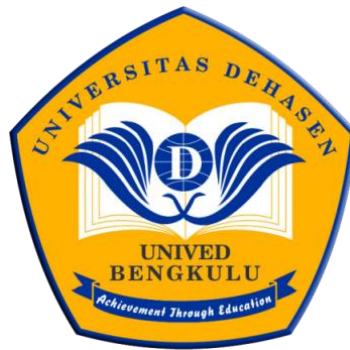


**REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS  
SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA *WEB*  
*SERVER***

**SKRIPSI**



**Oleh :**

**ANNE LESTYEA**  
**NPM. 21010095P**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DEHASEN  
BENGKULU  
2023**

**REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS  
SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA *WEB*  
*SERVER***

**SKRIPSI**

**OLEH :**

**ANNE LESTYEA  
NPM. 21010095P**

*Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer*

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DEHASEN  
BENGKULU  
2023**

**REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS  
SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA *WEB*  
*SERVER***

**SKRIPSI**

**OLEH :**

**ANNE LESTYEA**  
**NPM. 21010095P**

**Disetujui Oleh :**

**Pembimbing Utama,**



**Hari Asprivono, S.Kom, M.Kom**  
**NIDN. 02.060587.05**

**Pembimbing Pendamping,**



**Abdussalam Al Akbar, S.Kom, M.Kom**  
**NIDN. 02.051092.01**

**Mengetahui:**

**Ketua Program Studi,**



**Liza Yulianti, S.Kom, M.Kom**  
**NIDN. 02.160772.01**

**REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS  
SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA *WEB***

***SERVER***

**SKRIPSI**


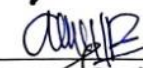


**OLEH :**

**ANNE LESTYEA  
NPM. 21010095P**

Telah dipertahankan didepan TIM Penguji Fakultas Ilmu Komputer Pada :

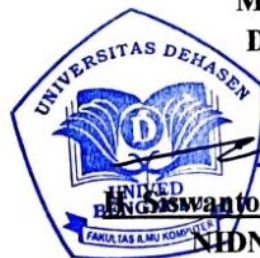
Hari : **Rabu**

Tanggal : **23 Agustus 2023**

<b>Penguji</b>	<b>Nama</b>	<b>NIDN</b>	<b>Tanda Tangan</b>
<b>Ketua</b>	Hari Aspriyono, S.Kom, M.Kom	02.060587.05	
<b>Anggota</b>	Abdussalam Al Akbar, S.Kom, M.Kom	02.051092.01	
<b>Anggota</b>	Khairil, S.Kom, M.Kom	02.130475.01	
<b>Anggota</b>	Deri Lianda, S.Kom, M.Kom	02.250489.04	

**Mengetahui :**

**D e k a n,**



**H. Siswanto, SE., S.Kom, M.Kom**

**NIDN. 02.240363.01**

## RIWAYAT HIDUP



Penulis dilahirkan di Desa Layang Lekat Kecamatan Pagar Jati Kabupaten Bengkulu Tengah, pada tanggal 25 Agustus 1999. Putri pertama dari dua bersaudara. Ayah bernama Z. Abidin dan Ibu bernama Supilawati.

Pendidikan yang pernah ditempuh yaitu Sekolah Dasar (SD) Negeri 6 Kota Bengkulu, Sekolah Menengah Pertama (SMP) Negeri 7 Kota Bengkulu, Sekolah Menengah Atas (SMA) Negeri 6 Kota Bengkulu, Universitas Dehasen Bengkulu Program Strata 1 (S1) Program Studi Informatika Fakultas Ilmu Komputer.

Selama berkuliah di Universitas Dehasen Bengkulu, penulis juga aktif di kegiatan organisasi kampus khususnya di Himpunan Mahasiswa Informatika, dengan jabatan terakhir sebagai Bendahara.

Dengan ketekunan, semangat yang tinggi, dan kerja keras, penulis telah berhasil menyelesaikan pengerjaan skripsi ini. Semoga penulisan skripsi ini mampu memberikan kontribusi positif bagi dunia pendidikan.

Akhir kata penulis mengucapkan rasa syukur yang sebesar-besarnya atas selesainya skripsi yang berjudul **“REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA WEB SERVER”**.

## MOTTO DAN PERSEMBAHAN

### **Motto:**

- Aku mencintai ayahku seperti bintang – bintang. Dia adalah contoh yang bersinar terang dan kebahagiaan yang berkelip-kelip di hatiku.
- Aku menyayangi Ibu karena telah memberikanku segalanya.

Karya ini saya persembahkan untuk :

- Ayah saya, Bapak Z.Abidin
- Ibu saya, Ibu Supilawati
- Saudari saya, Delta Ria Puspita
- Dosen pembimbing
- Serta Almamater tercinta

## PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : ANNE LESTYEA

Npm : 21010095P

Prodi : Informatika

Menyatakan dengan sesungguhnya bahwa :

1. Selama melakukan penelitian dan pembuatan skripsi ini saya tidak melakukan pelanggaran etika akademik dalam bentuk apapun atau pelanggaran lainnya yg bertentang dengan etika akademik
2. Skripsi yang saya buat merupakan karya ilmiah saya sebagai penulis, bukan jiplakan atau karya orang lain
3. Apabila di kemudian hari ditemukan bukti yang meyakinkan bahwa dalam proses pembuatan skripsi ini terdapat pelanggaran etika akademik atau skripsi ini hasil jiplakan atau skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi akademik yang ditetapkan oleh Universitas Dehasen Bengkulu

Demikian pernyataan ini saya buat dengan sebenarnya untuk dipergunakan bilamana perlu

Bengkulu, 19 Agustus 2023

Yang menyatakan,



**ANNE LESTYEA**  
**NPM.21010095P**

## ABSTRAK

### REKAYASA PROTOTIPE SISTEM ALARM NOTIFIKASI BERBASIS SNORT DAN COWRIE UNTUK MENDETEKSI SERANGAN PADA *WEB SERVER*

Oleh :

Anne Lestyca <sup>1)</sup>

Hari Aspriyono, S.Kom, M.Kom <sup>2)</sup>

Abdussalam Al Akbar, S.Kom, M.Kom <sup>3)</sup>

Penggunaan internet di era digitalisasi memberikan peluang terjadinya serangan yang dapat membebani kinerja perangkat komputer. Selain itu, informasi krusial yang dimiliki pengguna juga dapat disalahgunakan secara ilegal apabila berada di tangan pihak yang tidak bertanggung jawab. Oleh karena itu, dibutuhkan pencegahan supaya hal tersebut tidak memberikan dampak yang lebih besar.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan ide untuk merekayasa prototipe sistem alarm notifikasi menggunakan *Intrusion Detection System (IDS)* Snort dan *honeypot* Cowrie. Snort berperan untuk mendeteksi serangan yang menyerang *web server*. Sedangkan Cowrie berperan untuk menjebak dan mengalihkan penyerang ke sistem tiruan supaya seluruh aktivitas penyerang dapat diawasi dan tercatat dalam *log* Cowrie. Selain itu, prototipe ini dapat mengirimkan notifikasi serangan kepada administrator melalui Telegram dan Whatsapp secara *real-time*. Dari serangan yang terjadi, sistem akan melakukan perhitungan dan mengirimkan laporan per periode tertentu kepada administrator melalui *platform* yang sama.

Hasil pengujian yang dilakukan terhadap prototipe ini menunjukkan bahwa sistem dapat mendeteksi serangan dengan baik dan menjebak penyerang ke sistem tiruan. Setelah mendeteksi serangan, sistem juga mampu mengirimkan notifikasi serangan secara *real-time* dalam waktu yang singkat. Kemudian, melakukan perhitungan dan mengirimkan hasil perhitungan sesuai dengan jadwal yang telah ditetapkan.

Kata kunci : Snort, Cowrie, *Web Server*, Notifikasi, Telegram, Whatsapp

Keterangan :

1. Calon Sarjana Komputer
2. Pembimbing



**ABSTRACT**

**PROTOTYPE ENGINEERING OF SNORT AND COWRIE-BASED  
NOTIFICATION ALARM SYSTEM TO DETECT ATTACKS  
ON WEB SERVER**

**By:**

Anne Lestyca<sup>1)</sup>

Hari Aspriyono<sup>2)</sup>

Abdussalam Al Akbar<sup>3)</sup>

The use of the internet in the era of digitalization provides opportunities for attacks that can burden the performance of computer devices. In addition, critical information owned by users can also be illegally misused if it is in the hands of irresponsible parties. Therefore, prevention is needed therefore this does not have a bigger impact. Based on these problems, this research proposes an idea to engineer a prototype notification alarm system using the Snort Intrusion Detection System (IDS) and Cowrie honeypot. Snort's role is to detect attacks that attack web servers. Meanwhile, Cowrie's role is to trap and divert attackers to an artificial system therefore all attacker activities can be monitored and recorded in the Cowrie log. In addition, this prototype can send attack notifications to administrators via Telegram and Whatsapp in real-time. From attacks that occur, the system will carry out calculations and send reports per certain period to the administrator via the same platform. The results of testing carried out on this prototype show that the system can detect attacks well and trap attackers into a dummy system. After detecting an attack, the system is also able to send real-time attack notifications in a short time. Then, carry out calculations and send the calculation results according to a predetermined schedule.

**Keywords:** *Snort, Cowrie, Web Server, Notifications, Telegram, Whatsapp.*

**Information :**

1. Student
2. Supervisors



## KATA PENGANTAR

Alhamdulillah penulis panjatkan kehadiran Allah SWT yang melimpahkan rahmat dan karunia-Nya sehingga Skripsi yang berjudul “**Rekayasa Prototipe Sistem Alarm Notifikasi Berbasis Snort dan Cowrie untuk Mendeteksi Serangan pada Web Server**” ini dapat diselesaikan dalam waktu yang telah ditetapkan.

Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan Skripsi ini terutama kepada :

1. Bapak Prof. Dr. Husaini, SE., M.Si., Ak., CA, CRP selaku Rektor Universitas Dehasen Bengkulu.
2. Bapak Siswanto, SE., S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
3. Ibu Liza Yulianti, S.Kom, M.Kom, selaku Ketua Prodi Informatika Fakultas Ilmu Komputer yang telah banyak membantu dalam urusan administrasi akademik selama berkuliah.
4. Bapak Hari Aspriyono, S.Kom, M.Kom, selaku Dosen Pembimbing I yang telah banyak memberikan arahan dan bimbingan kepada penulis dalam pengerjaan proyek penelitian.
5. Bapak Abdussalam Al Akbar, S.Kom, M.Kom, selaku Dosen Pembimbing II telah banyak memberikan arahan dan bimbingan kepada penulis.

6. Bapak Khairil, S.Kom, M.Kom, selaku Kepala UPT. Pusat Komputer Universitas Dehasen Bengkulu yang telah berkenan menjadi narasumber informasi dan memberikan saran terhadap penelitian yang dilakukan.
7. Bapak Z. Abidin (Ayah), sebagai support system yang selama ini menjadi alasan saya untuk bertahan dan selalu semangat menghadapi apapun.
8. Ibu Supilawati (Ibu), seseorang yang menjadikan saya sebagai sosok yang terus memperbaiki diri hingga saat ini dan senantiasa belajar sesuatu yang baru demi masa depan saya.

Semoga segala bantuan dan bimbingan yang telah diberikan kepada penulis mendapat imbalan yang berlimpah dari Tuhan Yang Maha Esa.

Penulis mengharapkan kritik dan saran yang sifatnya membangun dari berbagai pihak. Akhirnya semoga Skripsi ini dapat bermanfaat bagi penulis khususnya, dan bagi pembaca umumnya.

Bengkulu, Agustus 2023

Penulis

## DAFTAR ISI

Daftar	Halaman
<u>HALAMAN JUDUL</u> .....	i
<u>HALAMAN SAMPUL DALAM</u> .....	i
<u>HALAMAN PERSETUJUAN</u> .....	ii
<u>LEMBAR PENGESAHAN</u> .....	iii
<u>RIWAYAT HIDUP</u> .....	iv
<u>MOTTO DAN PERSEMBAHAN</u> .....	v
<u>PERNYATAAN KEASLIAN SKRIPSI</u> .....	vi
<u>ABSTRAK</u> .....	vii
<u>ABSTRACT</u> .....	viii
<u>KATA PENGANTAR</u> .....	ix
<u>DAFTAR ISI</u> .....	xi
<u>DAFTAR GAMBAR</u> .....	xiii
<u>DAFTAR TABEL</u> .....	xvi
<u>DAFTAR LAMPIRAN</u> .....	xvii
<b><u>BAB I PENDAHULUAN</u></b> .....	<b>1</b>
<u>1.1. Latar Belakang</u> .....	1
<u>1.2. Rumusan Masalah</u> .....	3
<u>1.3. Batasan Masalah</u> .....	3
<u>1.4. Tujuan Penelitian</u> .....	4
<u>1.5. Manfaat Penelitian</u> .....	5
<b><u>BAB II LANDASAN TEORI</u></b> .....	<b>6</b>
<u>2.1. Server</u> .....	6
<u>2.2. Secure Web Server</u> .....	8
<u>2.3. Intrusion Detection System (IDS) Berbasis Snort</u> .....	9
<u>2.4. Honeypot Cowrie</u> .....	11
<u>2.5. Sistem Notifikasi dan Deteksi</u> .....	13
<u>2.6. Serangan Distributed Denial of Service (DDoS)</u> .....	17
<u>2.7. Serangan Port Scanning</u> .....	21
<u>2.8. Serangan Brute-Force</u> .....	22

