**SIMULASI KEAMANAN SERVER DI PT. AGIVA INDONESIA MENGGUNAKAN HONEYPOT**

**DAN ELK STACK**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan

Sarjana Strata 1



**Oleh**

**R. Muhammad Sukmawan 16201146**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI DAN DESAIN**

**INSTITUT TEKNOLOGI DAN BISNIS ASIA MALANG**

**2020**

**PERSETUJUAN TUGAS AKHIR**

Judul :Simulasi Keamanan Server di PT. Agiva

Indonesia Menggunakan Honeypot dan ELK Stack

Oleh : R. Muhammad Sukmawan

NIM : 16201146

Program Studi : Teknik Informatika

Malang, 5 Desember 2020

Menyetujui

**Dosen Pembimbing**

Dosen Pembimbing, S.Kom., M.Kom

**Ketua Prodi Teknik Informatika**

Suastika Yulia Riska, S.Pd.,M.Kom

**KETERANGAN LULUS UJIAN**

Yang bertanda tangan dibawah ini menerangkan bahwa:

Nama : R. Muhammad Sukmawan

NIM : 16201146

Program Studi : Teknik Informatika

Telah lulus ujian Tugas Akhir pada tanggal 5 Desember 2020 di Institut Teknologi dan Bisnis ASIA Malang.

Malang, 5 Desember 2020

Tim Penguji

**Ketua Tim Penguji**

Dosen Penguji, M.Kom

**Penguji 1**

Dosen Pembanding, M.Kom

**Penguji 2**

Dosen Pembimbing, M. Kom

**PERNYATAAN KEASLIAN**

Yang bertanda tangan dibawah ini adalah:

Nama : R. Muhammad Sukmawan

NIM : 16201146

Tempat/Tgl Lahir : Malang, 03 Agustus 1995

Program Studi : Teknik Informatika

Alamat : Perum II PT. GPM Blok E. 048,

Kecamatan Bandar Mataram, Lampung

Tengah

Menyatakan bahwa Karya Ilmiah/ Tugas Akhir yang berjudul:

**“Simulasi Keamanan Server di PT. Agiva Indonesia Menggunakan Honeypot dan ELK Stack”**

Adalah bukan merupakan karya tulis orang lain, baik sebagian maupun keseluruhan, kecuali dalam bentuk kutipan yang disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar, saya bersedia menerima sanksi akademik.

Malang, 5 Desember 2020

Mengetahui,

**Dosen Pembimbing**

Dosen Pembimbing, S.Kom., M.Kom

**Yang menyatakan,**

R. Muhammad Sukmawan **PERNYATAAN PERSETUJUAN PUBLIKASI**

Sebagai Civitas Akademik Institut Teknologi dan Bisnis Asia Malang, saya yang bertanda tangan di bawah ini:

Nama : R. Muhammad Sukmawan

NIM : 16201146

Program Studi : Teknik Informatika

Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi dan Bisnis Asia Malang Hak Bebas Royalti atas tugas akhir saya yang berjudul:

**“Simulasi Keamanan Server di PT. Agiva Indonesia Menggunakan Honeypot dan ELK Stack”**

beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti ini, Institut Teknologi dan Bisnis Asia Malang berhak untuk menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan Tugas Akhir saya tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Malang, 5 Desember 2020

**Yang menyatakan**

Materai

Rp. 6000

R. Muhammad Sukmawan

**ABSTRAKSI**

R. Muhammad Sukmawan. 16201146

**SIMULASI KEAMANAN SERVER DI PT. AGIVA INDONESIA MENGGUNAKAN HONEYPOT DAN ELK STACK**

Teknik Informatika, Institut Teknologi dan Bisnis ASIA Malang, 2020

Kata Kunci : *Honeypot*, ELK *Stack*, *cyber*, keamanan server

(xiv + 82)

PT. Agiva Indonesia merupakan perusahaan konsultan yang bergerak di bidang teknologi dan informasi. PT. Agiva memiliki sebuah server database yang digunakan untuk menyimpan informasi klien maupun informasi perusahaan. Tetapi dalam implementasinya PT. Agiva belum memiliki sistem pendeteksi serangan *cyber* yang baik sehingga ketika *server* disusupi atau diserang oleh pihak yang tidak bertanggung jawab, pihak administrator kurang mengetahui informasi mengenai serangan *cyber*, seperti siapa yang menyerang, bagaimana mereka menyerang, dan kapan serangan dilakukan.

Salah satu upaya yang dapat dilakukan adalah menambahkan keamanan di *server*, salah satunya dengan menggunakan *honeypot*. *Honeypot* seolah-olah menjadi sistem yang berhasil disusupi oleh penyerang, padahal penyerang tidak masuk ke sistem yang sebenarnya, tetapi masuk ke dalam sistem yang palsu, sehingga *honeypot* akan mencatat segala *log* illegal dari penyerang. Tetapi dalam penerapannya log yang dihasilkan oleh honeypot masih berupa huruf dan angka sehingga ditambahkan sistem ELK *Stack* yang berguna untuk memvisualisasikan *log* dari *honeypot*.

Setelah dilakukan pengujian, *honeypot cowrie* dan *dionaea* mampu mendeteksi adanya serangan yang ditujukan ke *server*. Selanjutnya sistem ELK mampu memvisualisasikan *log* dari *honeypot cowrie* secara *real-time*, sedangkan *log* yang berasal dari *honeypot dionaea* masih mendapati waktu *delay* pada pengiriman dengan rata-rata 3,75 detik.

# Daftar Pustaka (2014 – 2019)

# ABSTRACT

# R. Muhammad Sukmawan. 16201146

**SERVER SAFETY SIMULATION IN PT. AGIVA INDONESIA USING HONEYPOT AND ELK STACK**

Informatics, Institute of Technology and Business ASIA Malang, 2020

Keywords: Honeypot, ELK Stack, cyber attacks, server security

(xiv + 82)

PT. Agiva Indonesia is a consulting company engaged in technology and information. PT. Agiva has a database server that is used to store client information and company information. But in its implementation, PT. Agiva does not yet have a good cyber attack detection system, so when servers are compromised or attacked by irresponsible parties, administrators do not know information about cyber attacks, such as who attacked them, how they attacked, and when the attacks were carried out.

One effort that can be done is to add security on the server, one of which is by using Honeypot. The honeypot seems to be a system that has been infiltrated by the attacker, even though the attacker does not enter the real system, but has entered into a fake system, so the honeypot will record all illegal logs from the attacker. But in practice the logs generated by the honeypot are still letters and numbers, so the authors add the ELK Stack system which is useful for visualizing logs from the honeypot.

After testing, the cowrie and dionaea honeypots were able to detect an attack aimed at the server. Furthermore, the ELK system is able to visualize the logs from the cowrie honeypot in real-time, while the logs from the dionaea honeypot still find the delay time on delivery with an average of 3.75 seconds.

Bibliography (2014 - 2019)

**KATA PENGANTAR**

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah memberikan rahmat, hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan mengambil judul “Simulasi Keamanan Server di PT. Agiva Indonesia Menggunakan Honeypot dan ELK Stack”.

Penyusunan laporan tugas akhir ini merupakan tugas mandiri yang harus diselesaikan oleh mahasiswa Institut Teknologi dan Bisnis Asia Malang program studi Teknik Informatika yang merupakan bagian dari prasyarat kelulusan program kesarjanaan.

Pada kesempatan ini penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu terselesainya tugas akhir sampai dengan penyusunan laporan, yaitu :

1. Ibu Risa Santoso, B.A., M. Ed., selaku Rektor Institut Teknologi dan Bisnis Asia Malang.
2. Bapak Handry Rochmad Dwi Happy, S.Sn, M.Sn Dekan Fakultas Teknologi dan Desain.
3. Bapak Suastika Yulia Riska, S.Pd.,M.Kom selaku Ketua Prodi Informatika.
4. Ibu Fransiska Sisilia Mukti, ST., MT, selaku Dosen Pembimbing.
5. Ibu Lilis Widayanti Spd., Mpd., selaku Dosen Wali.
6. Keluarga tercinta yang telah memberikan kasih sayang, dan dukungan.

Kepada semuanya yang telah membantu penulis dalam menyelesaikan tugas akhir ini, semoga mendapat balasan dengan berkat dan karunia yang berlimpah dari Allah SWT.

Penulis menyadari bahwa laporan tugas akhir ini jauh dari sempurna, oleh karena itu penulis selalu mengharapkan saran dan kritik yang bersifat membangun dari segenap pembaca. Akhirnya, semoga laporan tugas akhir ini dapat berguna dan bermanfaat terutama bagi pihak-pihak yang tertarik untuk mengkaji dan mengembangkannya.

