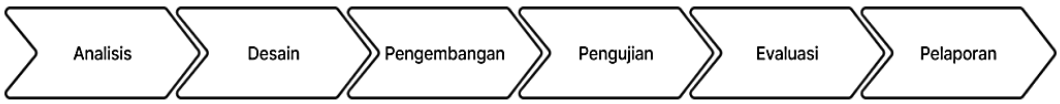
BAB III
METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah Design and Development (D&D). Metode ini merupakan suatu pendekatan penelitian yang secara sistematis mengkaji proses perancangan, pengembangan, serta evaluasi. Menurut Ellis dan Levy (2010), D&D adalah studi sistematis mengenai tahapan desain, pengembangan, dan evaluasi yang bertujuan membangun dasar empiris dalam pembuatan suatu produk atau perangkat, baik yang bersifat instruksional maupun non-instruksional, serta dapat menghasilkan model baru ataupun penyempurnaan model yang telah ada. Pemilihan metode D&D pada penelitian ini didasarkan pada kelebihannya, yaitu mampu memberikan kerangka kerja yang terstruktur untuk merancang, mengembangkan, sekaligus mengevaluasi produk atau solusi yang dihasilkan. Adapun alur penelitian dengan metode D&D ditunjukkan



Gambar 3.1. Alur Metode Penelitian D&D

Berikut adalah penjelasan tahapan metode penelitian yang dilakukan.

1. Tahapan analisis, yaitu proses identifikasi kebutuhan serta permasalahan dalam pengelolaan keamanan server, khususnya terkait deteksi dan pencegahan berbagai serangan siber. Analisis ini mencakup potensi ancaman berupa ICMP *Flood*, SYN *Flood*, UDP *Flood*, *Ping of Death*, *Brute Force*, SQL *Injection*, XSS *Attack, Port Scanning*, serta ARP *Spoofing*.
2. Tahapan Desain, yaitu merancang solusi keamanan jaringan dengan memanfaatkan integrasi antara Zeek untuk deteksi serangan, IPtables untuk pencegahan dengan Zabbix sebagai platform monitoring. Pada tahap ini juga dilakukan perancangan arsitektur jaringan, skenario pengujian

serangan, serta mekanisme notifikasi melalui Telegram dan WhatsApp agar setiap insiden dapat dipantau secara *realtime*.

3. Tahapan pengembangan, sistem yang telah dirancang diimplementasikan dalam lingkungan nyata. Proses ini mencakup instalasi dan konfigurasi Zeek dengan skrip deteksi serangan, pembuatan skrip shell untuk mengeksekusi aturan IPtables dalam memblokir IP penyerang, serta integrasi log hasil deteksi ke dalam Zabbix yang ditampilkan melalui *dashboard* untuk memudahkan pemantauan.
4. Tahapan pengujian dilakukan dengan melakukan simulasi sembilan jenis serangan, meliputi ICMP *Flood*, SYN *Flood*, UDP *Flood*, *Ping of Death*, *Brute Force*, SQL *Injection*, XSS *Attack*, *Port Scanning*, serta ARP *Spoofing*.
5. Tahapan evaluasi, yaitu proses dimana hasil pengujian kemudian dianalisis guna menilai efektivitas sistem dalam mendeteksi, mencegah, memberikan notifikasi, serta merekam log sebagai bahan analisis lebih lanjut.
6. Tahapan Pelaporan, yaitu seluruh proses penelitian dan hasil implementasi didokumentasikan dalam bentuk laporan skripsi. Dokumentasi ini berisi uraian hasil implementasi, hasil pengujian, serta analisis mendalam yang diharapkan dapat memberikan kontribusi ilmiah dalam pengembangan sistem keamanan server berbasis monitoring dan pencegahan otomatis.

3.2 Analisis Kebutuhan Sistem

Pada tahapan ini, analisis kebutuhan sistem terbagi menjadi 3 bagian yaitu identifikasi masalah, studi literatur, dan spesifikasi alat. Berikut merupakan penjelasan tiap sub-bab nya:

3.2.1 Identifikasi Masalah

Pada tahap ini, peneliti melakukan identifikasi terhadap permasalahan yang berkaitan dengan meningkatnya jumlah serta kompleksitas serangan siber. Ancaman seperti ICMP Flood, SYN Flood, UDP Flood, Ping of Death, Brute Force, SQL *Injection*, XSS Attack, Port Scanning, serta ARP Spoofing menunjukkan adanya kebutuhan yang mendesak untuk menerapkan sistem keamanan yang lebih andal. Rendahnya tingkat respons serta potensi dampak

yang ditimbulkan dari serangan-serangan tersebut menjadi perhatian penting dalam upaya meningkatkan kecepatan dan efektivitas respons keamanan. Identifikasi masalah ini menjadi landasan bagi penelitian untuk mengembangkan solusi yang lebih efektif melalui implementasi Zeek sebagai sistem deteksi, IPtables sebagai mekanisme pencegahan otomatis, serta integrasi dengan Zabbix untuk monitoring dan notifikasi melalui Telegram dan WhatsApp, sehingga keamanan server dapat terjaga dari berbagai ancaman siber.

3.2.2 Studi Literatur

Pada tahapan ini, penulis mengumpulkan studi literatur yang berkaitan dengan implementasi Zeek dan IPtables, notifikasi Telegram dan WhatsApp, serta monitoring Zabbix menggunakan referensi dari jurnal ilmiah, buku, dan artikel yang membahas implementasi sistem monitoring server, deteksi serta mitigasi serangan siber, dan integrasi notifikasi keamanan melalui WhatsApp serta Telegram. Studi literatur ini memberikan dasar teoritis serta konteks ilmiah yang mendukung integrasi sistem.

3.2.3 Spesifikasi Perangkat yang Digunakan

Dalam mendukung proses eksperimental dan analisis data dalam penelitian ini, berbagai perangkat keras atau alat yang digunakan. Instrumen-instrumen ini diperlukan untuk membangun lingkungan eksperimental, melaksanakan pengujian, memantau aktivitas jaringan, mengidentifikasi gangguan, serta mengevaluasi data yang dikumpulkan. Setiap alat yang digunakan memiliki peran penting dalam mencapai tujuan penelitian secara tepat. Berikut Tabel 3.1 menyajikan informasi perangkat keras.

Tabel 3.1. Informasi Perangkat Keras

No	Perangkat Keras	Spesifikasi	Deskripsi
1.	PC #1 (Server)	Intel Celeron CPU 3350 1.10GHz 1.1GHz, Intel HD Graphics 500, RAM 4 GB	Perangkat yang digunakan berfungsi sebagai server dengan Zeek, IPtables dan Zabbix yang telah terpasang

No	Perangkat Keras	Spesifikasi	Deskripsi
2.	PC#2 (Penyerang dan Monitoring)	Intel Core i5-8365U CPU 1.60GHz 1.9GHz, Intel UHD Graphics 620, 8 GB RAM	Perangkat digunakan untuk melancarkan serangan.

Pada penelitian ini juga dilakukan dengan bantuan menggunakan perangkat lunak atau bahan yang berperan penting dalam proses implementasi dan pengujian sistem. Seluruh daftar perangkat lunak dan bahan yang digunakan dapat dilihat pada Tabel 3.2, sebagai berikut.