2.9. <a href="#">Raspberry Pi 3 Model B+</a> .....	22
<b><a href="#">BAB III METODOLOGI PENELITIAN</a></b> .....	<b>24</b>
3.1. <a href="#">Gambaran Umum Subyek Penelitian</a> .....	24
3.1.1. <a href="#">Tempat dan Waktu Penelitian</a> .....	25
3.1.2. <a href="#">Struktur Organisasi</a> .....	25
3.1.3. <a href="#">Tugas dan Wewenang</a> .....	26
3.2. <a href="#">Metode Penelitian</a> .....	26
3.3. <a href="#">Analisis Kebutuhan</a> .....	32
3.4. <a href="#">Metode Pengumpulan Data</a> .....	35
3.5. <a href="#">Metode Perancangan Sistem</a> .....	36
<b><a href="#">BAB IV HASIL DAN PEMBAHASAN</a></b> .....	<b>Error! Bookmark not defined.</b>
4.1. <a href="#">Hasil</a> .....	<b>Error! Bookmark not defined.</b>
4.2. <a href="#">Pembahasan</a> .....	<b>Error! Bookmark not defined.</b>
4.3. <a href="#">Pengujian</a> .....	<b>Error! Bookmark not defined.</b>
<b><a href="#">BAB V KESIMPULAN DAN SARAN</a></b> .....	<b>Error! Bookmark not defined.</b>
5.1. <a href="#">Kesimpulan</a> .....	<b>Error! Bookmark not defined.</b>
5.2. <a href="#">Saran</a> .....	<b>Error! Bookmark not defined.</b>
<b><a href="#">DAFTAR PUSTAKA</a></b>	
<b><a href="#">LAMPIRAN</a></b>	

## DAFTAR GAMBAR

Gambar	Halaman
<a href="#">2.1 Arsitektur Snort (Ombase et al., 2018)</a> .....	10
<a href="#">2.2 Alur Sistem pada Penelitian Atmojo (Atmojo, 2018)</a> .....	13
<a href="#">2.3 Prosedur penelitian Rupiati et al.</a> .....	14
<a href="#">2.4 Skenario Pengujian (Rupiati et al., 2020)</a> .....	15
<a href="#">2.5 Flowchart sistem pada penelitian Hakim et al. (Hakim et al., 2020)</a> .....	16
<a href="#">2.6 Ilustrasi Serangan DDoS (Geges &amp; Wibisono, 2015)</a> .....	18
<a href="#">2.7 Ilustrasi serangan Ping of Death (Hakim et al., 2020)</a> .....	20
<a href="#">2.8 Ilustrasi serangan SYN Flood (Hakim et al., 2020)</a> .....	21
<a href="#">2.9 Skema serangan Port Scanning (A. Utomo, 2018)</a> .....	21
<a href="#">2.10 Skema serangan Brute-Force (A. Utomo, 2018)</a> .....	22
<a href="#">2.11 Raspberry Pi 3 Model B+ (RaspberryPi, 2016)</a> .....	23
<a href="#">3.1 Gambaran Umum Sistem</a> .....	24
<a href="#">3.2 Waterfall Development (Roth, Dennis, 2012, 2015)</a> .....	26
<a href="#">3.3 Kerangka Penelitian</a> .....	31
<a href="#">3.4 Topologi Mesh (Wagiu et al., 2016)</a> .....	36
<a href="#">3.5 Topologi Jaringan</a> .....	38
<a href="#">3.6 Format Notifikasi Real-Time Telegram</a> .....	40
<a href="#">3.7 Format Notifikasi Per Periode Telegram</a> .....	40
<a href="#">3.8 Format Notifikasi Real-Time Whatsapp</a> .....	41
<a href="#">3.9 Format Notifikasi Per Periode Whatsapp</a> .....	41
<a href="#">3.10 Rancangan Pengujian</a> .....	43
<a href="#">4.1 Versi Snort yang berhasil diimplementasikan</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.2 Cowrie berhasil diimplementasikan</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.3 Tampilan pesan notifikasi Telegram</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.4 Tampilan pesan notifikasi Whatsapp</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.5 Notifikasi jumlah serangan Telegram</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.6 Notifikasi jumlah serangan Whatsapp</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.7 Aplikasi Pendukung dan mkdir Snort</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.8 Install dan autoreconf DAQ Snort</a> .....	<b>Error! Bookmark not defined.</b>

<a href="#">4.9 <i>Install dan configure Snort</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.10 Pembuatan dan pengaturan penyimpanan <i>file rules Snort</i></a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.11 Menyalin <i>file konfigurasi</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.12 Proses pengunduhan <i>rules Snort</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.13 Pembuatan <i>file</i> untuk menyimpan log <i>Snort</i></a> ..	<b>Error! Bookmark not defined.</b>
<a href="#">4.14 Konfigurasi <i>file snort.conf</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.15 Hasil Validasi Konfigurasi <i>snort.conf</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.16 <i>ID @BotFather</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.17 Perintah untuk membuat Telegram <i>Bot</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.18 Membuat nama Telegram <i>Bot</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.19 <i>Username Telegram Bot</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.20 Token Telegram Bot yang berhasil dibuat</a> ....	<b>Error! Bookmark not defined.</b>
<a href="#">4.21 <i>ID Chat User dan ID Chat Group Telegram Bot</i></a>	<b>Error! Bookmark not defined.</b>
	<b>defined.</b>
<a href="#">4.22 Pendaftaran Akun Twilio</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.23 Halaman <i>login</i> pada Twilio</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.24 Pengujian Kirim Pesan Whatsapp dari Twilio</a>	<b>Error! Bookmark not defined.</b>
	<b>defined.</b>
<a href="#">4.25 Pengiriman Pesan Kode Whatsapp</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.26 Pesan Berhasil Diterima</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.27 Membuat “<i>Appointment Reminders</i>”</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.28 Membuat <i>request</i> untuk kirim pesan</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.29 Administrator menerima pesan</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.30 <i>Script CURL</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.31 File <i>script.txt</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.32 Mengunduh <i>script bot-tele.sh</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.33 Konfigurasi <i>file bot-tele.sh</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.34 Konfigurasi <i>file bot-tele.sh (lanjutan)</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.35 Instalasi aplikasi pendukung Cowrie</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.36 Mengunduh direktori Cowrie</a> .....	<b>Error! Bookmark not defined.</b>

[4.37 Mengaktifkan \*virtual environment\* untuk Cowrie](#)**Error! Bookmark not defined.**

[4.38 Upgrade pip dan skrip requirements.txt](#).....**Error! Bookmark not defined.**

[4.39 Pengubahan \*destination address\* Cowrie](#).....**Error! Bookmark not defined.**

[4.40 Instalasi authbind](#) .....**Error! Bookmark not defined.**

[4.41 Menjalankan Cowrie](#) .....**Error! Bookmark not defined.**

[4.42 Pengaturan perintah Crontab](#).....**Error! Bookmark not defined.**

[4.43 Skrip notifikasi penghitung serangan audit.sh](#)**Error! Bookmark not defined.**

[4.44 Penyerang melakukan serangan \*SYN Flood\*](#)...**Error! Bookmark not defined.**

[4.45 Log serangan \*SYN Flood\*](#).....**Error! Bookmark not defined.**

[4.46 Notifikasi serangan \*SYN bot\* Telegram](#) .....**Error! Bookmark not defined.**

[4.47 Notifikasi serangan \*SYN grup\* Telegram](#).....**Error! Bookmark not defined.**

[4.48 Notifikasi serangan \*SYN\* Whatsapp](#).....**Error! Bookmark not defined.**

[4.49 Penyerang melakukan serangan \*Port Scanning\*](#)**Error! Bookmark not defined.**

[4.50 Log serangan \*Port Scanning\*](#) .....**Error! Bookmark not defined.**

[4.51 Notifikasi serangan \*Port Scanning bot\* Telegram](#)**Error! Bookmark not defined.**

[4.52 Notifikasi serangan \*Port Scanning grup\* Telegram](#)**Error! Bookmark not defined.**

[4.53 Notifikasi serangan \*Port Scanning\* Whatsapp](#)**Error! Bookmark not defined.**

[4.54 Tampilan \*log\* Cowrie saat memonitoring serangan \*Port Scanning\*](#) ..... **Error! Bookmark not defined.**

[4.55 Penyerang melakukan serangan \*SSH Brute Force\*](#)**Error! Bookmark not defined.**

[4.56 Log serangan \*SSH Brute Force\*](#).....**Error! Bookmark not defined.**

[4.57 Notifikasi serangan \*SSH Brute Force bot\* Telegram](#)**Error! Bookmark not defined.**

[4.58 Notifikasi serangan \*SSH Brute Force grup\* Telegram](#)**Error! Bookmark not defined.**



<a href="#">4.59 Notifikasi serangan SSH Brute Force Whatsapp</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.60 Akses Putty web server</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.61 Kondisi komputer penyerang ketika berhasil masuk Putty</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.62 Tampilan log Cowrie saat penyerang akses Putty</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.63 Command unknown ifconfig</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.64 Command unknown sudo -i, ls, apt-get --u</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.65 Log Cowrie saat penyerang memasukkan command unknown</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.66 Kapasitas memori terpakai saat sistem belum dijalankan</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.67 Kapasitas memori terpakai saat sistem dijalankan</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.68 Notifikasi jumlah serangan bot Telegram</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.69 Notifikasi jumlah serangan grup Telegram</a>	<b>Error! Bookmark not defined.</b>
<a href="#">4.70 Notifikasi jumlah serangan Whatsapp</a>	<b>Error! Bookmark not defined.</b>

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
<a href="#">3.1 Hasil Identifikasi Kebutuhan Sistem</a> .....	32
<a href="#">3.2 Daftar Kebutuhan Perangkat Keras</a> .....	34
<a href="#">3.3 Daftar Kebutuhan Perangkat Lunak</a> .....	34
<a href="#">3.4 Konfigurasi Server Utama</a> .....	42
<a href="#">3.5 Konfigurasi Prototipe Sistem Alarm Notifikasi</a> .....	42
<a href="#">3.6 Konfigurasi PC Penyerang</a> .....	43
<a href="#">3.7 Tabel Rencana Pengujian</a> .....	45
<a href="#">4.1 Informasi Telegram Bot</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.2 Informasi Whatsapp API</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.3 Tabel Keterangan serangan SYN</a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.4 Tabel Keterangan serangan <i>Port Scanning</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.5 Tabel Keterangan serangan <i>SSH Brute Force</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.6 Tabel waktu pengiriman notifikasi serangan <i>SYN Flood</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.7 Tabel waktu pengiriman notifikasi serangan <i>Port Scanning</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.8 Tabel waktu pengiriman notifikasi serangan <i>SSH Brute Force</i></a> .....	<b>Error! Bookmark not defined.</b>
<a href="#">4.9 Tabel Hasil Pengujian</a> .....	<b>Error! Bookmark not defined.</b>

## DAFTAR LAMPIRAN

### Lampiran

1. Rencana Kegiatan
2. Surat Penetapan Pembimbing
3. Surat Izin Penelitian
4. Kartu Bimbingan
5. Surat Persetujuan Izin Penelitian
6. Surat Keterangan Menyelesaikan Penelitian
7. *Source Code*
8. Struktur Organisasi
9. Form Wawancara

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pada era digitalisasi saat ini, banyak aspek kehidupan masyarakat yang telah beralih dari model konvensional menjadi serba elektronik. Hal tersebut tentunya menyebabkan masyarakat untuk mengakses kebutuhan yang diinginkan melalui perangkat dan media digital. Dalam melakukan akses digital, dibutuhkan koneksi internet ke perangkat yang digunakan supaya mendapatkan layanan dari server yang dituju (Fadhlorrohman et al., 2021; Geges & Wibisono, 2015).

Server yang terhubung dengan jaringan internet memiliki kerentanan terhadap serangan yang dapat terjadi pada rentang waktu tertentu. Beberapa serangan tersebut adalah *Denial of Service/Distributed Denial of Service* (DoS/DDoS), *Port Scanning*, dan *Brute Force Attack* (Geges & Wibisono, 2015; Mardiyanto et al., 2016; Rupiati et al., 2020). Serangan-serangan tersebut dapat menyebabkan berbagai macam masalah mulai dari penurunan kinerja pada server dengan membebani sistem server, mengetahui celah keamanan sistem, hingga mendapatkan informasi krusial yang kemudian dapat disalahgunakan (Mardiyanto et al., 2016; Rupiati et al., 2020).

Berdasarkan uraian tersebut, solusi yang sesuai untuk diterapkan menurut Aryachandra et al. adalah menggunakan pendeteksi serangan. Metode deteksi serangan yang paling umum digunakan adalah *Intrusion Detection System* (IDS). Berbagai studi menunjukkan bahwa IDS adalah alat fundamental yang telah diusulkan selama bertahun-tahun sebagai pendeteksi keamanan yang efisien dan mekanisme pertahanan yang kuat. IDS dapat mendeteksi pola intrusi dengan

memeriksa paket jaringan secara kritis, menerapkan *signature* dan menghasilkan *alert* untuk administrator sistem (Aryachandra *et al.*, 2016).

Jenis IDS yang paling umum digunakan dalam melakukan pendeteksian serangan adalah Snort. Snort dipilih karena dapat melakukan deteksi serangan yang terjadi pada target berdasarkan *rules* yang ada (Aryachandra *et al.*, 2016; Hakim *et al.*, 2020; Hassan *et al.*, 2018; Ombase *et al.*, 2018). Kemudian, untuk meningkatkan keamanan pada server, akan ditambahkan *honeypot* Cowrie yang berperan sebagai wadah yang menjebak penyerang, melakukan identifikasi terhadap perilaku penyerang yang ingin mengakses server, dan mencatat segala aktivitas yang dilakukan penyerang (A. Utomo, 2018; Cahyanto *et al.*, 2016; Rupiati *et al.*, 2020; Sulaksono & Suharyanto, 2020). Cowrie dipilih karena dapat diimplementasikan pada Raspberry Pi dan mampu bekerja dengan efektif dan efisien dalam menemukan permasalahan jaringan nirkabel (A. Utomo, 2018; Sulaksono & Suharyanto, 2020).

Kemudian, untuk membantu sistem dalam mengirimkan pemberitahuan kepada administrator jaringan, penelitian ini juga akan menggunakan Telegram *Bot* dan Whatsapp API untuk mengirimkan pesan dari prototipe ke Telegram dan Whatsapp. Telegram dipilih karena memiliki keunggulan dari sudut pandang keamanan komunikasi. Whatsapp dipilih karena merupakan media sosial yang paling populer di kalangan pengguna *smartphone* dunia. Tujuan dari penambahan fitur ini adalah supaya administrator mendapatkan notifikasi terkait dengan adanya indikasi serangan, karena cara kerja IDS cenderung hanya melakukan pendeteksian saja (Hakim *et al.*, 2020).