Malang, 5 Desember 2020

Penulis

**DAFTAR ISI**

**Halaman**

# Halaman Sampul i

# Persetujuan Tugas Akhir ii

# Keterangan Lulus Ujian iii

# Pernyataan Keaslian iv

# Pernyataan Persetujuan Publikasi v

# Abstraksi vi

# Abstract vii

# Kata Pengantar viii

# Daftar Isi ix

# Daftar Gambar xi

# Daftar Tabel xii

# Daftar Kode Program xiii

# BAB I PENDAHULUAN 1

# Latar Belakang Masalah 1

# Rumusan Masalah 3

# Batasan Masalah 3

# Tujuan Dan Manfaat 3

# Tujuan Penelitian 3

# Manfaat Bagi Penulis 3

# Manfaat bagi PT. Agiva Indonesia 3

# Manfaat bagi Institut Asia Malang 3

# Metodologi Penelitian 4

# Sistematika Penulisan 5

# BAB II LANDASAN TEORI 7

# Jaringan Komputer 7

# Protokol Jaringan 7

# Jenis-jenis Protokok Jaringan 7

# Domain Name System (DNS) 10

# IP Address 11

# Kelas IP Address 11

# Network Address dan Broadcast Address 13

# Subnetting 14

# Keamanan Jaringan 15

**Halaman**

# Klasifikasi keamanan jaringan 16

# Jenis-Jenis Serangan 17

# Honeypot 20

# Kategori Honeypot 21

# Klasifikasi Honeypot 22

# Cowrie 23

# Dionaea 24

# Penempatan Honeypot 25

# ELK Stack 26

# Elasticsearch 26

# Logstash 27

# Kibana 28

# Filebeat 29

# SNORT 30

# BAB III PEMBAHASAN 33

# Keamanan Server pada PT. Agiva Indonesia 33

# Topologi Jaringan pada PT. Agiva Indonesia 33

# Analisa Sistem Keamanan Server di PT. Agiva Indonesia 34

# Perancangan Simulasi Keamanan Server di PT. Agiva Indonesia 36

# Perancangan Alur Sistem Kerja Honeypot 36

# Perancangan Alur Sistem Kerja ELK Stack 38

# Perancangan Pengujian Sistem Simulasi Keamanan

# Server di PT. Agiva Indonesia 39

# Rancangan Pengujian Sistem Honeypot 43

# Rancangan Pengujian Sistem ELK 45

# Rancangan Pengujian Brute Force SSH 47

# Rancangan Pengujian Remote Server 49

# Rancangan Eksploitasi Layanan 50

# Rancangan Pengujian Visualisasi Kibana 51

# BAB IV IMPLEMENTASI DAN PENGUJIAN 54

# Spesifikasi Implementasi 57

# Konfigurasi Server Utama 57

# Konfigurasi Snort 58

# Konfigurasi Filebeat 60

**Halaman**

# Konfigurasi Server Honeypot 62

# Konfigurasi IP Statis 62

# Konfigurasi Dionaea 63

# Konfigurasi Cowrie 64

# Konfigurasi Filebeat 65

# Konfigurasi Port Forwarding 66

# Konfigurasi Server ELK Stack 67

# Konfigurasi IP Statis 67

# Instalasi Oracle Java 68

# Konfigurasi Elasticsearch 69

# Konfigurasi Logstash 70

# Konfigurasi Kibana 71

# Pengujian Sistem 72

# Pengujian Sistem Honeypot 73

# Pengujian Sistem ELK 74

# Pengujian Brute Force 75

# Pengujian Remote SSH 76

# Pengujian Remote Telnet 77

# Pengujian Remote FTP 78

# Pengujian Eksploitasi MS04\_011 79

# Pengujian Eksploitasi MS03\_026 80

# Pengujian Visualisasi Kibana 80

# BAB V PENUTUP 81

# Kesimpulan 81

# Saran 82

# Daftar Pustaka 83

# Riwayat Hidup 85

**DAFTAR GAMBAR**

**Gambar Halaman**

# Perbandingan TCP/IP dengan OSI 8

# IP Address Kelas A 12

# IP Address Kelas B 12

# IP Address Kelas C 13

# IP Address Kelas D 13

# IP Address Kelas E 13

# Contoh log pada Honeypot 19

# Honeypot Eksternal 23

# Honeypot Internal 24

# Honeypot pada DMZ 24

# Proses ELK Stack 25

# Contoh Tampilan Pada Kibana 26

# Topologi Jaringan PT. Agiva Indonesia 31

# Desain Sistem Keamanan Server yang Akan Dibangun

# di PT. Agiva Indonesia 32

# Cara Kerja Sistem Honeypot & ELK Stack .. 34

# Perancangan Alur Sistem Honeypot 35

# Perancangan Alur Sistem ELK Stack 36

# Rancangan Pengujian Sistem Honeypot 37

# Rancangan Pengujian Sistem ELK 37

# Rancangan Pengujian Brute Force SSH 38

# Rancangan Pengujian Remote Server Honeypot 38

# Rancangan Pengujian Eksploitasi Layanan 40

# Konfigurasi local.rules 51

# Konfigurasi Filebeat.yml Pada Server Utama 52

# Konfigurasi IP Statis Server Honeypot 52

# Konfigurasi Filebeat Pada Server Honeypot.yml 54

# Konfigurasi IP statis server ELK 55

# Tes elasticsearch pada web browser 57

# Konfigurasi input Logstash 58

# Konfigurasi filter cowrie logstash 59

# Konfigurasi filter dionaea logstash 62

**DAFTAR TABEL**

**Tabel Halaman**

# Kelas IP Address A-E 11

# Perbedaan Honeypot dengan konsep keamanan 21

# Perbandingan Honeypot 22

# Spesifikasi perangkat pada VMware 31

# Tabel deteksi serangan dan selisih waktu 64

**DAFTAR KODE PROGRAM**

**Kode Halaman**

# Konfigurasi Snort Pada Server Utama 58

# Konfigurasi Filebeat 59

# Konfigurasi IP Statis Server Honeypot 60

# Konfigurasi Dionaea 61

# Konfigurasi Filebeat Pada Server Honeypot 62

# Konfigurasi Port Forwarding 63

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Salah satu aspek yang penting untuk diperhatikan dalam membangun sebuah infrastruktur jaringan adalah keamanan jaringan. Keamanan jaringan yaitu proses pencegahan yang dilakukan oleh penyerang untuk terhubung ke dalam jaringan komputer melalui akses yang tidak sah, atau penggunaan secara illegal dari komputer dan jaringan. Keamanan jaringan diperlukan untuk menghindari ancaman-ancaman dari pihak yang tidak bertanggung jawab, yang bertujuan untuk menyerang, merusak, dan bahkan mengambil data-data pribadi yang menyebabkan kerugian bagi pemiliknya **(Agustino & Priyoatmojo, 2017).**

Salah satu masalah keamanan yang sangat signifikan pada jaringan komputer adalah serangan yang mengganggu kinerja server. Jenis serangan yang biasa dilakukan antara lain *bruteforce, scanning*, Ddos (*Denial of Service Attacks*), serta malware **(Agustino & Priyoatmojo, 2017)**. Serangan-serangan ini umum terjadi pada sebuah server, termasuk pada PT. Agiva Indonesia. PT. Agiva Indonesia merupakan perusahaan konsultan yang bergerak dalam bidang teknologi informasi. Belum adanya sistem pendeteksi serangan *cyber* yang baik, merupakan salah satu masalah yang saat ini sedang dihadapi oleh administrator server PT. Agiva Indonesia. Sehingga pihak administrator kurang mengetahui informasi mengenai serangan *cyber* yang ditujukan ke server, seperti siapa yang menyerang, bagaimana mereka menyerang, dan kapan serangan dilakukan.

Salah satu upaya yang dapat dilakukan adalah menambahkan keamanan di server, salah satunya dengan menggunakan *Honeypot*. *Honeypot* merupakan sebuah sistem atau komputer yang sengaja dijadikan umpan untuk menjadi target serangan dari penyerang **(Mardiyanto & Indriyani, 2016)**. *Honeypot* seolah-olah menjadi sistem yang berhasil disusupi oleh penyerang, padahal penyerang tidak masuk ke sistem yang sebenarnya, tetapi masuk ke dalam sistem yang palsu, sehingga *honeypot* akan mencatat segala log illegal dari penyerang.

Dalam penerapannya, log yang dihasilkan dari *honeypot* hanya berupa huruf dan angka, sehingga akan menyulitkan dalam menganalisis log tersebut. Untuk memudahkan administrator dalam menganalisis log, maka diusulkan adanya sistem visualisasi dengan menggunakan ELK *stack*. ELK *stack* merupakan gabungan komponen yaitu elasticsearch, logstash, dan kibana, yang berguna untuk mengumpulkan log yang akan divisualisasikan dalam bentuk grafik **(Arifin & Susilowati, 2018)**. Kolaborasi antara Honeypot dan ELK stack dapat menjadi sebuah solusi sistem keamanan dalam mendeteksi serangan sekaligus memberikan visualisasi kepada administrator. Hasil akhir dari penelitian ini diharapkan mampu membantu pihak administrator server PT. Agiva Indonesia dalam mengevaluasi tingkat keamanan untuk server yang dibangun.

* 1. **Rumusan Masalah**

Berdasarkan uraian latar belakang, dapat dirumuskan sebuah masalah yaitu bagaimana mensimulasikan keamanan server di PT. Agiva Indonesia menggunakan *honeypot* dan ELK *stack*.

* 1. **Batasan Masalah**

Pembahasan dalam penelitian ini cukup luas, untuk itu penelitian perlu dibatasi agar tidak menyimpang dari tujuan penelitian. Batasan masalah pada penelitian ini sebagai berikut:

1. *Honeypot* yang dibangun bersifat *low interaction*.
2. Penelitian sistem keamanan ini hanya untuk mendeteksi adanya serangan, tidak untuk menanggulangi adanya serangan.
3. Simulasi yang dilakukan hanya bentuk rekomendasi tanpa adanya keharusan untuk diimplementasikan di PT. Agiva Indonesia.
4. Peneliltian ini menggunakan beberapa software *freeware* untuk membangun sistem seperti Cowrie, Dionaea, Filebeat, Logstash, Elasticsearch, Kibana, dan Snort.
5. Pengujian serangan dilakukan menggunakan IP lokal dan hanya memakai 1 IP address penyerang dengan menggunakan protokol 22, 21, 23, 445, 135.
   1. **Tujuan Dan Manfaat Penelitian**
      1. **Tujuan Penelitian**

Tujuan dari penelitian ini yaitu untuk membantu pihak PT. Agiva Indonesia dalam mendeteksi serangan *cyber* dan sistem monitoring keamanan server melalui sebuah sistem yang mampu mendeteksi sebuah serangan menggunakan *honeypot*, dan kemudian memvisualisasikannya pada ELK *stack*.

* + 1. **Manfaat Bagi Penulis**

Manfaat yang dapat diambil bagi penulis ialah:

1. Mengaplikasikan disiplin ilmu yang telah diperoleh selama belajar di Institut Asia Malang Program Studi Informatika.
2. Dapat mengasah otak dalam berfikir secara cepat dan teliti untuk mencari penyelesaian masalah.
   * 1. **Manfaat Bagi PT. Agiva Indonesia**
3. Meningkatkan sistem keamanan server dalam mendeteksi ancaman.
4. Mempermudah dalam memonitoring sistem keamanan.
   * 1. **Manfaat Bagi Institut Asia**
5. Dapat mengukur sejauh mana keberhasilan proses belajar mengajar di dalam kelas, dan capaian materinya.
6. Menjadi bahan kajian yang dapat dikembangkan dikemudian hari.
   1. **Metodologi Penelitian**

Untuk mendukung penyelesaian laporan tugas akhir ini digunakan beberapa metodologi penelitian yaitu:

1. Studi Literatur

Dengan mempelajari buku-buku referensi dan jurnal yang berkaitan dengan permasalahan penelitian yang diangkat serta mencari solusi yang terbaik. Topik bahasan utama yang dibutuhkan diantarannya adalah *honeypot* dan ELK *stack*.

1. Analisa dan Desain Sistem (Perancangan)

Berdasarkan studi literatur yang dilakukan, dapat ditentukan bagaimana desain dari sistem yang akan dibuat. Pada tahapan ini, dilakukan dengan perancangan sistem yang akan dibangun.

1. Implementasi Sistem

Pada tahapan ini dilakukan impementasi sesuai dengan perancangan yang telah dibuat, dimulai dari instalasi perangkat lunak yang akan dipakai dan konfigurasi pada sistem yang dibangun.

1. Pengujian Sistem

Pengujian ini dilakukan untuk membuktikan sejauh mana kesesuaian sistem yang dibangun dapat bekerja dengan baik sesuai dengan tujuan. Hal yang diuji antara lain apakah sistem mampu mendeteksi dan memberikan informasi serangan, hasil log yang ditangkap oleh *honeypot*, dan visualisasi log dari *honeypot* yang ditampilkan oleh ELK *stack*.

* 1. **Sistematika Penulisan**

Agar pembahasan bersifat komunikatif dan mudah dipahami maka dalam laporan penelitian ini disusun dengan sistematika yang terstruktur. Sistematika penulisan dalam laporan penelitian ini dibagi menjadi 5 bab yang dijelaskan sebagai berikut:

**BAB I Pendahuluan**

Dalam bab ini menjelaskan secara umum penyusunan laporan tugas akhir ini yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, metodologi penulisan dan sistematika penulisan laporan.

**BAB II Landasan Teori**

Berisi tentang uraian-uraian secara teoritis disiplin ilmu yang mendukung pokok bahasan laporan yaitu mengenai keamanan jaringan, *honeypot*, dan ELK *stack*.

**BAB III Pembahasan**

Bab ini mencakup analisis masalah yang terjadi di PT. Agiva Indonesia termasuk topologi jaringan yang ada. Selain itu dijelaskan beberapa perancangan sistem yang akan dibuat, perancangan tersebut meliputi rancangan sistem keamanan dan skenario pengujian.

**BAB IV Implementasi dan Pengujian**

Bab ini menjelaskan spesifikasi hardware dan sofrware yang digunakan dalam membuat simulasi sistem keamanan jaringan di PT. Agiva Indonesia. Selanjutnya menjelaskan implementasi dan pengujian terhadap sistem. Tahap ini dilakukan setelah sistem dirancang dan dianalisis pada perancangan sistem.

**BAB V Penutup**

Pada bab ini berisi kesimpulan-kesimpulan yang didapat dalam analisa permasalahan, serta saran-saran untuk perbaikan di kemudian hari.