Tabel 3.2. Informasi Perangkat Lunak

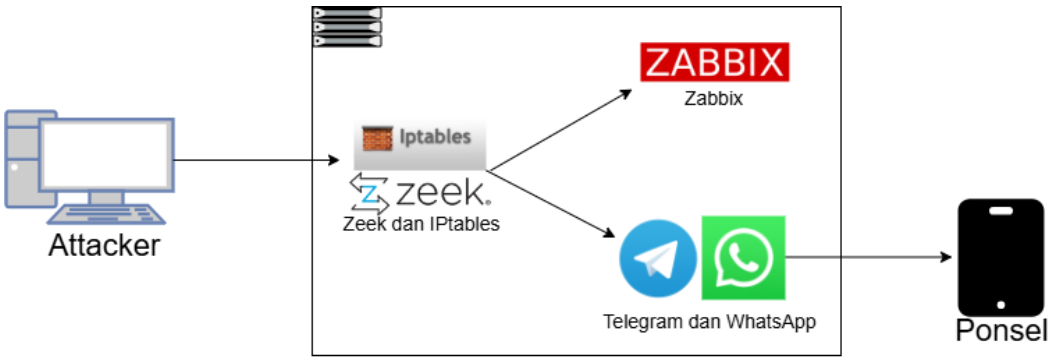
No	Perangkat Lunak	Versi	Deskripsi
1.	PC #1 (Ubuntu Server)	Ubuntu server LTS 24.04 (64-bit)	Sistem operasi ini dimanfaatkan dalam penelitian sebagai server (ntrusion Detection and Pr*event*ion System (IDPS)
2.	PC#2 (Kali Linux)	Linux 6.12.13-amd64	Sistem operasi untuk melancarkan serangan.
3.	Zeek	*Version* 7.2.0-dev.314	Perangkat lunak yang digunakan untuk mendeteksi serangan.
4.	IPtables	IPtables v1.8.10	Perangkat lunak yang digunakan untuk memblokir serangan
5.	Zabbix Server	Zabbix_server (Zabbix) 7.0.16	Perangkat lunak ini dimanfaatkan dalam penelitian untuk melakukan *monitoring* server.

No	Perangkat Lunak	Versi	Deskripsi
6.	Zabbix Agent	Zabbix_agentd (daemon) (Zabbix) 7.0.16	Perangkat lunak untuk mengirimkan data kinerja dan status server ke Zabbix Server.
7.	Zabbix Sender	Zabbix_sender (Zabbix) 7.0.16	Perankat lunak untuk mengirimkan data secara manual ke Zabbix Server dari skrip atau proses tertentu.
7.	Ngrok	Ngrok version 3.24.0	Perangkat lunak untuk mengakses server lokal dari jaringan eksternal untuk proses pengujian sistem.
8.	FTP Sever	Vsftpd: version 3.0.5	Perangkat lunak yang digunakan untuk menyerupai lingkungan server.
11.	Web Server	Apache/2.4.58 (Ubuntu)	Perangkat lunak ini dimanfaatkan dalam penelitian untuk mendeteksi dan mengatasi serangan
13.	MySQL	Ver 8.0.42-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))	Perangkat lunak ini dimanfaatkan dalam penelitian untuk mendeteksi dan mengatasi serangan
14.	Mail Server	Dovecot: version 2.3.21	Perangkat lunak yang digunakan untuk menyerupai lingkungan server.
15.	Telegram	Versi 11.14.0	Perangkat lunak untuk menerima notifikasi keamanan.

No	Perangkat Lunak	Versi	Deskripsi
16.	WhatsApp	Versi 2.25.20.82	Perangkat lunak untuk menerima notifikasi keamanan.
17.	Hping3	version 3.0.0-alpha-2	*Tool* untuk mengirim serangan pada penelitian ini.
18.	Hydra	Hydra v9.5	*Tool* untuk mengirim serangan pada penelitian ini.
19.	Nmap	Nmap 7.95	*Tool* untuk mengirim serangan pada penelitian ini.
20.	Sqlmap	Sqlmap 1.9.2	*Tool* untuk mengirim serangan pada penelitian ini.

3.3 Desain Sistem

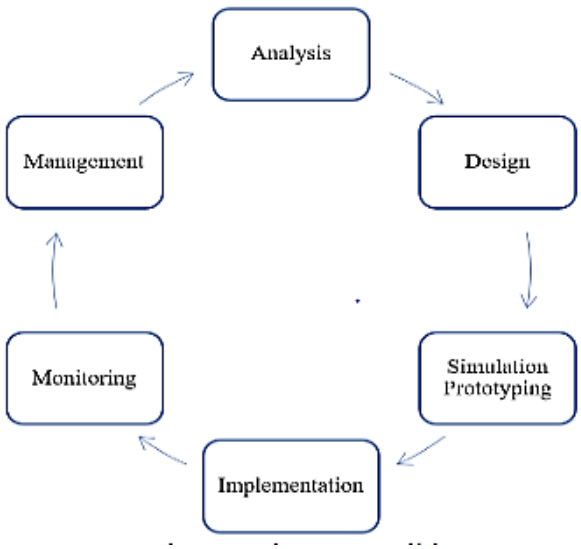
Pada tahap ini, desain sistem didefinisikan sebagai proses perancangan arsitektur serta alur kerja sistem yang disesuaikan dengan kebutuhan penelitian. Tujuan dari tahap ini adalah memberikan gambaran yang jelas mengenai interaksi antara komponen perangkat keras, perangkat lunak, dan jaringan dalam mendukung implementasi sistem deteksi serta pencegahan serangan. Sistem ini memanfaatkan Zeek sebagai detektor serangan, IPtables sebagai mekanisme pemblokiran, Zabbix sebagai platform monitoring, serta integrasi notifikasi melalui Telegram dan WhatsApp. Adapun Gambar 3.2 berikut menunjukkan arsitektur sistem pada implementasi penelitian ini.



Gambar 3.2. Arsitektur Sistem

3.4 Metode Pengembangan Sistem

Menurut Baba dkk. (2021), menjelaskan metode penelitian Network *Development Life Cycle* (NDLC). NDLC merupakan sebuah metode yang digunakan untuk mengembangkan atau mengoptimalkan infrastruktur jaringan guna meningkatkan kinerja dan efisiensi operasional, terutama dalam mendukung kebutuhan perusahaan. Metode ini mengadopsi pendekatan *continuous improvement*, di mana hasil analisis yang diperoleh pada setiap tahap siklus digunakan sebagai dasar untuk perbaikan yang berkelanjutan. NDLC adalah model yang mencakup elemen-elemen kunci dalam pengembangan sistem jaringan, yang terdiri dari berbagai fase, tahapan, langkah-langkah, atau mekanisme yang didefinisikan secara spesifik. Istilah *cycle* dalam NDLC merujuk pada proses pengembangan sistem jaringan yang berkelanjutan dan dinamis, yang mencakup seluruh tahapan dari analisis hingga pengelolaan, dan memungkinkan pengembangan sistem secara bertahap dan berkesinambungan.