Berdasarkan uraian di atas, penulis tertarik untuk mengusulkan judul penelitian yaitu “**Rekayasa Prototipe Sistem Alarm Notifikasi Berbasis Snort dan Cowrie untuk Mendeteksi Serangan pada Web Server**”.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini, yaitu:

- a. Apakah prototipe sistem alarm notifikasi ini dapat meneruskan paket *alert* yang tersimpan dalam *file* snort.log dan memberikan notifikasi kepada administrator jaringan ketika terjadi indikasi serangan pada *web server*?
- b. Apakah Cowrie pada prototipe sistem alarm notifikasi ini mampu mendeteksi dan memberi respon terhadap serangan serta memberi informasi palsu kepada penyerang?
- c. Bagaimana efisiensi dari prototipe sistem alarm notifikasi ini saat mengidentifikasi serangan dan mengirimkan notifikasi kepada administrator jaringan?

## 1.3. Batasan Masalah

Berikut ini dijelaskan batasan masalah yang terdapat pada penelitian ini:

- a. Prototipe ini berfungsi untuk meneruskan *alert* kepada administrator ketika terdapat indikasi serangan pada *web server* dan melakukan mitigasi sementara dengan mengalihkan serangan ke server tiruan.

- b. Metode pengujian dilakukan dengan simulasi serangan dan *performance testing*. Jenis serangan yang digunakan adalah *SYN Flood attack*, *Port Scanning*, dan *SSH Brute Force Attack*.
- c. Notifikasi yang dikirimkan kepada administrator ketika terdapat indikasi serangan pada *web server* berupa pesan notifikasi ke aplikasi Telegram dan Whatsapp.
- d. Notifikasi ini berisi informasi tentang jenis serangan, waktu serangan, sumber IP penyerang, sumber *port* penyerang, dan target serangan.
- e. Perangkat yang digunakan untuk membuat prototipe alarm ini adalah Raspberry Pi 3 Model B+.
- f. Parameter yang akan diuji dalam *performance testing* meliputi waktu eksekusi, penggunaan memori pada Raspberry Pi, dan keberhasilan sistem dalam mengirimkan notifikasi kepada administrator.

#### **1.4. Tujuan Penelitian**

Berikut ini dijelaskan tujuan umum dan khusus dari penelitian ini.

##### **1.4.1. Tujuan Umum**

Adapun tujuan umum dari penelitian ini adalah untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer pada Program Studi Informatika di Universitas Dehasen Bengkulu.

##### **1.4.2. Tujuan Khusus**

Adapun tujuan khusus dari penelitian ini adalah membuat prototipe sistem alarm notifikasi untuk mendeteksi serangan dan menjebak

penyerang ke dalam server tiruan kemudian mengirimkan notifikasi kepada administrator ketika terdapat indikasi serangan pada *web server*.

### **1.5. Manfaat Penelitian**

Manfaat dari penelitian ini adalah untuk menyediakan sarana yang dapat direkomendasikan sebagai alat pendukung dalam melakukan penanggulangan ketika terjadi indikasi serangan pada *web server*.



## BAB II

### LANDASAN TEORI

#### 2.1. Server

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer (Prakoso & Asmunin, 2018; Suryana, 2018). Sebuah server didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus (Prakoso & Asmunin, 2018). Macam-macam server atau jenis – jenis server dapat digolongkan dalam beberapa golongan jika dilihat dari fungsinya (Prakoso & Asmunin, 2018; Suryana, 2018), misalnya :

1. Server aplikasi (*application server*)

Server aplikasi adalah server yang digunakan untuk menyimpan berbagai macam aplikasi yang dapat diakses oleh *client* (Prakoso & Asmunin, 2018).

2. Server data (*data server*)

Server data digunakan untuk menyimpan data baik yang digunakan *client* secara langsung maupun data yang diproses oleh server aplikasi (Prakoso & Asmunin, 2018).

3. Server Proxy (*Proxy server*)

Server proxy adalah server yang berfungsi untuk mengatur lalu lintas di jaringan melalui pengaturan proxy. Orang awam lebih mengenal *proxy server* untuk mengkoneksikan komputer *client* ke internet (Prakoso & Asmunin, 2018).

#### 4. *Web Server*

*Web server* merupakan server yang melayani permintaan HTTP atau HTTPS dari *browser* dan mengirimkan kembali permintaan tersebut dalam bentuk halaman *web*. *Web server* memiliki fungsi utama untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan (Prakoso & Asmunin, 2018).

#### 5. *Database Server*

*Database server* merupakan server yang memiliki fitur pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang memiliki model *client/server*. Adapun fungsi dari *database server* adalah sebagai analisis data, penyimpanan data, pengarsipan data, dan lain-lain (Prakoso & Asmunin, 2018).

#### 6. *FTP Server*

*File Transfer Protocol* (FTP) merupakan protokol server atau *client* yang berfungsi untuk mentransfer data atau tukar-menukar data atau *file* antar komputer yang ada dalam satu *network* atau jaringan. Fungsi dari server ini adalah memberikan layanan yang menangani perpindahan *file* (*transfer file*) (Sungkar *et al.*, 2020).

#### 7. *Mail Server*

Sesuai dengan namanya, *server e-mail* adalah pusat kendali sistem *e-mail*. Sebuah *mail server* biasanya terdiri dari area penyimpanan, set konfigurasi *user*, daftar *user*, dan seri modul komunikasi lainnya (Basorudin, 2018).

## 8. *DNS Server*

*Domain Name System* (DNS) adalah sebuah sistem yang menyimpan informasi tentang nama *host* dan nama *domain* dalam bentuk basis data terdistribusi (*distributed database*) di dalam jaringan komputer, misalkan *Internet*. DNS memberikan fasilitas berupa alamat IP untuk setiap nama *host* (*Client*) dan mendata setiap server transmisi surat (*mail exchange server*) yang menerima surat elektronik (*email*) untuk setiap *domain*-nya (Basorudin, 2018).

## 9. *DHCP Server*

*Dynamic Host Configuration Protocol* (DHCP) adalah sebuah protokol yang digunakan untuk dapat memberikan IP *address* secara otomatis ke komputer yang terhubung aktif ke jaringan TCP/IP. Untuk dapat mengimplementasikan DHCP ini dibutuhkan sebuah *DHCP server*, yaitu komputer yang digunakan untuk melayani permintaan akan IP *address* tersebut. Server DHCP juga berperan dalam memudahkan pekerjaan administrator jaringan supaya tidak membagi IP secara manual (Adipratama & Gunawan, 2005).

## 2.2. *Secure Web Server*

*Secure web server* adalah kombinasi yang awalnya merupakan *web server* saat berkomunikasi dengan klien menggunakan protokol *Secure Socket Layer* (SSL). Artinya struktur dasar server sama dengan *web server* biasa selain itu pemrograman untuk SSL juga terlibat oleh sebelumnya. Server berkomunikasi dengan klien melalui protokol SSL (Prayogo *et al.*, 2019).

Protokol HTTP adalah pesan teks biasa, jadi jika paket HTTP dari IP tertentu dipantau oleh beberapa alat penangkap paket, data pengguna dapat dengan mudah dicegat dan menyebabkan kebocoran informasi. Ini adalah praktik umum untuk menggunakan *Secure Socket Layer* (SSL) untuk meningkatkan keamanan web. Adapun *Internet Information Server* (IIS) dan Apache, praktik umum adalah menerbitkan sertifikat *web server* untuk IIS dengan menggunakan "layanan sertifikat" di bawah sistem operasi Windows 2000/2003 dan IIS untuk situs web yang aman; atau di bawah sistem operasi Linux, gunakan *OpenSSL* untuk menerbitkan sertifikat *web server* untuk Apache dan mencapai situs web yang aman (Andriani *et al.*, 2018).

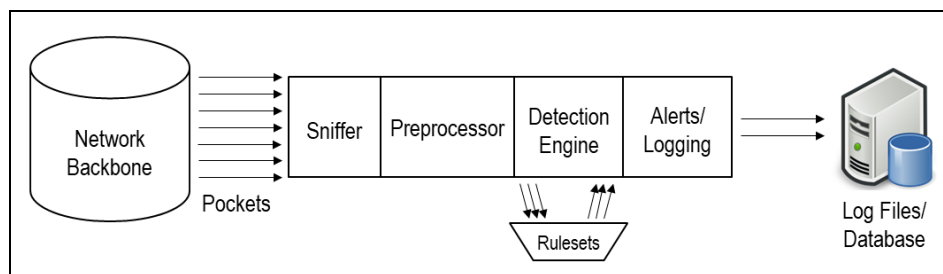
Untuk mengatasi masalah komunikasi web yang aman, Netscape mengedepankan Protokol *Secure Socket Layer* (SSL) pada tahun 1994, yang dapat memastikan transmisi informasi yang aman dan rahasia melalui internet. Protokol HTTP yang menggunakan teknologi SSL disebut HTTPS, dan digunakan secara luas di web. Kemudian protokol distandarisasi oleh IETF dan diberi nama lain *Transport Layer Security* (TLS) (Andriani *et al.*, 2018).

### **2.3. *Intrusion Detection System* (IDS) Berbasis Snort**

Pada penelitian yang dilakukan oleh Aryachandra *et al.*, dijelaskan bahwa metode deteksi serangan yang paling umum digunakan adalah *Intrusion Detection System* (IDS) (Aryachandra *et al.*, 2016). IDS menganalisis lalu lintas dan menampilkan *header* paket protokol internet dan menggunakan *counter* untuk mempertahankan alamatnya, kemudian membandingkannya dengan *threshold* yang telah ditentukan sebelumnya ketika *counter* melebihi *threshold* arus lalu

lintas adalah serangan. Jenis IDS yang paling banyak digunakan adalah Snort (Aryachandra *et al.*, 2016).

Snort adalah IDS jaringan sumber terbuka dan menggunakan pola aktivitas yang telah ditentukan sebelumnya terkait dengan serangan yang diketahui untuk mengidentifikasi dan memblokir lalu lintas yang terinfeksi; pola ini dikenal sebagai *signature* atau *rules*. Snort berisi seperangkat *rules* di mana *rules* mengontrol lalu lintas, memeriksa paket dan mendeteksi serangan. Ketika Snort mendeteksi serangan, akan menghasilkan *alert* untuk memberikan informasi serangan kepada administrator jaringan. *Rules* pada Snort dapat dimodifikasi sesuai kebutuhan pengguna (Hassan *et al.*, 2018). Selain itu, *alert* yang dimaksud adalah catatan serangan pada deteksi serangan berupa penyimpanan log yang tersimpan dalam *database*.



Gambar 2.1 Arsitektur Snort (Ombase *et al.*, 2018)

Gambar 2.1 menjelaskan tentang arsitektur dari Snort yang memiliki 4 komponen, yaitu *sniffer*, *pre-processor*, *detection engine*, dan *alerts/logging*. Pertama, *sniffer* menerima paket dan menganalisis berbagai protokol jaringan. Snort menyimpan paket tersebut kemudian diproses nanti. Mekanisme ini menggunakan *pcap library (libpcap)* untuk mendapatkan paket dari perangkat jaringan. *libpcap* menyediakan informasi, seperti panjang paket, waktu

pengambilan paket, jenis *link*, dan sebagainya. Kemudian, *packet decoder* mendekode paket yang diterima sesuai dengan jenis *layer link* dan mengatur *pointer* di berbagai bagian paket, hal ini memungkinkan akses cepat dari berbagai bagian jaringan. Setelah proses dekode paket selesai, paket dikirim ke *pre-processor*. Ada sejumlah *pre-processor* yang tersedia di Snort. Setelah *preprocessing*, paket tersebut diteruskan ke *search engine*. Tahap ini mencocokkan paket dengan *database signature* yang disertakan dalam Snort dengan *walking signature tree* sampai *search engine* berhasil melakukan pencocokan dengan *rules* yang telah ditentukan. Jika *rules* cocok, Snort menghasilkan *alert* dan mengirimkannya ke *database* (Ombase *et al.*, 2018).

#### **2.4. Honeypot Cowrie**

Berdasarkan penelitian yang dilakukan oleh Cahyanto *et al.*, *honeypot* merupakan suatu sistem palsu yang dibuat untuk menjebak dan mengalihkan perhatian penyerang yang berusaha menangkal usaha-usaha yang dapat merugikan sistem atau layanan. Sistem ini membuat penyerang seolah-olah berhasil mendapatkan sekumpulan data dari suatu sumber, padahal data tersebut tidak penting dan lokasi tersebut sudah terisolir (Cahyanto *et al.*, 2016). Menurut Cahyanto *et al.*, *honeypot* juga dapat menganalisis perilaku dan aktivitas penyerang saat melakukan akses ke sistem server. Berikut ini merupakan tipe *honeypot*:

- a. *Low Interaction Honeypot*, tipe ini dibuat untuk mensimulasikan layanan yang mirip dengan sistem pada server asli. Contohnya, layanan *File Transfer Protocol* (FTP), *Telnet*, *Hypertext Transfer Protocol* (HTTP), dan lainnya.

