Flow 12
2.10. InfluxDB 13
2.11. Grafana OSS 13
2.12. Bot Telegram 13
2.13. OpenSSH 13

2.14. Shell Script (BASH) 14
2.15. Nginx 14
2.16. PostgreSQL 14
BAB III METODE PENELITIAN 16
3.1. Rancangan Penelitian 16
3.2. Tahapan Penelitian 16
3.3. Objek Penelitian 17
BAB IV HASIL DAN PEMBAHASAN 18
4.1. Analisis Kebutuhan 18
4.2. Perancangan Sistem 20
4.2.1. Diagram Blok Sistem 20
4.2.2. Flowchart Sistem 21
4.2.3. Topologi Jaringan 23
4.2.4. Pengalamatan IP Address 23
4.2.5. Spesifikasi Virtual Machine 24
4.2.6. Spesifikasi Hardware & Software 25
4.3. Implementasi Sistem 26
4.3.1. Instalasi Virtual Machine pada VMWare 26
4.3.2. Instalasi dan Konfigurasi FastNetMon 29
4.3.3. Konfigurasi IP Traffic Flow pada Mikrotik 31
4.3.4. Instalasi dan Konfigurasi InfluxDB 32
4.3.5. Instalasi dan Konfigurasi Grafana 33
4.3.6. Konfigurasi Bot Telegram 36
4.3.7. Pembuatan Script BASH 37
4.3.8. Konfigurasi Server IoT 39
4.4. Pengujian 42
4.4.1. Deskripsi Pengujian 42
4.4.2. Prosedur Pengujian 45
4.4.3. Data Hasil Pengujian 50
4.4.4. Analisis Data Pengujian 68
BAB V PENUTUP 75
5.1 Simpulan 75
5.2 Saran 75

DAFTAR PUSTAKA 77

LAMPIRAN 81

Lampiran 1 - Script Bash Notifikasi Telegram 81
Lampiran 2 - Script Python pengujian trafik SmartDoor 82
Lampiran 3 – Konfigurasi FastNetMon 84
Lampiran 4 – Konfigurasi InfluxDB 86
Lampiran 5 – Dokumentasi pengerjaan di JTIK 87

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu 5
Tabel 2.2 Perbandingan Tools NIDS 11
Tabel 4.1 Analisis kebutuhan 18
Tabel 4.2 Tabel IP Address 23
Tabel 4.3 Spesifikasi Virtual Machine 24
Tabel 4.4 Tabel Spesifikasi hardware dan software 25
Tabel 4.5 Contoh notifikasi serangan oleh Bot Telegram 47
Tabel 4.6 Data hasil deteksi serangan DDoS 50
Tabel 4.7 Data hasil pengujian visualisasi Grafana 52
Tabel 4.8 Data hasil pengujian notifikasi telegram 55
Tabel 4.9 Data Resource-Usage hasil pengujian serangan TCP Syn Flood 56
Tabel 4.10 Data Resource-Usage hasil pengujian serangan UDP Flood 56
Tabel 4.11 Data Resource-Usage hasil pengujian serangan ICMP Flood 57
Tabel 4.12 Data Resource-Usage hasil pengujian serangan DRipper 58
Tabel 4.13 Data Resource-Usage hasil pengujian serangan MHDDoS 59
Tabel 4.14 Data waktu deteksi serangan TCP Syn Flood 59
Tabel 4.15 Data waktu deteksi serangan UDP Flood 60
Tabel 4.16 Data waktu deteksi serangan ICMP Flood 61
Tabel 4.17 Data waktu deteksi serangan DDoS Ripper 61
Tabel 4.18 Data waktu deteksi serangan MHDDoS 62
Tabel 4.19 Data waktu notifikasi serangan TCP Syn Flood 63
Tabel 4.20 Data waktu notifikasi serangan UDP Flood 63
Tabel 4.21 Data waktu notifikasi serangan ICMP Flood 64
Tabel 4.22 Data waktu notifikasi serangan DDoS Ripper 64
Tabel 4.23 Data waktu notifikasi serangan MHDDoS 65
Tabel 4.24 Data Resource-usage pengujian 500 user 66
Tabel 4.25 Data Resource-usage pengujian 1000 user 66
Tabel 4.26 Data Resource-usage pengujian 2000 user 67