**BAB II**

**LANDASAN TEORI**

* 1. **Jaringan** **Komputer**

Jaringan Komputer dapat diartikan sebagai suatu himpunan sejumlah komputer yang saling terkoneksi. Dua buah komputer dikatakan membentuk suatu jaringan komputer disaat keduanya dapat saling bertukar informasi **(Supriyadi & Gartina, 2015).** Dalam jaringan komputer, terdapat jenis-jenis jaringan yang berbeda, diantaranya **(Wongkar & Sinsuw, 2015)**:

* 1. Local Area Nework (LAN)

LAN adalah singkatan dari *local area network*. Jenis jaringan LAN ini sangat sering ditemui di warnet-warnet, kampus, sekolah ataupun perkantoran yang membutuhkan hubungan atau koneksi antara dua komputer atau lebih dalam suatu ruangan. Jaringan LAN juga merupakan jaringan yang sangat di pengaruhi oleh topologi jaringannya.

* 1. Metropolitan Area Network (MAN)

MAN singkatan dari *metropolitan area network*. Jenis jaringan komputer MAN ini adalah suatu jaringan komputer dalam suatu kota dengan transfer data berkecepatan tinggi yang menghubungkan suatu lokasi seperti sekolah, kampus, perkantoran dan pemerintahan. Sebenarnya jaringan MAN ini adalah gabungan dari beberapa jaringan LAN. Jangkauan dari jaringan MAN ini bisa mencapai 10 - 50 kilo meter.

* 1. Personal Area Network (PAN)

PAN adalah singkatan dari *personal area network*. Jenis jaringan komputer PAN adalah hubungan antara dua atau lebih sistem komputer yang berjarak tidak terlalu jauh. Biasanya jenis jaringan yang satu ini hanya berjarak 4 sampai 6 meter saja. Jenis jaringan ini sangat sering digunakan, contohnya menghubungkan HP dengan komputer.

* 1. Wide Area Network (WAN)

WAN singkatan dari *wide area network*. WAN adalah jenis jaringan komputer yang mencakup area yang cukup besar. contohnya adalah jaringan yang menghubugkan suatu wilayah atau suatu negara dengan negara lainnya.

* 1. Wireless LAN (WLAN)

Wireless LAN atau kadang disingkat dengan WLAN adalah sebuah sistem komunikasi data yang fleksibel yang dapat diaplikasikan sebagai ekstensi ataupun sebagai alternatif pengganti untuk jaringan LAN kabel. Wireless LAN menggunakan teknologi frekuensi radio, mengirim dan menerima data melalui media udara, dengan meminimalisasi kebutuhan akan sambungan kabel. Dengan begitu, wireless LAN telah dapat mengkombinasikan antara konektivitas data dengan mobilitas user. Wireless LAN adalah sebuah alternatif dimana untuk alternatif LAN kabel sulit atau tidak mungkin dibangun.

* 1. **Protokol** **Jaringan**

Protokol jaringan merupakan protokol yang mengatur komunikasi data di dalam jaringan data tersebut. Setiap jaringan data publik dapat secara bebas mengatur sendiri protokol jaringan yang akan dipakai. Hal ini tergantung pada topologi dan kondisi jaringan setempat. Pada dasarnya protokol dalam jaringan berfungsi untuk **(Prastiyanto, 2015)** :

* + 1. Mengatur penyambungan dan pemutusan hubungan antara pemakai jaringan.
    2. Pengontrol kesalahan dan pengatur prioritas pemakai jaringan.
    3. Sinkronisasi proses serta pengatur aliran data (*routing*).
    4. Menghindari duplikasi.
    5. Memaksimalkan dan mengefisienkan kerja jaringan.
    6. **Jenis-jenis Protokol Jaringan**

Terdapat beberapa macam protokol dalam jaringan sesuai dengan fungsi dan kegunaannya masing-masing. Berikut merupakan jenis-jenis protokol yang sering digunakan:

1. *Telecommunication Network* (Telnet)

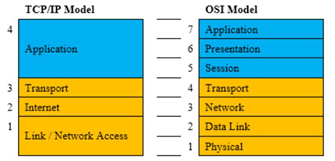
Telnet merupakan sebuah protokol jaringan yang digunakan di koneksi Internet atau *local area network*. Telnet dikembangkan pada 1969 dan distandarisasi sebagai IETF STD 8, salah satu standar Internet pertama.

1. *Secure Shell* (SSH)

SSH (*secure shell*) adalah protokol jaringan yang berada di lapisan aplikasi pada protokol TCP/IP, memfasilitasi sistem komunikasi yang aman diantara dua sistem yang menggunakan arsitektur klien server dengan menyediakan kerahasiaan dan integritas data melalui teknik enkripsi dan dekripsi yang dilakukan secara otomatis didalam koneksinya. Protokol ini mempunyai kelebihan dibanding protokol yang sejenis seperti Telnet dan FTP, karena SSH memiliki sistem otentikasi, otorisasi, dan ekripsinya sendiri. Dengan begitu keamanan sebuah sesi komunikasi melalui bantuan SSH ini menjadi lebih terjamin.

1. TCP/IP (*Transmission Control Protocol/Internet Protocol*)

TCP/IP merupakan standar komunikasi data yang digunakan dalam proses tukar-menukar data dari satu komputer ke komputer lain. TCP/IP merupakan jaringan terbuka yang bersifat independen terhadap mekanisme transport pada jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (software) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP *stack*. Pada TCP/IP terdapat beberapa protokol sub yang menangani masalah komunikasi antar komputer.



**Gambar 2.1** Perbandingan TCP/IP dengan OSI **(Prastiyanto, 2015)**

1. *User Datagram Protocol* (UDP)

UDP adalah kependekan dari *User Datagram Protocol* merupakan bagian dari *internet protocol*. Dengan UDP, aplikasi komputer dapat mengirimkan pesan kepada komputer lain dalam jaringan lain tanpa melakukan komunikasi awal. UDP bersifat unreliable. Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan *acknowledgment*. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah host dalam jaringan yang menggunakan TCP/IP. Header UDP berisi *field source process identification* dan *destination process identification*. UDP melakukan komunikasi secara sederhana dengan mekanisme yang sangat minimal. Ada proses *checksum* untuk menjaga integritas data.

1. *Internet Control Message Protocol* (ICMP)

ICMP adalah salah satu protokol inti. ICMP berbeda tujuan dengan TCP dan UDP, ICMP tidak digunakan secara langsung oleh aplikasi jaringan milik pengguna. salah satu pengecualian adalah aplikasi ping yang mengirim pesan ICMP *Echo Request* (dan menerima *Echo Reply*) untuk menentukan apakah komputer tujuan dapat dijangkau dan berapa lama paket yang dikirimkan dibalas oleh komputer tujuan. protokol internet. ICMP utamanya digunakan oleh sistem operasi komputer jaringan untuk mengirim pesan kesalahan yang menyatakan, sebagai contoh, bahwa komputer tujuan tidak bisa dijangkau.

1. *File Transfer Protocol* (FTP)

FTP adalah sebuah protokol internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*file*) komputer antar mesin-mesin dalam sebuah *internetwork*. FTP atau *Transmission Control Protocol* (TCP) untuk komunikasi data antara klien dan server, sehingga diantara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum transfer data dimulai. FTP hanya menggunakan metode autentikasi standar, yakni menggunakan Username dan pasword-nya yang dikirim dalam bentuk tidak terenkripsi. Pengguana terdaftar dapat menggunakan username dan password-nya untuk mengakses, men-download, dan meng-upload berkas-berkas yang dikehendaki. Umumnya, para pengguna daftar memiliki akses penuh terdapat berapa direktori.

Pengguna yang belum terdaftar dapat juga menggunakan metode *anonymous login*, yakni dengan menggunakan nama pengguna *anonymous* & password yang diisi dengan menggunakan alamat e-mail. Sebuah server FTP diakses dengan menggunakan *Universal Resource Identifier* (URL) dengan menggunakan format ftp://namaserver. Klien FTP dapat menghubungi server FTP dengan membuka URL tersebut. Tujuan FTP server adalah sebagai berikut :

1. Untuk men-sharing data.
2. Untuk menyediakan *indirect* atau *implicit remote computer*.
3. Untuk menyediakan tempat penyimpanan bagi User.
4. Untuk menyediakan transfer data yang *reliable* dan efisien.
   * 1. ***Domain Name System* (DNS)**

*Domain Name System* (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang mengunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti web browser atau e-mail, dimana DNS membantu memetakan host name sebuah komputer ke IP *address*. Selain digunakan di internet, DNS juga dapat di implementasikan ke *private network* atau intranet dimana DNS memiliki keunggulan seperti :

* 1. Mudah, DNS sangat mudah karena user tidak lagi direpotkan untuk mengingat IP *address* sebuah komputer cukup *host name* (nama Komputer).
  2. Konsisten, IP *address* sebuah komputer bisa berubah tapi host name tidak berubah.
  3. Simple, user hanya menggunakan satu nama domain untuk mencari baik di Internet maupun di Intranet.

Domain Name System (DNS) sangat membantu manusia dalam mengakses sebuah komputer dalam jaringam yang besar seperti internet. Manusia akan lebih mudah mengingat sebuah nama dari pada deretan angka-angka. Misalnya http://google.com/ jauh lebih mudah diingat dari pada http://74.125.224.72.

* 1. **IP** ***Address***

IP *Address* adalah sederetan angka biner 32 bit yang terbagi menjadi 4 kelompok, masing-masing kelompok terdiri atas biner 8 bit yang dipisahkan dengan tanda titik atau disebut *dotted-decimal format*. IP address merupakan identitas setiap *host* pada jaringan internet **(Kustanto & Saputro, 2015).** Contoh :

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx (label huruf x)

11000000.10101000.00000000.00000001 (jika x bernilai 1 atau 0)

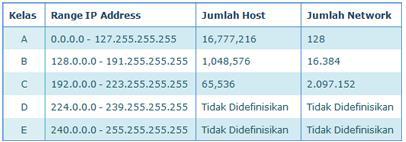
192.168.0.1 (Konversi biner ke desimal)

IP *Address* yang digunakan untuk keperluan internet disebut sebagai IP *Address public*. Sedangkan IP *Address* yang digunakan untuk keperluan LAN/Intranet disebut sebagai IP *Address private*.

* + 1. **Kelas IP *Address***

Dalam pengelolaannya IP *Address* dibagi menjadi 5 kelas, yaitu kelas A, B, C, D, E, dimana masing-masing kelas akan mempunyai fungsi bit yang berbeda-beda. Untuk kelas A, B, C disebut sebagai IP *Address Unicast* yang digunakan untuk keperlulan umum. Pengiriman paket ke alamat *unicast* hanya diterima oleh sebuah Host yang dituju.

**Tabel 2.1** Kelas IP Address A-E **(Kustanto & Saputro, 2015)**



Untuk IP *Address* kelas D disebut sebagai IP *Address Multicast*. Pengiriman paket ke alamat *multicast* akan diterima oleh sejumlah *host* penerima. Salah satu aplikasi yang memanfaatkan IP *Address* kelas D adalah *real time video conferencing*. Kemudian IP *Address* kelas E digunakan untuk keperluan riset. Pembagian ke 5 kelas dari IP Address ditunjukkan pada tabel 2.1 **(Kustanto & Saputro, 2015).**

* 1. Kelas A

Kelas A mempunyai 8 bit *network*, 24 bit *host*, dan biasa digunakan untuk jaringan dengan skala besar. Bit pertama di dalam IP *address* kelas A selalu diset dengan nilai 0 (nol). Bit kedua sampai bit ke delapan merupakan sebuah *Network* ID. Dua puluh empat bit sisanya (atau tiga oktet terakhir) merepresentasikan *Host* ID. Dengan jumlah *Host* ID sampai 24 bits, artinya kelas A memiliki 16,777,214 *host*.



**Gambar 2.2** IP *Address* Kelas A **(Kustanto & Saputro, 2015)**

* 1. Kelas B

Kelas B mempunyai 16 bit *network*, 16 bit *host* dan biasa digunakan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B biasanya berupa bilangan biner 10. Empat belas bit berikutnya merupakan *Network* ID. Sisa 16 bit merepresentasikan *Host* ID. IP address kelas B memiliki 65,534 *Host*.