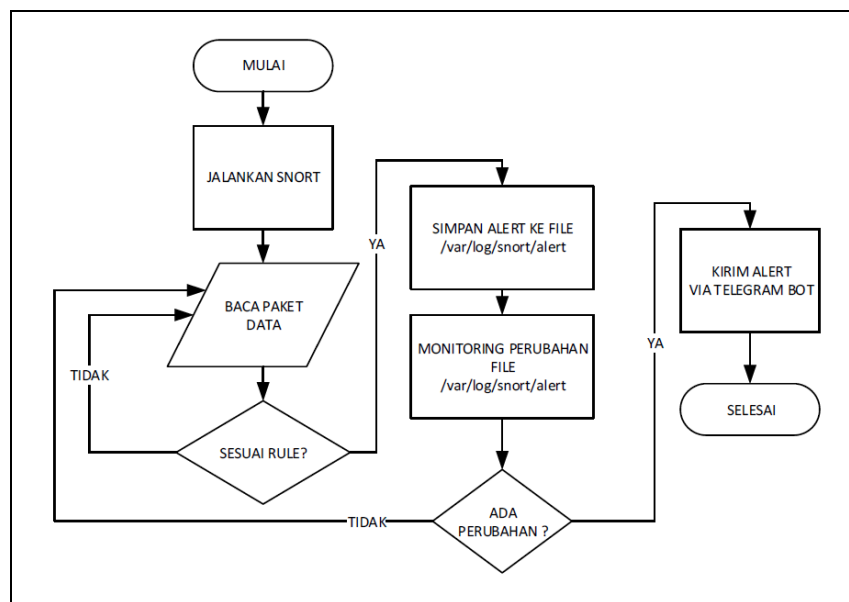
- b. *Medium Interaction Honeypot*, tipe ini merupakan layanan yang dibuat sebagai sistem operasi palsu yang dimaksud untuk menjebak penyerang. Layanan ini akan melewatkan beberapa perintah dari *honeypot* namun seluruh informasi dari penyerang akan tercatat dan dapat dievaluasi oleh administrator jaringan. Fungsi ini biasanya terdapat pada *honeypot* Cowrie (Sulaksono & Suharyanto, 2020).
- c. *High Interaction Honeypot*, tipe ini merupakan sistem tiruan yang dibangun dengan spesifikasi layanan dan sistem operasi yang sama persis seperti sistem pada server aslinya.

Tercatat pada tahun 2020, Sulaksono & Suharyanto telah melakukan penelitian tentang Cowrie. Dalam penelitian tersebut disebutkan bahwa Cowrie merupakan suatu perangkat lunak yang berfungsi untuk memudahkan dalam melakukan inisialisasi pada *honeypot* dan dapat digunakan untuk menyamarkan layanan pada server *openssh*. Cowrie termasuk salah satu contoh *honeypot* tipe interaksi sedang yang dapat mendeteksi dan meriwayatkan serangan *SSH brute force*, *telnet* dan *openssh server* (Sulaksono & Suharyanto, 2020). Cara kerja Cowrie menggunakan konsep pengalihan, yang mana ketika *openssh* berhasil diserang maka Cowrie akan mengarahkan penyerang untuk masuk ke sistem tiruan atau layanan palsu milik *honeypot*. Sehingga penyerang akan mengira bahwa penyerangan tersebut telah berhasil, padahal penyerang hanya masuk dalam perangkat *honeypot*. Kemudian, Cowrie juga memiliki *log* yang dapat melakukan proses *logging* untuk mencatat semua kegiatan yang dilakukan oleh penyerang selama berada di dalam sistem tiruan (*honeypot*). Sehingga

adminstrator jaringan dapat mengetahui kegiatan apa saja yang dilakukan penyerang pada sistem palsu tersebut (Sulaksono & Suharyanto, 2020).

## 2.5. Sistem Notifikasi dan Deteksi

Pada penelitian yang dilakukan oleh Atmojo pada tahun 2018, sistem alarm notifikasi yang dibangun menggunakan perangkat Raspberry Pi 3 (Atmojo, 2018). Penelitian ini menerapkan IDS jenis Snort untuk melakukan pendeteksian serangan. Saat aktivitas serangan yang terjadi pada jaringan berhasil dideteksi oleh sistem, maka akan dilakukan pencocokan terhadap *rules* yang sudah diatur. Jika hasil deteksi cocok dengan *rules* yang diatur, maka sistem akan mengirimkan notifikasi kepada administrator menggunakan *Bot Alert Snort* pada aplikasi Telegram (Atmojo, 2018). Gambar 2.2 menunjukkan diagram alir sistem yang diterapkan pada penelitian Atmojo.



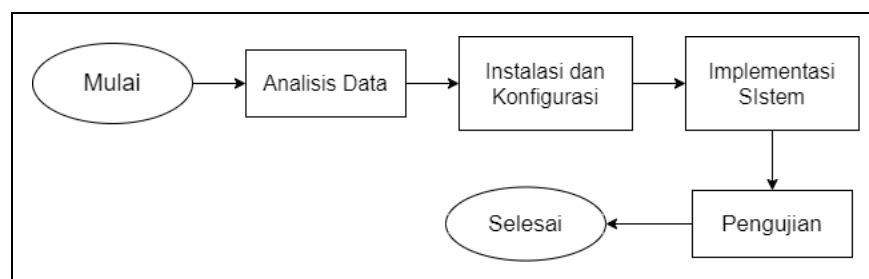
Gambar 2.2 Alur Sistem pada Penelitian Atmojo (Atmojo, 2018)

Berdasarkan Gambar 2.2, sistem bekerja diawali dengan dijalankannya Snort untuk mendeteksi serangan pada jaringan komputer (Atmojo, 2018).

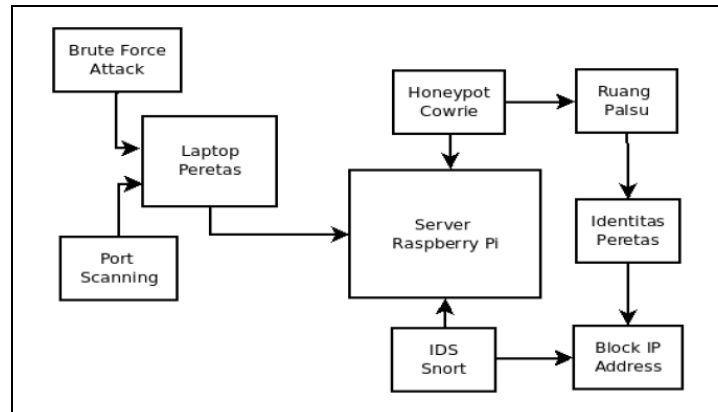


Kemudian, sistem akan membaca paket data yang telah terdeteksi dan membandingkan dengan *rules* serangan yang telah diatur pada sistem. Setelah melakukan pencocokan *rules*, sistem akan melakukan pencatatan log aktivitas serangan dan menyimpannya ke dalam direktori `/var/log/snort/alert` yang berisi tentang informasi mengenai jenis *rules* yang sesuai. Aplikasi *Swatch* berjalan untuk melakukan monitoring setiap perubahan yang terjadi pada direktori log Snort, dan mengirimkan notifikasi ke aplikasi Telegram melalui *bot* API setiap menitnya (Atmojo, 2018).

Penelitian selanjutnya adalah penelitian yang dilakukan pada tahun 2020 oleh Rupiati *et al.* Penelitian ini juga menerapkan IDS berbasis Snort dalam melakukan pendeteksian serangan pada sistem yang dibangun. Dalam melakukan peningkatan keamanan pada sistem ini, peneliti mengimplementasikan *honeypot* Cowrie. Cowrie berperan dalam menjebak penyerang dengan membangun server tiruan yang memberikan layanan seperti server aslinya. Hal ini bertujuan untuk mengidentifikasi motif penyerangan dan mencatat seluruh riwayat aktivitas serangan yang dilakukan (Rupiati *et al.*, 2020). Gambar 2.3 dan Gambar 2.4 menunjukkan prosedur penelitian dan pengujian yang dilakukan oleh Rupiati *et al.*



Gambar 2.3 Prosedur penelitian Rupiati *et al.*



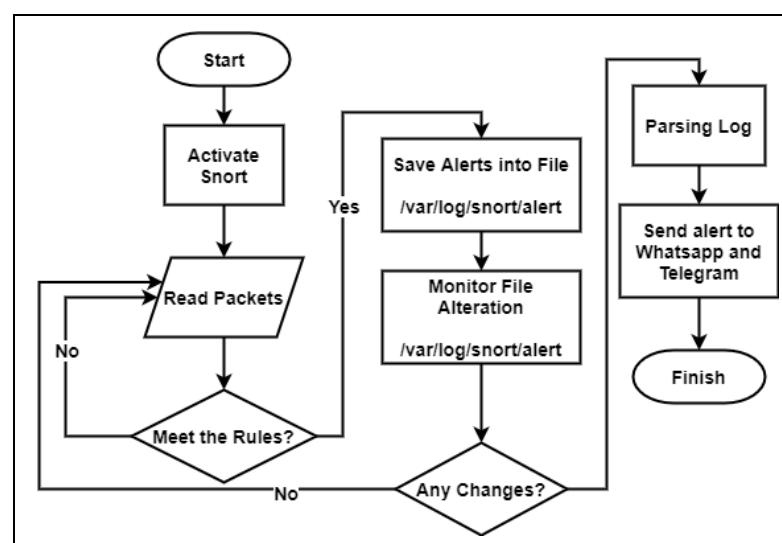
Gambar 2.4 Skenario Pengujian (Rupiat *et al.*, 2020)

Gambar 2.3 menjelaskan bagaimana alur penelitian yang dilakukan oleh Rupiat *et al.* Hal pertama yang dilakukan adalah menganalisis data yang dikumpulkan dari proses wawancara dan *survey*, serta merincikan kebutuhan penelitian yang meliputi bahan-bahan dan peralatan apa saja yang digunakan selama penelitian. Kemudian penelitian dilanjutkan dengan proses instalasi dan konfigurasi aplikasi pendukung yang digunakan pada perangkat Raspberry Pi. Selanjutnya, mengimplementasikan sistem sesuai rancangan pada server setelah instalasi dan konfigurasi aplikasi pendukung yang dilakukan pada tahap sebelumnya. Setelah sistem berhasil diimplementasikan, tahap berikutnya adalah pengujian pada sistem yang telah dibangun pada server Raspberry Pi. Untuk skenario pengujian seperti yang ditunjukkan pada Gambar 2.4, pengujian menggunakan aplikasi Nmap untuk melakukan serangan port scanning yang bertujuan untuk mengumpulkan informasi penting dari suatu jaringan, seperti *port 22 ssh (secure shell)* (Rupiat *et al.*, 2020).

Selanjutnya, sistem diuji menggunakan *tools hydra* dan *medusa* untuk melakukan serangan *brute force* yang bertujuan untuk memperoleh informasi berupa kumpulan *username* dan *password* secara ilegal dari sistem server. Setelah

penyerang melakukan dua teknik serangan tersebut, *honeypot* Cowrie dan IDS Snort yang sudah dibangun di dalam server Raspberry Pi berperan untuk mendeteksi serangan yang masuk ke dalam sistem server. Kemudian *honeypot* Cowrie berperan untuk menjebak penyerangan dengan mengalihkan penyerang ke sistem tiruan yang telah dibuat. Selama penyerang berada di dalam sistem, *honeypot* Cowrie mencatat seluruh kegiatan penyerang dan melakukan analisis terhadap identitas penyerang. Setelah itu, IDS Snort memblokir penyerang sesuai hasil analisis Cowrie dan menahan paket serangan yang masuk ke dalam sistem server (Rupiat et al., 2020).

Kemudian, penelitian lainnya yang juga menerapkan sistem notifikasi adalah penelitian yang dilakukan oleh Hakim *et al.* pada tahun 2020. Pada penelitian ini dilakukan implementasi IDS untuk mendeteksi serangan yang terjadi pada suatu sistem. Penelitian ini menerapkan Snort untuk melakukan analisis lalu lintas jaringan berdasarkan *real-time traffic* dan log paket IP pada jaringan (Hakim *et al.*, 2020). Gambar 2.5 menunjukkan alur dari penelitian Hakim *et al.*



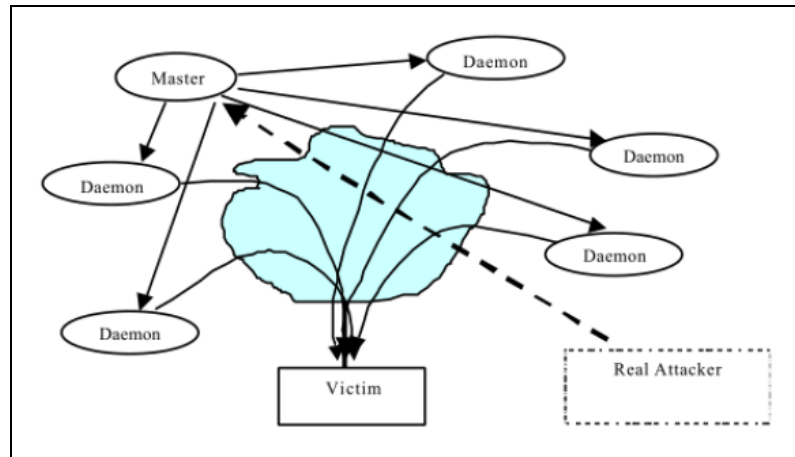
Gambar 2.5 Flowchart sistem pada penelitian Hakim *et al.* (Hakim et al., 2020)

Berdasarkan Gambar 2.5, alur sistem dimulai dengan menjalankan Snort yang telah dikonfigurasi dengan *rules* untuk mengurutkan paket yang telah diterima. Jika ada paket yang memenuhi *rules*, maka akan disimpan ke direktori di */var/log/snort/alert*. Setiap perubahan yang terjadi dalam direktori ini akan diamati oleh *tools* Swatchdog. Pada saat yang sama, perubahan log akan diurai dan sistem akan mengirimkan pemberitahuan tentang log yang telah diurai ke WhatsApp dan Telegram (Hakim *et al.*, 2020).

## **2.6. Serangan *Distributed Denial of Service* (DDoS)**

Pada tahun 2013, jumlah serangan DDoS meningkat hingga 50%, tetapi perusahaan yang menggunakan layanan perlindungan DDoS dapat memblokir serangan ini. Serangan DoS biasanya terjadi pada server, sistem, atau jaringan yang berkinerja buruk untuk meretas aset target, sehingga pengguna yang berwenang tidak mungkin menggunakan sumber daya. Tujuan dari serangan DDoS sama dengan serangan DoS, tetapi serangan DDoS adalah bentuk serangan DoS yang terdistribusi, yang berarti bahwa serangan tersebut terjadi dari lokasi yang berbeda untuk menyerang satu korban. Adapun kondisi yang menunjukkan bahwa sistem diserang oleh DoS atau DDoS (Hassan *et al.*, 2018) yaitu:

- a. Penurunan kualitas rutinitas jaringan, terutama ketika kita mencoba mengakses direktori atau catatan yang disimpan di jaringan atau untuk mengakses situs web.
- b. Tidak mengunjungi situs tertentu.
- c. Masalah yang ditemukan di situs web dan lebih dari jumlah email spam biasanya.