DAFTAR GAMBAR

Gambar 2.1 IoT SmartDoor 8
Gambar 3.1 Tahapan Penelitian 16
Gambar 4.1 Diagram Blok Sistem 20
Gambar 4.2 Flowchart deteksi serangan DDoS 21
Gambar 4.3 Flowchart monitoring lalu lintas jaringan 22
Gambar 4.4 Topologi jaringan 23
Gambar 4.5 Spesifikasi Server IoT 26
Gambar 4.6 Konfigurasi IP Address Server IoT 27
Gambar 4.7 Spesifikasi Server Monitoring 27
Gambar 4.8 Konfigurasi IP Address Server Monitoring 27
Gambar 4.9 Spesifikasi RouterOS 28
Gambar 4.10 Konfigurasi IP Address RouterOS 28
Gambar 4.11 Spesifikasi Host Attacker 28
Gambar 4.12 Konfigurasi IP Address Host Attacker 29
Gambar 4.13 Status fastnetmon 29
Gambar 4.14 Konfigurasi IP Traffic Flow pada RouterOS 31
Gambar 4.15 Instalasi InfluxDB 32
Gambar 4.16 Templates measurement 32
Gambar 4.17 Data Sources pada Grafana 33
Gambar 4.18 Import Dashboard FastNetMon 34
Gambar 4.19 Dashboard FastNetMon 34
Gambar 4.20 Panel traffic total bps dan pps 35
Gambar 4.21 Panel traffic bps per host 35
Gambar 4.22 Panel traffic pps per host 35
Gambar 4.23 Panel traffic bps per subnet 36
Gambar 4.24 Panel traffic pps per subnet 36
Gambar 4.25 Pembuatan Bot Telegram 36
Gambar 4.26 ID Chat Group 37
Gambar 4.27 Fungi Send Telegram Notification 38
Gambar 4.28 Fungsi perulangan 38
Gambar 4.29 Konfigurasi file pg_hba.conf 40

Gambar 4.30 Konfigurasi Nginx Reverse Proxy 40
Gambar 4.31 Migrasi database..... 41
Gambar 4.32 Start aplikasi SmartDoor 41
Gambar 4.33 Dashboard aplikasi SmartDoor 42
Gambar 4.34 Perintah mengirim paket TCP Syn 45
Gambar 4.35 Perintah mengirim paket UDP 46
Gambar 4.36 Perintah mengirim paket ICMP..... 46
Gambar 4.37 Command serangan TCP Syn Flood 47
Gambar 4.38 Command serangan UDP Flood..... 48
Gambar 4.39 Command serangan ICMP Flood..... 48
Gambar 4.40 Command serangan DDoS Ripper 49
Gambar 4.41 Command serangan MHDDoS 49
Gambar 4.42 Command pengujian traffic SmartDoor..... 50
Gambar 4.43 Perintah Fastnetmon_client..... 51
Gambar 4.44 Contoh deteksi lalu lintas tinggi..... 51
Gambar 4.45 Contoh file log..... 51
Gambar 4.46 Panel host traffic keluar (Bps)..... 52
Gambar 4.47 Panel host traffic masuk (Bps) 53
Gambar 4.48 Panel host traffic keluar (Pps) 53
Gambar 4.49 Panel host traffic masuk (Pps)..... 53
Gambar 4.50 Panel subnet traffic keluar (Bps)..... 54
Gambar 4.51 Panel subnet traffic masuk (Bps) 54
Gambar 4.52 Panel subnet traffic keluar (Pps) 54
Gambar 4.53 Panel subnet traffic masuk (Pps)..... 54
Gambar 4.54 Visualisasi traffic 500 user 67
Gambar 4.55 Visualisasi traffic 1000 user 68
Gambar 4.56 Visualisasi traffic 2000 user 68
Gambar 4.57 Data penggunaan CPU 69
Gambar 4.58 Data penggunaan Memory 70
Gambar 4.59 Data penggunaan traffic jaringan 70
Gambar 4.60 Data waktu deteksi serangan DDoS..... 71
Gambar 4.61 Data waktu notifikasi serangan DDoS 72

Gambar 4.62 Data Resource usage traffic SmartDoor 73
Gambar 4.63 Data traffic jaringan SmartDoor 74

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, penggunaan teknologi *Internet of Things* (IoT) semakin populer. IoT merupakan sebuah konsep yang memiliki tujuan untuk menyatukan berbagai macam objek di dunia nyata agar dapat saling berkomunikasi dan menjadi bagian dari sistem yang terintegrasi dengan menggunakan internet sebagai penghubung (Pridiatama and Agustin, 2021). Penggunaan IoT semakin meluas di berbagai sektor, seperti pendidikan, industri, dan bisnis. Namun, semakin meluasnya penggunaan IoT juga membuat kejahatan pada teknologi ini ikut meningkat. Salah satunya adalah *Distributed Denial of Service* (DDoS) yang menyerang server IoT. Laporan Kaspersky Lab yang dirilis pada tahun 2020 berjudul "*DDoS attacks in Q3 2020*" membahas tentang tren serangan DDoS selama kuartal ketiga tahun 2020. Laporan tersebut menyebutkan bahwa terjadi peningkatan jumlah serangan DDoS sebesar 10,7% dibandingkan dengan periode yang sama pada tahun 2019 pada server IoT selama periode tersebut (Kuprev, Gutnikov and Badovskaya, 2020). DDoS merupakan serangan yang dilakukan dengan cara membanjiri lalu lintas jaringan dengan traffic palsu dengan jumlah yang besar, sehingga server menjadi down dan tidak bisa diakses. Serangan ini menyebabkan kegagalan server karena server menerima banyak permintaan dalam waktu yang singkat (Islam et al., 2022). Kasus seperti ini pernah terjadi pada sistem *Control Traffic Light* di Melbourne, Australia yang menyebabkan kemacetan lalu lintas di beberapa wilayah (The Age, 2019). Kondisi ini dapat mengancam keselamatan dan keamanan penggunaan IoT, karena ketika server IoT tidak dapat diakses, maka data dan informasi yang dihasilkan oleh perangkat IoT tidak dapat dipantau dan dikelola.