**Gambar 2.3** IP *Address* Kelas B **(Kustanto & Saputro, 2015)**

* 1. Kelas C

Kelas C mempunyai 24 bit *network*, 8 bit *host* dan digunakan untuk jaringan berskala kecil. Tiga bit pertama bernilai biner 110. Kemudian 21 bit selanjutnya merupakan *network* *identifier*. Dengan begitu IP *address* kelas C memiliki 254 *host* untuk setiap *network*-nya.



**Gambar 2.4** IP *Address* Kelas C **(Kustanto & Saputro, 2015)**

* 1. Kelas D

Dalam jaringan kelas D semua bit digunakan untuk keperluan *multicasting*. Bit yang bernilai paling tinggi berada pada bit yang paling kiri dan selalu bernilai 1110.



**Gambar 2.5** IP *Address* Kelas D **(Kustanto & Saputro, 2015)**

* 1. Kelas E

Dalam kelas E bit yang nilainya paling tinggi berada pada bit paling kiri dan selalu bernilai 11110.



**Gambar 2.6** IP *Address* Kelas E **(Kustanto & Saputro, 2015)**

* + 1. **Network Address dan Broadcast Address**

Dalam sebuah alokasi IP *address*, *network address* merupakan IP *address* yang merepresentasikan alamat sebuah *network*. Semua *host* dalam satu *network* memiliki *network address* yang sama. *Network address* merupakan IP pertama dalam sebuah *subnet* IP. Sedangkan *broadcast address* merupakan jenis IP *address* yang digunakan untuk mengirim data ke semua *host* yang masih berada dalam satu *network*. *Broadcast address* adalah IP terakhir dalam sebuah subnet IP. *Network address* dan *broadcast address* tidak dapat dipasang dalam sebuah perangkat. Contoh, IP *address* 192.168.0.1 dengan subnet mask 255.255.255.0 memiliki *network address* 192.168.0.0 dan *broadcast address* 192.168.0.255 **(Kustanto & Saputro, 2015).**

Dalam pengalamatan IP, terdapat dua bagian penting yaitu alamat *network* dan *host*. Untuk mengetahui alamat *network* pada sebuah alamat IP diperlukan sebuah *network mask*. Sebagai contoh alamat IP 192.168.2.122, maka dapat diketahui bahwa ini adalah kelas C dengan alamat *network* 192.168.2.0. Komputer memerlukan perhitungan untuk mendapatkan nilai tersebut, caranya adalah dengan melakukan proses AND antara alamat IP dengan *Network mask*-nya **(Kustanto & Saputro, 2015**).

IP Address = 192.168.2.122

= 11000000. 10101000. 00000010. 01111010

Net Mask = 255.255.255.0

= 11111111. 11111111. 11111111. 00000000

AND = 11000000. 10101000. 00000010. 00000000

Network Address = 192.168.2.0

* + 1. ***Subnetting***

*Subnetting* adalah suatu metode untuk memperbanyak network ID dari suatu network ID yang telah dimiliki. *Subnetting* digunakan untuk efisiensi IP *Address*. Berfungsi sebagai jembatan perbedan topologi fisik, mempermudah proses manajemen *security network*, dan mengisolasi trafik **(Kustanto & Saputro, 2015).**

Dalam jaringan *subnetting* digunakan untuk menentukan *prefix*, *Network* ID, *Broadcast* ID, dan *Range* IP. Misalnya dalam suatu jaringan dubutuhkan IP *Address* untuk 30 user dengan IP *Gateway* 192.168.1.1 maka tentukan *prefix*, *Network* ID, *Broadcast* ID, dan *Range* IP.

1. Prefix

Untuk 30 user maka dibutuhkan satu IP untuk *Network* ID dan satu IP untuk Broadcast ID maka total IP yang dibutuhkan adalah 32 IP *Address*. Tiga puluh dua IP *Address* dapat diperoleh melalui perhitungan 25 sehingga untuk menentukan prefix yaitu 32-5 = /27

1. Network ID

Untuk mencari *Network* ID, IP *address* 192.168.1.1 dikonversi menjadi bilangan biner dan dihitung menggunakan logika AND serta semua bilangan biner diubah menjadi 1 kecuali 5 bilangan biner terakhir dijadikan 0 (nol).

1000000 010100000000001 00000001

1111111 111111111111111 11100000

1000000 0101000 00000001 00000000

92 68 1 0

1. Broadcast ID

Untuk mencari *broadcast* IP *Address* 192.168.1.1, dikonversi ke bilangan biner dan dihitung nilai dari 5 bilangan biner terakhir lalu tambahkan dengan nilai pada oktet terakhir IP *address* 1000000 10101000 00000001 00000001 1111111 11111111 11111111 11111111 16 + 8 + 4 + 2 + 1 = 31 + 1 = 32. Jadi *broadcast* ID = 192.168.1.31

1. Range IP

Range IP = 192.168.1.1 – 192.168.1.31

* 1. **Keamanan** **Jaringan**

Keamanan jaringan komputer merupakan hal penting di dalam perkembangan teknologi informasi. Keamanan jaringan yaitu proses pencegahan yang dilakukan oleh penyerang untuk terhubung ke dalam jaringan komputer melalui akses yang tidak sah, atau penggunaan secara illegal dari komputer dan jaringan. Keamanan jaringan diperlukan untuk menghindari ancaman-ancaman dari pihak yang tidak bertanggung jawab, yang bertujuan untuk menyerang, merusak, dan bahkan mengambil data-data pribadi yang menyebabkan kerugian bagi pemiliknya

Faktor-faktor penyebab resiko dalam jaringan komputer meliputi kelemahan manusia (*human error*), kelemahan perangkat keras komputer, kelemahan sistem operasi jaringan dan kelemahan sistem jaringan komunikasi. Beberapa ancaman di dalam jaringan komputer meliputi ancaman fisik berupa pencurian perangkat keras, kerusakan pada komputer dan perangkat komunikasi jaringan, *wiretapping* dan bencana alam. Ancaman yang bersifat logik berupa kerusakan pada sistem operasi atau aplikasi, virus, dan *sniffing*. Ancaman lain berupa *sniffer* (peralatan yang memonitor proses yang sedang berlangsung), *spoofing* (penggunaan komputer untuk meniru, dengan cara menimpa identitas MAC *Address* atau alamat IP), *remote attack*, *hacker* dan *cracker* **(Agustino & Priyoatmojo, 2017).**

Pada dasarnya tujuan dari keamanan jaringan adalah **(Mandiri, 2018)**:

1. *Confidentiality*, merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. *Confidentiality* biasanya berhubungan dengan informasi yang diberikan ke pihak lain.
2. *Integrity*, keaslian pesan yang dikirim melalui sebuah jaringan dan dapat dipastikan bahwa informasi yang dikirim tidak dimodifikasi oleh orang yang tidak berhak dalam perjalanan informasi tersebut.
3. *Availability*, aspek *availability* atau ketersediaan berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.
   * 1. **Klasifikasi Keamanan Jaringan**

Sistem keamanan dapat diklasifikasikan menjadi empat macam yaitu **(Mandiri, 2018)**:

1. Keamanan fisik (*Physical Security*)

Suatu keamanan yang meliputi seluruh sistem beserta peralatan, peripheral, dan media yang digunakan. Biasanya seorang penyerang akan melakukan *wiretapping* (proses pengawasan dan penyadapan untuk mendapatkan *password* agar bisa memiliki hak akses).

1. Keamanan dari pihak luar

Memanfaatkan faktor kelemahan atau kecerobohan dari orang yang berpengaruh (memiliki hak akses) merupakan salah satu tindakan yang diambli oleh seorang *hacker* maupun *cracker* untuk dapat masuk pada sistem yang menjadi targetnya. Hal ini biasa disebut *social engineering*. *Social engineering* merupakan tingkatan tertinggi dalam dunia *hacking* maupun *cracking*. Biasanya orang yang melakukan *social engineering* akan menyamar sebagai orang yang memakai sistem dan lupa *password*, sehingga akan meminta kepada orang yang memiliki hak akses pada sistem untuk mengubah atau mengganti *password* yang akan digunakan untuk memasuki sistem tersebut.

1. Keamanan data dan media

Pada keamanan ini penyerang akan memanfaatkan kelemahan yang ada pada *software* yang digunakan untuk mengolah data. Biasanya penyerang akan menyisipkan virus pada komputer target melalui *attachment* pada e-mail. Cara lainnya adalah dengan memasang *backdoor* atau *trojan* *horse* pada sistem target. Tujuannya untuk mendapatkan dan mengumpulkan informasi berupa *password* administrator. *Password* tersebut nantinya digunakan untuk masuk pada *account* administrator.

1. Keamanan dalam operasi

Merupakan salah satu prosedur untuk mengatur segala sesuatu yang berhubungan dengan sistem keamanan pasca serangan. Dengan demikian, sistem tersebut dapat berjalan baik atau menjadi normal kembali. Biasanya para penyerang akan menghapus seluruh *log-log* yang tertinggal pada sistem target (*log cleaning*) setelah melakukan serangan.

* + 1. **Jenis-jenis Serangan**

Jenis serangan yang umumnya dilakukan melalui jaringan antara lain sebagai berikut **(Risyan, 2019)**:

1. *Phishing*

*Phishing* adalah jenis rekayasa sosial yang biasanya digunakan untuk mencuri data pengguna seperti nomor kartu kredit dan kredensial masuk. Itu terjadi ketika seorang penyerang, menyamar sebagai individu tepercaya, menipu korban untuk membuka pesan teks/ email.

1. *Malware*

Malware adalah kode yang dibuat untuk secara diam-diam memengaruhi sistem komputer yang disusupi tanpa persetujuan pengguna.

1. *Worms*

Worms berbeda dari virus karena mereka tidak melampirkan ke *file host*, tetapi merupakan program mandiri yang menyebar di seluruh jaringan dan komputer.

1. SQL *Injection*

SQL *Injection*, juga dikenal sebagai SQLI, adalah jenis serangan yang menggunakan kode jahat untuk memanipulasi database *backend* untuk mengakses informasi yang tidak dimaksudkan untuk ditampilkan.

1. *Denial* *of Service* (DDoS)

*Denial Of Service* (DDoS) bertujuan untuk mematikan jaringan atau layanan, menyebabkannya tidak dapat diakses oleh pengguna yang dituju. Serangan mencapai misi ini dengan membanjiri target dengan lalu lintas atau membanjirinya dengan informasi yang memicu kecelakaan.

1. *Brute* *Force*

Serangan *Brute Force* adalah serangan jaringan di mana penyerang mencoba masuk ke akun pengguna dengan secara sistematis memeriksa dan mencoba semua kata sandi yang mungkin sampai menemukan yang benar.

1. *Port* *Scanning*

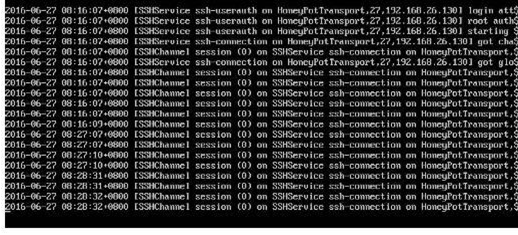
Melalui *port scanning* seorang *attacker* dapat melihat fungsi dan cara bertahan sebuah sistem dari berbagai macam *port*.

1. ICMP *Flooding*

Seorang penyerang melakukan eksploitasi sistem dengan tujuan untuk membuat suatu target *host* menjadi *hang*, yang disebabkan oleh pengiriman paket berukuran besar kearah target *host*. *Exploiting* sistem ini dilakukan dengan mengirimkan suatu *command* ping dengan tujuan *broadcast* atau *multicast* dimana si pengirim dibuat seolah-olah adalah target *host*.