Gambar 3.3. *Network Development Life Cycle* (Rahman dkk., 2024)

Penggunaan metode NDLC dalam penelitian ini didasarkan pada kesesuaiannya dengan kebutuhan penelitian yang berkaitan dengan jaringan. Tahapan dalam NDLC meliputi analisis, desain, implementasi, pengelolaan, dan pemantauan (*monitoring*), yang relevan dengan tujuan penelitian untuk merancang dan mengimplementasikan sistem *monitoring* server menggunakan Zabbix

berbasis Zeek dan IPtables dengan notifikasi melalui Telegram dan WhatsApp. Tahapan-tahapan ini memberikan kerangka kerja yang terstruktur untuk memastikan pengembangan sistem jaringan yang optimal dan efisien, hanya saja dalam penelitian ini peneliti tidak melaksanakan tahapan manajemen karena berkaitan dengan proses pemeliharaan. Berikut adalah tahapan metode penelitian yang dilakukan (Nugroho dan Herianto, 2022).

1. *Analysis* merupakan pengamatan dan pengolahan data hasil implementasi sistem, termasuk evaluasi kerja deteksi dan mitigasi serangan, kinerja perangkat lunak pendukung, serta keandalan notifikasi melalui Telegram dan WhatsApp. Informasi yang diperoleh digunakan sebagai dasar integrasi, pengembangan, dan penyempurnaan sistem *monitoring* server berbasis Zabbix, Zeek, dan IPtables agar mampu memberikan perlindungan optimal terhadap ancaman siber.

2. *Design* merupakan tahap yang dilakukan desain sistem yang mencakup perancangan topologi pengujian yang terintegrasi dengan Zabbix untuk sistem *monitoring* server berbasis Zeek dan IPtables. Desain ini juga mencakup penentuan alur kerja notifikasi yang akan dikirimkan melalui Telegram dan WhatsApp, seperti desain topologi yang dirancang agar sesuai dengan kebutuhan sistem *monitoring* server.

3. *Simulation Prototyping*, yaitu tahap yang mencakup pembuatan *prototipe* sistem *monitoring* server yang dirancang. Simulasi ini digunakan untuk menguji desain sistem, memastikan kompatibilitas komponen, dan memvalidasi fungsi deteksi dan mitigasi ancaman sebelum implementasi penuh dilakukan pada lingkungan jaringan yang sebenarnya.

4. *Implementation*, yaitu tahap implementasi yang melibatkan penerapan sistem keamanan yang dirancang ke dalam server yang relevan. Proses ini mencakup instalasi dan konfigurasi Zabbix, Zeek, IPtables dan integrasi notifikasi Telegram dan WhatsApp. Pada tahap ini, sistem yang telah dirancang diuji dalam kondisi yang mendekati lingkungan nyata, sehingga

dapat dievaluasi kinerjanya dalam mendeteksi dan merespon serangan siber.

5. *Monitoring* adalah tahap pengujian dilakukan untuk memastikan semua komponen sistem berfungsi sesuai dengan yang dirancang. Selain itu, sistem dipantau secara terus-menerus untuk mengidentifikasi potensi anomali dan mengevaluasi kinerja deteksi serta mitigasi ancaman.

6. *Management* adalah tahap akhir yang melibatkan pengelolaan sistem yang telah diimplementasikan. Pengelolaan mencakup pemeliharaan kebijakan keamanan, pembaruan perangkat lunak, pengelolaan lalu lintas jaringan, serta penyesuaian notifikasi yang dihasilkan oleh Zeek dan IPtables. Dengan pengelolaan yang baik, diharapkan sistem dapat berfungsi secara optimal dan berkelanjutan dalam menghadapi ancaman siber.

3.4.1 *Analysis*

Tahap analisis dilakukan setelah sistem *monitoring* selesai diimplementasikan dan proses pengujian berjalan. Tujuan tahap ini adalah mengevaluasi kinerja sistem berdasarkan hasil nyata dari proses *monitoring*, deteksi, mitigasi, dan pengiriman notifikasi yang terjadi selama pengujian. Penjelasan rinci analisis *pasca monitoring* yang akan digunakan sebagai berikut.

3.4.1.1 Jenis dan Sumber Data

Pada penelitian ini, jenis dan sumber data yang digunakan adalah data utama yang diperoleh langsung dari hasil rumusan masalah sistem. Data utama ini dikumpulkan dari berbagai tahapan pengujian terhadap sistem *monitoring* server yang dibangun menggunakan kombinasi Zeek, IPtables, dan Zabbix, serta sistem notifikasi otomatis menggunakan Telegram dan WhatsApp.

Sumber data utama dalam penelitian ini diperoleh dari log sistem, catatan hasil pengujian, serta pengamatan langsung terhadap jalannya eksperimen. Log yang dikumpulkan berasal dari output Zeek, IPtables, dan Zabbix, sedangkan catatan pengujian disusun berdasarkan waktu pelaksanaan, jumlah serangan yang diuji, serta respon sistem terhadap setiap skenario. Pengamatan dilakukan secara

langsung selama proses pengujian berlangsung untuk memastikan keakuratan data yang diperoleh. Data utama yang dihimpun tersebut memberikan informasi mendalam mengenai performa sistem secara keseluruhan, serta kinerja integrasi antara komponen-komponen yang digunakan dalam upaya *monitoring* dan pencegahan serangan pada jaringan.

3.4.1.2 Teknik Pengumpulan Data

Penelitian ini menggunakan teknik pengumpulan data yang komprehensif untuk memperoleh informasi yang dibutuhkan dalam analisis Zabbix untuk sistem *monitoring* server berbasis Zeek dan IPtables dengan notifikasi melalui Telegram dan WhatsApp. Teknik pengumpulan data yang digunakan adalah sebagai berikut (Nadhipa, 2024).

1. Observasi dilakukan melalui pengamatan langsung terhadap implementasi Zeek, IPtables, dan Zabbix yang dilengkapi dengan sistem notifikasi Telegram dan WhatsApp. Pengamatan mencakup pemantauan operasional sistem, respon terhadap serangan siber, serta evaluasi kinerja keseluruhan sistem keamanan jaringan yang diuji. Zabbix digunakan sebagai alat pemantauan tambahan untuk mengawasi kinerja jaringan dan mendeteksi anomali dalam lalu lintas data.

2. Analisis log dilakukan dengan mempelajari data log yang dihasilkan oleh Zeek dan Zabbix terkait deteksi dan mitigasi serangan siber. Log ini memberikan informasi rinci mengenai jenis serangan yang terdeteksi, waktu respon sistem, serta langkah-langkah mitigasi yang dilakukan untuk menangani serangan. Zabbix membantu dalam mengumpulkan metrik performa jaringan dan mendeteksi anomali yang dapat menjadi indikasi potensi ancaman keamanan.