Dalam hal ini, diperlukan upaya untuk mengamankan server IoT dari serangan DDoS. Selain itu, penting juga untuk memiliki sistem pemantauan dan deteksi yang kuat, sehingga serangan DDoS dapat dideteksi dan ditangani secepat mungkin. Salah satu teknologi yang dapat digunakan adalah *Intrusion Detection System* (IDS) yang berfungsi untuk mendeteksi aktivitas dan memantau lalu lintas data pada

jaringan. FastNetMon merupakan salah satu tools IDS yang digunakan untuk monitoring dan pemantauan jaringan yang memfokuskan pada deteksi dan mitigasi serangan DDoS. FastNetMon memiliki beberapa kelebihan jika dibandingkan dengan tools IDS lainnya, seperti dapat melakukan pemantauan lalu lintas jaringan secara real-time, memberikan informasi mengenai serangan DDoS, melakukan tindakan secara otomatis untuk memblokir serangan DDoS(Fastnetmon, 2023). FastNetMon juga bisa diintegrasikan dengan protokol pemantauan jaringan seperti Netflow, aplikasi penampil data seperti Grafana, dan database time-series seperti InfluxDB. Penelitian mengenai Network Monitoring dan DDoS Detection sudah cukup banyak dilakukan dengan menggunakan teknologi/tools yang berbeda. Penelitian yang dilakukan oleh (Zaen, Tantoni and Ashari, 2021) membuat sistem mitigasi serangan DDoS dengan IDS menggunakan Bot Telegram. Pada sistem ini hanya menggunakan fitur Traffic Monitor pada Router Mikrotik untuk mendeteksi DDoS dan tidak memiliki Dashboard/user interface untuk memudahkan dalam monitoring traffic jaringan.

Berdasarkan latar belakang tersebut, maka dilakukanlah rancang bangun network traffic monitoring dan DDoS detection. Sistem ini memiliki tujuan untuk melakukan monitoring traffic yang masuk dan keluar dari server IoT dan memberikan informasi atau notifikasi jika terjadi serangan DDoS. Sistem ini terdiri dari beberapa perangkat lunak yang diintegrasikan, seperti penggunaan protokol NetFlow pada Router Mikrotik untuk mengirim data matriks ke Server Monitoring, penggunaan FastNetmon untuk menerima dan memproses data matriks dan mendeteksi serangan DDoS, penggunaan InfluxDB untuk menyimpan data secara real-time, dan penggunaan Grafana untuk memvisualisasikan data secara real-time, serta penggunaan Bot Telegram untuk pemberitahuan/notifikasi jika terjadi serangan DDoS.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, perumusan masalah yang terdapat pada penelitian ini adalah sebagai berikut:

1. Bagaimana cara membangun Network Traffic Monitoring dan DDoS Detection dengan menggunakan FastNetMon dan Grafana?

2. Seberapa efektif sistem dalam mendeteksi serangan DDoS berbasis flooding pada jaringan (TCP Syn, UDP, dan ICMP Flood) dan serangan DDoS berbasis resource seperti CPU dan RAM (DRipper dan MHDDoS)?
3. Bagaimana penggunaan resource pada Server Monitoring yang menggunakan fastnetmon untuk melakukan deteksi serangan DDoS?

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan, batasan masalah yang terdapat pada penelitian ini adalah sebagai berikut:

1. Server IoT yang akan digunakan adalah Server SmartDoor-Lock dengan menggunakan server dummy.
2. Menggunakan 1 VM untuk Server Monitoring, 1 VM untuk Server IoT, dan 1 VM untuk penyerang.
3. Menggunakan Mikrotik RouterOS untuk dijadikan Router/Gateway.
4. Menggunakan Bot Telegram untuk notifikasi.
5. Parameter pengujian adalah waktu deteksi serangan, waktu respon notifikasi, penggunaan CPU dan Memory pada Server Monitoring.
6. Pengujian serangan DDoS dilakukan dengan menggunakan tools hping3, DRipper, dan MHDDoS.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Membuat rancang bangun Network Traffic Monitoring dan DDoS Detection menggunakan FastNetMon dan Grafana pada IoT Server.
2. Menambahkan fitur notifikasi pada Network Traffic Monitoring dan DDoS Detection menggunakan Bot Telegram.

1.4.2 Manfaat

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Mempermudah System Administrator untuk melakukan pengawasan terhadap lalu lintas data pada Server IoT melalui Dashboard Grafana.

