BAB III
METODOLOGI

3.1 Analisis Kebutuhan Sistem

3.1.1 Kebutuhan Perangkat Keras

Pada penelitian yang dilakukan, adapun perangkat keras (*hardware*) yang dibutuhkan untuk menunjang implementasi pada sistem yang dibangun adalah sebagai berikut:

a. Komputer *Server* (IDS)

Komputer *Server* merupakan komputer yang digunakan sebagai pusat *monitoring* dan memiliki IDS di dalamnya. Kebutuhan minimal untuk membangun sistem tersebut yaitu:

1) CPU 1 Core, ruang penyimpanan 8 GB, memory 256 MB RAM
2) *Operating System* Linux 14.04 LTS

b. Komputer *Server*

Komputer *Server* merupakan komputer yang digunakan sebagai pusat layanan seperti FTP, SSH, *Web Server*. Komputer *server* ini akan dijadikan target penyerangan oleh *attacker*. Kebutuhan minimal untuk membangun sistem tersebut yaitu:

1) CPU 1 Core, ruang penyimpanan 8 GB, memory 256 MB RAM
2) *Operating System* Linux 14.04 LTS

c. Komputer *Attacker*

Komputer *Attacker* merupakan komputer yang digunakan untuk pengujian serangan terhadap IDS. Kebutuhan minimal untuk membangun sistem tersebut yaitu:

1) CPU 1 Core, ruang penyimpanan 8 GB, memory 256MB RAM
2) *Operating System* Kali Linux 1.0.9

3.1.2 Kebutuhan Perangkat Lunak

Sistem yang dibangun membutuhkan beberapa perangkat lunak (*software*) yang digunakan untuk menunjang fungsionalitas dan kinerja sistem. Adapun beberapa perangkat lunak yang harus dipasang dan dikonfigurasi pada komputer *server* maupun yang digunakan pada komputer *attacker* yaitu:

a. Snort 2.9.8.2

Snort adalah sebuah aplikasi yang berfungsi untuk mendeteksi adanya intrusi. Aplikasi *opensource* ini terdiri dari beberapa *engine* yang memiliki peran masing-masing terhadap adanya intrusi. Snort menggunakan *database rule* sebagai acuan untuk menetapkan paket sebagai sebuah intrusi. Paket yang terdeteksi sebagai intrusi akan dicatat dalam *file log* yang terletak pada direktori Snort.

b. Barnyard2 versi 2-1.14

Aplikasi pendukung pada sistem deteksi penyusupan yang menggunakan Snort yaitu Barnyard2. Aplikasi tersebut bekerja sebagai penerjemah untuk Snort unified2 *file output biner*. Fungsi utama dari Barnyard2 yaitu memungkinkan Snort untuk menulis ke *disk* secara efisien dan meninggalkan tugas parsing data biner ke dalam berbagai format sehingga tidak akan menyebabkan Snort kehilangan lalu lintas jaringan.

c. phpMyAdmin 4.6.0

phpMyAdmin adalah perangkat lunak (*software*) berbasis *web* bersifat *opensource* yang digunakan untuk menangani pengelolaan administrasi MySQL seperti *database*, tabel, relasi, perizinan, pengguna. Perangkat lunak phpMyAdmin menyediakan kemudahan untuk pengelolaan *database* serta dapat diakses dari mana saja. Dalam penggunaannya, pengguna tidak perlu repot untuk mengetik perintah seperti pada terminal. Data yang ditampilan phpMyAdmin berbentuk tabel yang dapat dikelola dengan sangat mudah sehingga mampu dipahami oleh pengguna secara jelas. Perangkat lunak tersebut dapat berjalan pada segala jenis *platform* mengingat phpMyAdmin berbasis *web* sehingga mampu dikelola dari lokasi manapun.

d. VMware 10.0.7

VMware merupakan salah satu *virtual machine* yang dapat digunakan untuk membuat virtualisasi seperti *server*. Dengan adanya VMware, sebuah *operating system* dapat dijalankan tanpa harus melakukan instalasi pada komputer fisik. VMware mampu membuat komputer *virtual* yang berjalan di dalam komputer fisik. Dengan keunggulan tersebut, penelitian ini akan memanfaatkan VMware untuk menjalankan *server* yang akan dibangun.

e. Nmap 7.12

Nmap merupakan aplikasi yang digunakan untuk melakukan pemindaian terhadap *port* pada sistem atau disebut juga *port scanning*. Aplikasi *opensource* tersebut telah tersedia dalam sistem operasi Kali Linux. Hasil dari *port scanning* yaitu untuk mengetahui celah yang terbuka dari sistem sehingga *attacker* dapat memanfaatkan celah tersebut untuk melakukan tindakan penyusupan. Nantinya, aplikasi nmap akan digunakan dalam pengujian untuk mengetahui kinerja IDS dalam mendeteksi adanya serangan.

f. Openssh-server 7.2

SSH (*Secure Shell*) digunakan untuk mengakses komputer secara *remote*. Dengan kata lain, pengguna tidak perlu repot untuk mengakses *server* secara langsung. Namun, hanya dengan ssh pengguna dapat mengontrol *server* dengan jarak jauh. Antarmuka SSH yaitu berbasis *command line*.

g. Vsftpd v3.0.2

FTP (*File Transfer Protocol*) berperan dalam proses pengiriman maupun peneriman *file*. Dengan kata lain, proses pengiriman *file* dijalankan oleh protokol yang disebut FTP. Salah satu aplikasi yang menjalankan proses tersebut yaitu Vsftpd.

h. Wkhtmltopdf

Wkhtmltopdf adalah aplikasi *opensource* (LGPLv3) yang dapat mengubah *file* HTML ke dalam format PDF menggunakan Qt WebKit rendering engine. Aplikasi tersebut dapat diunduh pada laman resmi

