IMPLEMENTASI *HONEYPOT COWRIE* DAN *SNORT* SEBAGAI ALAT DETEKSI SERANGAN PADA *SERVER*

Toriyansa Natanegara, Yusuf Muhyidin, Dayan Singasatia
Program Studi Teknik Informatika S1, STT Wastukancana Purwakarta
Jl Cikopak Sadang No 53 Purwakarta, Indonesia
Toriyansanatanegara07@wastukancana.ac.id

ABSTRAK

Internet telah membantu orang berbagi informasi, tetapi tidak semua informasi terbuka dan tersedia secara bebas. Ada *user* yang mencoba membuka informasi tersebut meskipun mereka tidak memiliki akses terhadap data tersebut. Oleh karena itu, diperlukan sistem keamanan jaringan untuk mendeteksi serangan dan melindungi sumber daya jaringan (hak akses, informasi/data, perangkat lunak dan/atau perangkat keras) untuk mencegah akses oleh berbagai penyusup yang mengancam sistem. Oleh karena itu penelitian ini menggunakan metode *Network Development Life Cycle* (NDLC) yang merupakan metode pengembangan atau perancangan sistem jaringan komputer dan memungkinkan sistem yang dirancang atau dikembangkan tersebut dimonitor untuk menentukan keefektifannya., Metode NDLC memiliki 6 tahapan yaitu Analisa, Desain, Simulasi Prototipe, Implementasi, Monitoring *dan Management.* Berdasarkan pengujian yang dilakukan oleh peneliti, dapat disimpulkan bahwa *Honeypot Cowrie* dan IDS *Snort* efektif dalam mendeteksi serangan yang masuk ke dalam server pada sistem operasi Ubuntu 20.04.2 LTS. Kedua sistem ini mampu mendeteksi serangan berupa DoS/DDOS, serta serangan *Bruteforce Attack* untuk *Snort* dan serangan *Bruteforce* SSH pada *Honeypot Cowrie*. Dalam pengujian tersebut, serangan-serangan tersebut dapat terdeteksi dengan cepat, memungkinkan tindakan tanggap yang cepat dalam menangani serangan DDOS dan *Bruteforce*.

Kata kunci: *Honeypot Cowrie*, *Intrusion Detection System* (IDS), *Network Development Life Cycle* (NDLC), IDS *Snort*, Serangan Jaringan

1. PENDAHULUAN

Perkembangan *internet* di Indonesia semakin meningkat. Berdasarkan hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pada periode 2022-2023, jumlah pengguna internet di Indonesia mencapai 215,63 juta orang. Angka ini mengalami peningkatan sebesar 2,67 persen dibandingkan periode sebelumnya yang mencatat 210,03 juta pengguna. Jumlah pengguna internet tersebut setara dengan 78,19 persen dari total penduduk Indonesia yang berjumlah 275,77 juta jiwa. Dalam perbandingan dengan periode riset sebelumnya, terjadi peningkatan tingkat penetrasi internet di Indonesia sebesar 1,17 persen pada tahun tersebut dibandingkan periode 2021-2022 yang sebesar 77,02 persen [1].

Internet telah membantu orang berbagi informasi, tetapi tidak semua informasi terbuka dan tersedia secara bebas. Ada *user* yang mencoba membuka informasi tersebut meskipun mereka tidak memiliki akses terhadap data tersebut. Oleh karena itu, diperlukan sistem keamanan jaringan untuk mendeteksi serangan dan melindungi sumber daya jaringan (hak akses, informasi/data, perangkat lunak dan/atau perangkat keras) untuk mencegah akses oleh berbagai penyusup yang mengancam sistem. Sistem jaringan komputer yang dirancang sedemikian rupa sehingga sumber daya dapat dibagi untuk penggunaan bersama. Saat berbagi sumber daya, keamanan dan kerahasiaan informasi adalah masalah yang paling penting. Jadi ketika serangan *cyber* terjadi, itu menyebabkan kerugian. Hal penting yang perlu diperhatikan adalah Kerahasiaan, Integritas dan Ketersediaan. yang merupakan standar untuk menerapkan kemanan dan mengevaluasi informasi [2].

Menerapkan *firewall* dapat membantu mengatasi masalah keamanan jaringan, tetapi tidak optimal untuk mendeteksi penyusup. *Honeypot* dapat menangkap penyusup dengan mengubah *port* yang bertindak sebagai jebakan/tempat di jaringan untuk menipu penyerang, mengumpulkan *log*, dan aktivitas serangan. *Cowrie* adalah jenis *honeypot* yang meniru protokol *Secure Shell* (SSH) di *server* dengan kredensial lemah dan mudah dibaca, menjebak penyerang di zona tempat *server* palsu dibuat. Alat Sistem Deteksi Intrusi (IDS) Alat deteksi intrusi lainnya adalah *snort*, yang dapat melakukan *real time traffic* dan analisis paket di jaringan *Internet* protokol (IP) [3].