2. Memudahkan System Administrator menentukan metode selanjutnya untuk menangani serangan DDoS.

1.5 Sistematika Penulisan

Berikut merupakan sistematika atau kerangka dalam penulisan penelitian ini:

a. BAB I PENDAHULUAN

   Bab I menguraikan tentang latar belakang dari penelitian, rumusan masalah yang didapat dari latar belakang, batasan masalah, dan manfaat serta tujuan dalam penelitian ini.

b. BAB II TINJAUAN PUSTAKA

   Bab II menguraikan tentang landasan teori dan konsep yang berkaitan dengan permasalahan pada penelitian yang diambil, selain itu juga terdapat beberapa penelitian relevan terkait dari penelitian-penelitian terdahulu untuk dikaji dalam penelitian ini.

c. BAB III METODE PENELITIAN

   Bab III pada penelitian ini berisikan tentang metode penelitian yang akan digunakan, tahapan yang akan dilakukan dalam penelitian, dan objek dari penelitian dalam penelitian ini.

d. BAB IV HASIL & PEMBAHASAN

   Bab IV pada penelitian ini berisikan pembahasan penelitian, mulai dari proses persiapan kebutuhan hingga proses uji dengan melakukan percobaan-percobaan terhadap sistem. Pengujian ini dilakukan untuk mengetahui apakah sistem berjalan sesuai yang diharapkan atau tidak.

e. BAB V PENUTUP

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

BAB V
PENUTUP

5.1 Simpulan

Berdasarkan dari hasil rancang bangun Network Traffic Monitoring dan DDoS Detection dengan menggunakan FastNetMon dan Grafana dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Sistem Network Traffic Monitoring dan DDoS Detection dibangun dengan menggunakan beberapa perangkat lunak yang diintegrasikan, seperti netflow untuk merekam data lalu lintas jaringan, fastnetmon untuk mendeteksi serangan DDoS, influxDB dan grafana untuk menyimpan dan memvisualisasikan data lalu lintas jaringan, serta script bash untuk mengirimkan notifikasi melalui bot telegram. Dengan merancang sistem ini, hasilnya menunjukkan bahwa sistem berfungsi sesuai rancangan. FastNetMon dapat digunakan untuk mendeteksi serangan DDoS berdasarkan threshold yang telah diatur, Grafana memvisualisasikan lalu lintas jaringan melalui dashboard, dan notifikasi melalui bot Telegram dapat dikirimkan jika terdeteksi adanya serangan DDoS.
2. Sistem lebih efektif untuk mendeteksi serangan DDoS berbasis flooding pada jaringan (TCP Syn Flood, UDP Flood, dan ICMP Flood) dibandingkan dengan serangan DDoS berbasis resource seperti CPU dan RAM (DRipper dan MHDDoS). Waktu deteksi dan notifikasi untuk serangan DDoS berbasis flooding pada jaringan adalah sekitar 1-5 detik, sedangkan untuk serangan DDoS berbasis resource adalah sekitar 15-20 detik.
3. Penggunaan resource pada Server Monitoring yang menggunakan fastnetmon untuk melakukan deteksi serangan DDoS dianggap masih cukup aman. Dimana rata-rata penggunaan CPU adalah 7% dan rata-rata penggunaan Memory adalah sebesar 31,3%.

5.2 Saran

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

1. Menambahkan fungsi mitigasi DDoS pada sistem untuk melakukan tindakan secara otomatis jika terjadi serangan DDoS.

2. Mengkaji penggunaan router selain MikroTik RouterOS dalam implementasi sistem.

DAFTAR PUSTAKA

cybersecurity.att.com, 2020. *Open Source IDS Tools: Comparing Suricata, Snort, Bro (Zeek), Linux*. [online] cybersecurity.att.com. Available at: <https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview> [Accessed 4 August 2023].

debian, 2021a. *Introduction to Debian*. [online] Available at: <https://www.debian.org/intro/about> [Accessed 14 February 2023].

debian, 2021b. *Release Bullseye*. [online] Available at: <https://www.debian.org/releases/bullseye/> [Accessed 14 February 2023].

Efendi, Y., 2018. Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer*, 4(2), pp.21–27. https://doi.org/10.35329/jiik.v4i2.41.

Fastnetmon, 2023. *Fast, Reliable, Automated DDoS Detection, Prevention, and Mitigation*. [online] FastNetMon. Available at: <https://fastnetmon.com/ddos-detection-and-mitigation/> [Accessed 9 February 2023].

Fernando, N., Humaira and Asri, E., 2020. Monitoring Jaringan dan Notifikasi dengan Telegram pada Dinas Komunikasi dan Informatika Kota Padang. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 1(4), pp.121–126. https://doi.org/10.30630/jitsi.1.4.17.

gnu.org, 2022. *Bash Reference Manual*. [online] Available at: <https://www.gnu.org/software/bash/manual/bash.html> [Accessed 8 May 2023].

grafana, 2023. *Introduction to Grafana*. [online] Available at: <https://grafana.com/docs/grafana/latest/introduction/> [Accessed 8 February 2023].