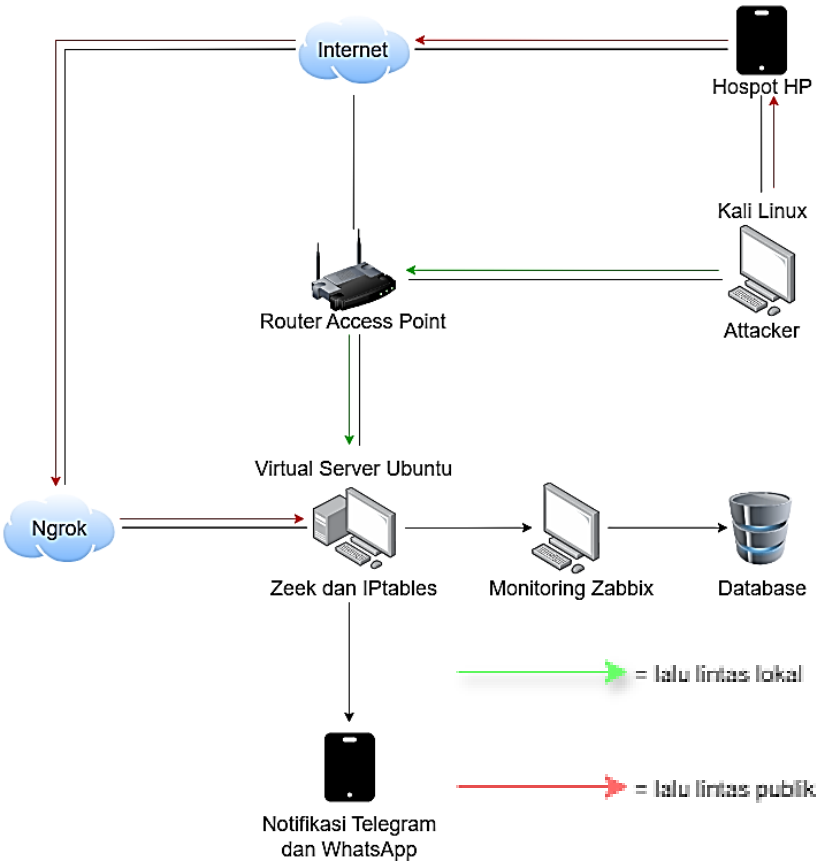
3. Dokumentasi teknis mencakup pengumpulan informasi dari panduan teknis, petunjuk konfigurasi, dan dokumentasi lainnya yang relevan dengan implementasi Zeek, IPtables, Zabbix, serta integrasi notifikasi Telegram dan WhatsApp. Dokumentasi ini membantu memahami proses

implementasi secara lebih mendalam dan memastikan konfigurasi sistem berjalan dengan optimal.

3.4.2 *Design*

3.4.2.1 Topologi Pengujian

Pada Gambar 3.4., peneliti melakukan perancangan topologi pengujian yang akan digunakan sebagai dasar simulasi dalam penelitian ini. Perancangan dilakukan dengan memanfaatkan *Virtual Machine* (VM), yaitu *VirtualBox*, untuk menciptakan lingkungan virtual yang menyerupai kondisi nyata.



Gambar 3.4. Topologi Pengujian

Pada gambar ini, peneliti memperlihatkan dua skenario pengujian IP Lokal dan IP Publik dengan arah serangan yang diilustrasikan menggunakan panah hijau untuk lalu lintas lokal, serta panah merah untuk lalu lintas publik. Berikut penjelasan dari kedua skenario pada Gambar 3.4. diatas:

1. Pada skenario IP local, perangkat *monitoring* Zabbix, Ubuntu Server, dan *Attacker* berada dalam satu subnet lokal yang sama melalui Access Point/Router bermode bridge *adapter* melalui *Wireless Fidelity* (Wi-Fi). Sehingga ketika Attacker meluncurkan serangan, paket langsung diarahkan ke Ubuntu Server melalui jaringan LAN. Zeek kemudian memeriksa setiap paket berdasarkan signature yang telah ditetapkan, lalu, apabila pola serangan terdeteksi, menginstruksikan IPtables untuk melakukan pemblokiran.
2. Pada skenario IP Publik, Ubuntu Server menjalankan Ngrok client untuk membuka tunnel aman ke Ngrok *Cloud Service*, sehingga menghasilkan endpoint publik. Melalui *path* publik inilah *monitoring* Zabbix dan *Attacker* yang kali ini berada di luar jaringan local mengakses server. Paket eksternal ditransmisikan oleh Ngrok *Cloud* ke VM, kemudian diinspeksi oleh Zeek dan diproses oleh IPtables dengan mekanisme yang sama seperti pada skenario lokal.

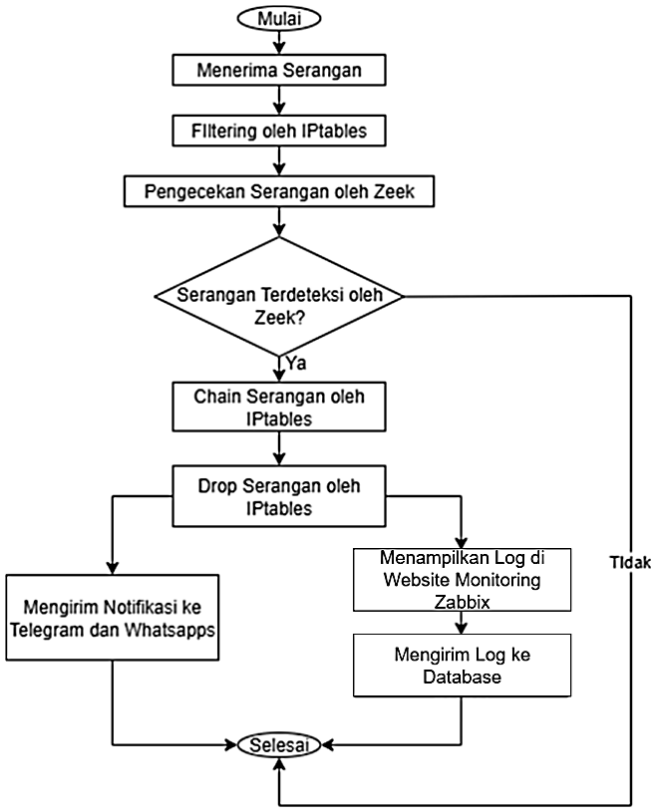
Seluruh hasil deteksi pada kedua skenario kemudian dicatat oleh Zeek dalam bentuk log. Berikutnya, Zabbix menarik data log tersebut untuk memantau status keamanan dan ditampilkan melalui antarmuka pemantauan Zabbix, Setelah itu, informasi tersebut secara otomatis disimpan ke dalam sistem basis data. Proses ini bertujuan untuk memastikan bahwa seluruh riwayat aktivitas jaringan, termasuk percobaan serangan, hasil inspeksi, serta tindakan pemblokiran, terdokumentasi dengan baik dan dapat diakses kembali sewaktu-waktu untuk keperluan analisis lanjutan atau audit keamanan. Lalu selain dikirim ke Zabbix, log juga diteruskan secara otomatis ke Telegram dan WhatsApp sebagai notifikasi. Dengan alur ini, mulai dari peluncuran serangan, inspeksi paket, pemblokiran, hingga penyampaian alert tercermin secara terpadu dalam sistem.

3.4.3 *Simulation*

3.4.3.1 Cara Kerja Sistem

Sstem yang digunakan dalam penelitian ini dibangun dengan menggabungkan beberapa komponen yang saling terintegrasi yang bisa dilihat pada Gambar 3.5., di mana masing-masing memiliki peran mulai dari Zeek

sebagai IDS, IPtables sebagai mitigasi serta penyaringan terhadap serangan, hingga Zabbix, Telegram, dan WhatsApp sebagai informasi melalui platform pemantauan dan layanan pemberitahuan otomatis ke Administrator sistem.