Berdasarkan permasalahan yang telah dijelaskan sebelumnya, dibutuhkan sistem kemanan yang dapat mendeteksi serangan pada *server*, maka dari itu peneliti menggunakan *Honeypot Cowrie* dan *Snort* sebagai alat deteksi serangan pada server, dan penelitian ini menggunakan metode *Network Development Life Cycle* (NDLC), Tujuan dari metode NDLC ini adalah mengembangkan atau merancang sistem jaringan komputer dan memungkinkan sistem yang dibuat dapat dimonitor sehingga efektivitasnya dapat diukur.

2. TINJAUAN PUSTAKA

2.1. Jaringan Komputer

Jaringan komputer merupakan suatu sistem yang terdiri dari dua komputer atau lebih yang terkoneksi melalui media transfer data atau media komunikasi tertentu, sehingga memungkinkan mereka untuk berbagi data, aplikasi, atau perangkat keras secara saling terhubung. Setiap pengguna di Internet memiliki alamat IP atau MAC. Alamat IP atau alamat MAC ini digunakan untuk mengidentifikasi alamat tertentu atau alamat pengiriman, Jaringan komputer memiliki banyak keuntungan, seperti kemampuan untuk berbagi sumber daya, meningkatkan efisiensi kerja, memfasilitasi komunikasi yang cepat, dan menyediakan akses ke informasi dan layanan yang luas. Namun, jaringan juga rentan terhadap risiko keamanan dan harus dilindungi dengan menggunakan langkah-langkah keamanan seperti *firewall*, enkripsi data, dan autentikasi pengguna [5].

2.2. *Internet*

Internet adalah kependekan dari *Interconnected Network*, yang merujuk pada berbagai jenis koneksi komputer yang membentuk jaringan komputer global (*global computer network*) melalui berbagai saluran komunikasi seperti telepon, seluler, satelit, dan lain-lain. Kata "INTERNET" berasal dari bahasa Latin "inter," yang berarti "antara." *Internet* merupakan dunia maya dari jaringan komputer yang terdiri dari miliaran komputer di seluruh dunia. Jaringan internet ini menghubungkan berbagai jenis komputer dan jaringan yang berbeda sistem operasi dan aplikasinya, dengan memanfaatkan kemajuan alat komunikasi seperti telepon dan satelit yang menggunakan protokol standar untuk berkomunikasi.

TCP/IP adalah protokol yang digunakan untuk mengatur integrasi dan komunikasi dalam jaringan komputer ini. TCP (*Transmission Control Protocol*) bertanggung jawab untuk memastikan koneksi berfungsi dengan baik, sementara IP (*Internet Protocol*) bertugas mentransfer data dari satu komputer ke komputer lainnya. Protokol TCP/IP bekerja dengan cara memilih rute terbaik untuk transmisi data, dan jika rute tersebut tidak dapat digunakan, maka akan memilih rute alternatif, mengatur, serta mengirim paket transmisi data.

Untuk menggunakan layanan Internet, harus mendaftar ke ISP (*Internet Service Provider*) atau penyedia layanan internet. Melalui penyedia internet, komputer di seluruh dunia dapat digunakan untuk berkomunikasi dan bertukar informasi, termasuk mengirim *email*, terhubung ke komputer lain, mengirim dan menerima file, serta berpartisipasi dalam diskusi topik tertentu di *newsgroup*, dan lain sebagainya. *Internet* memberi kesempatan bagi komputer di seluruh dunia untuk saling berinteraksi dan berbagi informasi secara luas dan cepat [6]

2.3. *Honeypot*

Honeypot adalah sistem keamanan *server* yang berfungsi dengan membuat replika layanan palsu dari *server* yang ingin dilindungi. Sistem ini telah dikembangkan sebagai perangkat *open-source* yang dapat diunduh secara gratis oleh pengguna yang berminat. *Honeypot* berperan sebagai garis pertahanan terakhir jika ada keterbatasan biaya dalam mengamankan *server web*. Terdapat tiga jenis *honeypot* yang dapat disesuaikan sesuai dengan tingkat potensi ancaman yang dihadapi oleh *server*, dan pengguna memiliki fleksibilitas untuk memilih salah satu dari ketiga layanan tersebut guna diimplementasikan pada sistem jaringan mereka [7]. ketiga layanan tersebut sebagai berikut.

a. Low Interaction Honeypot
Low Interaction Honeypot adalah jenis layanan pertama dalam *honeypot*, di mana *honeypot* menciptakan replika *server* dan administrator jaringan. Dalam hal ini, pemilik *server* tetap memiliki kontrol penuh atas aktivitas intrusi yang terjadi.

b. Medium Interaction Honeypot
Medium Interaction Honeypot merupakan layanan lain yang disediakan oleh *Honeypot* yang menciptakan sistem operasi palsu untuk menarik perhatian penyerang. Dalam layanan ini, sistem mengarahkan beberapa perintah penyerang ke *honeypot* palsu, dan sebaliknya, semua informasi mengenai penyerang disimpan dan dapat dievaluasi oleh administrator jaringan. Salah satu contoh penyedia layanan ini adalah *Cowrie*.

c. High Interaction *Honeypot*
High Interaction Honeypot merupakan layanan ketiga dalam *honeypot*, di mana administrator jaringan tidak perlu lagi secara aktif memantau aktivitas penyusupan. Pada layanan ini, *server* asli direplikasi sepenuhnya, sehingga penyerang dapat menyerang *server* replika yang berisi informasi palsu. Dengan demikian, penyerang akan merasa puas karena mengakses informasi secara ilegal, namun pada kenyataannya, *server* asli tetap aman tanpa terkena dampak apapun dari serangan tersebut.