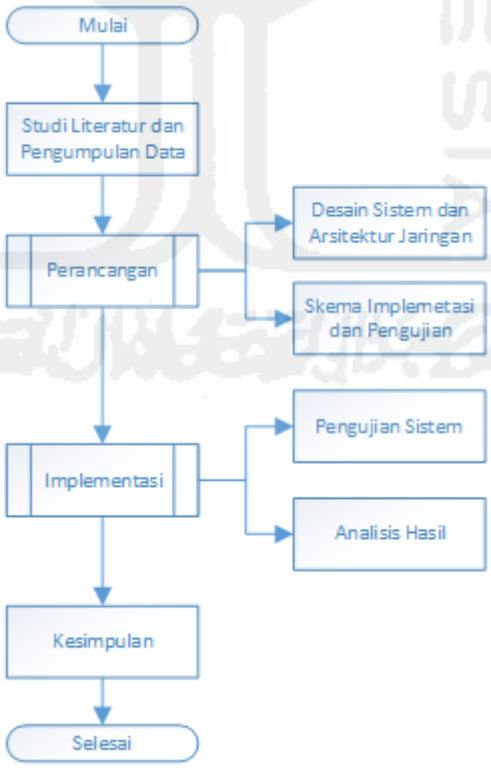
http://wkhtmltopdf.org/ secara gratis. Implementasinya, wkhtmltopdf ini membutuhkan aplikasi pendukung yaitu xvfb agar dapat mengkonversi. Hasil konversi dari aplikasi tersebut harus disertakan pada perintah di terminal.

i. BASE

BASE atau *Basic Analysis and Security Engine* merupakan aplikasi yang berfungsi sebagai *engine* yang menyimpan semua aktifitas serangan yang terdeteksi oleh Snort ke dalam *database*. Selain itu, BASE menyediakan antarmuka untuk menampilkan *log* dan menganalisis peringatan yang dihasilkan oleh Snort.

3.2 Diagram Alur Penelitian

Penelitian yang akan dilakukan melalui beberapa tahap yang dapat dilihat pada Gambar 3.1 di bawah ini:



Gambar 3.1 Diagram Alur Penelitian

Pada gambar alur penelitian di atas, tahapan studi literatur dan pengumpulan data dilakukan dengan mengumpulkan beberapa referensi seperti penelitian sebelumnya serta sumber-sumber lain yang mendukung penelitian ini. Pengumpulan data yang dilakukan bertujuan untuk memperoleh informasi mengenai keamanan sistem dan jenis serangan yang tergolong sebagai intrusi. Selain itu, dilakukan juga perbandingan antara penelitian yang sudah ada. Hal ini memungkinkan untuk penambahan atau bahkan mengurangi fitur yang ada.

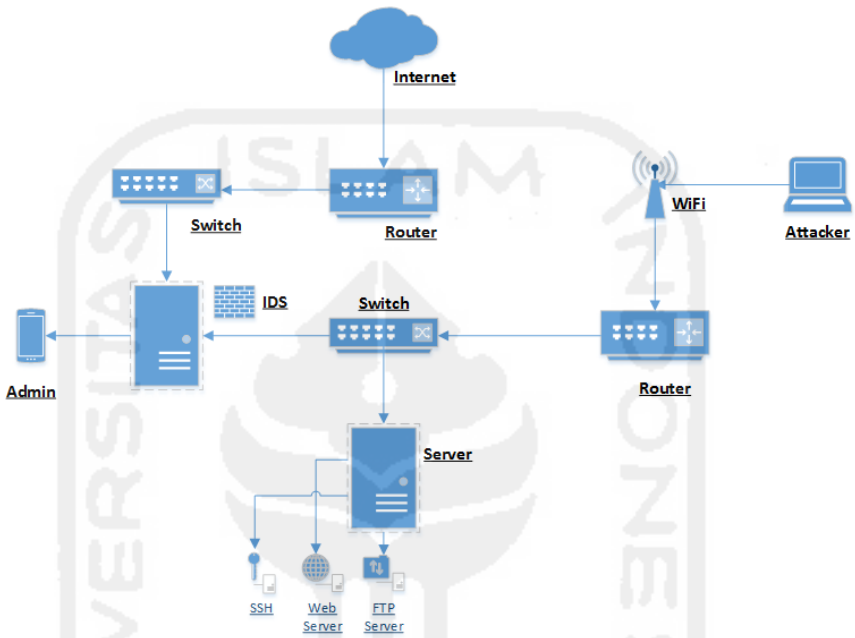
Setelah tahapan studi literatur, kemudian dilakukan tahapan mengenai perancangan, desain sistem dan arsitektur jaringan. Pada tahap perancangan, ditentukan kebutuhan serta komponen yang digunakan dalam sistem. Komponen tersebut dapat berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang akan digunakan. Selain dari segi kebutuhan dan komponen sistem, perancangan sistem dan arsitektur jaringan pun menjadi hal yang perlu diperhatikan. Desain arsitektur jaringan digambarkan bagaimana penempatan sistem Snort sebagai IDS yang dibangun sedangkan desain sistem akan diperlukan dalam pembuatan sistem.

Pada tahap implementasi, desain sistem maupun jaringan yang telah dikembangkan kemudian diterapkan.Tahap implementasi penelitian dimulai dari penyusunan rencana secara detail dan kebutuhan perangkat yang digunakan untuk membangun sistem hingga proses instalasi. Setelah tahap implementasi, dilakukan tahap pengujian untuk melakukan uji coba terhadap sistem yang telah dibangun dengan beberapa skenario pengujian yang telah ditetapkan. Tahapan ini menentukan bagaimana kinerja sistem berdasarkan skenario yang telah diimplementasikan. Dari tahap pengujian tersebut akan dianalisis dari berbagai aspek, mulai dari kinerja hingga sistem memberikan respon berupa *alert* akibat dari hasil pengujian.