Gambar 3.5. Flowchart Cara Kerja Sistem

Cara kerja sistem secara keseluruhan berlangsung melalui beberapa tahapan yang terstruktur, mulai dari penerimaan serangan hingga log serangan tersebut dapat ditampilkan di Zabbix dan diterima oleh Telegram dan WhatsApp. Adapun cara kerja sistem secara keseluruhan dapat dijelaskan melalui tahapan-tahapan berikut:

1. Penerimaan Serangan oleh Server

Langkah pertama dimulai ketika server menerima paket serangan dari *attacker*, dimana paket ini akan berasal dari IP lokal dan IP publik yang telah dijelaskan pada sub bab 3.4.2.1 mengenai Topologi Pengujian sebelumnya. Paket tersebut masuk melalui *interface* jaringan pada server yang telah dikonfigurasi untuk memantau lalu lintas secara terus-menerus. Pada tahap ini, semua serangan yang masuk, akan diteruskan ke sistem deteksi untuk

dianalisis lebih lanjut. Proses penerimaan ini menjadi titik awal bagi sistem untuk diproses oleh tahap selanjutnya.

2. *Filtering* oleh IPtables

Sebelum dianalisis lebih lanjut, lalu lintas tersebut terlebih dahulu melewati *Filtering* IPtables. Pada tahap ini, IPtables menerapkan *basic filtering* seperti menerima atau menolak paket berdasarkan IP *Address*, *port*, atau protokol tertentu. Proses ini berfungsi sebagai penyaringan awal untuk mengurangi beban analisis pada tahap berikutnya. Meskipun demikian, *filtering* pada tahap ini belum sepenuhnya mengeksekusi mitigasi terhadap serangan kompleks, melainkan meneruskan lalu lintas yang terindikasi mencurigakan ke komponen analisis yang lebih canggih, yaitu Zeek, untuk dilakukan deteksi dan klasifikasi ancaman secara mendalam.

3. Pemeriksaan oleh Zeek

Setelah melakukan *filtering* oleh IPtables, Zeek akan melakukan inspeksi mendalam terhadap paket-paket yang masuk. Zeek menganalisis setiap paket berdasarkan event handler yang telah didefinisikan di dalam skrip berformat .zeek, yang berisi aturan pendeteksian serangan. Jika Zeek menemukan pola yang tidak sesuai dengan kebijakan dalam skrip Zeek, maka paket tidak akan melanjutkan ke proses berikutnya dan dianggap selesai.

4. *Chain* Serangan oleh IPtables

Setelah Zeek mengonfirmasi adanya serangan, sistem akan menginstruksikan IPtables untuk memanggil *chain* khusus yang telah disiapkan. *Chain* ini berfungsi sebagai jalur pemrosesan terpisah, di mana setiap paket berbahaya yang terdeteksi akan dipisahkan dari lalu lintas normal. Dengan pemisahan ini, IPtables dapat menerapkan aturan lanjutan yang lebih spesifik sesuai jenis serangan. Pendekatan ini memastikan bahwa mitigasi dilakukan secara terstruktur dan tepat sasaran, sehingga meminimalkan risiko gangguan terhadap layanan normal.

5. Drop Serangan oleh IPtables

Setelah paket dikelompokkan berdasarkan tipe serangan, IPtables akan menjalankan proses pemblokiran sesuai aturan yang telah ditentukan. Aturan pemblokiran ini berasal dari *rule* IPtables yang dimasukkan secara otomatis melalui *event handler* pada skrip Zeek ketika serangan terdeteksi. Dengan mekanisme ini, IPtables tidak hanya berfungsi sebagai penyaring awal, tetapi juga sebagai komponen aktif dalam pencegahan. Paket yang terindikasi berbahaya akan langsung di-*drop*, sehingga tidak dapat mengakses layanan atau sumber daya di dalam server. Pendekatan ini memastikan bahwa setiap ancaman dapat dihentikan pada tahap awal sebelum berpotensi menimbulkan kerusakan lebih lanjut.

6. **Menampilkan Log di Web Monitoring Zabbix**

Setiap aktivitas yang berkaitan dengan serangan, baik yang terdeteksi maupun yang berhasil diblokir, akan dicatat oleh Zeek dalam bentuk log. Log ini kemudian dikirimkan ke Zabbix untuk ditampilkan. Hal ini bertujuan agar Administrator dapat *monitoring* server apalagi terdapat serangan.

7. Pengiriman *Log* ke *Database*

Setelah log ditampilkan pada Zabbix, data tersebut akan disimpan ke dalam sistem *database* secara otomatis. Penyimpanan ini bertujuan untuk mendokumentasikan riwayat deteksi dan respon terhadap serangan. Lalu memungkinkan dilakukannya analisis lanjutan berdasarkan data historis yang tersimpan.

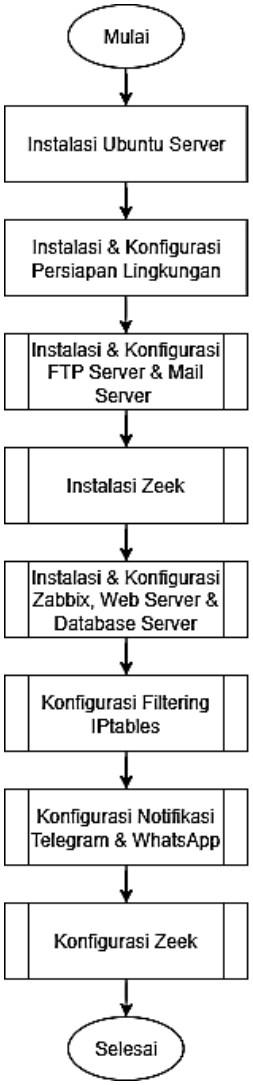
8. Pengiriman Notifikasi ke Telegram dan WhatsApp

Sistem bekerja dengan mekanisme notifikasi otomatis yang dipicu saat aktivitas mencurigakan terdeteksi. Zeek secara otomatis mengeksekusi skrip yang berisi instruksi untuk mengakses kredensial dan API dari layanan Telegram dan WhatsApp. Skrip ini kemudian memproses dan mengirimkan pesan peringatan secara cepat kepada Administrator melalui kedua platform

tersebut, sehingga memungkinkan respon cepat terhadap potensi ancaman jaringan.

3.4.4 *Implementation*

Pada Gambar 3.6. memperlihatkan alur implementasi sistem yang dilakukan secara bertahap dalam penelitian ini. Setiap tahapan dirancang agar sistem *monitoring* dan deteksi intrusi dapat berjalan secara optimal. Adapun dam langkah-langkah implementasinya dijelaskan sebagai berikut.



Gambar 3.6. Alur Implementasi Sistem

1. Instalasi Ubuntu Server

Proses implementasi diawali dengan instalasi sistem operasi Ubuntu Server 24.04 LTS pada server. Sistem operasi ini dipilih karena stabil, ringan, dan mendukung kebutuhan konfigurasi sistem *monitoring* dan keamanan jaringan yang akan dibangun.