2.4. *Cowrie*

Cowrie adalah perangkat lunak yang berfungsi sebagai alat bantu untuk mengimplementasikan *honeypot* dan digunakan untuk menyamarkan layanan di *server openssh*. Perangkat lunak ini termasuk dalam kategori *honeypot Medium Interaction* yang digunakan untuk mendeteksi dan mencatat serangan *brute force* pada *server* SSH, *Telnet*, dan *OpenSSH*. *Cowrie* beroperasi dengan menggunakan konsep *redirection*, yang berarti setelah serangan *openssh* berhasil, *Cowrie* mengarahkan penyerang ke layanan *honeypot* palsu. Hal ini membuat penyerang mengira bahwa serangannya berhasil, padahal sebenarnya mereka hanya terperangkap dalam *honeypot* palsu.

Fitur utama *Cowrie* adalah kemampuan untuk melakukan *log* atau pencatatan aktivitas. Dengan fitur *logging* ini, semua aktivitas penyerang tercatat dalam sistem palsu. Hal ini memungkinkan administrator jaringan untuk mengetahui secara detail apa yang dilakukan oleh penyerang pada sistem palsu tersebut [7].

2.5. *Intrusion Detection System*

Intrusion Detection System merupakan suatu mekanisme keamanan yang diprogram untuk melakukan pengawasan terhadap titik akses, aktivitas *host*, dan upaya penyusupan. Cara paling sederhana untuk menjelaskan IDS adalah dengan menggambarkannya sebagai alat khusus yang dapat membaca dan menginterpretasi konten dari *file log* yang berasal dari *router, firewall, server*, dan perangkat jaringan lainnya. Secara lebih khusus, IDS adalah sebuah sistem yang dapat mengidentifikasi penggunaan yang tidak diizinkan dalam suatu jaringan. IDS bekerja dengan menganalisis paket data untuk menentukan apakah itu termasuk dalam upaya penyusupan atau tidak [8].

2.6. *Snort*

Snort adalah program *Network-based Intrusion Detection System*, yang berfungsi untuk mendeteksi upaya *hacking* pada sistem jaringan komputer. *Snort* memiliki sifat *open source* dengan lisensi GNU *General Public License*, yang memungkinkan penggunaannya untuk mengamankan sistem *server* tanpa biaya. Sebuah sistem IDS haruslah lintas *platform*, memiliki jejak sistem (*footprinting*) yang ringan, dan mudah dikonfigurasi oleh *administrator* sistem untuk menerapkan solusi keamanan dalam waktu singkat. Solusi implementasi tersebut mencakup berbagai *software* yang dapat dihubungkan untuk merespons situasi keamanan tertentu [8].

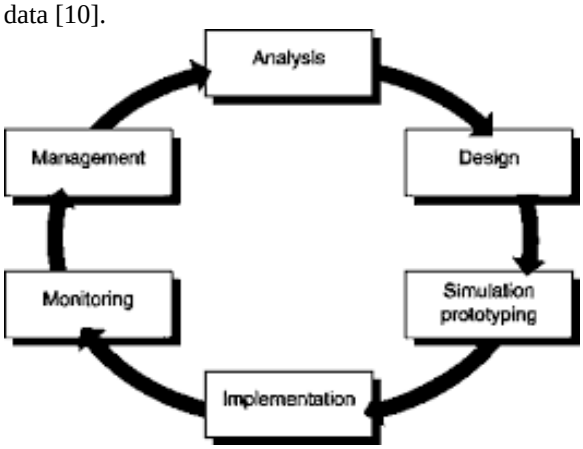
2.7. Keamanan Jaringan

Keamanan jaringan adalah aktivitas yang dilakukan pada jaringan komputer yang memengaruhi keamanan sistem dan bertentangan langsung atau tidak langsung dengan kebijakan keamanan sistem. Secara umum, kesalahan keamanan jaringan dapat diklasifikasikan: *Probing, scanning, account compromise, root compromise, packet sniffers, denial of service, trust exploits, malicious code and infrastructure attacks.*

Keamanan jaringan komputer sangat penting untuk mengontrol akses jaringan dan mencegah penyalahgunaan sumber daya jaringan yang tidak *sah*. *Administrator* jaringan mengelola tugas keamanan jaringan [9].