* 1. ***Honeypot***

*Honeypot* adalah suatu alat untuk mendapatkan informasi tentang penyerang. Selanjutnya administrator jaringan dapat mempelajari aktifitas-aktifitas yang dapat merugikan dan melihat kecenderungan dari aktifitas tersebut. *Honeypot* adalah sebuah sistem yang dirancang untuk diperiksa dan diserang. Sistem dengan *Honeypot* akan menipu dan atau memberikan data palsu apabila ada orang yang memiliki maksud yang tidak baik masuk ke suatu sistem. Secara teori *Honeypot* tidak akan mencatat trafik yang legal. Sehingga bisa dilihat bahwa yang berinteraksi dengan *Honeypot* rata-rata adalah user yang menggunakan sumber daya sistem secara ilegal. Jadi Honeypot menjadi sistem yang seolah-olah berhasil disusupi oleh penyerang. Padahal penyerang tidak masuk kesistem yang sebenarnya, tetapi malah masuk kedalam sistem yang palsu. Gambar 2.7 merupakan salah satu contoh log yang dihasilkan oleh honeypot **(Mustofa & Aribowo, 2015).**



**Gambar 2.7** Contoh log pada Honeypot **(Mustofa & Aribowo, 2015)**

..

..

..

…….

dst

**BAB III**

**PEMBAHASAN**

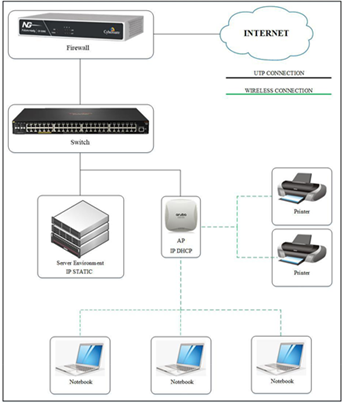
* 1. **Keamanan Server di PT. Agiva Indonesia**

PT. Agiva Indonesia merupakan salah satu perusahaan layanan dan konsultan yang bergerak di bidang teknologi informasi. Mereka melayani segala macam masalah yang berhubungan dengan teknologi informasi baik itu jaringan, software, maupun hardware. Dalam penerapannya PT. Agiva Indonesia memiliki sebuah server yaitu database server yang digunakan untuk meyimpan data-data. Baik itu data perusahaan ataupun data milik customer.

Pada PT. Agiva Indonesia belum ada sistem pendeteksi serangan cyber yang baik, sehingga sangat rentan bagi sebuah server khususnya database server yang menyimpan data-data penting untuk diserang dan disusupi oleh seorang *attacker*. Sistem keamanan *firewall* yang ada dirasa masih belum cukup dalam melakukan pendeteksian penyusupan, khususnya penyusupan yang dilakukan dari dalam jaringan (jaringan internal). Kurangnya informasi yang diperoleh administrator mengenai serangan *cyber*, seperti siapa yang menyerang, bagaimana mereka menyerang, dan kapan serangan dilakukan, merupakan salah satu tantangan yang saat ini sedang dihadapi oleh administrator server PT. Agiva Indonesia.

* + 1. **Topologi Jaringan pada PT. Agiva Indonesia**

Data utama untuk penelitian ini adalah topologi jaringan dari PT. Agiva Indonesia. Internet terhubung ke *firewall* menggunakan koneksi *Wide Area Network* (WAN). Kemudian langsung terhubung ke jaringan lokal (LAN) dengan menggunakan *switch*. Dari *switch*, diteruskan ke *server* dan *Access Point*, sehingga *client* dapat terhubung ke jaringan melalui WLAN. Pada sisi *server* sendiri, terdapat *database server* yang digunakan untuk keperluan penyimpanan data-data, baik itu data-data perusahan maupun data customer dari PT. Agiva Indonesia. Gambar 3.1 menunjukkan topologi jaringan yang digunakan oleh PT. Agiva Indonesia.

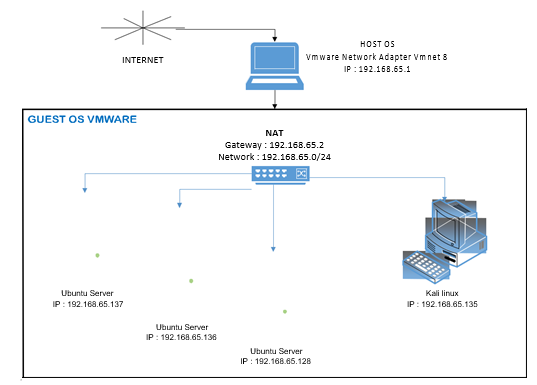


**Gambar 3.1** Topologi Jaringan PT. Agiva Indonesia

* 1. **Analisa Sistem Keamanan Server pada PT. Agiva Indonesia**

Berdasarkan permasalahan yang telah dijelaskan diatas, maka dibutuhkan sebuah sistem pendeteksi serangan pada PT. Agiva Indonesia. Sehingga pihak administrator dapat mengetahui aktivitas serangan yang ditujukan ke server PT. Agiva Indonesia. Dalam hal ini *honeypot* dan ELK *Stack* menjadi upaya yang ditawarkan sebagai sistem pendeteksi serangan sekaligus memberikan visualisasi kepada administrator. Hal ini berfungsi untuk mengoptimalkan keamanan server pada PT. Agiva Indonesia.

Konsep solusi yang diberikan terkait dengan permasalahan yang sudah dijabarkan sebelumnya yaitu dengan mengimplementasikan sistem *honeypot* guna melindungi *server* dari aktivitas serangan. *Honeypot* berfungsi untuk menjebak penyerang lalu membuat *log* aktifitas penyerang, kemudian ELK *Stack* akan memproses *log* dari *honeypot* dan akan divisualisasikan sehingga pihak administrator dapat dengan mudah memonitor log.



**Gambar 3.2** Desain Sistem Keamanan Server yang Akan Dibangun di PT. Agiva Indonesia

Sistem yang dibangun bersifat simulasi, dengan membuatnya dalam *virtual network*. Dalam hal ini memakai aplikasi VMware untuk membuat simulasi sistem. Dapat dilihat pada gambar 3.2, *network* dari *Host* OS diteruskan ke dalam VMware menggunakan konfigurasi NAT dengan *network address* 192.168.65.0/24 dan *gateway* 192.168.65.2. Didalam VMware akan diinstal 3 server, yaitu server utama *honeypot* *server*, dan ELK *server*. Pada *server* utama akan disimulasikan layaknya *server* pada PT. Agiva yaitu dengan membuka *port* untuk *remote* SSH (22) dan juga *port* untuk database Mysql (3306). Pada *server honeypot* akan dipasang *low-interaction* *honeypot* yaitu *cowrie & dionaea* untuk membuka *port-port* palsu, beserta *filebeat* untuk pengiriman *log*. Kemudian pada *server* ELK diinstal *elasticsearch*, *logstash* dan kibana sebagai server visualisasi *log honeypot*.

Dan juga pada VMware akan diinstal PC penyerang yang menggunakan OS kali linux agar memudahkan dalam proses penyerangan, karena sudah terdapat tools yang akan digunakan untuk proses penyerangan. Honeypot dalam kasus ini bertindak sebagai production honeypot karena bertujuan untuk mengurangi resiko serangan ke server utama, dengan cara melakukan forwarding port-port dari honeypot (21, 22, 23, 135, 445) ke server utama, sehingga penyerang ketika melakukan serangan ke server utama akan teralihkan ke server honeypot.

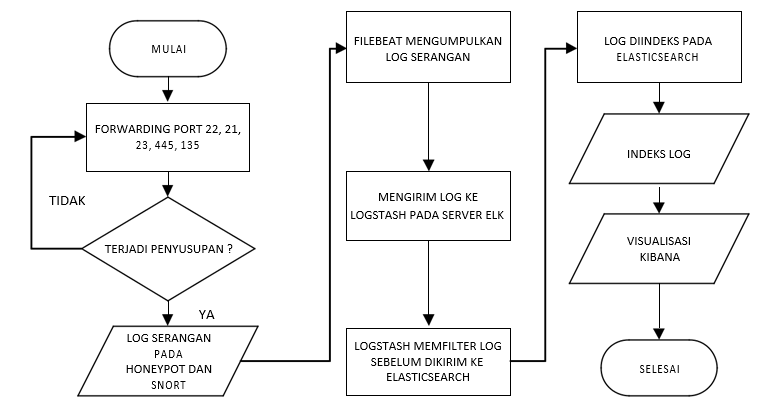
Untuk spesifikasi perangkat yang digunakan dalam VMware dapat dilihat pada tabel 3.

**Tabel 3.1** Spesifikasi Perangkat Pada Vmware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Perangkat** | **Server** | **Honeypot Server** | **ELK Server** | **Penyerang** |
| OS | Ubuntu server 18.04 64 bit | Ubuntu server 14.04 64 bit | Ubuntu server 16.04 64 bit | Kali Linux 2020.2a 64 bit |
| Harddisk | 20 GB | 20GB | 40GB | 40GB |
| RAM | 4 GB | 2 GB | 8 GB | 4 GB |
| Processor | 2 | 2 | 4 | 4 |
| IP | 192.168.65.  128 | 192.168.65.  136 | 192.168.65.  137 | 192.168.65.  135 |
| Perangkat Lunak | Open SSH server  MySQL Snort Filebeat 6.6.0 | Dionaea 0.6.0 Cowrie 1.6.0  Filebeat 6.6.0 | Elastics earh 6.6.0  Logstash 6.6.0  Kibana 6.6.0 | Xhydra  Metasploit |

* 1. **Perancangan Simulasi Keamanan Server di PT. Agiva Indonesia**

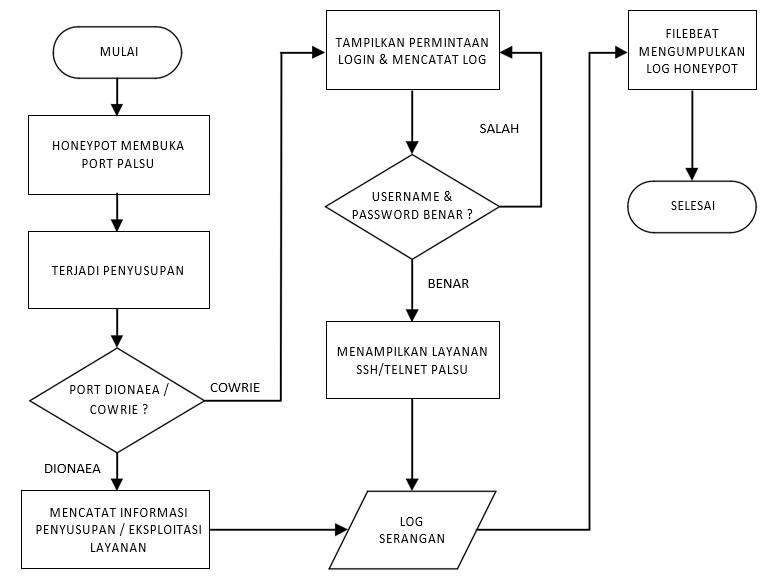
Pada gambar 3.3, merupakan flowchart cara kerja sistem secara umum. Dimulai dari honeypot yang membuka *port* palsu, lalu di lakukan *forwarding port* dari *honeypot* ke *server* utama. Ketika belum terjadi penyusupan, *honeypot* akan tetap membuka *port* palsu untuk menarik penyerang. Ketika terjadi penyusupan maka akan direkam dan menjadi *log* serangan. *Log* serangan nantinya akan di kumpulkan oleh *filebeat* baik itu dari *server honeypot* dan *server* utama, kemudian akan dikirimkan ke *logstash* yang terdapat pada *server* ELK melalui *port* 5504. Di *logstash*, *log-log* tersebut akan difilter menurut proses pemfilteran *logstash* yang nanti akan dikonfigurasi. Setelah *log* tersebut di filter, barulah dikirim ke *elasticsearch*, yang kemudian di olah oleh *elasticsearsh* menjadi indeks-indeks *log*, yang nantinya akan memudahkan dalam pemvisualisasian pada kibana.