Berdasarkan hasil analisa pengujian sistem, kemudian dirangkum dan ditarik kesimpulan terhadap sistem yang telah dibangun. Hasil kesimpulan tersebut menentukan apakah sistem IDS yang dibangun berjalan secara efektif dan sesuai dengan tujuan dari penelitian ini.

3.3 Gambaran Umum Sistem

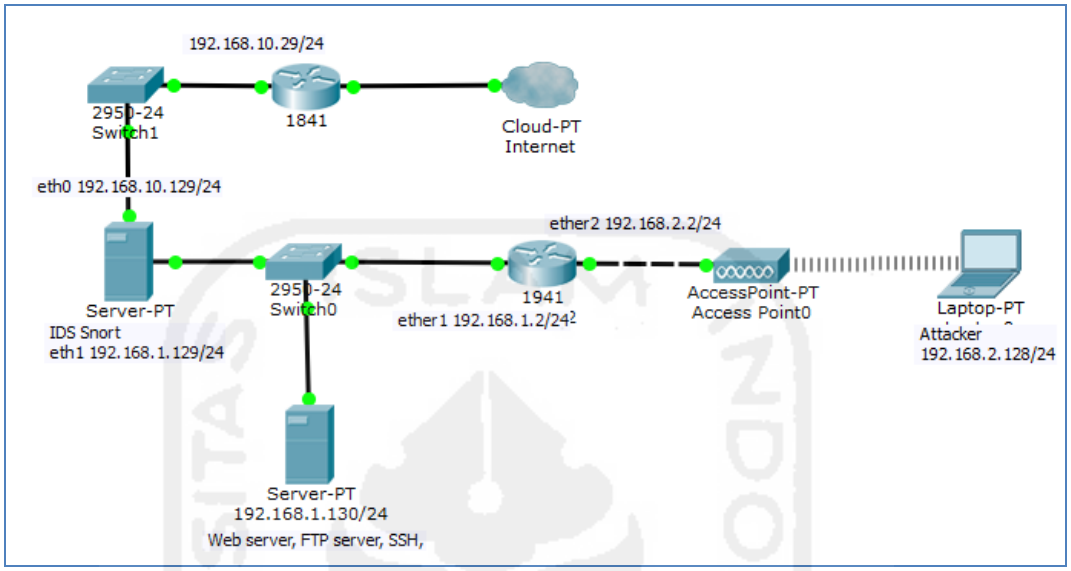
Sistem IDS yang dibangun dalam penelitian ini memiliki arsitektur jaringan yang memberikan gambaran secara jelas mengenai interkoneksi antar perangkat satu dengan perangkat lainnya. Arsitektur IDS tersebut dapat dilihat pada Gambar 3.2



Gambar 3.2 Implementasi IDS

Sistem IDS yang dibangun menggunakan Snort merupakan sistem yang dapat mendeteksi adanya intrusi pada jaringan. IDS tersebut akan menganalisa paket sebelum dilanjutkan ke *server*. *Server* yang dibangun menggunakan *virtual machine* yang artinya *server* tersebut merupakan komputer virtual yang kemampuannya sama seperti komputer asli pada umumnya. Sistem IDS yang dibangun memiliki kapasitas 8 GB dengan dua buah *network interface card* (NIC). Masing-masing NIC tersebut terhubung dengan komputer *server* dan jaringan luar. Komputer server diberikan alamat IP yang bersifat statis. Komputer server tersebut memiliki layanan seperti Web Server, FTP, dan SSH. Ketika terdeteksi adanya intrusi maupun serangan ke server, maka paket data yang masuk akan dianalisa oleh komputer IDS. Jika terindentifikasi sebagai sebuah intrusi, selanjutnya akan dihasilkan peringatan yang disimpan pada *log file* dan *database*

yang kemudian dikirimkan kepada Administrator secara *real time*. Secara rinci, pengalamatan masing-masing komputer dapat dilihat pada Gambar 3.3



Gambar 3.3 Topologi Jaringan

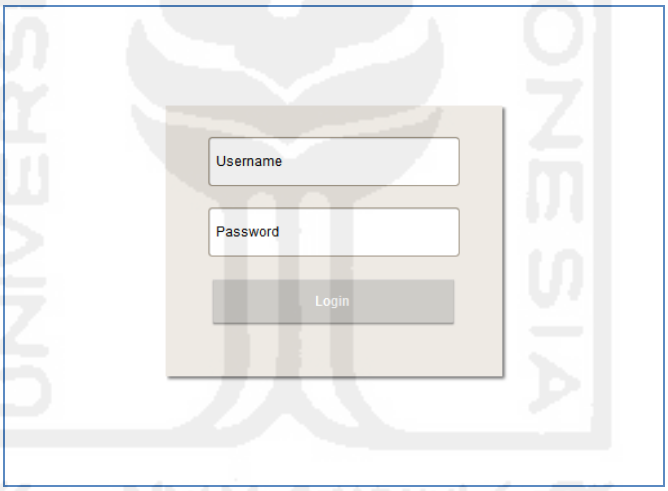
Pada gambar di atas menunjukkan rancangan topologi jaringan yang akan diimplementasikan pada penelitian ini. Pada IDS akan terkoneksi *Internet* karena untuk mengirimkan notifiksi melalui Telegram. Pada IDS akan menggunakan IP yang bersifat NAT dari *virtual machine* yaitu 192.168.10.129/24 dan IP statis yaitu 192.168.1.129/24. Kemudian pada komputer *server* akan diberikan IP statis yaitu 192.168.1.130. Sedangkan untuk pengujian yang digunakan oleh *attacker* memiliki alamat 192.168.2.128/24. Karena *server* target dan *attacker* berbeda jaringan, maka dibutuhkan sebuah router agar kedua jaringan tersebut dapat berkomunikasi.