Husna, M.A. and Rosyani, P., 2021. Implementation of Network and Server Monitoring System Using Zabbix Integrated with Grafana and Telegram. *Jurnal Riset Komputer)*, [online] 8(6), pp.2407–389. https://doi.org/10.30865/jurikom.v8i6.3631.

idcloudhost, 2020. *Mengenal Apa Itu Serangan dan Definisi Denial-of-service DDos Attack*. [online] Available at: <https://idcloudhost.com/mengenal-apa-itu-serangan-dan-definisi-denial-of-service-ddos-attack> [Accessed 8 February 2023].

Indrarto, S.A. and Basuki, A., 2022. Penerapan Platform Visualisasi dan Analisis Trafik Jaringan menggunakan Elastic Stack. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, [online] 6(9), pp.2548–964. Available at: <http://j-ptiik.ub.ac.id>.

Islam, H.Z., Faiqurahman, M., Dwi, F. and Sumadi, S., 2022. Analisis Dampak Serangan DoS Pada Software Defined Network. 4(1), pp.71–82.

Juniper.net, 2020. *How many Packets per Second per port are needed to achieve Wire-Speed?* [online] supportportal.Juniper.net. Available at: <https://supportportal.juniper.net/s/article/How-many-Packets-per-Second-per-port-are-needed-to-achieve-Wire-Speed?> [Accessed 1 August 2023].

kali.org, 2022. *hping3*. [online] Available at: <https://www.kali.org/tools/hping3/> [Accessed 14 February 2023].

Kuprev, O., Gutnikov, A. and Badovskaya, E., 2020. *DDoS attacks in Q3 2020*. [online] SECURELIST by Kaspersky. Available at: <https://securelist.com/ddos-attacks-in-q3-2020/99171/> [Accessed 1 February 2023].

MatrixTM, 2023. *MHDDoS*. [online] Github. Available at: <https://github.com/MatrixTM/MHDDoS> [Accessed 8 July 2023].

Mulyanto, A.D., 2020. Pemanfaatan Bot Telegram Untuk Media Informasi Penelitian. *Matics*, 12(1), p.49. https://doi.org/10.18860/mat.v12i1.8847.

Muqorobin, M., Hisyam, Z., Mashuri, M., Hanafi, H. and Setiyantara, Y., 2019. Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open Cloud Computing. *Majalah Ilmiah Bahari Jogja*, 17(2), pp.1–9. https://doi.org/10.33489/mibj.v17i2.205.

Netmonk, 2019. *Apa Itu NetFlow?* [online] Available at: <https://netmonk.id/apa-itu-netflow> [Accessed 14 February 2023].

Nginx, 2023. *Welcome to NGINX Wiki!* [online] Available at: <https://www.nginx.com/resources/wiki/> [Accessed 30 May 2023].

openssh, 2023. *About OpenSSH*. [online] Available at: <https://www.openssh.com/manual.html> [Accessed 14 February 2023].

palahsu, 2021. *DDoS-Ripper*. [online] Github. Available at: <https://github.com/palahsu/DDoS-Ripper> [Accessed 8 July 2023].

PostgreSQL, 2023. *What is PostgreSQL?* [online] Available at: <https://www.postgresql.org/about/> [Accessed 30 May 2023].

Pridiatama, F. and Agustin, M., 2021. Rancang Bangun Smart Bathroom Berbasis Raspberry Pi. *Jurnal SIMADA (Sistem Informasi dan Manajemen Basis Data)*, 4(2), pp.128–138. https://doi.org/10.30873/simada.v4i2.3008.

Rusli, M., Usman, C.M., Mulya, M.F. and Widyaningsih, T.W., 2022. Aplikasi Sistem Monitoring Server Menggunakan Device Orange Pi Berbasis Web Service Studi Kasus PT. MNC Televisi Indonesia – MNC Group. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, 5(2), pp.24–35. https://doi.org/10.47970/siskom-kb.v5i2.276.

SonicWall.com, 2019. *How can I calculate supported maximum PPS (Packets Per Second) of UTM appliance?* [online] Sonicwall.com. Available at: <https://www.sonicwall.com/support/knowledge-base/how-can-i-calculate-supported-maximum-pps-packets-per-second-of-utm-appliance/170504889979238/> [Accessed 1 August 2023].

Techopedia, 2019. *Network Monitoring*. [online] Available at: <https://www.techopedia.com/definition/24149/network-monitoring> [Accessed 10 February 2023].

The Age, 2019. *Melbourne traffic lights hit by massive cyberattack*. [online] The Age. Available at: <https://www.theage.com.au/national/victoria/melbourne-traffic-lights-hit-by-massive-cyber-attack-20190128-p50u95.html> [Accessed 2 February 2023].

Umar, R. and Prasetyo Marsaid, A., 2023. Analisis Keamanan Jaringan LAN Terhadap Kerentanan Jaringan Ancaman DDoS Menggunakan Metode Penetration Testing. *Jurnal Riset Komputer)*, [online] 10(1), pp.2407–389. https://doi.org/10.30865/jurikom.v10i1.5835.

VMware, I., 2023. *VMware Workstation Pro*. [online] Available at: <https://www.vmware.com/products/workstation-pro.html> [Accessed 7 February 2023].