2.8. NDLC

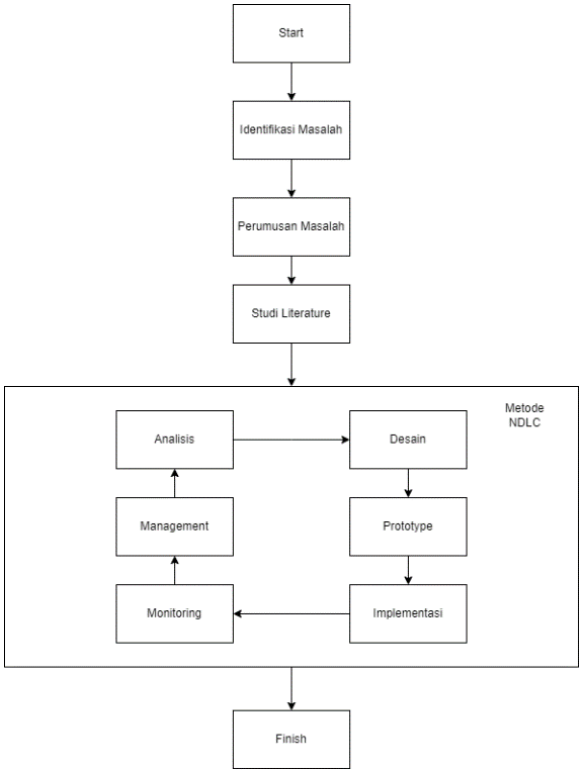
Network Development Life Cycle (NDLC) adalah sebuah metode yang digunakan dalam pengembangan atau perancangan jaringan. Metode ini terdiri dari enam tahap, yaitu Analisis, Desain, Simulasi, Implementasi, Monitoring, dan Manajemen. NDLC digunakan untuk mengembangkan atau merancang sistem jaringan komputer dan memungkinkan sistem yang dibuat dapat dimonitor sehingga efektivitasnya dapat diukur. Selain itu, NDLC juga merupakan metode yang berlandaskan pada proses pengembangan sebelumnya seperti perencanaan bisnis strategis, siklus hidup pengembangan aplikasi, dan analisis distribusi data [10].



Gambar 1. NDLC

3. METODE PENELITIAN

Metodologi *Network Development Life Cycle* (NDLC) digunakan peneliti dalam mengkaji, terdiri dari 6 tahapan yaitu *Analysis*, *Design*, *Simulation Prototype*, *Implementation*, *Monitoring*, dan *Management* tetapi untuk penelitian saat ini hanya sampai pada tahap *Implementation*, Berikut adalah diagram kerangka penelitian pada metode NDLC:



Gambar 2. Diagram Alur Penelitian

3.1. Identifikasi Masalah

Pada tahap ini, penulis mengidentifikasi masalah utama yang terkait dengan penelitian, yang dituangkan dalam latar belakang masalah, yang kemudian dilakukan untuk menemukan solusi.

3.2. Perumusan Masalah

Pada tahap ini, peneliti menelaah permasalahan yang ada terkait objek penelitian yang akan dirumuskan untuk melakukan penelitian.

3.3. Studi Literatur

Pada tahap ini penulis mengumpulkan informasi dan pengetahuan berupa teori melalui literatur, jurnal, sumber bacaan elektronik, serta menggunakan berbagai literatur yang berkaitan dengan penelitian yang diteliti untuk mendapatkan informasi pendukung.

3.4. Analisa

Pada tahap ini, penulis menganalisis permasalahan yang muncul, menganalisis kebutuhan sistem dan aplikasi pendukung lainnya. Masalah dan kebutuhan sistem (kebutuhan *Hardware* dan *Software server* dan klien) ditentukan dengan menggunakan metode analisis konten berdasarkan masalah yang diperiksa dari berbagai sumber.

3.5. Desain

Pada tahap design ini merupakan tahap lanjutan dari tahap sebelumya, dimana tahap sebelumnya telah menjelaskan analisis sistem yang berjalan yang telah memberikan informasi atau gambaran untuk sistem yang akan diusulkan, dan dilanjutkan dengan pembuatan topologi awal dan topologi usulan.

3.6. Simulasi prototipe

Dalam tahap ini penulis melakukan persiapan untuk simulasi langsung dengan menggunakan *Ubuntu Desktop* dan *Kali Linux* di *virtual box* sebagai penyerang dan bertahan *system* keamanan yang akan di bangun.

3.7. Implementasi

Dalam tahap ini penulis melakukan implementasi *system* keamanan jaringan komputer yang akan diterapkan di *Virtual Private Server*, untuk mengetahui system keamanan yang sudah dipasang yaitu *Honeypot Cowrie* dan *Intrusion Detection System* (IDS) *Snort* berhasil atau tidak dalam mendeteksi serangan yang masuk pada *server*.