Gambar 2.6 Ilustrasi Serangan DDoS (Geges & Wibisono, 2015)

Berdasarkan karakteristik serangan DDoS yang diilustrasikan pada Gambar 2.6, terdapat empat komponen pada serangan ini, yaitu *attacker*, program kontrol utama, *daemon* serangan/*bots*, dan *victim*. Pertama, melibatkan *victim*, yaitu *host* target yang telah dipilih untuk menerima beban serangan. Kedua, melibatkan kehadiran agen serangan (*daemon*) yaitu program *agent* yang melakukan serangan secara langsung terhadap *victim*. *Daemon* biasanya ditempatkan di komputer inang/perantara. Instalasi *daemon* pada komputer inang mengharuskan *attacker* untuk mendapatkan akses dan berhasil menyusup ke komputer yang menjadi inang *daemon*. Komponen ketiga dari serangan DDoS adalah program kontrol utama. Tugasnya adalah untuk mengkoordinasikan serangan. Akhirnya, ada *attacker* yang menjadi aktor utama di balik serangan DDoS (Geges & Wibisono, 2015).

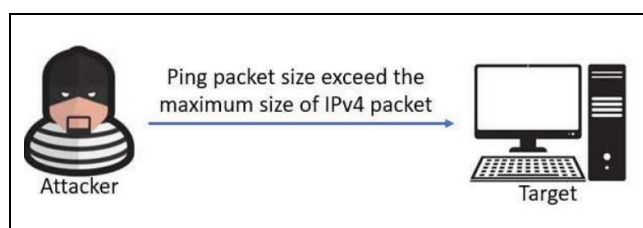
Secara umum, paket data yang beredar di jaringan menggunakan protokol TCP/IP untuk transmisinya. Paket ini sendiri tidak berbahaya, tetapi jika ada terlalu banyak paket yang abnormal, maka perangkat jaringan atau server akan mengalami kelebihan beban/*overload*. Kondisi ini dapat dengan cepat mengkonsumsi sumber daya sistem. Kasus lain adalah jika paket serangan

memanfaatkan celah keamanan pada protokol tertentu (misalnya *request* layanan yang tidak lengkap atau penyalahgunaan formasi protokol). Tindakan ini juga dapat menyebabkan kegagalan perangkat jaringan atau server. Kedua pendekatan serangan ini sama-sama mengakibatkan DoS. Kedua pendekatan ini merupakan prinsip-prinsip dasar serangan DDoS. Alasan utama mengapa sulit untuk mencegah serangan DDoS adalah karena pada suatu jaringan, lalu lintas yang sah dan yang ilegal tercampur. Identifikasi akan menjadi semakin sulit, ketika paket data serangan terlihat seperti paket data normal. Misalnya, dalam sistem IDS berbasis pencocokan pola *signature*, mungkin sulit untuk membedakan pesan ilegal dari pesan yang sah pada awal koneksi. Dalam banyak kasus, abnormalitas pada jaringan baru terlihat ketika serangan terjadi (Geges & Wibisono, 2015).

Serangan DDoS merupakan serangan DoS yang didistribusikan dari lokasi berbeda ketika komputasi awan memiliki faktor *availability*, *scalability*, *flexibility*, dan *accessibility*. Sebagian besar host terlibat dalam serangan. Istilah yang digunakan dalam serangan ini adalah *handler* dan *agent*. Serangan dimulai dengan dua atau lebih kontrol pemrosesan dan mengelola agent untuk memulai serangan DDoS (Hassan *et al.*, 2018).

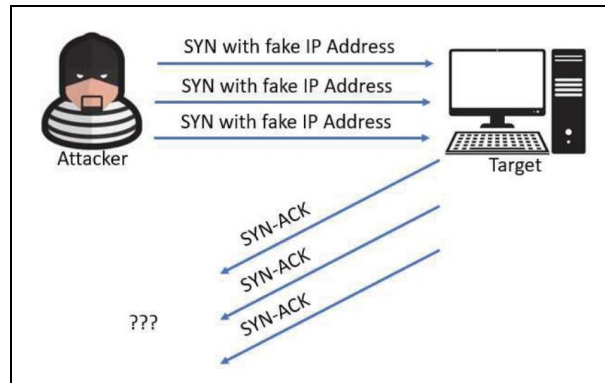
Berdasarkan penelitian yang dilakukan oleh Hakim *et al.*, terdapat beberapa jenis serangan DDoS yang dapat terjadi, diantaranya *Ping of Death attack* dan *SYN Flood attack*. Serangan *Ping of Death* adalah salah satu serangan sederhana namun efektif untuk membuat sistem target *crash* dan berpotensi diinjeksi dengan kode berbahaya. Ide dasar serangan ini adalah untuk memanfaatkan kelemahan desain sistem komputer dalam menangani paket IP yang masuk, yang tidak dapat ditangani jika ukuran paket lebih besar dari ukuran

paket maksimum paket IPv4 yaitu 65535 *byte*. Paket *Ping* umumnya 56 *bytes* atau 64 *bytes* jika menyertakan *header* IP. Dalam skenario serangan ini, penyerang akan mengirimkan paket ping dengan ukuran yang melebihi ukuran maksimum. Jadi, ketika paket diterima dan dipasang kembali, sistem target akan mengalami memori *buffer overflow* atau sistem *crash* (Hakim *et al.*, 2020). Berikut Gambar 2.7 menunjukkan ilustrasi dari serangan *Ping of Death*.



Gambar 2.7 Ilustrasi serangan *Ping of Death* (Hakim *et al.*, 2020)

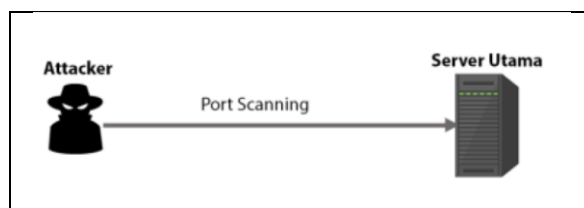
*SYN Flood attack* merupakan salah satu jenis dari serangan DDoS. Ide dasar dari serangan ini adalah melanggar protokol "*TCP Three-way Handshake*" pada koneksi TCP. *Three-Way Handshake* mengatur transmisi data TCP pada komunikasi *client-server*. *Client* mengirimkan paket SYN ke server, kemudian server merespon dengan mengirimkan paket SYN-ACK ke *client*, dan *client* mengirimkan paket ACK ke server. Setelah tiga langkah ini, koneksi antara klien dan server dapat dibuat. Skenario serangan ini dilakukan dengan penyerang menggunakan alamat IP palsu untuk mengirimkan sejumlah besar paket SYN berulang kali ke *port* acak di server target. Oleh karena itu, server akan mengalami kesulitan dalam menutup koneksi sambil terus menanggapi sekumpulan paket SYN yang masuk ke *client* yang sah. Kondisi ini akan menyebabkan server menjadi *crash* (Hakim *et al.*, 2020). Pada Gambar 2.8 ditunjukkan ilustrasi dari serangan *SYN Flood*.



Gambar 2.8 Ilustrasi serangan SYN Flood (Hakim *et al.*, 2020)

## 2.7. Serangan *Port Scanning*

*Port Scanning* merupakan tahap *information gathering* (pengumpulan informasi) dalam *penetration testing* yang dilakukan oleh *attacker* untuk melakukan enumerasi terhadap setiap *port* yang terbuka pada server utama. Tujuan dari serangan ini untuk mengetahui apakah *port 22* (SSH) pada server utama terbuka (*Open*) atau tidak (*Closed / Filtered*) dan menguji sistem dari *honeypot* yang telah dibuat berjalan dengan benar atau tidak (A. Utomo, 2018; Mardiyanto *et al.*, 2016).



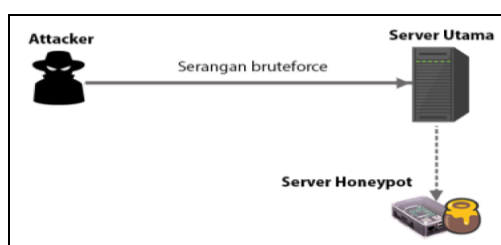
Gambar 2.9 Skema serangan *Port Scanning* (A. Utomo, 2018)

Berdasarkan Gambar 2.9, pada tahap ini server utama meneruskan permintaan *attacker* terhadap *port 22* di server utama ke *port 22* pada server tiruan (*honeypot*), sehingga *attacker* akan menerima hasil *port 22* yang terbuka pada server *honeypot*, walaupun *IP Address* yang diserang adalah *IP Address* server utama (A. Utomo, 2018).



## 2.8. Serangan *Brute-Force*

Serangan *brute-force* adalah metode serangan untuk meretas *password* yang ada dengan cara melakukan enumerasi huruf atau angka atau juga dengan menggunakan daftar list (*Dictionary / Wordlist*) *username-password* pada fitur atau fungsi *login* (A. Utomo, 2018). Tujuan dari serangan ini adalah untuk mendapatkan kombinasi *username* dan *password* secara ilegal (Rupiat *et al.*, 2020). Skema serangan *brute-force* ditunjukkan pada Gambar 2.10.

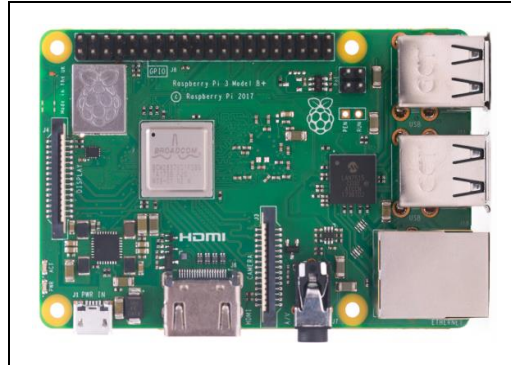


Gambar 2.10 Skema serangan *Brute-Force* (A. Utomo, 2018)

## 2.9. Raspberry Pi 3 Model B+

Raspberry Pi 3 Model B+ adalah model komputer papan tunggal terbaru dari Raspberry Pi generasi ketiga. Perangkat ini seperti ditunjukkan pada Gambar 2.11 dilengkapi dengan *chipset* Broadcom BCM2873BO Cortex A53 64-bit sehingga dapat menghasilkan kecepatan hingga 1.4GHz, *dual-band wireless Local Area Network (LAN)* 2.4Ghz dan 5Ghz, Bluetooth 4.2/BLE, Gigabit Ethernet dengan USB 2.0 maksimum keluaran 300 Mbps, dan 4 x USB 2.0 *port* (RaspberryPi, 2016). Perangkat ini dipilih karena teknologi terbaru didalamnya dapat mengendalikan panas saat pemakaian dengan kinerja maksimal. Perangkat ini dinilai mampu bekerja secara efektif dan efisien ketika diimplementasikan pendeteksi jaringan seperti IDS dan *honeypot*. Selain itu, perangkat ini juga terjangkau dan relatif murah dibandingkan dengan perangkat *firewall* lainnya

sehingga dapat menghemat biaya dan sumber daya yang dikeluarkan saat implementasi proyek (A. Utomo, 2018).



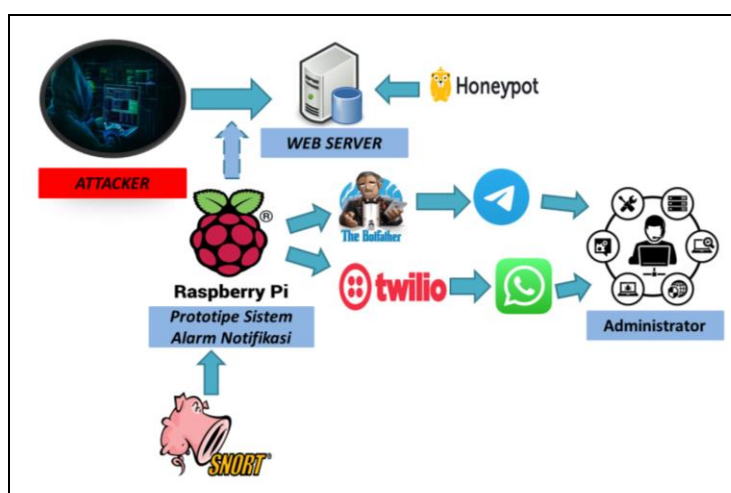
Gambar 2.11 Raspberry Pi 3 Model B+ (RaspberryPi, 2016)

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Gambaran Umum Subyek Penelitian

Berikut merupakan gambaran umum subyek penelitian yang akan dibangun pada penelitian ini seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Gambaran Umum Sistem

Pada Gambar 3.1 dijelaskan bahwa penyerang melakukan serangan untuk menghentikan dan membebani kinerja dari sistem pada *web server*. Ketika serangan tersebut terjadi, Snort akan melakukan pendeteksian sesuai dengan *rules* yang telah diatur. Kemudian, *alert* serangan yang telah dicatat oleh Snort akan dikirimkan ke prototipe sistem alarm notifikasi yang dalam hal ini dilakukan oleh Raspberry Pi 3 Model B+. Setelah *alert* serangan diidentifikasi, sistem akan mengalihkan penyerang ke server sistem yang dalam hal ini diolah oleh *honeypot* Cowrie. Hasil identifikasi serangan ini akan dicatat pada masing-masing log Snort dan Cowrie, kemudian dikirimkan dalam bentuk pesan notifikasi kepada administrator melalui Telegram *Bot* dan Whatsapp API.

Pengiriman pesan notifikasi pada Telegram dan Whatsapp administrator jaringan memanfaatkan *Application Programming Interface* (API). API berfungsi untuk membuat sistem dapat mengirimkan notifikasi secara otomatis melalui Telegram dan Whatsapp. Pada Telegram terdapat fitur Telegram *Bot*, dimana *Bot* merupakan singkatan dari robot yang memungkinkan pengguna untuk berinteraksi dengan mesin dalam mengirim baris perintah dan pesan. Pengguna dapat mengakses fitur tersebut melalui permintaan HTTPS dan *Bot* API. Sedangkan untuk Whatsapp, fitur API yang dapat digunakan pengguna untuk membuat Whatsapp API adalah Twilio. Dengan adanya kedua fitur tersebut, sistem alarm notifikasi akan mengirimkan pesan notifikasi kepada administrator (Hakim *et al.*, 2020).