3.4 Desain Sistem

Hasil dari perencanaan yang telah dipaparkan pada gambaran umum sistem, desain implementasi sistem dirancang menyesuaikan keadaaan lingkungan objek penelitian. Desain implementasi harus memperhatikan faktor kesederhanaan infrastruktur namun tetap menjaga kinerja sistem secara optimal.

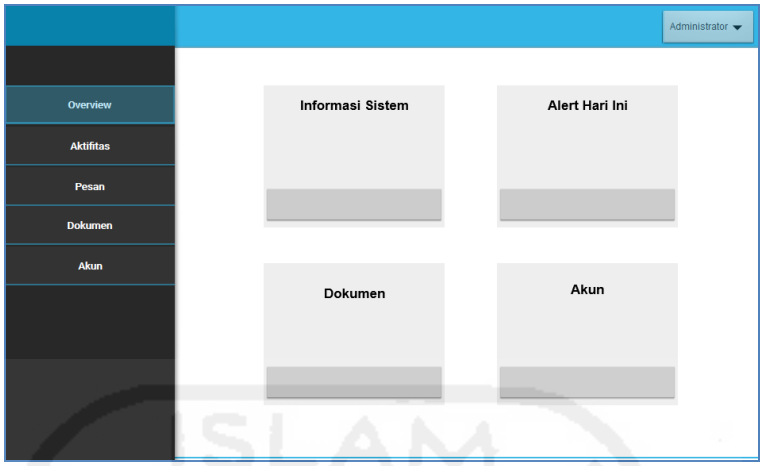
3.4.1 Desain Antarmuka

Laporan mengenai deteksi intrusi dapat diakses melalui *web-based*. Dengan adanya laporan berbasis *web* tersebut, administrator dapat melihat maupun melakukan pengaturan terhadap akun Telegram yang digunakan. Akan terdapat dua *web* yang dapat digunakan, yaitu *web* yang menampilkan laporan singkat beserta akun Telegram dengan memanfaatkan *framework* Code Igniter dan *web* untuk menganalisa hasil deteksi Snort yaitu *Basic Analysis and Security Engine* (BASE). Laporan berbasis *web* dibuat dengan antarmuka (*interface*) sederhana, tetapi dapat dimanfaatkan sebaik mungkin oleh Administrator. Pada BASE, Administrator akan disajikan informasi lengkap megenai insiden yang terdeteksi. Antarmuka awal yaitu halaman login yang dapat dilihat pada Gambar 3.4.



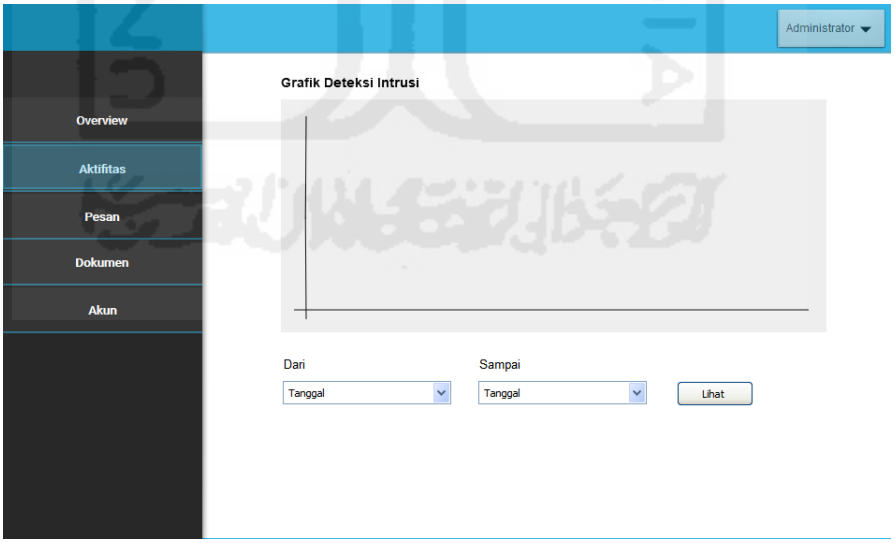
Gambar 3.4 Antarmuka Halaman *Login*

Pada Gambar 3.4 merupakan tampilan antarmuka dimana terdapat kolom yang harus diisi ketika pengguna akan masuk ke dalam sistem untuk melihat laporan hasil deteksi intrusi. Kolom tersebut berkaitan dengan informasi seperti *username* dan *password*, kemudian terdapat tombol *login* untuk mengeksekusi perintah. Setelah berhasil *login*, maka akan diarahkan ke halaman dashboard yang ditampilkan pada Gambar 3.5.



Gambar 3.5 Antarmuka Halaman *Dashboard*

Pada halaman Administrator yang ditunjukkan Gambar 3.5 terdapat beberapa menu yang memiliki fungsi tertentu. Menu-menu tersebut mempunyai kegunaan yang berbeda, seperti pada menu Overview akan menampilkan informasi terkait informasi sistem, jumlah serangan hari ini maupun total serangan serta informasi akun. Selain itu, juga terdapat menu Aktifitas yang ditunjukkan pada Gambar 3.6