2. Konfigurasi IP *Address*

Setelah sistem operasi terpasang, server dikonfigurasi menggunakan pengaturan *Dynamic Host Configuration Protocol* (DHCP) agar secara otomatis memperoleh alamat IP dari jaringan. Selain itu, antarmuka jaringan server telah disetting dengan mode *bridge adapter* melalui koneksi Wi-Fi *host*, sehingga server dapat langsung terhubung dengan jaringan lokal maupun internet. Konfigurasi ini memungkinkan server berperan sebagai target *monitoring* serta pengujian terhadap berbagai jenis serangan.

3. Instalasi dan Konfigurasi FTP Server dan Mail Server

. Tahap berikutnya adalah instalasi serta konfigurasi layanan tambahan yang meliputi FTP Server menggunakan vsftpd dan Mail Server menggunakan Postfix. FTP Server digunakan untuk mendukung simulasi transfer dan akses data antar perangkat dalam jaringan, sedangkan Mail Server berfungsi sebagai sarana komunikasi email antarsistem. Kedua layanan ini dibutuhkan agar lingkungan server yang dibangun lebih representatif terhadap kondisi nyata, khususnya dalam menguji efektivitas sistem *monitoring* terhadap aktivitas lalu lintas data dan komunikasi.

4. Instalasi Zeek

Instalasi Zeek dilakukan sebagai langkah awal untuk menyiapkan sistem deteksi intrusi yang dapat memantau lalu lintas jaringan secara langsung pada server. Instalasi dilakukan pada sistem operasi Ubuntu Server, dengan menyiapkan terlebih dahulu semua dependensi yang diperlukan dan pustaka pendukung lainnya. Proses instalasi dilakukan dengan mengunduh source code Zeek versi terbaru, lalu dikompilasi secara manual untuk memastikan

kompatibilitas dengan sistem yang digunakan. Setelah proses kompilasi berhasil, Zeek kemudian diatur untuk berjalan pada *interface* jaringan yang telah ditentukan. Instalasi ini bertujuan untuk memastikan bahwa Zeek dapat berfungsi secara optimal dan siap dikonfigurasi lebih lanjut untuk kebutuhan deteksi serangan dan integrasi sistem *monitoring*.

5. Instalasi dan Konfigurasi Zabbix, Web Server, dan Database Server

Setelah layanan dasar terpasang, dilakukan instalasi dan konfigurasi Zabbix sebagai sistem *monitoring* utama. Dalam tahap ini, Database Server yang digunakan adalah MySQL, yang berfungsi untuk menyimpan seluruh data hasil *monitoring* dan log serangan. Sementara itu, Web Server yang digunakan adalah Apache2, yang berperan sebagai antarmuka pengguna untuk menampilkan informasi *monitoring* secara visual melalui *dashboard* Zabbix. Kombinasi keduanya memungkinkan Zabbix beroperasi secara optimal dalam mencatat, menyimpan, dan menyajikan data pemantauan jaringan.

6. Konfigurasi Filtering IPtables

Filtering IPtables dikonfigurasi melalui skrip shell yang memuat aturan lalu lintas jaringan pada server berdasarkan jenis serangan yang diuji. Filtering ini disusun sedemikian rupa agar dapat memblokir koneksi yang berbahaya sesuai hasil deteksi Zeek, namun tetap menjaga agar lalu lintas penting tidak ikut terblokir. Hal ini bertujuan untuk mendukung proses pengujian secara optimal tanpa mengganggu konektivitas yang sah.

7. Konfigurasi Notifikasi Telegram dan WhatsApp

. Selanjutnya untuk mendukung respon cepat terhadap serangan, sistem dilengkapi dengan fitur notifikasi otomatis melalui Telegram dan WhatsApp. Integrasi dengan Telegram dilakukan menggunakan *BotFather*, yang menghasilkan *bot* token untuk mengirim pesan secara langsung ke pengguna. Sementara itu, untuk WhatsApp digunakan layanan API pihak ketiga yaitu *CallMeBot*, yang memungkinkan pengiriman pesan melalui endpoint HTTP secara mudah, sehingga Administrator dapat segera menerima peringatan meskipun sedang tidak mengakses sistem secara langsung.

8. Konfigurasi Zeek

Tahap terakhir adalah konfigurasi Zeek agar mampu mendeteksi berbagai jenis ancaman server. Konfigurasi dilakukan dengan menyusun skrip khusus yang berisi serangkaian *event* handler. Skrip tersebut tidak hanya berfungsi untuk melakukan deteksi, tetapi juga dirancang untuk menjalankan tindakan lanjutan secara otomatis. Salah satu tindakan tersebut adalah pemblokiran IP penyerang melalui eksekusi perintah IPtables yang dipanggil langsung dari dalam skrip Zeek. Dengan pendekatan ini, sistem dapat memberikan respon cepat terhadap ancaman tanpa perlu intervensi manual. Selain itu, skrip Zeek juga dikembangkan untuk mengirimkan notifikasi ke platform Telegram dan WhatsApp. Mekanisme ini dilakukan dengan memanggil skrip eksternal yang telah terhubung dengan API masing-masing platform, sehingga setiap serangan yang terdeteksi dapat langsung diberitahukan kepada Administrator jaringan melalui pesan instan. Pesan yang dikirimkan berisi informasi terkait jenis serangan, alamat IP sumber, dan waktu kejadian. Sebagai bagian dari integrasi dengan sistem *monitoring*, Zeek juga dikonfigurasi untuk mengirimkan data log ke Zabbix. Seluruh konfigurasi ini dilakukan untuk membentuk sistem keamanan jaringan yang aktif, otomatis, dan terintegrasi, di mana Zeek berperan sebagai komponen utama dalam proses deteksi, mitigasi, dan pengiriman notifikasi dan log.

3.4.5 *Monitoring*

Monitoring merupakan salah satu tahapan penting NDLC yang berfungsi untuk memantau jalannya sistem serta memastikan bahwa seluruh komponen berjalan sesuai dengan perencanaan. Dalam penelitian ini, *monitoring* dilakukan dengan menggunakan Zabbix sebagai sistem utama untuk mengawasi aktivitas server dan jaringan. Sistem ini juga bekerja sama dengan Zeek sebagai alat deteksi serta IPtables sebagai alat pencegahan serangan jaringan, dan didukung oleh notifikasi otomatis melalui Telegram dan WhatsApp. *Monitoring* tidak hanya mencatat lalu lintas dan aktivitas serangan server, tetapi juga menjadi dasar dalam melakukan evaluasi terhadap efektivitas dan keandalan sistem.

3.4.6 Rancangan Monitoring

Monitoring dilakukan dengan memanfaatkan fitur-fitur yang ada dalam Zabbix sebagai platform utama untuk memantau aktivitas server. Zabbix diinstal pada server dan dikonfigurasi untuk menerima data dari log yang dihasilkan oleh Zeek dan IPtables. Proses *monitoring* ini melibatkan beberapa langkah penting sebagai berikut.