### **3.1.1. Tempat dan Waktu Penelitian**

Penelitian ini akan dilaksanakan di UPT. Pusat Komputer Universitas Dehasen Bengkulu, yang beralamat di Jalan Meranti Raya No. 32 Sawah Lebar Kota Bengkulu. Waktu penelitian ini dilaksanakan mulai dari bulan September 2022 sampai dengan Agustus 2023.

### **3.1.2. Struktur Organisasi**

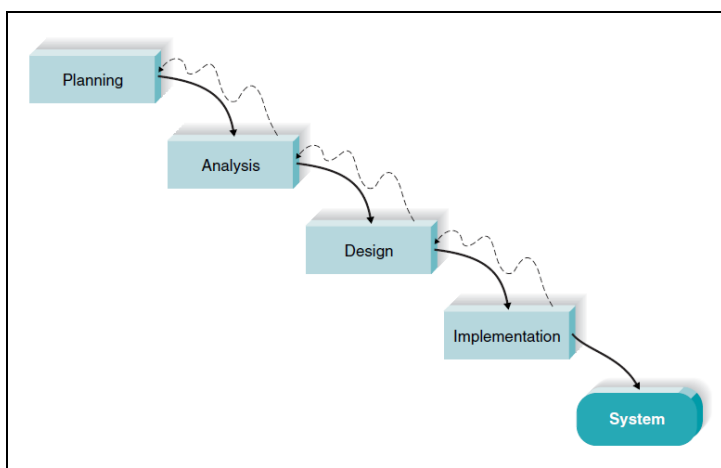
Struktur organisasi adalah suatu kerangka sistem yang digunakan untuk menunjukkan kedudukan dan koordinasi masing-masing unsur pelaksana organisasi, tanggung jawab, jalur hubungan masing-masing hierarki, dan uraian tugas masing-masing unsur pelaksana. Adapun struktur organisasi dari UPT. Pusat Komputer (Puskom) Universitas Dehasen (UNIVED) Bengkulu sebagaimana terlampir pada lampiran.

### 3.1.3. Tugas dan Wewenang

Tugas dan wewenang dari UPT. Puskom UNIVED Bengkulu adalah untuk mengelola bidang pekerjaan tertentu dalam mendukung pelaksanaan kegiatan proses belajar dan mengajar di UNIVED Bengkulu. Bidang pekerjaan tersebut meliputi teknologi informasi, sistem informasi, pelaksanaan mata kuliah praktikum di seluruh bidang studi, serta menunjang kegiatan penelitian yang diselenggarakan oleh dosen dan mahasiswa.

### 3.2. Metode Penelitian

Metodologi penelitian yang akan digunakan pada penelitian ini adalah model pengembangan *System Development Life Cycle (SDLC)* dengan pendekatan *Waterfall Development*. Pendekatan *Waterfall* dipilih karena sistem yang dibuat harus memenuhi parameter keberhasilan yang ada. Sehingga tidak perlu ada lagi kembali ke tahap sebelumnya secara berulang dalam melakukan pembangunan sistem di tahap selanjutnya. Adapun proses dari *Waterfall* ditunjukkan pada Gambar 3.2 berikut ini:



Gambar 3.2 *Waterfall Development* (Roth, Dennis, 2012, 2015)

Berdasarkan Gambar 3.2 tersebut, proses pembangunan sistem dilakukan bertahap dan harus memenuhi kriteria keberhasilan dari setiap prosesnya. Ketika kriteria tersebut berhasil dipenuhi, maka akan lebih mudah untuk beralih ke tahap selanjutnya. Hal ini sesuai dengan kebutuhan pada penelitian ini dimana sistem harus dibangun per tahap dan harus berhasil sehingga dapat beralih ke tahap pembangunan selanjutnya. Apabila kriteria keberhasilan tidak dapat dipenuhi maka akan sulit untuk kembali ke tahap sebelumnya. Pendekatan ini memungkinkan pengguna untuk mendapatkan sistem yang sesuai dengan diharapkan pada tujuan penelitian.

SDLC merupakan proses memahami bagaimana sistem informasi dapat mendukung kebutuhan bisnis, merancang sistem, membangun sistem, dan memberikannya kepada pengguna (Roth, Dennis, 2012, 2015). *Waterfall Development* merupakan model yang memiliki sifat sistematis dalam penerapannya. Setiap tahap yang dilakukan pada model ini harus bisa dipastikan bahwa semua kebutuhan yang akan dilakukan sudah dipenuhi dengan sempurna (*fixed*), sehingga memungkinkan pada tahap selanjutnya tidak ada lagi sistem ulangan-ulangan untuk kembali pada tahap-tahap sebelumnya. Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun sistem (Agarwal *et al.*, 2010; Roth, Dennis, 2012, 2015). Untuk mendapatkan sistem yang sesuai, dilakukan dengan cara menyelesaikan seluruh tahapan SDLC kemudian dilakukan evaluasi terhadap hasil yang diperoleh dengan menggunakan kebutuhan yang ditetapkan. Berikut penjelasan dari masing-masing tahapan yang dilakukan.

### **3.2.1. Planning**

Merupakan tahapan pertama yang dilakukan untuk memahami alasan mengapa sistem dibangun dan menentukan bagaimana cara untuk membangun sistem (Agarwal *et*

*al.*, 2010; Roth, Dennis, 2012, 2015). Tahap ini juga membantu peneliti untuk mengevaluasi anggaran yang akan dikeluarkan dan keuntungan (manfaat) dari ide penelitian yang akan dilakukan (Agarwal *et al.*, 2010). Pada penelitian ini, tahap *planning* dilakukan dengan observasi, wawancara, dan studi literatur terkait dengan keamanan pada *web server*, serangan siber yang memiliki peluang terjadi pada *web server*, urgensi terkait dengan deteksi serangan sebelum akhirnya dilakukan penanggulangan. Observasi dan wawancara dilakukan dengan berkonsultasi dengan dosen pembimbing, narasumber di tempat penelitian, dan para ahli di bidang terkait yang dapat memberikan saran serta masukan dari tema penelitian yang akan dikerjakan. Sedangkan, Studi literatur dilakukan dengan telaah kepustakaan dari teori-teori pada makalah/*paper* penelitian sebelumnya yang mendukung penelitian ini, mengidentifikasi masalah dari penelitian terkait yang menjadi rujukan sehingga didapatkan rumusan masalah yang dapat diatasi pada penelitian ini. Hasil yang diharapkan berupa penentuan latar belakang, rumusan masalah, pembatasan masalah, dan tahapan/rencana penelitian.

### **3.2.2. Analysis**

Merupakan tahapan untuk menentukan jawaban dari siapa yang akan menggunakan sistem yang dibangun, cara kerja sistem, dimana dan kapan sistem akan digunakan (Roth, Dennis, 2012, 2015). Tahap ini merupakan tahap penting dan kritis pada model *waterfall* serta diharuskan untuk melakukan analisis kebutuhan dari sistem yang dibangun (Agarwal *et al.*, 2010). Tujuannya untuk menentukan fungsi, struktur, kegunaan, performa dan model dalam merancang sistem (Agarwal *et al.*, 2010).

Proses analisis memiliki dua kebutuhan yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang melekat pada informasi yang

dibutuhkan pada sistem. Sedangkan kebutuhan non-fungsional adalah kebutuhan yang berhubungan dengan karakteristik, spesifikasi, perilaku yang harus diterapkan pada sistem.

Pada tahap ini dilakukan analisis hasil observasi, wawancara, dan studi literatur terhadap *planning* dan observasi terhadap perangkat-perangkat yang memiliki fungsi untuk mendukung pengerjaan penelitian ini, seperti Raspberry Pi 3 Model B+ sebagai sistem alarm notifikasi dan langkah-langkah atau *tools* yang akan digunakan. Hasil yang diharapkan berupa analisa sistem aktual, analisa sistem baru, daftar kebutuhan meliputi kebutuhan fungsional dan non-fungsional yang digunakan dalam melakukan perancangan, gambaran umum sistem, dan topologi jaringan yang akan dibuat.

### 3.2.3. *Design*

Merupakan tahap penentuan mengenai bagaimana sistem akan beroperasi, perangkat yang akan digunakan untuk mengoperasikan sistem, *interface* pengguna dan semua yang dibutuhkan dalam membangun sistem (Roth, Dennis, 2012, 2015). Tahap ini bertujuan untuk mentransformasikan spesifikasi kebutuhan sistem yang terdapat dalam dokumen *Software Requirement System* (SRS) ke struktur rancangan yang sesuai dengan beberapa bahasa pemrograman (Agarwal et al., 2010). Pada tahap *design* akan dihasilkan rancangan dari skema sistem prototipe alarm notifikasi yang akan dibuat dengan fungsi yang dapat menerima *alert* dan mengirimkan notifikasi kepada administrator ketika terdapat indikasi serangan pada *web server*.

*Design* dilakukan menggunakan aplikasi *draw.io* untuk menghasilkan visualisasi skema sistem, *flowchart* dan gambaran umum dari sistem yang akan dibangun. Hasil dari



tahap *design* adalah gambaran terkait rancangan sistem prototipe alarm notifikasi dan topologi jaringan yang akan dibangun dalam bentuk gambar dan bagan alir.

#### **3.2.4. Implementation**

Merupakan tahap akhir pada SDLC (Roth, Dennis, 2012, 2015). Tahapan ini memiliki dua langkah yang harus dilakukan yaitu, pembangunan sistem dan pengujian sistem. Tujuan dari tahap ini adalah untuk menghasilkan data dari implementasi sesuai dengan rancangan sistem yang telah dibuat dan pengujian terhadap sistem yang dibangun (Agarwal *et al.*, 2010).

Dalam simulasi sistem, akan dilakukan menggunakan Virtualbox yang akan mensimulasikan *environment web server* yang sudah dilakukan konfigurasi menggunakan Apache. Kemudian, prototipe alarm notifikasi akan menggunakan modul Raspberry Pi 3 Model B+ yang diimplementasikan Snort untuk mendeteksi serangan sesuai dengan *rules* yang diatur dan *honeypot* Cowrie sebagai server tiruan yang akan menjebak penyerang. Selanjutnya, prototipe ini mengirimkan notifikasi kepada administrator melalui aplikasi Telegram dan Whatsapp.

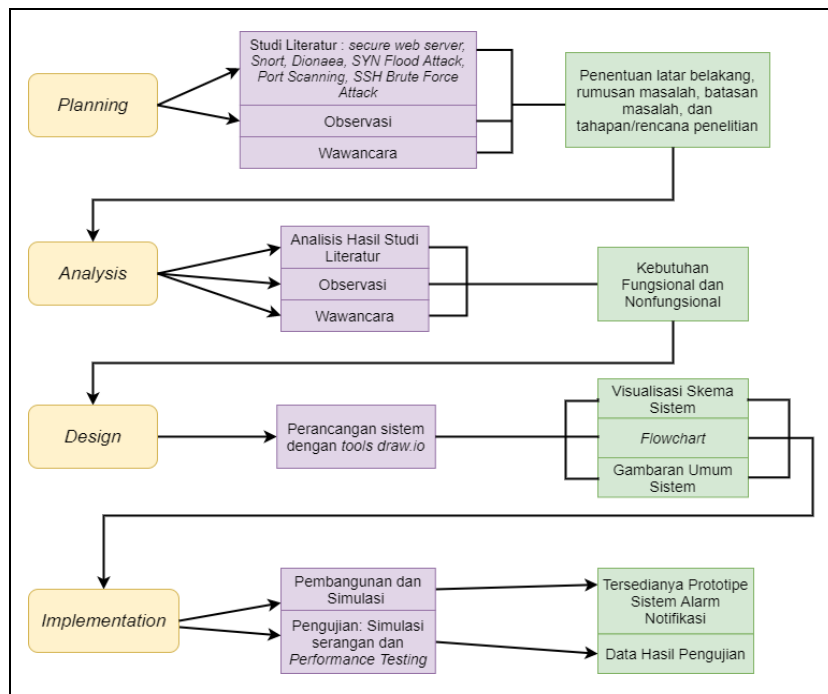
Selain itu, pengujian akan dilakukan dengan simulasi serangan dan pengujian fungsional sistem pada prototipe. Jenis serangan yang akan digunakan adalah *SYN Flood attack*, *Port Scanning*, dan *SSH Brute Force Attack*. Proses ini bertujuan untuk menyimpulkan bahwa sistem dapat mendeteksi serangan yang dilakukan. Sedangkan, pengujian fungsional sistem pada prototipe dilakukan dengan metode *performance testing*. Dari serangkaian metode pengujian yang dilakukan akan menghasilkan data yang selanjutnya akan digunakan sebagai pertimbangan dalam merumuskan kesimpulan akhir dari penelitian ini.

### a. Simulasi Serangan

Metode simulasi serangan dilakukan dengan tujuan untuk mengetahui apakah sistem yang dibuat dapat mendeteksi ketika terjadi serangan pada *web server*. Jenis serangan yang digunakan adalah *SYN Flood attack*, *Port Scanning*, dan *SSH Brute Force Attack*.

### b. Performance testing

Metode *performance testing* dilakukan untuk mengetahui tingkat keberhasilan sistem dalam menjalankan fungsinya dan waktu yang dibutuhkan untuk mengolah data. Parameter yang digunakan dalam metode ini adalah waktu eksekusi program dan kapasitas memori yang digunakan pada komponen prototipe alarm notifikasi.



Gambar 3.3 Kerangka Penelitian

### 3.3. Analisis Kebutuhan

Pada bagian analisis kebutuhan akan dibahas mengenai identifikasi kebutuhan pada sistem yang akan dibangun, kebutuhan fungsional dan non-fungsional, serta lingkungan implementasi perangkat keras dan perangkat lunak.