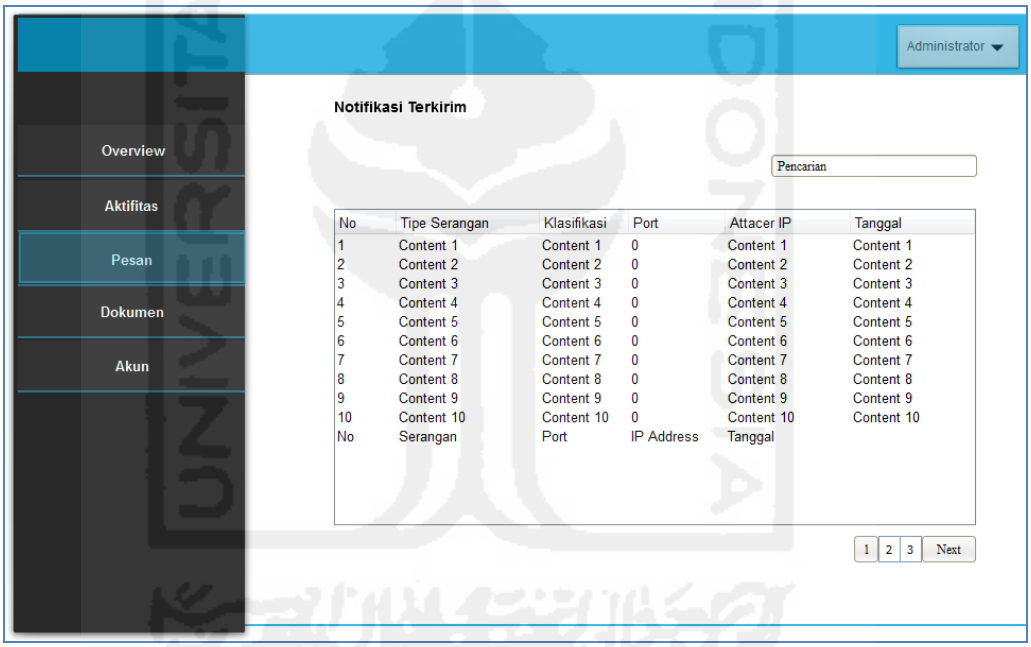


Gambar 3.6 Aktifitas Laporan Insiden

Pada antarmuka halaman aktifitas laporan insiden (Gambar 3.6) ditampilkan laporan hasil deteksi terhadap paket yang teridentifikasi sebagai intrusi. Laporan

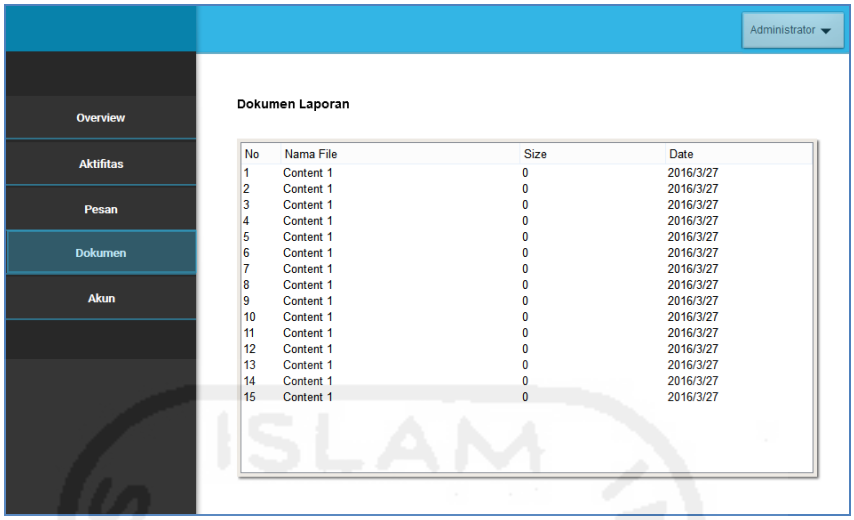
tersebut disajikan dalam bentuk grafik yang menunjukan kuantitas serangan. Di bawah grafik tersebut terdapat informasi lengkap dalam bentuk tabel. Terdapat fitur untuk menampilkan informasi serangan dalam jangja waktu tertentu.

Selain menu aktifitas, terdapat menu pesan berfungsi untuk menampilkan notifikasi yang telah dikirim ke administrator mengenai adanya intrusi. Selain mendapatkan peringatan melalui aplikasi *instant messaging* Telegram. administrator juga dapat melihat informasi notifikasi yang telah terkirim melalui menu notifikasi dari menu administrator. Antarmuka menu notifikasi dapat dilihat pada Gambar 3.7



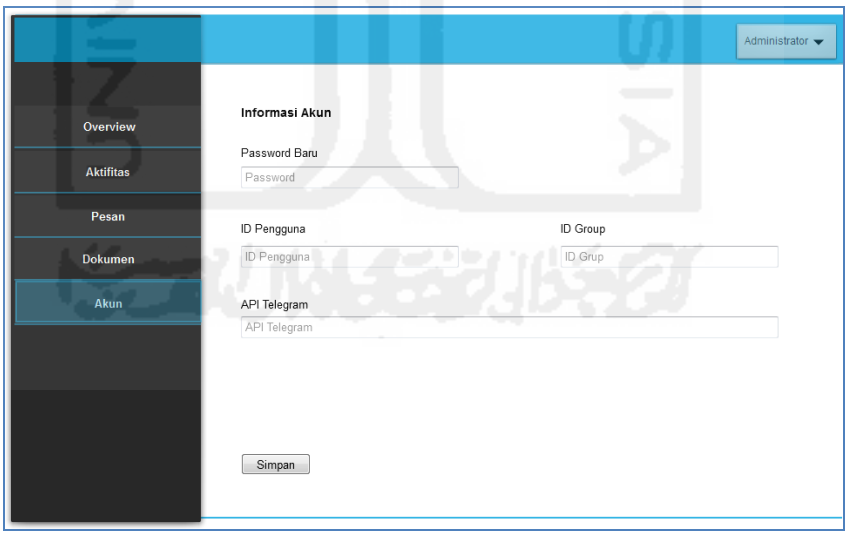
Gambar 3.7 Menu Notifikasi Terkirim

Informasi yang terdapat pada menu notifikasi tersaji dalam sebuah tabel. Tabel tersebut berisikan notifikasi insiden yang dialami oleh sistem yaitu mengenai jenis serangan, klasifikasi serangan, *IP Address*, isi pesan, dan tanggal tepat insiden penyusupan terjadi. Selain itu, terdapat menu dokumen yang menampilkan dokumentasi laporan mengenai keamanan sistem dalam jangka waktu tertentu yang dapat dilihat pada Gambar 3.8.