1. Pengaturan Logging: Zeek dikonfigurasi untuk mencatat setiap aktivitas server secara detail, termasuk koneksi mencurigakan, upaya serangan, serta lalu lintas yang dianggap tidak wajar. Log yang dihasilkan oleh Zeek berisi informasi penting serangan yang terjadi. Log yang dihasilkan oleh Zeek ini kemudian dikirim ke Zabbix. Dengan demikian, log yang tampil di Zeek dan Zabbix memiliki informasi yang sama. Sementara itu, log dari IPtables mencatat informasi penting serangan yang diblokir. Log ini digunakan untuk memastikan bahwa aturan mitigasi berjalan sesuai dengan konfigurasi dan dapat dievaluasi untuk penyesuaian lebih lanjut jika diperlukan. Sebagai langkah verifikasi tambahan, digunakan pula *tcpdump* untuk menangkap lalu lintas jaringan secara lansung sebagai pembuktian bahwa serangan yang tercatat benar-benar terjadi dan bukan hasil dari deteksi palsu.
2. Integrasi dengan Telegram dan WhatsApp: Untuk mempercepat respon terhadap ancaman yang terdeteksi, sistem *monitoring* ini diintegrasikan dengan layanan notifikasi otomatis melalui Telegram dan WhatsApp. Ketika Zeek atau IPtables mendeteksi adanya aktivitas berbahaya, informasi log akan diteruskan dalam bentuk notifikasi kepada Administrator melalui kedua platform tersebut. Hal ini memungkinkan Administrator segera mengambil tindakan mitigasi, bahkan ketika tidak sedang berada didepan sistem *monitoring*.
3. Analisis Visual dan Laporan Harian: Zabbix menyajikan data *monitoring* yang terdiri dari tabel log serangan masuk, diagram jumlah serangan, grafik CPU, *bandwidth*, dan RAM, serta informasi saat Zeek dimatikan dan dinyalakan. Laporan harian ini dibentuk berdasarkan akumulasi data

log dari Zeek yang dikumpulkan secara berkelanjutan. Dengan demikian, setiap kejadian, baik berupa serangan maupun aktivitas mencurigakan lainnya, terdokumentasi dan terarsipkan dalam sistem Zabbix. Analisis ini berguna untuk mengidentifikasi tren serangan atau pola aktivitas abnormal yang bisa digunakan sebagai bahan evaluasi keamanan server jangka panjang.

3.5 Pengujian Sistem

Dalam penelitian ini digunakan tiga skema pengujian yang dirancang untuk memastikan bahwa implementasi sistem dapat berjalan sesuai dengan fungsinya. Adapun skema pengujian yang diterapkan disajikan sebagai berikut.

3.5.1 Pengujian Fungsionalitas Intergrasi Sistem Keseluruhan

Pengujian ini dilaksanakan sebagai tahap validasi awal untuk memastikan seluruh komponen sistem dapat berfungsi dengan baik serta terintegrasi secara optimal, baik pada lingkungan IP lokal maupun IP publik. Adapun aspek-aspek yang menjadi fokus dalam pengujian fungsionalitas sistem adalah sebagai berikut.

1. Fungsionalitas Integrasi Zeek dan IPtables: Pada pengujian ini, peneliti memastikan bahwa Zeek dapat berjalan dengan baik dalam mendeteksi lalu lintas jaringan yang mencurigakan, serta memverifikasi bahwa skrip yang dibuat berhasil memicu IPtables untuk memblokir alamat IP penyerang berdasarkan log yang dihasilkan oleh Zeek. Pengujian dilakukan baik pada lalu lintas yang berasal dari IP lokal maupun IP publik.
2. Fungsionalitas Integrasi Zabbix: Pengujian ini dilakukan untuk memastikan bahwa log hasil deteksi dari Zeek dapat dikirimkan dan diolah oleh Zabbix, kemudian divisualisasikan dengan benar melalui *dashboard*. Validasi dilakukan dengan memicu serangan dari IP lokal maupun IP publik, lalu memeriksa apakah status serangan tersebut tercatat dan ditampilkan di sistem monitoring Zabbix.
3. Fungsionalitas Integrasi Telegram dan WhatsApp: Pengujian ini dilakukan untuk memastikan bahwa sistem dapat mengirimkan notifikasi serangan

secara otomatis melalui Telegram dan WhatsApp kepada administrator jaringan. Pengujian dilakukan dengan memicu serangan dari IP lokal maupun IP publik, lalu memeriksa apakah notifikasi diterima secara *realtime* sesuai dengan hasil deteksi Zeek.

3.5.2 Pengujian Impelementasi dan Kinerja Sistem

Pengujian implementasi dan kinerja sistem merupakan tahapan yang bertujuan untuk memastikan bahwa sistem yang telah dirancang tidak hanya berhasil diterapkan, tetapi juga mampu berfungsi secara optimal sesuai dengan tujuan penelitian. Pada tahap ini dilakukan serangkaian uji coba untuk memverifikasi efektivitas integrasi komponen sistem, tingkat akurasi deteksi, kecepatan respons terhadap serangan, serta keandalan dalam memberikan notifikasi dan monitoring secara cepat. Adapun pengujian implementasi dan kinerja yang dilakukan meliputi:

1. Implementasi Zeek dan IPtables: berisi pengujian pendeteksian serangan oleh sistem Zeek serta tindakan pencegahan yang dilakukan oleh IPtables. Data ini juga mencatat jenis serangan yang berhasil dideteksi dan dicegah, serta waktu respon sistem terhadap setiap aktivitas mencurigakan.
2. Kinerja Zeek dan IPtables: pengujian performa sistem yang dilakukan sebanyak tiga kali dengan variasi jumlah serangan yang berbeda. Data ini digunakan untuk mengukur kinerja Zeek dan IPtables dalam mendeteksi dan mencegah serangan dengan tingkat intensitas yang bervariasi serta memperhatikan pengaruh serangan terhadap sumber daya sistem, seperti penggunaan CPU, RAM, dan bandwidth server. Untuk mempermudah analisis, data kinerja dibagi menjadi dua kategori berdasarkan karakteristik serangan:
    a. Serangan DoS: Serangan ini mengandalkan jumlah paket yang sangat besar (flooding attack), sehingga indikator keberhasilan pencegahan lebih relevan diukur berdasarkan persentase paket yang berhasil terdeteksi dan diblokir dibandingkan dengan total paket yang dikirimkan. Dengan cara ini, kinerja sistem dapat

dinilai dari kemampuannya menangani trafik yang sangat tinggi secara cepat.

b. Serangan Non-DoS: Serangan ini umumnya tidak bergantung pada jumlah paket yang besar, melainkan memanfaatkan celah atau teknik tertentu. Pada kategori ini, indikator keberhasilan diukur berdasarkan jumlah serangan yang berhasil dideteksi dan dimitigasi pada setiap skenario uji. Metode ini dipilih karena jumlah paket tidak selalu mencerminkan tingkat ancaman pada serangan jenis ini.

Pembagian kategori ini dilakukan agar metode pengukuran keberhasilan sesuai dengan karakteristik masing-masing serangan, sehingga hasil analisis lebih akurat dan relevan.

3. Kinerja Zabbix: pengujian difokuskan pada Zabbix dalam memantau hasil deteksi serangan yang diperoleh dari Zeek. Aspek yang diuji mencakup keberhasilan terkirimnya log dari Zeek ke Zabbix dan mengukur total respon kecepatan log tampil di Zabbix serta akurasi visualisasi pada *dashboard*
4. Kinerja WhatsApp dan Telegram: berisi catatan waktu pengiriman notifikasi dari sistem ke media komunikasi pengguna. Data ini mencakup kecepatan sistem dalam mengirimkan notifikasi setelah deteksi atau pencegahan dilakukan.