#### 3.3.1. Identifikasi Kebutuhan

Identifikasi kebutuhan dihasilkan melalui hasil analisis literatur yang digunakan sebagai referensi pada penelitian ini. Identifikasi kebutuhan dilakukan terhadap tiga referensi yang menerapkan sistem notifikasi menggunakan IDS berbasis Snort dan *honeypot* Cowrie diterapkan pada Raspberry Pi 3 Model B+. Berdasarkan analisis literatur yang telah dilakukan, didapatkan hasil identifikasi kebutuhan sistem dari penelitian yang dirujuk sebagai referensi. Tabel 3.1 menunjukkan hasil identifikasi kebutuhan dari literatur pada penelitian Hakim *et al.*, Atmojo, dan Utomo *et al.*

Tabel 3.1 Hasil Identifikasi Kebutuhan Sistem

Penelitian Hakim <i>et al.</i>	Penelitian Atmojo	Penelitian Utomo <i>et al.</i>
Sistem dapat melakukan deteksi serangan pada jaringan komputer	Sistem dapat melakukan deteksi serangan pada jaringan komputer	Sistem dapat melakukan deteksi serangan pada jaringan komputer
Sistem dapat melakukan monitoring dan <i>parsing log</i>	Sistem dapat melakukan monitoring terhadap perubahan log serangan	Sistem menjebak penyerang dengan membuat server tiruan
Sistem dapat mengirimkan notifikasi kepada administrator jaringan melalui Telegram dan Whatsapp	Sistem dapat mengirimkan notifikasi kepada administrator jaringan melalui Bot Telegram	Sistem dapat mengidentifikasi serangan yang dilakukan dengan mencatat riwayat serangan

#### 3.3.2. Penentuan Kebutuhan Sistem

Setelah dilakukan identifikasi kebutuhan, perlu dilakukan penentuan kebutuhan sistem dalam tahap perancangan. Hal ini bertujuan untuk menentukan spesifikasi sistem yang akan dibangun. Spesifikasi sistem yang dibutuhkan dapat diuraikan dalam bentuk daftar kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional dan non-fungsional merupakan hal-hal yang diperlukan pada sistem. Kebutuhan fungsional merupakan kebutuhan yang berhubungan langsung terhadap kegiatan yang dibutuhkan oleh sistem. Kemudian, kebutuhan non-fungsional merupakan properti yang dibutuhkan untuk mendukung kegiatan yang dilakukan oleh sistem. Berikut merupakan daftar kebutuhan fungsional dan non-fungsional pada prototipe sistem alarm notifikasi.

#### **a. Kebutuhan fungsional**

Kebutuhan fungsional dari sistem pada penelitian ini adalah sebagai berikut.

- a. Sistem dapat mendeteksi saat terjadi indikasi serangan pada *web server*.
- b. Sistem ini dapat mengirimkan notifikasi kepada administrator jaringan saat terdapat indikasi serangan pada *web server* berdasarkan paket *alert* yang tercatat pada log serangan pada Snort ke perangkat komunikasi administrator melalui aplikasi Telegram dan Whatsapp.
- c. Sistem ini dapat memberitahukan alamat IP yang terindikasi sebagai penyerang yang melakukan serangan.
- d. Sistem ini dapat menjebak penyerang dan mengalihkan serangan ke server tiruan serta mencatat seluruh aktivitas penyerang dalam *file log Cowrie*.

- e. Sistem ini dapat melakukan audit serangan secara berkala dan menghitung jumlah serangan yang terjadi pada periode waktu tertentu.

#### **b. Kebutuhan non-fungsional**

Kebutuhan non-fungsional dari sistem pada penelitian ini adalah sebagai berikut.

- a. Sistem ini menggunakan Raspberry Pi 3 Model B+.
- b. Sistem ini menggunakan Kali Linux / Windows 10 sebagai komputer penyerang.
- c. Sistem ini menggunakan IDS berbasis Snort.
- d. Sistem ini menggunakan *honeypot* Cowrie.
- e. Sistem ini menggunakan BotFather pada Telegram dan Twilio Whatsapp API untuk melakukan pengiriman pesan notifikasi kepada administrator.

#### **3.3.3. Lingkungan Implementasi Perangkat Keras**

Berikut merupakan spesifikasi perangkat keras yang digunakan untuk mendukung proses pembuatan sistem yang ditunjukkan pada Tabel 3.2.

Tabel 3.2 Daftar Kebutuhan Perangkat Keras

No	Perangkat	Spesifikasi
1	<i>Single Board Computer</i>	Raspberry Pi 3 Model B+
2	Laptop HP Elitebook 840	<i>Processor</i> : Intel Core i-5 <i>Operating system</i> : Windows 10 RAM : 8 GB
3	Monitor	Monitor PC

#### **3.3.4. Lingkungan Implementasi Perangkat Lunak**

Berikut merupakan spesifikasi perangkat lunak yang digunakan untuk mendukung proses pembuatan sistem yang ditunjukkan pada Tabel 3.3.

Tabel 3.3 Daftar Kebutuhan Perangkat Lunak

No	Perangkat	Spesifikasi
1	Editor Python Raspberry Pi	TonyPython
2	<i>Operating system</i>	Raspbian OS (Bullseye)
3	<i>Intrusion Detection System (IDS)</i>	Snort 2.9.20
4	<i>Honeypot</i>	Cowrie
5	<i>Database</i>	MySQL
6	<i>Web Server</i>	Apache

### 3.4. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian ini adalah observasi, wawancara, studi literatur, dan analisis hasil implementasi serta pengujian pada sistem.

#### 3.4.1. Observasi dan wawancara

Observasi dan wawancara bertujuan untuk mendapatkan informasi tentang sistem aktual yang akan dilakukan analisis permasalahan dan kemudian dijadikan sebagai landasan untuk penelitian selanjutnya. Informasi yang didapat saat observasi berasal dari tempat penelitian, yaitu UPT. Puskom UNIVED Bengkulu. Sedangkan, informasi yang didapat saat wawancara berasal dari keterangan yang diberikan oleh Kepala UPT. Puskom UNIVED Bengkulu sesuai dengan pertanyaan-pertanyaan yang diajukan.

#### 3.4.2. Studi Literatur

Studi literatur bertujuan untuk mendapatkan informasi berupa daftar kebutuhan baik fungsional maupun non-fungsional yang digunakan dalam melakukan perancangan dan gambaran umum sistem yang akan dibuat. Hasil yang diharapkan pada tahap ini dapat dilakukan dengan menganalisis teori-teori pendukung dari penelitian sebelumnya.

#### 3.4.3. Hasil Implementasi dan Pengujian

Analisis hasil implementasi dan pengujian bertujuan untuk menyimpulkan apakah penelitian yang dilakukan dapat menjawab permasalahan yang diangkat atau tidak.

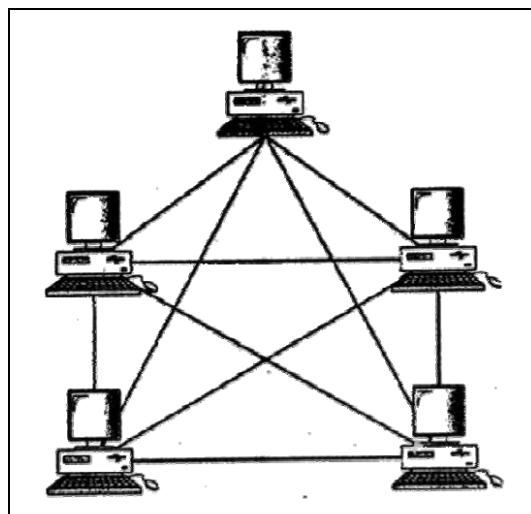
Informasi yang didapat pada tahap ini berasal dari data-data dan dokumentasi yang dilakukan saat melakukan implementasi dan pengujian prototipe sistem alarm notifikasi ini.

### 3.5. Metode Perancangan Sistem

Pada bagian ini akan dibahas tentang analisa sistem aktual, analisa sistem baru, dan perancangan pengujian yang akan dilakukan pada penelitian ini.

#### 3.5.1. Analisa Sistem Aktual

Analisa sistem aktual diisi berdasarkan data-data yang didapat dari hasil observasi dan wawancara di tempat penelitian. Analisa sistem aktual dilakukan untuk mengobservasi keadaan terkini jaringan yang ada pada UPT. Puskom UNIVED Bengkulu. Berdasarkan keterangan dari narasumber di UPT. Puskom UNIVED Bengkulu, yakni Kepala UPT. Puskom UNIVED Bengkulu Bapak Khairil, S.Kom, M.Kom., topologi jaringan yang digunakan di lingkungan UNIVED Bengkulu adalah topologi Mesh. Berikut ini merupakan topologi jaringan yang digunakan pada UPT. Puskom UNIVED Bengkulu:



Gambar 3.4 Topologi Mesh (Wagiu et al., 2016)

Berdasarkan gambar Gambar 3.4, topologi Mesh terdiri beberapa perangkat yang terhubung dan memiliki koneksi atau biasa disebut sebagai *link* secara *point-to-point* dengan setiap perangkat lainnya. Oleh karena itu, topologi Mesh memiliki  $\frac{n(n-1)}{2}$  kanal fisik untuk menghubungkan  $n$  perangkat. Dalam mengelola banyak *link* tersebut, seluruh perangkat dalam jaringan tersebut harus memiliki *port input/output* (I/O) sebanyak  $n$  atau sebanding dengan jumlah perangkat yang ada (Wagiu et al., 2016).

Topologi ini dapat memberikan kemampuan koneksi yang kuat karena ketika satu *link* dalam topologi Mesh tidak stabil maka tidak akan mempengaruhi *link* yang lain sehingga tidak menyebabkan seluruh sistem terhenti. Kemudian, jika terdapat kerusakan pada salah satu *link*, maka *station* dapat mencari dan menggunakan *link* yang lain tanpa mengalami suatu kendala yang berarti. Selain itu, topologi ini juga menjamin kerahasiaan dan keamanan data, karena setiap pesan dikirim melalui suatu *link* khusus (Wibawanto, 2018).

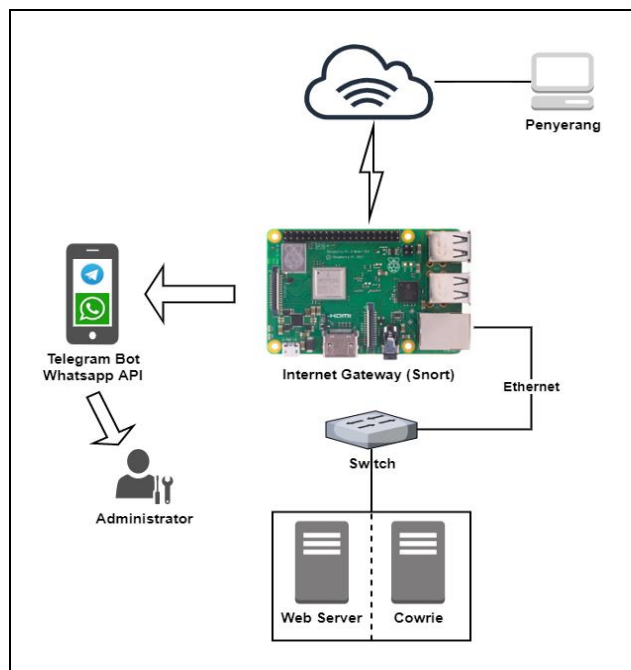
### **3.5.2. Analisa Sistem Baru**

Pada bagian ini akan dijelaskan tentang analisa sistem baru yang akan dibangun dalam penelitian ini. Bagian ini meliputi topologi jaringan dan perancangan pengujian.

#### **a. Topologi Jaringan**

Gambar 3.5 berikut ini merupakan topologi jaringan yang akan dibangun dalam penelitian ini.





Gambar 3.5 Topologi Jaringan

Berdasarkan Gambar 3.5, topologi jaringan pada penelitian ini terdiri dari server, PC yang akan melakukan serangan, prototipe sistem alarm notifikasi, dan perangkat komunikasi yang memiliki fitur aplikasi pesan singkat. Snort yang diimplementasikan pada prototipe ini akan mendeteksi serangan yang mencoba mengakses *web server*. Aktivitas serangan yang terjadi akan dicatat dalam log serangan pada Snort dan akan menghasilkan *alert*. Kemudian, prototipe sistem alarm notifikasi akan mengirimkan notifikasi serangan kepada administrator jaringan ketika terjadi indikasi serangan pada *web server*. *Output* dari notifikasi serangan bagi administrator jaringan berupa pesan singkat yang akan dikirimkan ke aplikasi Telegram dan Whatsapp.

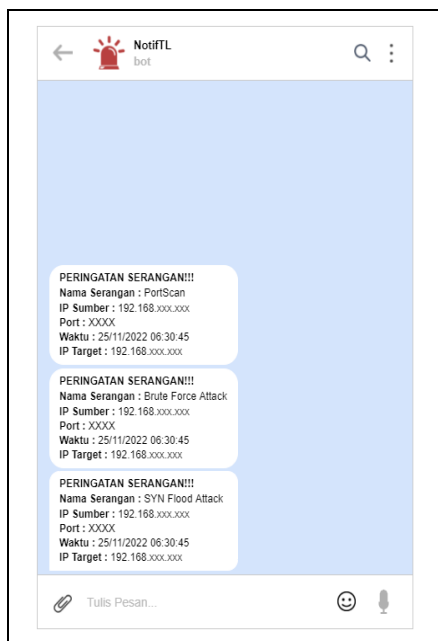
Pengiriman notifikasi serangan, memanfaatkan fitur API yang ada pada Telegram dan Whatsapp. API memungkinkan sistem dapat mengirimkan pesan secara otomatis kepada penerima yang dituju. Pada aplikasi Telegram terdapat fitur Telegram *Bot* atau fitur robot yang dapat membantu pengguna berinteraksi dengan mesin dalam baris

perintah dan pesan. Pengguna dapat mengakses fitur tersebut melalui permintaan HTTPS dan *Bot* API. Sedangkan untuk Whatsapp, fitur API yang dapat digunakan pengguna untuk membuat Whatsapp API adalah Twilio.