**Gambar 3.3** Cara Kerja Sistem Honeypot & ELK Stack

* + 1. **Perancangan Alur Sistem Kerja Honeypot**

Pada gambar 3.4 menjelaskan tentang alur sistem *honeypot* yang akan diterapkan pada server *honeypot*. Tahap awal ialah *honeypot* membuka *port-port* palsu. Pada sisi *cowrie port* yang akan diteruskan ke *server* utama yaitu port 22 dan 23. Sedangkan pada *dionaea*, *port* yang akan diteruskan ke *server* adalah *port* 21, 135, 445. Ketika terjadi penyusupan pada *honeypot cowrie* (SSH/telnet), maka *cowrie* akan bertindak seperti sistem yang asli, yaitu memberikan permintaan login *username & password*. *Cowrie* akan merekam informasi penyerang seperti IP dan waktu serangan disaat penyerang benar/salah dalam memasukkan kombinasi *username & password*. Ketika penyerang berhasil memasukkan kombinasi yang benar, maka *cowrie* akan menampilkan layanan SSH/ *telnet* palsu kepada penyerang.



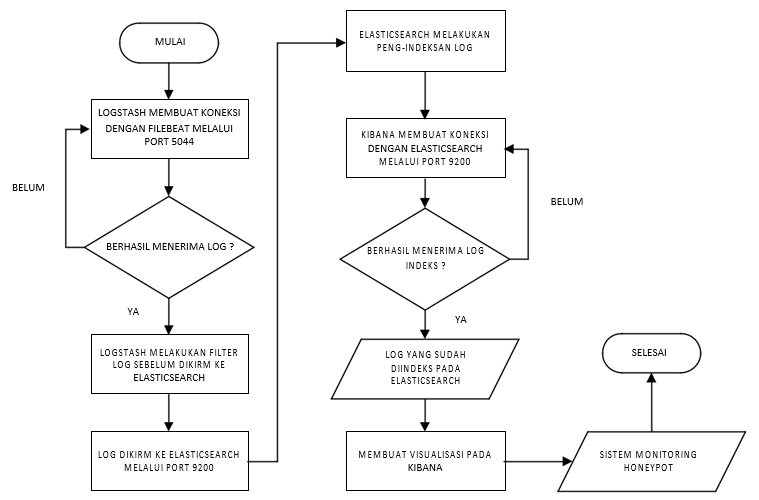
**Gambar 3.4** Perancangan Alur Sistem Honeypot

Kemudian ketika terjadi penyerangan pada *honeypot dionaea* (*port* 21, 135, 445) maka *honeypot dionaea* akan merekam informasi penyerangan dan juga eksploitasi layanan. Setelah terdapat *log* dari masing-masing *honeypot*, maka *filebeat* sebagai perantara *logstash* akan mengumpulkan *log honeypot* untuk dikirim ke *server* ELK.

* + 1. **Perancangan Alur Sistem ELK Stack**

Dari gambar 3.5, dapat dilihat bahwa *log* dari *server honeypot* dan *log* dari *snort server* utama akan diterima oleh *logstash*. Dengan meminta koneksi ke *filebeat* melalui *port* 5044. Ketika *logstash* berhasil menerima *log* dari *server honeypot* dan *snort*, maka akan dilakukan proses filter dan pengeditan file *log* agar terbaca oleh *elasticsearch*.

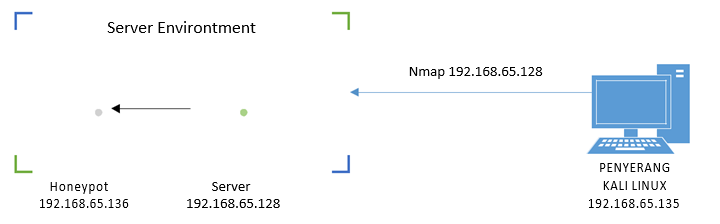
**Gambar 3.5** Perancangan Alur Sistem ELK Stack



Setelah difilter, *log* dikirim ke *elasticsearch* melalui *port* 9200. Di *elasticsearch* dilakukan pengindeksan *log*, agar nantinya mudah dibaca oleh admin. Setelah *log* sudah diindeks, *elasticsearch* menunggu permintaan dari kibana untuk mengambil *log*. Ketika kibana membuat koneksi dengan *elasticsearch*, maka terjadi pengambilan *log* dari kibana ke *elasticsearch* melalui *port* 9200. Di kibana *log* tersebut nantinya akan dibuatkan visualisasi oleh admin, yang pada akhirnya menjadi sistem monitoring *honeypot*. Apabila proses pengambilan log belum berhasil maka akan dicoba kembali sampai berhasil.

* 1. **Perancangan Pengujian Sistem Simulasi Keamanan Server di PT. Agiva Indonesia**

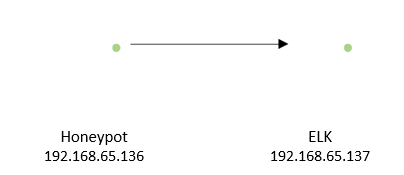
Dalam sub bab ini akan dijelaskan perancangan pengujian sistem meliputi pengujian sistem *honeypot*, pengujian sistem ELK, pengujian *brute force* SSH, *remote server*, eksploitasi layanan dan visualisasi kibana.



**Gambar 3.6** Rancangan Pengujian Sistem Honeypot

* + 1. **Perancangan Pengujian Sistem Honeypot**

Pada rancangan ini penyerang akan melakukan *port scanning* pada IP *address server* dengan menggunakan perintah nmap 192.168.65.128. Pengujian akan dilakukan dua kali yaitu sebelum *honeypot server* dijalankan dan sesudah *honeypot server* dijalankan. Pengujian ini untuk melihat apakah *honeypot server* dapat memberikan layanan palsu kepada *server* utama. Simulasi ditunjukkan pada gambar 3.6.



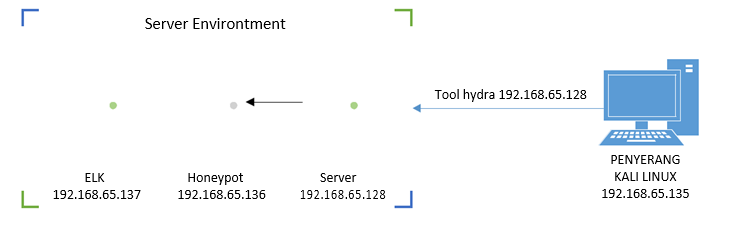
**Gambar 3.7** Rancangan Pengujian Sistem ELK

* + 1. **Perancangan Pengujian Sistem ELK**

Pada rancangan pengujian sistem ELK akan diuji apakah *log* serangan dari server *honeypot* dapat terkirim ke *server* ELK dan dapat divisualisasikan. Simulasi ditunjukkan pada gambar 3.7.

* + 1. **Perancangan Pengujian Brute Force SSH**

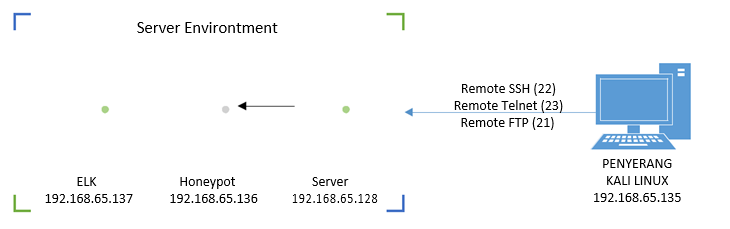
Pada rancangan ini penyerang akan melakukan serangan *brute force* pada *port* 22 *server* utama dengan menggunakan *tool* *hydra*. Pengujian ini untuk melihat apakah serangan *brute force* dapat dialihkan menuju *server honeypot*, dan *log* serangan dapat dikirim ke *server* ELK dan divisualisasikan. Simulasi pengujian *brute force* ditunjukkan pada gambar 3.8.



**Gambar 3.8** Rancangan Pengujian Brute Force SSH

* + 1. **Perancangan Pengujian Remote Server**

Pada rancangan ini penyerang akan melakukan *remote server* menuju *port-port* default, yaitu *port* 22, 21, dan 23, yang telah dirancang untuk menipu penyerang, karena *port* SSH asli server telah diubah menjadi *port* 7722. Pengujian ini dilakukan untuk menguji apakah *remote server* (port 22, 21, 23) dapat dialihkan menuju ke *server honeypot*. Dan *server* ELK mampu mengambill *log honeypot* lalu memvisualisasikannya. Simulasi pengujian *remote server* ditunjukkan pada gambar 3.9.



**Gambar 3.9** Rancangan Pengujian Remote Server Honeypot

* + 1. **Perancangan Eksploitasi Layanan**

…..

…….

………

dst

**BAB IV**

**IMPLEMENTASI DAN PENGUJIAN**

* 1. **Spesifikasi Implementasi**

Untuk melakukan simulasi *virtual* sistem, dibutuhkan spesifikasi baik *hardware* maupun *software*. Berikut merupakan spesifikasi perangkat keras laptop dan perangkat lunak yang digunakan:

Processor : Intel Core i5 8250U + 1,6 GHz

Memory : DDR4 8 GB x 2

Harddisk : 1000 GB

OS : Windows 10 Pro 64 bit

Software : VMware Workstation 15 Pro

* 1. **Konfigurasi Server Utama**

Dalam konfigurasi *server* utama akan dijelaskan konfigurasi *port* SSH, konfigurasi *snort*, konfigurasi *filebeat* dan IP *forwarding*. Keseluruhan instalasi dilakukan sebagai *root* agar setiap *file* yang dihasilkan memiliki *root permission*.

* + 1. **Konfigurasi Snort**

Dalam tahap ini dilakukan proses instalasi dan konfigurasi *snort* pada *server* utama agar merekam IP penyerang yang mengakses *port* 21, 22, 23, 135 dan 445. Dimulai dari proses penginstalan *snort*, kemudian mengedit *file* snort.conf, supaya sistem *snort* hanya merekam aktifitas pada *server* utama yaitu 192.168.65.128. Dilanjutkan dengan proses pembuatan *rules*, dengan menambahkan beberapa *rules* pada *file* local.rules. Yang terakhir yaitu menjalankan sistem *snort*. Konfigurasi *snort* ditunjukkan pada kode 4.1.

**Kode Program 4.1** Konfigurasi Snort

1: agiva@agiva:$ apt-get install ethtool build-essential libpcap-dev libpcre3-dev libdumbnet-dev bison flex zlib1g-dev liblzma-dev openssl libssl-dev

2: agiva@agiva:$ apt update && sudo apt install snort

3: agiva@agiva:$ nano /etc/snort/snort.conf

ipvar HOME\_NET 192.168.65.128 #(line 45)

var RULE\_PATH /etc/snort/rules #(line 104)

var SO RULE PATH /etc/snort/so rules #(line 105)

include $RULE\_PATH/local.rules #(line 433)

04: agiva@agiva:$ nano /etc/snort/rules/local.rules

05: agiva@agiva:$ snort -T -c /etc/snort/snort.conf -i ens33

06: agiva@agiva:$ snort -A fast -c /etc/snort/snort.conf -q -i ens33



**Gambar 4.1** Konfigurasi local.rules

* + 1. **Konfigurasi Filebeat**

Dalam tahap ini dilakukan proses instalasi dan konfigurasi *file* filebeat.yml untuk berkomunikasi langsung dengan logstash pada server utama dengan alamat 192.168.65.137:5044.