Zaen, M.T.A., Tantoni, A. and Ashari, M., 2021. DDoS ATTACK MITIGATION WITH INTRUSION DETECTION SYSTEM (IDS) USING TELEGRAM BOTS. *JISA(Jurnal Informatika dan Sains)*, 4(2), pp.149–154. https://doi.org/10.31326/jisa.v4i2.1043.

DAFTAR RIWAYAT HIDUP PENULIS

Arif Setiawan,

Lahir di Purbalingga pada 11 April 2001. Penulis merupakan anak pertama dari dua bersaudara. Saat ini berdomisili di Kramat Jati, Jakarta Timur, DKI Jakarta.

Lulus dari SD Negeri 4 Serang pada tahun 2013, selanjutnya penulis melanjutkan sekolah menengah pertama di SMP Negeri 2 Karangreja dan lulus pada tahun 2016, kemudian menyelesaikan pendidikan menengah akhir di SMK Negeri 22 Jakarta pada tahun 2019. Pada tahun yang sama, penulis berkesempatan untuk melanjutkan pendidikan di Politeknik Negeri Jakarta Program Studi Teknik Multimedia dan Jaringan Jurusan Teknik Informatika dan Komputer.

LAMPIRAN

Lampiran 1 - Script Bash Notifikasi Telegram

```
send_telegram_notification() {
    TELEGRAM_BOT_TOKEN="YOUR_TELEGRAM_BOT_TOKEN"
    TELEGRAM_CHAT_ID="YOUR_TELEGRAM_CHAT_ID"
    MSG_TEXT="$1"

    curl -s -X POST "https://api.telegram.org/bot$TELEGRAM_BOT_TOKEN/sendMessage" \
        -d chat_id="$TELEGRAM_CHAT_ID" \
        -d text="$MSG_TEXT" \
        -d parse_mode="HTML"
}
```

```
while true; do
    current_files=($(ls "$folder"))

    for file in "${current_files[@]}"; do
        if [[ ! " ${files[@]} " =~ " ${file} " ]]; then
            timestamp=$(stat -c %y "$folder/$file" | awk -F "." '{print $1}')
            file_path="$folder/$file"
            file_content=$(cat "$file_path")

            ip=$(echo "$file_content" | grep -oP 'IP: \K[\d.]+')
            attack_type=$(echo "$file_content" | grep -oP 'Attack type: \K\w+')
            attack_protocol=$(echo "$file_content" | grep -oP 'Attack protocol: \K\w+')
            attack_direction=$(echo "$file_content" | grep -oP 'Attack direction: \K\w+')
            initial_attack_power=$(echo "$file_content" | grep -oP 'Initial attack power: \K[\d ]+')

            # Remove whitespace characters from initial_attack_power
            initial_attack_power=$(echo "$initial_attack_power" | tr -d ' ')

            # Check if the initial attack power exceeds 500
            if [ "$initial_attack_power" -gt 500 ]; then
                # Get the alert level and message emoji
                alert_level=$(levelAlert "$initial_attack_power")
                message_emoji=$(levelMessage "$alert_level")

                # Format the timestamp
                formatted_timestamp=$(date -d "$timestamp" +"%Y-%m-%d %H:%M:%S")

                # Format the message for Telegram
                message="<b>High $attack_direction packet detected</b>%0A"
                message+="IP: $ip%0A"
                message+="Type: $attack_type%0A"
                message+="Protocol: $attack_protocol%0A"
                message+="Direction: $attack_direction%0A"
                message+="Power: $initial_attack_power pps%0A"
                message+="Timestamp: $formatted_timestamp%0A"
                message+="Alert Level: $message_emoji%0A"
                message+="Attack details: $folder$file"

                # Send the notification to Telegram
                send_telegram_notification "$message"
            fi

            files+=("$file")
        fi
    done
    sleep 1
done
```