Gambar 3.8 Antarmuka Menu Dokumen

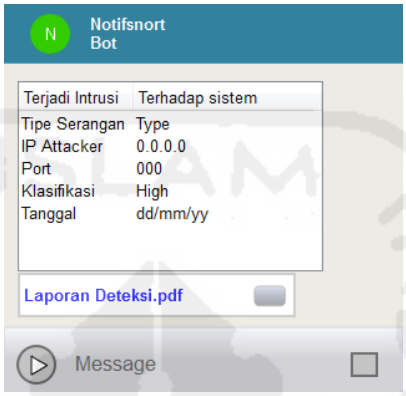
Dokumentasi yang ditampilkan merupakan kumpulan dokumen yang terletak pada sebuah direktori. Informasi yang tersaji berupa nama *file*, ukuran *file*, serta tanggal pembuatan *file*. Selanjutnya terdapat menu akun yang dapat dilihat pada Gambar 3.8.



Gambar 3.9 Antarmuka Menu Akun

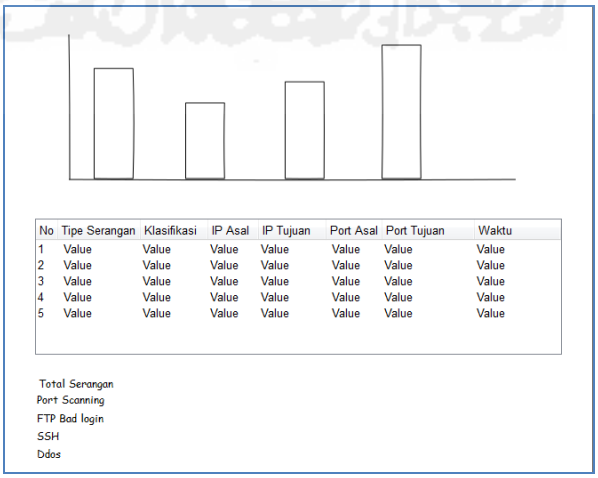
Menu akun menampilkan kolom untuk mengganti *password* akun dan mengisi kode ID pengguna maupun ID grup pada aplikasi *instant messaging* Telegram. ID Pengguna merupakan id yang digunakan dalam pengiriman

notifikasi ke administrator, sedangkan id grup digunakan untuk pemberitahuan terhadap grup Telegram. Pada menu akun terdapat fitur untuk mengaktifkan maupun menonaktifkan pengiriman notifikasi. Antarmuka pada aplikasi *instant messagig* Telegram dapat dilihat pada Gambar 3.9



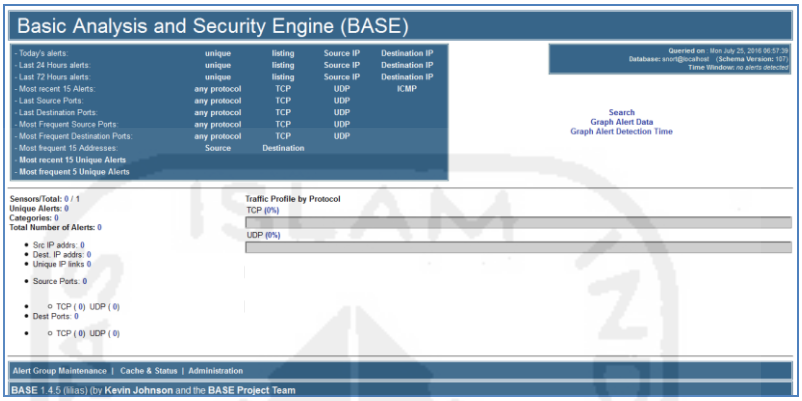
Gambar 3.10 Antarmuka Notifikasi Telegram

Pada gambar di atas merupakan ilustrasi notifikasi jika terdapat sebuah serangan pada *server*. Pesan yang dikirimkan sistem ke administrator berupa informasi singkat mengenai insiden terjadinya intrusi. Informasi tersebut berupa tipe serangan, klasifikasi serangan, *ip address attacker*, *port* tujuan, dan tanggal insiden terjadi. Selain itu, Administrator juga mendapatkan *attachment file* berupa dokumen digital degan format PDF yang dapat dilihat pada Gambar 3.10.



Gambar 3.11 Dokumen Deteksi Intrusi

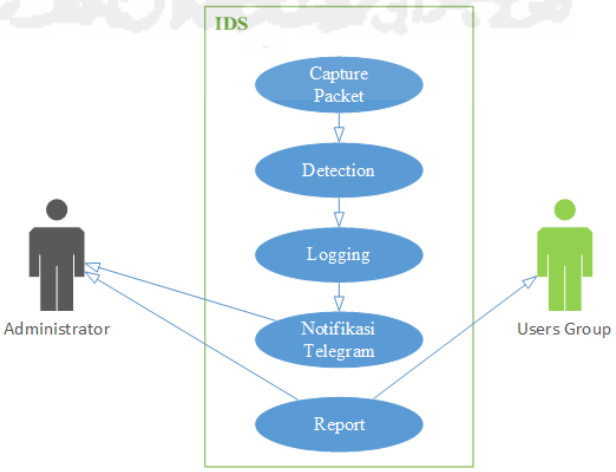
Informasi khusus untuk menganalisa insiden yang terjadi disajikan dalam BASE berbasis *web* yang memiliki tampilan sederhana dan mencakup semua *log* hasil deteksi Snort. Tampilan antarmuka BASE dapat dilihat pada gambar 3.12.



Gambar 3.12 Antarmuka BASE

3.5 Proses Kerja Sistem

Perancangan dibuat sebagai visualisasi dari alur kerja sistem dimana dapat menjadi sebuah standar atau disebut juga dengan *Unified Modelling Language* (UML). Diagram Use case merupakan salah satu jenis diagram UML yang menggambarkan interaksi antara sistem dan aktor. Cara kerja sistem yang akan dibangun dapat mudah dipahami oleh pengguna. Adapun perancangan sistem pada penelitian ini yang dapat dilihat pada Gambar 3.13.