4. HASIL DAN PEMBAHASAN

4.1. Analisis

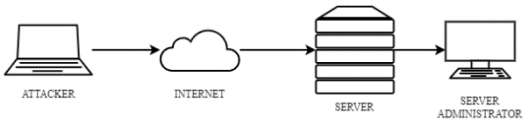
Pada tahap ini penulis melakukan Analisa konten yang menghasilkan beberapa kebutuhan sistem, analisa sistem yang berjalan, dan juga masalah yang dihadapi.

a. Identifikasi Masalah
   Ada *user* yang mencoba membuka informasi tersebut meskipun mereka tidak memiliki akses terhadap data tersebut. Oleh karena itu, diperlukan sistem keamanan jaringan untuk melindungi sumber daya jaringan (hak akses, informasi/data, perangkat lunak dan/atau perangkat keras) untuk mencegah akses oleh berbagai penyusup yang mengancam sistem. Sistem jaringan komputer yang dirancang sedemikian rupa sehingga sumber daya dapat dibagi untuk penggunaan bersama. Saat berbagi sumber daya, keamanan dan kerahasiaan informasi adalah masalah yang paling penting. Jadi ketika serangan *cyber* terjadi, itu menyebabkan kerugian.
b. Perumusan Masalah
   Berdasarkan identifikasi masalah diatas maka perumusan masalahnya adalah merancang sistem yang mampu mendeteksi serangan serangan pada *Virtual Private Server* menggunakan *honeypot Cowrie* dan *Snort*?
c. Studi Literatur
   Dalam penelitian ini, peneliti mengumpulkan data dan informasi dari berbagai jurnal, artikel ilmiah, dan sumber terpercaya lainnya yang terkait dengan topik yang sedang diteliti. Setelah data terkumpul, dilakukan analisis terhadap setiap referensi yang dikumpulkan. Analisis ini melibatkan membaca secara menyeluruh dan memahami isi dari setiap studi literatur, mencatat temuan penting, dan mengidentifikasi kerangka pemikiran serta metodologi yang digunakan dalam penelitian sebelumnya. Hal ini akan memberikan pemahaman yang lebih mendalam tentang implementasi *Honeypot Cowrie* dan *Snort*. Peneliti dapat mempelajari konsep kedua sistem keamanan server tersebut, mengidentifikasi kelebihan dan kekurangan penggunaan mereka, mengevaluasi kinerja dan kehandalan deteksi penyerangan yang dimiliki kedua sistem keamanan server tersebut, serta menemukan temuan-temuan penting dari penelitian terdahulu. guna membantu memahami konsep penelitian terkait implementasi Honeypot Cowrie dan Snort sebagai sistem keamanan pada Virtual Private Server.
d. Analisis Sistem Berjalan
   Analisis sistem yang berjalan membahas tentang jaringan internet dan jaringan komputer yang memiliki kerentanan untuk diserang DDOS dan BruteForce. Jika terus seperti ini maka akan mengganggu pelayanan jaringan internet dan jaringan komputer yang sedang berlangsung, karena belum adanya keamanan jaringan dan jaringan komputer untuk mendeteksi serangan sejak awal.
e. Analisis Sistem Usulan
   Dalam memperbaiki masalah kerentanan kemanan. Maka dibutuhkan penerapan Honeypot Cowrie dan Snort pada server untuk mencegah dan mendeteksi adanya serangan DDOS dan BruteForce ke server.

4.2. Design

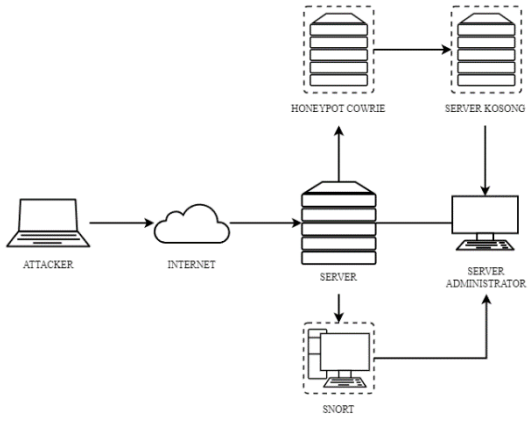
Pada tahap design ini merupakan tahap lanjutan dari tahap sebelumya, dimana tahap sebelumnya telah menjelaskan analisis sistem yang berjalan yang telah memberikan informasi atau gambaran untuk sistem yang akan diusulkan, Maka dari itu dibuatlah perancangan sistem yang akan dibangun dalam bentuk seperti gambar topologi dibawah ini

a. Design *Topology* Awal
   Design *topology* awal ini jaringan komputer belum menerapkan sistem keamanan jaringan komputer sehingga dapat diserang dengan serangan DDOS dan Bruteforce yang akan mengakibatkan jaringan komputer mengalamai perlambatan atau rusak dan juga penyerang bisa masuk ke dalam server dan mengambil data-data yang penting yang ada di dalam server.



Gambar 3. Desain *Topology* Awal

b. Design *Topology* Usulan
   Design *topology* usulan ini dimana penulis mengusulkan penerapan Honeypot Cowrie dan Intrusion Detection System (IDS) Snort. Honeypot Cowrie disini bertugas untuk mendeteksi dan mencatat serangan *bruteforce* pada *server* SSH, Telnet, dan *OpenSSH*, dan Snort bertugas untuk memantau *traffic* jaringan untuk mendeteksi tanda-tanda serangan atau aktivitas berbahaya seperti serangan DOS/DDOS, dan BruteForce. Sehingga dapat mempermudah Server Administrator dalam memantau server karena semua serangan tercatat oleh snort dan serangan BruteForce dicatat dalam log Cowrie catatan tersebut berupa IP penyerang, dan juga cara penyerang masuk ke server.