Lampiran 2 - Script Python pengujian trafik SmartDoor

```python
from urllib.request import Request
import requests
import inquirer
import xlsxwriter
import json
from concurrent.futures import ThreadPoolExecutor, thread
from time import perf_counter

# INFO: Prompt Question
questions = [
    inquirer.List(
        "test",
        message="What API testing program do you want to do?",
        choices=["Register User", "Logging In User", "Register Card", "Pairing User and Card", "User Request Room", "Admin Give Room Access", "Check In"],
    ),
]
test = inquirer.prompt(questions)
filePath = input("Testing file name: ")
outputFile = input("Output testing file name: ")
threadSize = int(input("How many thread you want to use: "))
url_ = "http://127.0.0.1:8000" #Local TIK #"http://127.0.0.1:80" #Local #for online url please user "https://smart-door-pnj.herokuapp.com"
secret = {
    "username"  : "dimasaulia",
    "password"  : "T4np4$4nd1",
    "api_id"    : "cl9xxy39x0000cklku2klje55",
    "api_key"   : "WCPaDR1F2cwSUMPWuXDm13JMYU-NeWO1"
}
buffData = []
RUID = "yuiOB"
DUID = "Hge40"
# INFO: Open File
file = open(f'{filePath}')
jsonData = json.load(file)
file.close()

# INFO: Starting XLSX
workbook = xlsxwriter.Workbook(f'{outputFile}.xlsx')


def fetchData(_payload, _endpoint, _no, _worksheet):
    with requests.Session() as client:
        resp = client.post(f'{url_}{_endpoint}', json=_payload)
        _worksheet.write(f"A{_no}", f"{_no-1}")
        _worksheet.write(f"B{_no}", str(_payload))
        _worksheet.write(f"C{_no}", str(resp.json()))
        _worksheet.write(f"D{_no}", round(float(resp.elapsed.total_seconds()),3)*1000)
        if(_endpoint == "/api/v1/user/register" or _endpoint == "/api/v1/user/login"):
            _payload["jwt"] = resp.cookies["jwt"]
            buffData.append(_payload)
```

```python
def fetchDataAdmin(_payload, _endpoint, _no, _worksheet):
    with requests.Session() as client:
        resp = client.post(f'{url_}{_endpoint}', json=_payload, cookies=adminAuth)
        _worksheet.write(f"A{_no}", f"{_no-1}")
        _worksheet.write(f"B{_no}", str(_payload))
        _worksheet.write(f"C{_no}", str(resp.json()))
        _worksheet.write(f"D{_no}", round(float(resp.elapsed.total_seconds()),3)*1000)

def fetchDataAdminAcc(_payload, _endpoint, _no, _worksheet):
    with requests.Session() as client:
        resp = client.post(f'{url_}{_endpoint}?ruid={RUID}&cardNumber={_payload["cardNumber"]}', json=_payload, cookies=adminAuth)
        _worksheet.write(f"A{_no}", f"{_no-1}")
        _worksheet.write(f"B{_no}", str(_payload))
        _worksheet.write(f"C{_no}", str(resp.json()))
        _worksheet.write(f"D{_no}", round(float(resp.elapsed.total_seconds()),3)*1000)

def fetchDataWCookie(_payload, _endpoint, _no, _worksheet):
    with requests.Session() as client:
        resp = client.post(f'{url_}{_endpoint}?ruid={RUID}&cardNumber={_payload["cardNumber"]}', cookies={"jwt":_payload["jwt"]})
        _worksheet.write(f"A{_no}", f"{_no-1}")
        _worksheet.write(f"B{_no}", str(_payload))
        _worksheet.write(f"C{_no}", str(resp.json()))
        _worksheet.write(f"D{_no}", round(float(resp.elapsed.total_seconds()),3)*1000)

if test.get("test") == "Register User":
    registerWorksheet = workbook.add_worksheet("REGISTER")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchData, jsonData, ["/api/v1/user/register"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)

    file = open(f'{filePath}', "w+")
    file.write(json.dumps(buffData))
    file.close()
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

if test.get("test") == "Logging In User":
    registerWorksheet = workbook.add_worksheet("REGISTER")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchData, jsonData, ["/api/v1/user/login"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)
```

Lanjutan

```
    file = open(f'{filePath}', "w+")
    file.write(json.dumps(buffData))
    file.close()
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

if test.get("test") == "Register Card":
    registerWorksheet = workbook.add_worksheet("CARD REGISTER")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchData, jsonData, [f"/api/v1/card/h/register?id={secret['api_id']}&key={secret['api_key']}"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] *
            len(jsonData))
        executor.shutdown(wait=True)
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

if test.get("test") == "Pairing User and Card":
    print("Logging in Admin first")
    login = requests.post(url =f'{url_}/api/v1/user/login', json=secret)
    adminAuth = login.cookies
    registerWorksheet = workbook.add_worksheet("PAIRING USER AND CARD")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    print("Start executing task")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchDataAdmin, jsonData, [f"/api/v1/user/pair"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

if test.get("test") == "User Request Room":
    registerWorksheet = workbook.add_worksheet("REQUEST ROOM ACCESS")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    print("Start executing task")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchDataWCookie, jsonData, [f"/api/v1/room/u/request"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )
```

```
if test.get("test") == "Admin Give Room Access":
    print("Logging in Admin first")
    login = requests.post(url =f'{url_}/api/v1/user/login', json=secret)
    adminAuth = login.cookies
    registerWorksheet = workbook.add_worksheet("GIVE ROOM ACCESS")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    print("Start executing task")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchDataAdminAcc, jsonData, [f"/api/v2/room/pair"] * len(jsonData), range(2, len(jsonData)+2), [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

if test.get("test") == "Check In":
    registerWorksheet = workbook.add_worksheet("CHECK IN")
    registerWorksheet.write("A1", "NO")
    registerWorksheet.write("B1", "PAYLOAD DATA")
    registerWorksheet.write("C1", "RESPONSE DATA")
    registerWorksheet.write("D1", "RESPON TIME")
    start = perf_counter()
    with ThreadPoolExecutor(threadSize) as executor:
        executor.map(fetchData, jsonData, [f"/api/v2/room/h/check-in/{DUID}?id={secret['api_id']}&key={secret['api_key']}"] * len(jsonData), range(2, len(jsonData)+2),
            [registerWorksheet] * len(jsonData))
        executor.shutdown(wait=True)
    print("Execution time:", f"{round((perf_counter() - start),3) * 1000}ms" )

workbook.close()
```