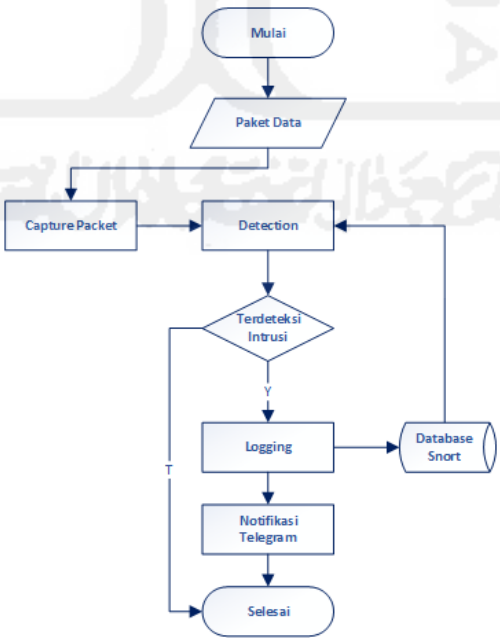


Gambar 3.13 Diagram Use Case Sistem

Pada Gambar 3.13 menunjukkan bagaimana gambaran cara kerja Snort sebagai *Intrusion Detection System*. Snort akan menangkap paket dalam lalu lintas jaringan menggunakan packet *capture-decode engine*. Kemudian akan dilakukan pendeteksian terhadap paket dengan membadingkan aturan yang ada. Jika teridentifikasi sebagai intrusi, maka akan dilakukan pencatatan yang nantinya dihasilkan sebuah peringatan secara *real time* maupun dalam bentuk dokumen digital.

Use case tersebut mempresentasikan sebuah interaksi antara aktor dengan sistem. Terdapat dua aktor yaitu admininstrator dan *users group* dimana memiliki peran yang sama untuk mendapatkan notifikasi mengenai deteksi terhadap intrusi. Namun, hal yang membedakan yaitu jenis notifikasai yang dikirimkan oleh sistem kepada kedua aktor. Selain mendapatkan notifikasi serangan secara real time, Adminisrtrator juga akan menerima laporan berupa dokumen digital dalam format PDF. Berbeda dengan Administrator, users group hanya mendapatkan laporan dalam format PDF. Berikut proses kerja sistem dari mendeteksi serangan hingga pengiriman notifikasi:

3.5.1 Alur Deteksi Serangan

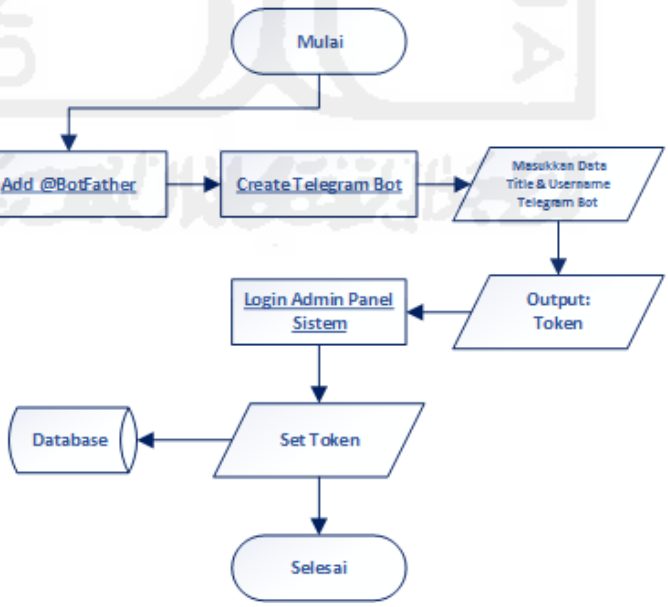


Gambar 3.14 *Flowchart* Deteksi Serangan

Pada Gambar 3.14 menggambarkan alur bagaimana sistem bekerja dalam mendeteksi intrusi dan memberika peringatan secara *real time*. Pertama, paket yang masuk akan ditangkap dan dianalisa oleh Snort berdasarkan aturan yang telah ditetapkan. Jika paket tersebut tidak terdeteksi sebagai sebuah intrusi atau serangan, maka paket tersebut akan dibuang dan proses berakhir. Namun, ketika paket tersebut terdeteksi sebagai sebuah intrusi, selanjutnya akan dilakukan pencatatan pada *file log* maupun *database*. Setelah dicatat dan disimpan dalam *database* maka akan terjadi sebuah *trigger* untuk mengeksekusi *file* php yang berfungsi untuk mengirimkan notifikasi Telegram terhadap Administrator.

3.5.2 Telegram *Bot Token* dan ID Pengguna

Untuk pemberitahuan peringatan intrusi dilakukan dengan memanfaatkan media *instant messaging* Telegram dengan menggunakan fitur Telegram *bot*. Hal yang utama dalam penggunaan fitur tersebut yaitu dengan memiliki *token* atau kode otentikasi yang dapat mengatur tindakan Telegram *bot*. Alur mendapatkan *bot* serta menyimpannya dalam *database* sistem dapat dilihat pada Gambar 3.15 berikut ini.