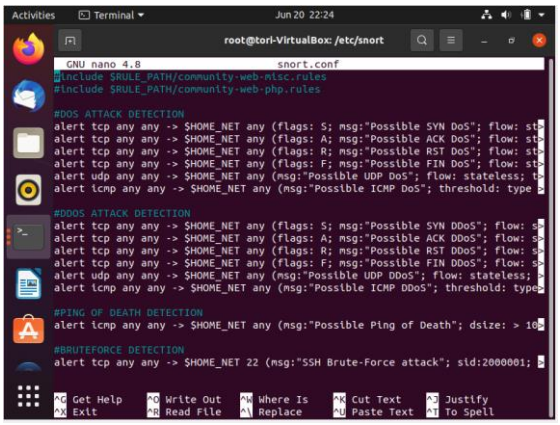
Gambar 4. Desain Topologi Usulan

4.3. *Simulation Prototype*

Dalam tahap ini penulis melakukan persiapan untuk simulasi langsung dengan menggunakan *Ubuntu Desktop* dan *Kali Linux* di *virtual box* sebagai penyerang dan bertahan *system* keamanan yang akan di bangun.

1) Instalasi Dan Konfigurasi *Honeypot Cowrie*
   a. Instal Dependensi:
      Cowrie membutuhkan beberapa dependensi untuk berjalan dengan baik. Instal dependensi yang diperlukan dengan menjalankan perintah di terminal:
      ” sudo apt install python3-virtualenv python3-dev libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind”
   b. Buat Pengguna *Cowrie*:
      Buat pengguna sistem baru untuk menjalankan *Cowrie*:
      “sudo adduser --disabled-password *Cowrie*”
   c. Unduh dan Instal *Cowrie*:
      Masuk ke direktori pengguna *Cowrie*:
      “cd /home/*Cowrie*”
      Unduh kode sumber *Cowrie* dari repositori resmi:
      “sudo -u *Cowrie* git clone https://github.com/*Cowrie*/*Cowrie*.git “
      Pindah ke direktori *Cowrie* yang baru dibuat:
      “cd *Cowrie*”
      Buat *virtual environment* untuk *Cowrie* dan aktifkan:
      “sudo -u *Cowrie* virtualenv *Cowrie*-env source *Cowrie*-env/bin/activate”
      Instal *Cowrie* dan dependensinya:
      “sudo -u *Cowrie* pip install -r requirements.txt “
      Salin file konfigurasi default:
      “cp *Cowrie*.cfg.dist *Cowrie*.cfg”
      Konfigurasi *Cowrie*:
      Edit file konfigurasi ***Cowrie*.cfg** untuk menyesuaikan pengaturan sesuai kebutuhan, seperti mengubah port, menyimpan *log*, atau menambahkan fitur tambahan.
      “nano *Cowrie*.cfg”
      Konfigurasi *Authbind* :
      Jika ingin menjalankan Cowrie di *port* 22, yang umumnya digunakan untuk SSH, Dapat menggunakan Authbind untuk mengizinkan *Cowrie* berjalan di port tersebut.
      Buat file Authbind dan berikan izin eksekusi:
      “touch /etc/authbind/byport/22 chmod 777 /etc/authbind/byport/22”
   d. Menjalankan *Cowrie*:
      Aktifkan virtual environment *Cowrie*:
      “source/home/*Cowrie*/*Cowrie*-env/bin/activate”
      Jalankan *Cowrie*:
      “sudo -u *Cowrie* ./start.sh”
   e. Instalasi Dan Konfigurasi *Snort*
      Buka terminal lalu masukan perintah “apt install snort -y”
2) Selanjutnya masukan perintah “cd /etc/snort/rules “ untuk masuk ke directory snort.

3) Masukan perintah "nano local.rules" untuk menkonfigurasi aturan snort yang akan dijalankan, berikut rules yang dipakai penulis.



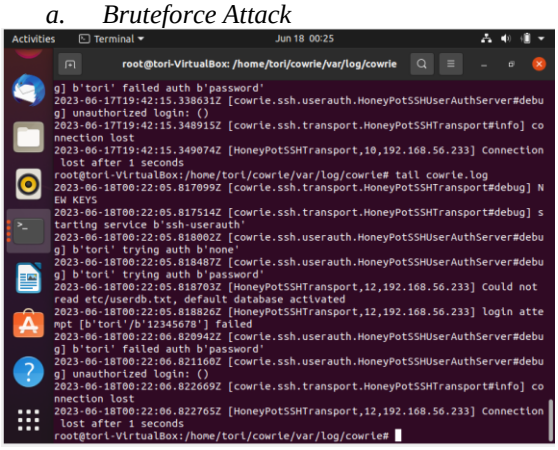
Gambar 5. Rules *Snort*

4) snort -T -c /etc/snort/snort.conf
5) snort -A console -c /etc/snort/snort.conf

4.4. *Implementation*

Dalam tahap ini penulis melakukan pengujian *system* keamanan jaringan komputer yang akan diterapkan di *Virtual Private Server*, untuk mengetahui *system* keamanan yang sudah dipasang yaitu *Honeypot Cowrie* dan *Intrusion Detection System* (IDS) *Snort* berhasil atau tidak dalam mendeteksi serangan yang masuk pada *server*.