Lampiran 3 – Konfigurasi FastNetMon

File /etc/fastnetmon.conf

```
### Main configuration params
### Logging configuration
logging_level = info
logging_local_syslog_logging = off
logging_remote_syslog_logging = off
logging_remote_syslog_server = 10.10.10.10
logging_remote_syslog_port = 514
disable_usage_report = off
enable_ban = on
enable_ban_ipv6 = on
process_incoming_traffic = on
process_outgoing_traffic = on
dump_all_traffic = off

dump_other_traffic = off
ban_details_records_count = 20
ban_time = 20
unban_only_if_attack_finished = off

networks_list_path = /etc/networks_list
white_list_path = /etc/networks_whitelist
check_period = 1

enable_connection_tracking = on

ban_for_pps = on
ban_for_bandwidth = on
ban_for_flows = off
threshold_pps = 1000
threshold_mbps = 1000
threshold_flows = 1000
threshold_tcp_mbps = 100000
threshold_udp_mbps = 100000
threshold_icmp_mbps = 100000
threshold_tcp_pps = 100000
threshold_udp_pps = 100000
threshold_icmp_pps = 100000
ban_for_tcp_bandwidth = off
ban_for_udp_bandwidth = off
ban_for_icmp_bandwidth = off
ban_for_tcp_pps = off
ban_for_udp_pps = off
ban_for_icmp_pps = off

### Traffic capture methods
mirror_afpacket = off
mirror_afxdp = off
poll_mode_xdp = off
xdp_set_promisc = on
zero_copy_xdp = off
force_native_mode_xdp = off
xdp_read_packet_length_from_ip_header = off
microcode_xdp_path = /etc/xdp_kernel.o
mirror_af_packet_custom_sampling_rate = 1
mirror_af_packet_fanout_mode = cpu
af_packet_read_packet_length_from_ip_header = off
mirror_netmap = off
netmap_sampling_ratio = 1
netmap_read_packet_length_from_ip_header = off
pcap = off
netflow = on
sflow = off
interfaces = ens33
average_calculation_time = 5
speed_calculation_delay = 1

netflow_port = 2055
netflow_host = 0.0.0.0
netflow_sampling_ratio = 1

sflow_port = 6343
sflow_host = 0.0.0.0
sflow_read_packet_length_from_ip_header = off
```

Lanjutan

```
### Actions when attack detected
notify_script_path = /usr/local/bin/notify_about_attack.sh
collect_attack_pcap_dumps = off

# InfluxDB integration
# More details can be found here: https://fastnetmon.com/docs/influxdb_integration/
influxdb = on
influxdb_host = 127.0.0.1
influxdb_port = 8086
influxdb_database = fastnetmon
influxdb_auth = off
influxdb_user = fastnetmon
influxdb_password = secure
influxdb_push_period = 1
graphite = on
graphite_host = 127.0.0.1
graphite_port = 2003
graphite_prefix = fastnetmon
graphite_push_period = 1
monitor_local_ip_addresses = on

pid_path = /var/run/fastnetmon.pid
cli_stats_file_path = /tmp/fastnetmon.dat
cli_stats_ipv6_file_path = /tmp/fastnetmon_ipv6.dat
enable_api = on

### Client configuration
sort_parameter = packets
max_ips_in_list = 7
```

Lampiran 4 – Konfigurasi InfluxDB

File /etc/influxdb/influxdb.conf

```
### Welcome to the InfluxDB configuration file.
# Changed by FastNetMon
reporting-disabled = true
bind-address = "127.0.0.1:8088"
[meta]
  dir = "/var/lib/influxdb/meta"

[data]
  dir = "/var/lib/influxdb/data"
  wal-dir = "/var/lib/influxdb/wal"
  max-values-per-tag = 0
  series-id-set-cache-size = 100

[coordinator]
[retention]
[shard-precreation]
[monitor]
[http]
  enabled = true
  bind-address = "127.0.0.1:8086"
  log-enabled = false

[logging]
[subscriber]
[[graphite]]
   enabled = true
   retention-policy = ""
   bind-address = "127.0.0.1:2003"
   protocol = "tcp"
   consistency-level = "one"
   batch-size = 5000
   batch-pending = 10
   batch-timeout = "1s"
   separator = "."
   templates = [
     "fastnetmon.hosts.* app.measurement.cidr.direction.function.resource",
     "fastnetmon.networks.* app.measurement.cidr.direction.resource",
     "fastnetmon.total.* app.measurement.direction.resource"
   ]

[[collectd]]
[[opentsdb]]
[[udp]]
[continuous_queries]
[tls]
```

Lampiran 5 – Dokumentasi pengerjaan di JTIK