3.5.3 Pengujian Serangan Siber

Berikut merupakan uraian lebih rinci terkait jenis-jenis serangan yang akan diuji dalam penelitian ini. Setiap serangan akan disimulasikan baik melalui IP Lokal maupun IP Publik guna mengevaluasi efektivitas sistem secara menyeluruh. Pengujian ini bertujuan untuk mengukur kemampuan Zeek dalam mendeteksi, IPtables dalam memblokir, serta Zabbix dalam memantau, yang kemudian dilengkapi dengan notifikasi melalui WhatsApp dan Telegram. Adapun pengujian serangan yang dilakukan meliputi:

1. ICMP *Flood*, yaitu serangan yang dilakukan dengan mengirimkan sejumlah besar paket ICMP untuk membanjiri server target,

mensimulasikan serangan DoS. Pengujian ini bertujuan untuk menilai kemampuan sistem dalam mendeteksi dan merespon serangan tersebut.

2. ARP *Spoofing*, yaitu serangan yang memanfaatkan pengiriman paket ARP palsu untuk memanipulasi tabel ARP korban dengan cara memetakan alamat IP sah ke alamat MAC milik penyerang, sehingga lalu lintas data dialihkan dan memungkinkan pengujian kemampuan sistem dalam mendeteksi serta memblokir upaya pemalsuan alamat *link* layer tersebut.
3. XSS *Attack*, yaitu serangan yang dilakukan dengan menyuntikkan skrip berbahaya ke dalam halaman web yang dilayani oleh server target. Tujuannya adalah untuk mengukur kinerja sistem dalam mendeteksi dan mencegah serangan injeksi kode dari sisi klien.
4. UDP *Flood*, yaitu serangan yang dilakukan dengan mengirimkan sejumlah besar paket UDP ke server target untuk mensimulasikan serangan DoS. Sistem diuji untuk mendeteksi dan mengelola trafik berlebih akibat serangan ini.
5. SQL *Injection*, yaitu serangan yang melibatkan penyuntikan perintah SQL melalui input aplikasi web dengan tujuan memanipulasi *database*. Sistem diuji untuk mendeteksi dan mencegah serangan injeksi ini.
6. *Port Scanning*, yaitu serangan yang dilakukan dengan memindai *port* pada server target untuk mengidentifikasi *port* yang terbuka. Hal ini digunakan untuk mengevaluasi kemampuan sistem dalam mendeteksi dan memperingatkan aktivitas pengintaian.
7. *Ping of Death*, yaitu serangan yang melibatkan pengiriman paket ICMP dengan ukuran berlebihan untuk melumpuhkan server target. Pengujian ini mengevaluasi kemampuan sistem dalam mengidentifikasi dan menangani serangan tersebut.
8. *Brute Force*, yaitu serangan yang dilakukan dengan mencoba berbagai kombinasi username dan password secara sistematis untuk mengakses server target. Sistem diuji untuk mendeteksi dan mencegah upaya *login* yang mencurigakan.
9. SYN *Flood*, yaitu serangan yang dilakukan dengan mengirimkan sejumlah besar paket SYN palsu ke server target untuk menghabiskan sumber daya

dengan koneksi TCP yang tidak lengkap. Pengujian ini mengevaluasi kinerja sistem dalam mendeteksi dan mengatasi serangan ini.

Setiap tahapan dalam skema pengujian melibatkan pemantauan dan pencatatan rinci terhadap kemampuan deteksi Zeek dan IPtables, serta respon sistem terhadap masing-masing jenis serangan. Hasil dari pengujian ini akan dianalisis untuk menentukan kekuatan dan kelemahan sistem dalam melindungi jaringan dari berbagai ancaman siber.

3.6 Evaluasi Kerja Monitoring

Evaluasi dilakukan untuk menilai konsistensi dan keandalan sistem dalam merespon dan menampilkan data serangan. Evaluasi ini mencakup beberapa aspek sebagai berikut.

1. Akurasi Kesesuaian Deteksi dan Mitigasi Serangan: Sistem diuji dengan cara mensimulasikan serangan beberapa kali. Tujuannya adalah memastikan bahwa setiap rule yang telah ditetapkan mampu mendeteksi dan merespon hanya terhadap jenis serangan yang dikirimkan, tanpa memicu deteksi terhadap serangan lain yang tidak terjadi. Evaluasi ini menilai tingkat akurasi penerapan rule di Zeek dan IPtables.
2. Ketepatan dan Stabilitas Notifikasi: Pengujian dilakukan untuk memastikan bahwa sistem notifikasi mampu mengirimkan pesan secara cepat dan tepat setiap kali ada ancaman. Evaluasi ini mencakup ketepatan waktu dan isi pesan yang dikirim ke Telegram dan WhatsApp.
3. Responivitas Sistem: Waktu yang dibutuhkan sistem untuk menampilkan log sebagai tanda deteksi dan mitigasi serangan, serta waktu pengirimkan log ke Zabbix menjadi indikator utama dalam evaluasi ini.
4. Pengaruh terhadap Sumber Daya Sistem: Selama proses pengujian, pemantauan dilakukan terhadap penggunaan CPU, RAM, dan *bandwidth* untuk melihat apakah terjadi peningkatan beban ketika sistem melakukan deteksi serangan, mencatat log, dan mengirimkan notifikasi. Tujuan dari evaluasi ini adalah untuk mengetahui seberapa besar pengaruh sistem

monitoring terhadap performa server, terutama saat terjadi serangan dengan intensitas tinggi.

5. Akurasi Kesesuaian Monitoring: Sistem diuji dengan memantau aktivitas serangan melalui platform Zabbix. Tujuannya adalah untuk memastikan akurasi *monitoring* dalam menampilkan data penggunaan CPU, RAM, dan *bandwidth* yang sesuai dengan aktivitas serangan yang terjadi. Evaluasi ini juga digunakan untuk menilai ketepatan aktivasi *trigger* Zeek pada saat serangan terdeteksi, serta kesesuaian diagram jumlah serangan yang ditampilkan dengan data hasil pengujian.

Dengan proses *monitoring* dan evaluasi yang menyeluruh, sistem ini diharapkan mampu memberikan pengawasan yang akurat, cepat, dan dapat dipercaya dalam menangani ancaman jaringan. Hasil evaluasi ini menjadi dasar untuk menjawab rumusan masalah yang berkaitan dengan integrasi keseluruhan sistem, kienrja dan implementasi Zeek dan IPtables, dan kinerja Zabbix serta notifikasi Telegram dan WhatsApp.

3.7 Pelaporan

Hasil penelitian yang diperoleh akan disusun dalam bentuk laporan tertulis berupa skripsi, yang memuat uraian mengenai perancangan sistem, proses implementasi, serta hasil pengujian. Laporan ini diharapkan dapat memberikan kontribusi bagi pengembangan sistem keamanan server yang lebih optimal, khususnya pada aspek deteksi dan pencegahan intrusi.