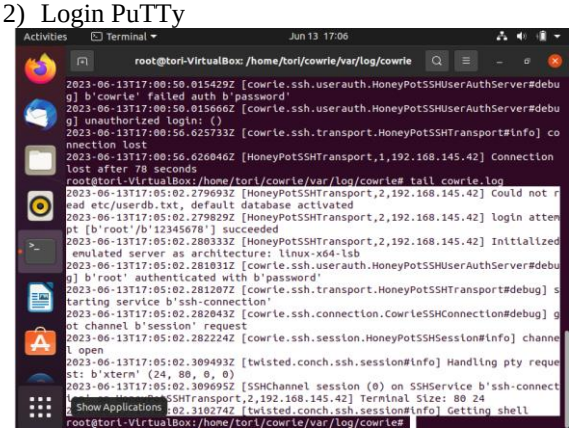
1) Skenario Penyerangan pada *Honeypot Cowrie*
   a. Bruteforce Attack



Gambar 6. *Bruteforce Attack* pada *Honeypot Cowrie*

Pada saat *Honeypot Cowrie* belum diterapkan pada *Ubuntu Desktop*, tidak dapat mendeteksi adanya serangan *Bruteforce Attack* terhadap *server*, dan setelah *Honeypot Cowrie* diterapkan pada *Ubuntu Desktop*, dapat mendeteksi adanya serangan *Bruteforce Attack* terhadap *Server* dengan adanya notifikasi "*Cowrie.ssh.userauth*" dan "*Cowrie.ssh.Transport*" serta mengetahui ip penyerang, dalam *Log Cowrie* di *Ubuntu* penyerang dengan Ip *Address* 192.168.56.233

2) Login PuTTy

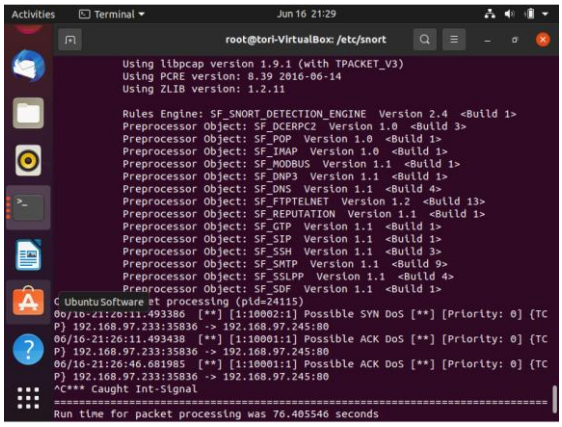


Gambar 7. *Login* PuTTy Pada *Honeypot Cowrie*

Disini terlihat aktivitas *login* yang dilakukan tercatat pada *log Cowrie* dengan beberapa kali percobaan untuk melihat perbedaan pada *log Cowrie* yang bisa menunjukan apakah percobaan *login* tersebut berhasil atau gagal, jika *login* gagal akan muncul pesan jika *password* salah "*failed auth b "password*"" jika *login* berhasil akan muncul pesan "*login attempt "succeeded*""

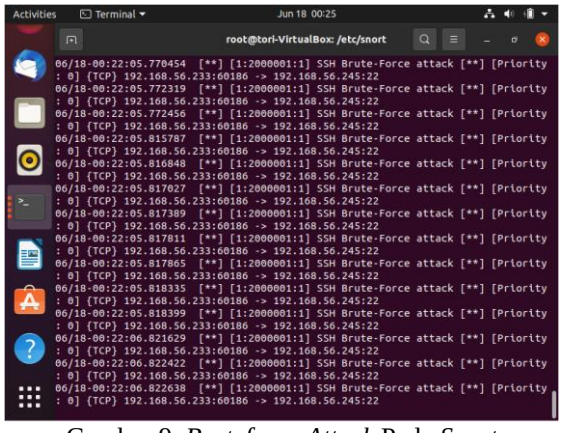
a. Skenario Penyerangan pada *Snort*

- DDOS



Gambar 8. DDOS *Attack* Pada Snort

b. Brutefoce Attack



Gambar 9. *Bruteforce Attack* Pada *Snort*

Pada saat *snort* belum diterapkan pada *Ubuntu Desktop,* tidak dapat mendeteksi adanya serangan DDOS terhadap *server*, dan setelah *snort* diterapkan pada *Ubuntu Desktop,* dapat mendeteksi adanya serangan DDOS terhadap *Server* dengan adanya notifikasi *Possible SYN DoS* dan *Possible ACK DoS*, serta mengetahui ip penyerang, dalam notifikasi *snort* di *Ubuntu* penyerang dengan Ip *Address* 192.168.97.233.