**Kode Program 4.2** Konfigurasi Filebeat

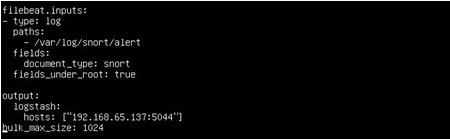
1: agiva@agiva:$ wget https://artifacts.elastic.co/ downloads/filebeat/filebeat-6.6.0-amd64.deb

2: agiva@agiva:$ dpkg -i filebeat-6\*

3: agiva@agiva:$ cd /etc/filebeat/

4: agiva@agiva:$ nano filebeat.yml

5: agiva@agiva:$ service filebeat start



**Gambar 4.2** Konfigurasi Filebeat.yml Pada Server Utama

* 1. **Konfigurasi Server Honeypot**

Dalam konfigurasi *server honeypot* akan dijelaskan proses instalasi dan konfigurasi setiap *honeypot*, yaitu *dionaea* dan *cowrie*. Keseluruhan instalasi dilakukan sebagai *root* agar setiap *file* yang dihasilkan memiliki *root permission*.

* + 1. **Konfigurasi IP Statis**

Dalam tahap ini dilakukan proses konfigurasi ip static server honeypot, menjadi 192.168.65.136.

**Kode Program 4.3** Konfigurasi IP Statis Server Honeypot

01: elk@ubuntu:$ nano /etc/network/interfaces



**Gambar 4.3** Konfigurasi IP Statis Server Honeypot

* + 1. **Konfigurasi Dionaea**

Dalam tahap ini dilakukan proses instalasi dan konfigurasi *honeypot* *dionaea* supaya dapat berjalan pada sisi *server honeypot*. Diawali dengan menambahkan repositori *dioanea*, update paket, kemudian menginstal *dionaea*. Agar *dionaea* nantinya menghasilkan *log* dengan format JSON, maka perlu dilakukan pemindahan *file* log\_json.yaml dari *ihandlers-available* ke *ihandlers-enable*. Setelah dipindahkan, maka langkah berikutnya yaitu mengedit *file* log\_json.yaml supaya *log* JSON nantinya berada pada direktori /opt/dionaea/var/dionaea/json.

Selanjutnya memberi tanda *comment* (#) pada layanan *telnet*, karena *port* 23 nantinya akan dipakai oleh *honeypot cowrie*. Kemudian memulai *dionaea* untuk berjalan di *server honeypot*.

**Kode Program 4.4** Konfigurasi Dionaea

1: honeypot@ubuntu:$ apt-get install software-properties-common

2: honeypot@ubuntu:$add-apt-repository ppa:honeynet/nightly

3: honeypot@ubuntu:$ apt-get update

4: honeypot@ubuntu:$ apt-get install dionaea

5: honeypot@ubuntu:$ cd /opt/dionaea/etc/dionaea/ihandlers-available

6: honeypot@ubuntu:$ cp log\_json.yaml /opt/dionaea/etc/dionaea/ ihandlers-enabled/

7: honeypot@ubuntu:$ nano /opt/dionaea/etc/dionaea/ihandlers-enabled/log\_json.yml

name: log\_json

config:

handlers:

-file:/opt/dionaea/var/dionaea/json/dionaea.json

8: honeypot@ubuntu:$ nano /opt/dionaea/etc/dionaea/services-available/blackhole.yml

services:

# Telnet #(line 4)

# -port: 23 #(line 5)

# protocol: tcp #(line 6)

09: honeypot@ubuntu:$ service dionaea start

* + 1. **Konfigurasi Cowrie**

Dalam tahap ini dilakukan proses instalasi dan konfigurasi *honeypot cowrie* supaya dapat berjalan pada sisi *server honeypot*. Diawali dengan mengganti *port* SSH asli *server* ke *port* 7000, karena *port* 22 nantinya akan di pakai oleh *honeypot cowrie*. Langkah selanjutnya yaitu proses penginstalan file-file yang dibutuhkan dalam proses instalasi *honeypot cowrie*. Berikutnya yaitu mengedit konfigurasi *cowrie* pada *file* cowrie.cfg, dengan mengubah *hostname* *server* palsu menjadi agiva, lalu mengaktifkan layanan *telnet* dan mengganti *port* supaya memakai *port* 22 dan 23.

*File* *userdb.txt* berfungsi untuk memberikan *username* dan *password* pada layanan SSH dan *telnet* palsu. Disini digunakan *username* = agiva dan *password* = agiva123 untuk bisa masuk ke layanan palsu. Yang terakhir ialah memberikan akses ke *honeypot* *cowrie* untuk memakai *port* 22 dan 23, kemudian memulai *honeypot* *cowrie* untuk berjalan di *server honeypot*.

**Kode Program 4.4** Konfigurasi Cowrie

1: honeypot@ubuntu:$ nano /etc/ssh/sshd\_config port 7000

2: honeypot@ubuntu:$ service ssh restart

3: honeypot@ubuntu:$ apt-get install git python-virtualenv

libssl-dev libffi-dev build-essential libpython-dev python 2.7-minimal authbind

4: honeypot@ubuntu:$ adduser cowrie –disabled-password cowrie

5: honeypot@ubuntu:$ su cowrie

6: cowrie@ubuntu:$ cd

7: cowrie@ubuntu:$ git clone http://github.com/cowrie/cowrie

8: cowrie@ubuntu:$ cd cowrie

9: cowrie@ubuntu:$ virtualenv cowrie-env

10: cowrie@ubuntu:$ source cowrie-env/bin/activate

11: (cowrie-env):$ pip install –-upgrade pip

12: (cowrie-env):$ pip install –-upgrade –r requirements.txt

13: (cowrie-env):$ deactivate

14: cowrie@ubuntu:$ cd /etc

15: cowrie@ubuntu:$ cp cowrie.cfg.dist cowrie.cfg

16: cowrie@ubuntu:$ cp userdb.example userdb.txt

17: cowrie@ubuntu:$ nano

cowrie.cfg hostname = agiva

[telnet]

enabled = true [ssh]

listen\_endpoints = tcp:22:interface=0.0.0.0

[telnet]

Listen\_endpoints = tcp:23:interface=0.0.0.0

18: cowrie@ubuntu:$ nano userdb.txt agiva:x:agiva123

19: cowrie@ubuntu:$ exit

20: honeypot@ubuntu:$ touch /etc/authbind/byport/22

21: honeypot@ubuntu:$ chown cowrie:cowrie /etc/authbind/byport/22

22: honeypot@ubuntu:$ chmod 770 /etc/authbind/byport/22

23: honeypot@ubuntu:$ touch /etc/authbind/byport/23

24: honeypot@ubuntu:$ chown cowrie:cowrie /etc/authbind/byport/23

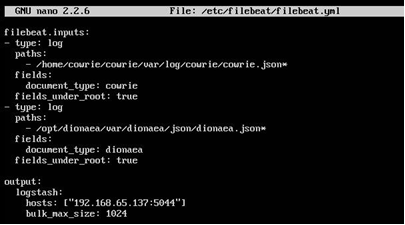
25: honeypot@ubuntu:$ chmod 770 /etc/authbind/byport/23

26: honeypot@ubuntu:$ su cowrie

27: cowrie@ubuntu:$ /home/cowrie/cowrie/bin/cowrie start

* + 1. **Konfigurasi Filebeat**

Dalam tahap ini dilakukan proses instalasi dan konfigurasi *file* *filebeat.yml* untuk berkomunikasi langsung dengan *logstash* pada *server honeypot* dengan alamat 192.168.65.137:5044.



**Gambar 4.4** Konfigurasi Filebeat Pada Server Honeypot.yml

**Kode Program 4.5** Konfigurasi Filebeat

1: honeypot@ubuntu:$ wget https://artifacts.elastic.co/downloads /filebeat/filebeat-6.6.0-amd64.deb

2: honeypot@ubuntu:$ dpkg -i filebeat-6\*

3: honeypot@ubuntu:$ cd /etc/filebeat/

4: honeypot@ubuntu:$ nano filebeat.yml

5: honeypot@ubuntu:$ service filebeat start

* + 1. **Konfigurasi Port Forwarding**

Dalam tahap ini dilakukan proses *forwarding* port-port pada *server honeypot* ke server utama, yang digunakan untuk menipu penyerang. *Port* yang akan di-*forward* yaitu *port* 21, 22, 23, 135 dan 445.

**Kode Program 4.6** Konfigurasi Port Forwarding

1: honeypot@ubuntu:$ iptables -t nat -A PREROUTING -p tcp --dport 22 -j DNAT --to-destination 192.168.65.128:22

2: honeypot@ubuntu:$ iptables -t nat -A PREROUTING -p tcp --dport 21 -j DNAT --to-destination 192.168.65.128:21

3: honeypot@ubuntu:$ iptables -t nat -A PREROUTING -p tcp --dport 23 -j DNAT --to-destination 192.168.65.128:23

4: honeypot@ubuntu:$ iptables -t nat -A PREROUTING -p tcp --dport

135 -j DNAT --to-destination 192.168.65.128:135

5: honeypot@ubuntu:$ iptables -t nat -A PREROUTING -p tcp --dport

445 -j DNAT --to-destination 192.168.65.128:445

6: honeypot@ubuntu:$ sudo iptables -t nat -A POSTROUTING -j

MASQUERADE

* 1. **Konfigurasi Server ELK Stack**

Dalam konfigurasi *server* ELK *stack* akan dijelaskan proses instalasi dan konfigurasi elemen-elemen ELK *stack* yaitu *elasticsearch*, *logstash*, dan *kibana*. keseluruhan instalasi dilakukan sebagai *root* agar setiap file yang dihasilkan memiliki *root permission*.

……

…….

…………..

Dst

* 1. **Pengujian Sistem**

Dalam sub bab pengujian sistem akan dibahas tentang pengujian sistem yang telah dibangun, diantaranya pengujian sistem *honeypot*, pengujian sistem ELK, pengujian *brute force*, *login* SSH, *login telnet*, *login* FTP, eksploitasi MS04\_011, eksploitasi MS03\_026, dan pengujian visualisasi kibana.

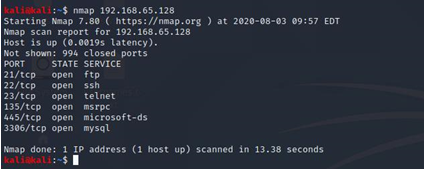
* + 1. **Pengujian Sistem Honeypot**

Pengujian sistem *honeypot* bertujuan untuk mengetahui apakah *honeypot* benar-benar ter-implementasi pada server *honeypot*. Untuk itu dilakukan pengujian *port scanning* pada sisi penyerang sebelum dan sesudah *honeypot* dijalankan. Pengujian dilakukan dengan men-*scan* IP *address server* utama menggunakan perintah nmap 192.168.65.128.



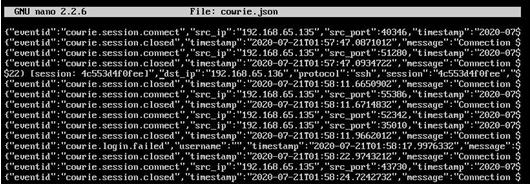
**Gambar 4.17** Hasil Port Scanning Sebelum Honeypot Dinyalakan

Gambar 4.17 menunjukkan hasil *port scanning* pada saat *honeypot* belum dijalankan. Setelah dilakukan *port scanning*, dapat dilihat pada gambar 4.18 bahwa hanya port 3306 saja yang terlihat. Pengujian berikutnya dilakukan setelah *honeypot* dijalankan pada jaringan dan dilakukan *port forwarding*. Untuk perintahnya masih sama menggunakan nmap 192.168.65.128.



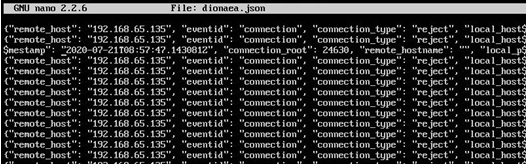
**Gambar 4.18** Hasil Port Scanning Sesudah Honeypot Dijalankan

Dari gambar 4.19 dapat dilihat bahwa sistem *honeypot* dapat berjalan dengan baik, yaitu dengan memberikan port-port palsu, sehingga memiliki kerentanan sistem dan menjadi pengalihan yang dapat diserang oleh penyerang. Selain itu *honeypot* juga telah mendapatkan data penyerang yang disimpan pada log honeypot, yaitu pada *cowrie.json*, dapat dilihat pada gambar 4.20.