Setelah itu, Cowrie akan berperan sebagai server tiruan untuk mengalihkan serangan yang dilakukan oleh penyerang. Selama berada dalam server tiruan, Cowrie akan mencatat seluruh riwayat akses yang dilakukan oleh penyerang. Riwayat aktivitas yang dilakukan penyerang selama berada dalam Cowrie akan tersimpan dalam *file* log. Data-data yang terdapat pada *file* log Cowrie selanjutnya akan dilakukan analisis oleh administrator terkait dengan perilaku dan anomali yang ditinggalkan penyerang selama melakukan aktivitas atau serangan terhadap *web server*. Berdasarkan data-data tersebut dapat diambil kebijakan dalam penanganan dan penanggulangan serangan oleh tim manajemen insiden.

Selain itu, prototipe ini dilengkapi dengan fitur audit serangan secara berkala yang akan menghitung jumlah serangan yang terjadi pada periode waktu tertentu dan akan dikirimkan kepada administrator yang juga dalam bentuk pesan notifikasi. Fitur ini akan dilakukan oleh Crontab sebagai *tools* pada Linux untuk melakukan penjadwalan saat menjalankan perintah atau skrip program secara otomatis. Perintah otomatis ini dapat dilakukan dalam periode waktu tertentu setiap hari, minggu, atau lainnya sesuai dengan kebutuhan sistem.

Adapun format notifikasi yang akan dikirimkan ke Telegram adalah sebagaimana ditunjukkan pada Gambar 3.6 dan Gambar 3.7.



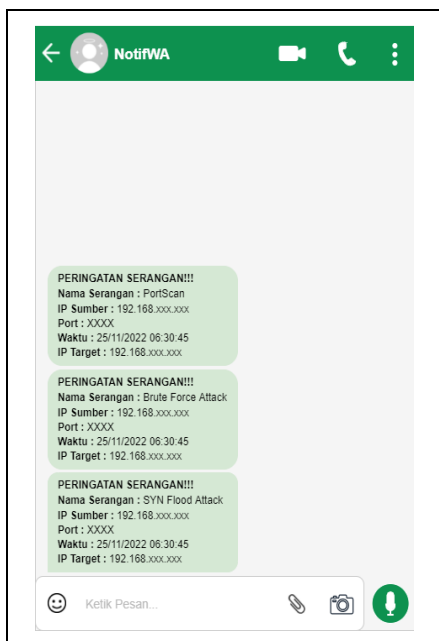
Gambar 3.6 Format Notifikasi  
*Real-Time* Telegram



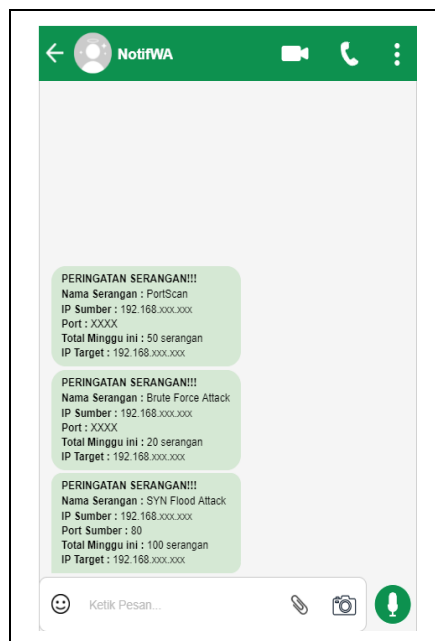
Gambar 3.7 Format Notifikasi  
Per Periode Telegram

Gambar 3.6 merupakan format notifikasi *real-time* yang akan dikirimkan ke aplikasi Telegram. Informasi yang disampaikan berupa nama serangan, IP sumber sebagai alamat IP penyerang, keterangan *port* sebagai *port* yang diserang, waktu menunjukkan kapan serangan tersebut terjadi, dan IP target merupakan alamat IP dari perangkat yang diserang. Sedangkan Gambar 3.7 menunjukkan format notifikasi per periode pada Telegram. Informasi yang dikirim sama dengan format notifikasi *real-time* hanya berbeda pada poin total serangan minggu ini, yang mana menunjukkan jumlah serangan yang terjadi pada periode selama satu minggu.

Adapun format notifikasi yang akan dikirimkan ke Whatsapp adalah sebagaimana ditunjukkan pada Gambar 3.8 dan Gambar 3.9.



Gambar 3.8 Format Notifikasi  
*Real-Time* Whatsapp



Gambar 3.9 Format Notifikasi  
Per Periode Whatsapp

Gambar 3.8 merupakan format notifikasi *real-time* yang akan dikirimkan ke aplikasi Whatsapp. Informasi yang disampaikan berupa nama serangan, IP sumber sebagai alamat IP penyerang, keterangan *port* sebagai *port* yang diserang, waktu menunjukkan kapan serangan tersebut terjadi, dan IP target merupakan alamat IP dari perangkat yang diserang. Sedangkan Gambar 3.9 menunjukkan format notifikasi per periode pada Whatsapp. Informasi yang dikirim sama dengan format notifikasi *real-time* hanya berbeda pada poin total serangan minggu ini, yang mana menunjukkan jumlah serangan yang terjadi pada periode selama satu minggu.

Arsitektur jaringan dibuat sesuai dengan topologi jaringan. Berdasarkan arsitektur tersebut, terdapat spesifikasi perangkat, konfigurasi alamat IP, dan sistem operasi yang digunakan. Konfigurasi alamat IP pada server akan dijelaskan pada Tabel 3.4.

Tabel 3.4 Konfigurasi Server Utama

<i>Network Adapter</i>	enp0s2
<i>IP Address</i>	192.168.20.100
<i>Network</i>	192.168.20.10
<i>Netmask</i>	255.255.255.0
Sistem Operasi	Ubuntu 20.04 LTS

Berdasarkan Tabel 3.4, alamat IP dari server utama adalah 192.168.20.100. Sistem operasi yang digunakan oleh server utama adalah Ubuntu 20.04 LTS. Konfigurasi alamat IP dari prototipe sistem alarm notifikasi dapat dilihat pada Tabel 3.5.

Tabel 3.5 Konfigurasi Prototipe Sistem Alarm Notifikasi

<i>Network Adapter</i>	eth0 eth1
<i>IP Address</i>	192.168.20.10 192.168.10.1
<i>Network</i>	192.168.20.10 192.168.10.1
<i>Netmask</i>	255.255.255.0
Sistem Operasi	Raspbian Bullseye (Debian 11)

Berdasarkan Tabel 3.5, prototipe sistem alarm menggunakan dua *interface*, yaitu eth0 dan eth1. Alamat IP dari masing-masing *interface* adalah 192.168.20.10 dan 192.168.10.1. Sistem operasi yang digunakan oleh prototipe sistem alarm notifikasi adalah Raspbian Bullseye. Konfigurasi alamat IP dari PC Penyerang dapat dilihat pada Tabel 3.6.

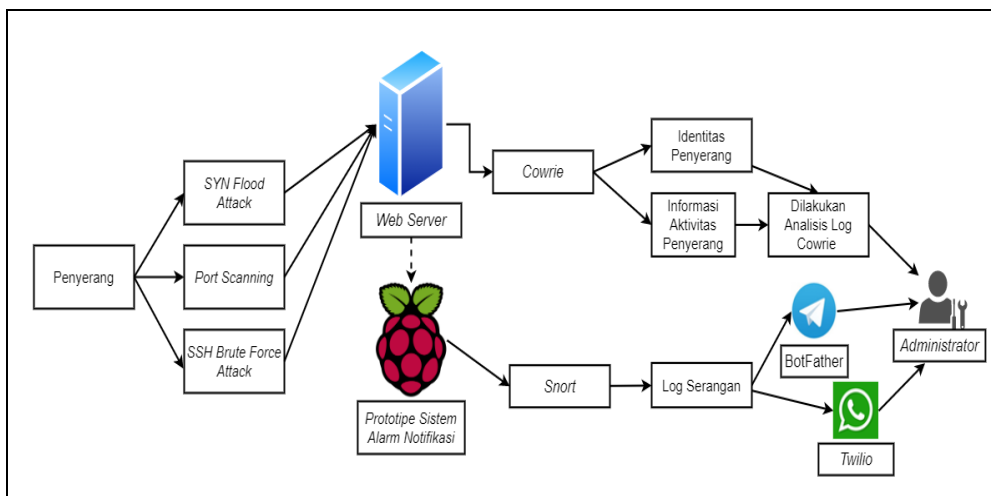
Tabel 3.6 Konfigurasi PC Penyerang

<i>Network Adapter</i>	eth0
<i>IP Address</i>	192.168.10.30
<i>Network</i>	192.168.10.1
<i>Netmask</i>	255.255.255.0
<i>Sistem Operasi</i>	Kali Linux

Berdasarkan Tabel 3.6, alamat IP dari PC 2 adalah 192.168.10.30. Sistem operasi yang digunakan oleh PC Penyerang adalah Kali Linux.

### b. Perancangan Pengujian

Gambar 3.10 berikut ini merupakan rancangan pengujian yang akan dilakukan pada penelitian ini.



Gambar 3.10 Rancangan Pengujian

Berdasarkan pada Gambar 3.10, pada penelitian ini akan dilakukan pengujian dengan 2 metode, yaitu simulasi serangan dan *performance testing*. Simulasi serangan bertujuan untuk menguji fungsi Snort apakah dapat mendeteksi serangan yang terjadi pada *web server* sesuai dengan *rules* yang ditentukan atau tidak. Setelah melakukan pendeteksian serangan, sistem ini diharapkan dapat melakukan analisis lebih detail

oleh Cowrie dan menjebak penyerang ke sistem tiruan yang dibuat menyerupai *web server* yang asli.

Kemudian, sistem ini juga diharapkan mampu mengirimkan pesan notifikasi kepada administrator yang berisi peringatan dini terkait serangan yang terjadi pada *web server* melalui Telegram dan Whatsapp menggunakan fitur API pada kedua aplikasi tersebut. Serangan yang digunakan pada simulasi ini adalah *SYN Flood Attack*, *Port Scanning*, dan *SSH Brute Force Attack*. Hasil yang diharapkan pada tahap ini adalah prototipe sistem alarm notifikasi ini dapat mendeteksi serangan sesuai dengan *rules* yang telah ditentukan, menjebak penyerang dengan sistem tiruan dari Cowrie, kemudian mengirimkan notifikasi kepada administrator melalui Telegram dan Whatsapp menggunakan BotFather dan Twilio Whatsapp API.

Selanjutnya, prototipe sistem alarm notifikasi ini akan diuji dari segi performa sistem saat menjalankan fungsinya menggunakan metode *performance testing*. Metode ini akan mengukur performa sistem berdasarkan waktu saat sistem mengeksekusi program untuk mengirim notifikasi melalui Telegram dan Whatsapp serta kapasitas memori yang digunakan pada prototipe sistem alarm notifikasi ini. Hasil yang diharapkan pada tahap ini adalah sistem dapat mengeksekusi program secara optimal dan mengirimkan notifikasi dalam rentang waktu tercepat sesuai dengan kemampuan maksimal perangkat sistem yang digunakan. Selain itu, kapasitas memori juga diperhitungkan dan digunakan seminimal mungkin karena dapat mempengaruhi performa kinerja sistem itu sendiri.

Adapun skenario pengujian yang akan dilakukan terhadap sistem yang dibangun sebagaimana tercantum pada Tabel 3.7.

Tabel 3.7 Tabel Rencana Pengujian

No	Komponen Pengujian	Skenario Pengujian	Hasil yang diharapkan	Hasil yang dicapai
1	Sistem Deteksi (Snort)	<ul style="list-style-type: none"> <li>- Penyerang menggunakan <i>SYN Flood Attack</i> untuk menurunkan kinerja <i>web server</i></li> <li>- Penyerang menggunakan <i>Brute Force Attack</i> untuk mendapatkan data krusial pengguna</li> <li>- Penyerang menggunakan serangan <i>Port Scanning</i> untuk mengetahui celah keamanan pada <i>web server</i></li> </ul>	Sistem dapat mendeteksi serangan sesuai dengan <i>rules</i> yang telah dikonfigurasi pada <i>file .conf</i> pada Snort	
2	Server tiruan <i>honeypot</i> (Cowrie)	<ul style="list-style-type: none"> <li>- Penyerang menggunakan serangan <i>Port Scanning</i> untuk mengetahui celah keamanan pada <i>web server</i></li> <li>- Penyerang memasukkan beberapa baris perintah yang dapat dieksekusi oleh Cowrie sebagai server tiruan</li> </ul>	Sistem dapat menjebak penyerang dan mengalihkan serangan ke <i>honeypot</i> (server tiruan) dan mencatat seluruh aktivitas yang dilakukan oleh penyerang selama dimonitor oleh Cowrie dan menyimpannya dalam <i>file log</i> Cowrie	
3	Fitur notifikasi serangan	Prototipe berhasil mendeteksi serangan sesuai dengan <i>rules</i> pada <i>file .conf</i>	<ul style="list-style-type: none"> <li>- Sistem dapat mengirimkan notifikasi setelah berhasil mendeteksi serangan secara <i>real-time</i> melalui aplikasi Telegram dan Whatsapp</li> <li>- Sistem dapat mengirimkan notifikasi per periode waktu tertentu sesuai dengan penjadwalan menggunakan</li> </ul>	



			Crontab	
4	Fungsi secara keseluruhan pada Prototipe Sistem Alarm Notifikasi	<ul style="list-style-type: none"> <li>- Menghitung waktu saat sistem melakukan pendeteksian serangan hingga mengirim notifikasi</li> </ul>	<ul style="list-style-type: none"> <li>- Mendapatkan hasil perhitungan waktu paling singkat mulai dari mendeteksi serangan hingga mengirimkan notifikasi serangan</li> </ul>	
		<ul style="list-style-type: none"> <li>- Menghitung kapasitas memori yang terpakai</li> </ul>	<ul style="list-style-type: none"> <li>- Mendapatkan hasil penggunaan memori seminimal mungkin saat menyalakan sistem alarm</li> </ul>	