Pada saat *snort* belum diterapkan pada *Ubuntu Desktop,* tidak dapat mendeteksi adanya serangan *Bruteforce Attack* terhadap *server*, dan setelah *snort* diterapkan pada *Ubuntu Desktop,* dapat mendeteksi adanya serangan *Bruteforce Attack* terhadap *Server* dengan adanya notifikasi *SSH Brute-Force attack*, serta mengetahui ip penyerang, dalam notifikasi *snort* di *Ubuntu* penyerang dengan Ip *Address* 192.168.56.233

4.5. Hasil Pengujian Deteksi Sistem Keamanan

Tabel 1. Hasil Pengujian Deteksi Sistem Keamanan

No	*System Keamanan*	Dapat mendeteksi serangan DDOS	Dapat mendeteksi serangan *Bruteforce*
1	*Honeypot Cowrie*	Tidak	Bisa
2	*Snort*	Bisa	Bisa

Dari hasil pengujian *Honeypot Cowrie* tidak dapat mendeteksi DDOS *Attack* yang masuk ke dalam *server* karena *Honeypot Cowrie* digunakan untuk mendeteksi dan mempelajari serangan terhadap protokol SSH, beberapa serangan SSH meliputi *Bruteforce Attack, Credential Stuffing, Command Injection, Shellcode Execution,* dan *Remote Code Execution. Honeypot Cowrie* dirancang untuk menarik perhatian penyerang dan merekam aktivitas mereka, sehingga dapat memberikan wawasan tentang teknik dan alat yang digunakan oleh penyerang serta potensi kerentanan di sistem yang sebenarnya. Sedangkan *Snort* dapat mendeteksi serangan DDOS maupun serangan *Bruteforce* karena *snort* merupakan IDS yang memang diperuntukan untuk mendeteksi serangan serangan yang masuk, Tetapi *Snort IDS* hanya dapat mendeteksi dan tidak dapat menindak lanjuti serangan yang masuk.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian yang dilakukan oleh peneliti, dapat disimpulkan bahwa *Honeypot Cowrie* dan IDS *Snort* berhasil mendeteksi serangan yang masuk ke dalam *server* pada sistem operasi Ubuntu 20.04.2 LTS. Serangan yang berhasil dideteksi meliputi serangan-serangan DoS/DDOS, dan *Bruteforce Attack* untuk *snort*, dan serangan *Bruteforce* SSH pada *Honeypot Cowrie* yang telah dilakukan dimana serangan tersebut dapat diketahui dengan cepat sehingga serangan DDOS dan *Bruteforce* dapat ditangani secara cepat.

Berdasarkan hasil implementasi yang telah dilakukan untuk pengembangan lebih lanjut berikut adalah saran yang bisa diberikan yaitu dengan menambahkan *Intrusion Preventive System* IPS sehingga setelah serangan dicatat lalu bisa ditindak lanjuti dengan pemblokiran, Menggunakan IDS/*Honeypot* lainnya seperti IDS *Suricata* dan *Honeypot Dionaea*, karena kemampuan *multi-threading* yang tinggi pada ids *suricata* dan *Honeypot Dionaea* yang berfokus kepada serangan *malware* dan serangan berbasis *worm*, dan Menggunakan GNS3 untuk simulasi topologi dan pengujian

DAFTAR PUSTAKA

[1] APJI, "Survei Pengguna Internet 2023," *Asosiasi Penyelenggara Jasa Intenet Indoneia (APJI)*, 2023. [Online]. Available: www.apjii.or.id/survei. [Accessed: 20-Mar-2023].

[2] Tati Ernawati and Fikri Faiz Fadhlur Rachmat, "Keamanan Jaringan dengan Cowrie Honeypot dan Snort Inline-Mode sebagai Intrusion Prevention System," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 180–186, 2021.

[3] A. Vishnevsky and P. Klyucharev, "A survey of game-theoretic approaches to modeling honeypots," *CEUR Workshop Proc.*, vol. 2081, no. September, pp. 139–142, 2017.

[4] R. Rupiat, S. Faisal, T. Al Mudzakir, S. Arum Puspita Lestari, and P. Karawang, "Seminar Nasional Hasil Riset Prefix-RTR DETEKSI SERANGAN PERETAS MENGGUNAKAN HONEYPOT COWRIE DAN INTRUSION DETECTION SYSTEM SNORT," no. Ciastech, pp. 727–736, 2020.

[5] I. K. Astuti, "Fakultas Komputer INDAH KUSUMA ASTUTI Section 01," *Jar. Komput.*, p. 8, 2018.

[6] A. G. Gani, "Pengenalan Teknologi Internet Serta Dampaknya," *J. Sist. Inf.*, vol. 2, no. 2, pp. 71–72, 2018.

[7] W. A. Sulaksono and C. E. Suharyanto, "Implementasi Honeypot Sebagai Sistem Keamanan Jaringan Pada Virtual Private Server," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 90–95, 2020.

[8] S. S. Mohammad Affandi, "Implementasi Snort sebagai alat pendeteksi," 2020.

[9] J. D. Santoso, "Keamanan Jaringan Nirkabel Menggunakan Wireless Intrusion Detection System," *Infos*, vol. 1, no. 3, pp. 44–50, 2019.

[10] A. R. Machdi, "Analisa dan Implementasi Sistem Kemananan Jaringan Intrusion Detection System (IDS) Berbasis Mikrotik," vol. 1, no. 1, pp. 16–21, 2021.