**Gambar 4.19** Log cowrie.json

Dan juga *log dionaea.json* untuk *honeypot dionaea*, yang dapat dilihat pada gambar 4.20.



**Gambar 4.20** Log dionaea.json

* + 1. **Pengujian Sistem ELK**

Pengujian sistem ELK , untuk mengetahui bahwa apakah ELK sudah terimplementasi pada *server* ELK, dan juga dapat mengambil *log* dari *server honeypot*. Untuk itu dilakukan pengambilan data *log* *port scanning* pada *server honeypot*. Sebelumnya dilakukan perintah *netstat* –plntu pada *server* ELK untuk melihat apakah sistem *elasticsearch* (9200), *kibana* (5601), dan *logstash* (5044) sudah berjalan, yang dapat ditunjukkan pada gambar 4.21.



**Gambar 4.21** Sistem ELK Pada Server ELK

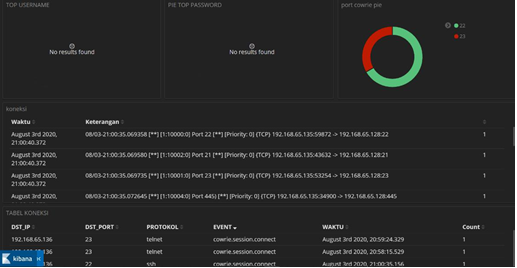
Ketika semua elemen sudah berjalan, dilanjutkan dengan membuka monitoring pada *kibana*. Semua *log* yang berhasil diterima oleh sistem ELK akan terlihat pada menu *discover kibana*. Pada gambar 4.22 merupakan *discover* pada *honeypot cowrie* setelah dilakukan serangan *port scanning*.

Dapat dilihat setiap session serangan dikelompokkan per baris sehingga memudahkan dalam pembuatan visualisasi. Terdapat dua sumber waktu dalam *discover* ini, pada bagian kiri merupakan waktu untuk *log* berhasil dikirimkan, kemudian bagian kanan merupakan waktu dari *log* *honeypot-*nya. Dalam pengujian *port scanning* *honeypot cowrie*, tidak ada perbedaan waktu yang signifikan, sehingga dapat diambil kesimpulan sistem ELK dapat menerima *log* secara *real time* pada kasus *port scanning honeypot cowrie*.



**Gambar 4.22** Discover Serangan Port Scanning Pada Cowrie

Pada gambar 4.23, merupakan visualisasi *kibana* yang dibuat. Terdapat informasi *source* IP penyerang, *source port* penyerang, *destination address*, dan *port* yang diserang. Untuk *username* dan *password* belum terdapat *log*-nya karena belum melakukan serangan yang melibatkan *username* dan *password*.



**Gambar 4.23** Visualisasi Serangan Port Scanning Pada Cowrie

Pada gambar 4.24 merupakan *discover log* yang diterima oleh *honeypot dionaea*. Dapat dilihat terdapat perbedaan waktu 4 detik dari waktu *log honeypot dionaea* (21:00:04) dengan waktu *log* ELK menerima *log* (21.00.08), dapat diambil kesimpulan bahwa dalam pengujian *port scanning* untuk *honeypot dionaea*, sistem ELK masih belum secara *real-time* menerima *log*.

…..

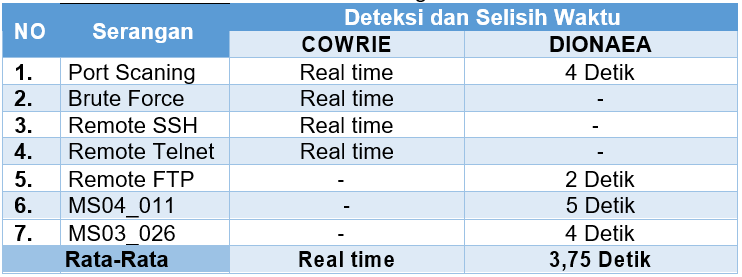
…….

………

dst

Dari semua pengujian yang telah dilakukan dirangkum dalam tabel yang berisi informasi deteksi serangan setiap *honeypot*, dan selisih waktu dalam pengiriman *log* ke *server* ELK, yang dapat dilihat pada tabel 4.1

**Tabel 4.1** Tabel deteksi serangan dan selisih waktu



Dari data pada tabel 4, dapat disimpulkan bahwa dari 7 pengujian serangan yang dilakukan, *honeypot cowrie* mendeteksi 4 serangan secara *real time*, diantaranya *port scanning*, *brute force*, *remote* SSH, dan *remote telnet*. Sedangkan untuk *honeypot dionaea* masih belum secara *real time* dalam mengirimkan *log* ke *server* ELK, dengan rata-rata waktu *delay* yaitu 3,75 detik. Untuk serangan yang dapat direkam oleh *honeypot dionaea* diantaranya *port scanning*, *remote* FTP, eksploitasi MS04\_011, dan eksploitasi MS03\_026.

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

D`ari hasil pengujian yang telah dilakukan didapatkan kesimpulan sebagai berikut :

1. Server *honeypot* yang diimplementasikan, mampu membuat layanan palsu dengan membuka port-port pada server utama.
2. Server ELK yang diimplementasikan, mampu menerima *log* serangan yang berasal dari *server honeypot* dan dapat memvisualisasikannya.
3. Serangan yang dideteksi oleh *honeypot cowrie* antara lain *port scanning*, *brute force* (22), *remote* SSH (22), dan *remote telnet* (23). Serangan yang dideteksi oleh *honeypot dionaea* antara lain *port scanning*, *remote* FTP (21), eksploitasi MS04\_011 (445), dan eksploitasi MS03\_026 (135).
4. Dari pengujian yang dilakukan, server ELK mampu menerima *log* serangan dari 4 serangan yang dideteksi oleh *honeypot cowrie* secara *real time*. Sedangkan dari 4 serangan yang dideteksi oleh *honeypot dionaea* dapat diterima dengan rata-rata delay 3,75 detik. Hal ini menandakan sistem ELK belum secara *real-time* menerima *log* dari *honeypot dionaea*.
5. Sistem *honeypot* dan ELK mampu menjadi pengalihan target serangan pada *server* dan juga sebagai sistem deteksi serangan.
6. Sistem virtual yang dibuat mampu bekerja sesuai dengan perancangan yang telah ditentukan. Hasil yang didapatkan mampu memberikan gambaran nyata terhadap kegiatan serangan bila terjadi pada jaringan komputer yang sebenarnya.
   1. **Saran**

Mengingat berbagai kekurangan dan keterbatasan yang ada pada sistem yang telah dibuat, maka diperlukan beberapa masukan guna perbaikan dan untuk meningkatkan hasil yang lebih baik, maka disarankan pada penelitian berikutnya yaitu :

1. Agar kedepannya sistem ini dapat diimplementasikan diluar jaringan lokal.
2. Agar kedepannya menggunakan *high-interaction honeypot*, agar mendapatkan informasi serangan yang lebih luas.
3. Agar kedepannya sistem ELK dapat menerima semua *log* secara *real-time*.
4. Penambahan sistem pencegahan serangan, agar dapat mencegah serangan yang masuk, sehingga tidak hanya sebagai sistem pendeteksi serangan.

**DAFTAR PUSTAKA**

Agustino, D. P., & Priyoatmojo, Y. (2017). *Implementasi Honeypot Sebagai Pendeteksi Serangan dan Melindungi Layanan Cloud Computing*. Bali: E-Proceedings KNS&I STIKOM Bali. Konferensi Nasional Sistem & Informatika 2017, 196–201.

Arifin, M. N., & Susilowati, E. (2018). *Desain dan Implementasi Log Event Management Server Menggunakan Elasticserach Logstash Kibana (ELK Stack)*. Fakultas Teknik Universitas Muhammadiyah Jakarta, Prosiding Semnastek.

Arkaan, N., & Sakti, D. V. S. Y. (2019). *Implementasi Low Interaction Honeypot Untuk Analisa Serangan Pada Protokol SSH*. Jurnal Nasional Teknologi Dan Sistem Informasi, 5(2). 112–120.

Elastic. (2018). *ELK Stack*. [online]. Available at: https://www.elastic.co [diakses 12 September 2020]

Hermawan. (2019). *Pengertian Server, Fungsi Server Beserta Cara Kerja dan Jenis-Jenis Server*. [online]. Available at: www.nesabamedia.com/pengertian-server-dan-fungsi-server [diakses 10 Agustus 2020]

Kustanto, & Saputro, D. T. (2015). *Belajar Jaringan Komputer Berbasis Mikrotik OS*. Yogyakarta: Gava Media.

Mardiyanto, B., & Indriyani, T. (2016). *Analisis Dan Implementasi Honeypot Dalam Mendeteksi Serangan Distributed Denial-Of-Services ( DDOS ) Pada Jaringan Wireless*. Jurnal Ilmu Komputer dan Informasi 9 (1), 35-42. Fakultas Teknik Elektro dan Teknologi Informasi Institut Teknologi Adhi Tama.

Mustofa, M. M., & Aribowo, E. (2015). *Penerapan Sistem Keamanan Honeypot dan IDS Pada Jaringan Nirkabel (Hotspot)*. Jurnal Sarjana Teknik Informatika 1 (1), 111-118.

Nawrocki, M. (2016). *A Survey on Honeypot Software and Data Analysis*. [online]. Available at: http://arxiv.org/abs/1608.06249 [diakses 5 Agustus 2020].

Prastiyanto, D. (2015). *Struktur Jaringan Komunikasi Data Paket Berdasar Protokol X.25.* Jurnal Teknik Elektro 2(2). Universitas Negeri Semarang.

S, A. K., Bandyopadhyay, A., Bhoomika, H., Singhania, I., & Shah, K. (2018). *Analysis of Network Traffic and Security through Log Aggregation*. International Journal of Computer Science and Information Security (IJCSIS) 16(6).

Supriyadi, A., & Gartina, D. (2015). *Memilih Topologi Jaringan dan Hardware dalam Desain Sebuah Jaringan Komputer*. Informatika Pertanian 16(2), 1037-1053. Sekretariat Badan Penelitian dan Pengembangan Pertanian.

Tambunan, B., & Raharjo, W. S. (2016). *Desain dan Implementasi Honeypot dengan Fwsnort dan PSAD sebagai Intrusion Prevention System*. Jurnal ULTIMA Computing, 5(1). Fakultas Teknik dan Informatika Universitas Multimedia Nusantara Tangerang.

Wongkar, S., & Sinsuw, A. (2015). *Analisa Implementasi Jaringan Internet Dengan Menggabungkan Jaringan LAN Dan WLAN Di Desa Kawangkoan Bawah Wilayah Amurang II*. E-Journal Teknik Elektro Dan Komputer, 4(6). Fakultas Teknik Elektro Universitas Sam Ratulangi Manado.

DAFTAR RIWAYAT PENULIS

**Data Pribadi**



Nama : R.Muhammad Sukmawan

Tmpt/tgl lahir : Lampung, 01 Juni 1998

Jenis kelamin : Laki-laki

Agama : Islam

Alamat` : PERUM II GPM Blok E. 048,

Bandar Mataram, Lampung Tengah, Lampung

Telepon (WA) : 082233732408

Email : muhammadrangga64@gmail.com

**Data Keluarga**

Nama Ayah : Mahmudi

Nama Ibu : Sugiati

Pekerjaan : Karyawan wiraswasta

Alamat : PERUM II GPM Blok E. 048, Bandar

Mataram, Lampung Tengah, Lampung

**Riwayat Pendidikan**

SMK Sugar Group, lulus tahun 2016

SMP Sugar Group, lulus tahun 2013.

SDS 02 GPM, lulus tahun 2010