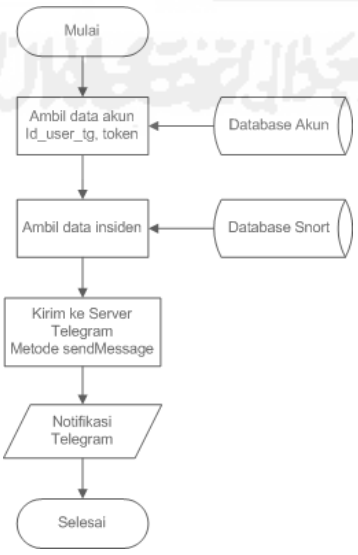


Gambar 3.15 *Flowchart* Set Telegram *Bot Token*

Flowchart yang ditunjukkan pada Gambar 3.15 menjelaskan bagaimana mendapatkan *token* Telegram *bot* dan menyimpannya ke dalam *database* sistem. Sebelum mendapatkan *token*, pengguna diwajibkan memiliki aplikasi *instant messaging* Telegram yang telah terpasang pada *smartphone.* Untuk mendapatkan *token*, pengguna dapat menambahkan @BotFather atau mengakses alamat *website* http://telegram.me/BotFather. *Token* tersebut kemudian disimpan dalam *database* sistem IDS untuk digunakan sebagai media pemberitahuan alarm peringatan adanya intrusi.

Selain membutuhkan *token*, hal yang harus diketahui yaitu ID pengguna telegram yang digunakan oleh administrator jaringan. Untuk mengetahui ID Pengguna, administrator harus memulai obrolan terhadap *bot* tersebut. Kemudian mengunjungi situs Telegram dengan alamat https://api.telegram.org/bot <TOKEN> /getUpdates dimana <TOKEN> diisi dengan token telegram *bot*. Metode *getUpdates* digunakan untuk memperoleh pembaharuan dari Telegram *bot*. Maka dari itu, didapatkan ID pengguna yang nantinya akan disimpan dalam *database* yang digunakan untuk mengetahui identitas administrator dan mengirimkan notifikasi dari sistem.

3.5.3 Alur Kirim Notifikasi

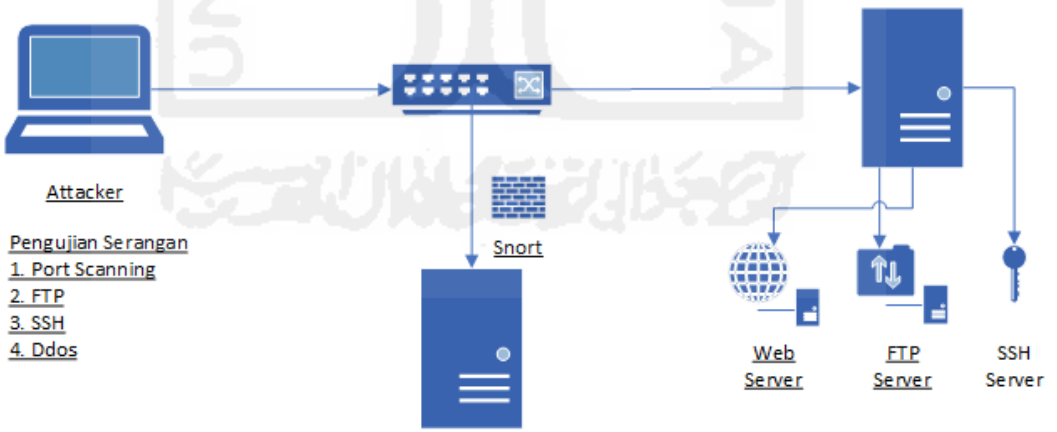


Gambar 3.16 Alur Kirim Notifikasi

Paket yang telah terdeteksi sebagai intrusi akan disimpan dalam *database* yang kemudian akan memicu proses pengiriman notifikasi kepada administrator. Metode yang digunakan oleh Telegram bot untuk mengirimkan pesan yaitu *sendMessage*. Perintah dasar untuk mengirimkan pesan yaitu dengan mengakses alamat API Telegram https://api.telegram.org/bot<TOKEN>/sendMessage?chat_id=<id_pengguna>&text=<notifikasi>. ID Pengguna dan token diambil dari *database* akun, sedangkan informasi insiden diambil dari *database* Snort. Parameter yang digunakan untuk mengirimkan isi pesan yaitu *text*. Format pesan notifikasi yang dikirimkan kepada administrator yaitu informasi mengenai tipe serangan, klasifikasi, *IP Address attacker*, dan waktu insiden terjadi.

3.6 Pengujian Sistem

Pengujian sistem merupakan tahapan terakhir untuk melakukan evaluasi atau uji coba dari penelitian yang yang telah dilakukan berdasarkan tahap implementasi. Evaluasi ini bertujuan untuk menentukan seberapa efektif sistem ini dapat mendeteksi adanya intrusi. Skema pengujian terhadap sistem dapat dilihat pada gambar berikut:



Gambar 3.17 Skenario Pengujian Sistem

Pada gambar di atas, pengujian dilakukan dengan beberapa tipe serangan dimana dilakukan ketika Snort telah diaktifkan. Selain itu, pengujian ini

memastikan bahwa sistem dapat memberikan peringatan berupa notifikasi kepada administrator atau tidak.

3.6.1 Skenario Pengujian

Pada tahap pengujian terdiri dari beberapa skenario pengujian yang dilakukan dalam beberapa kondisi, seperti:

a. Pengujian komputer *attacker* menggunakan Kali Linux untuk melakukan percobaan serangan menggunakan beberapa teknik yang berbeda yaitu *Port Scanning* (SYN,FIN,XMAS) , FTP *Bad login,* SSH *brute force*, Ddos *Attack.*

b. Pengujian kinerja IDS dilakukan secara bergantian untuk mengetahui efektifitas *alert* yang dikirimkan ke Administrator melalui aplikasi *instant messaging* Telegram secara *real time*.

c. Pengujian selanjutnya yaitu membuat laporan hasil deteksi dalam bentuk dokumen digital yang secara otomatis akan dikirimkan ke administrator dalam jangka waktu tertentu.

3.7 Pengujian Sistem

Dari hasil pengujian penelitian tersebut kemudian akan dilakukan analisa terhadap insiden serangan yang teridentifikasi. Analisa tersebut membandingkan waktu intrusi yang teridentifikasi dari masing-masing tipe serangan. Hal ini untuk mengetahui tingkat akurasi waktu dari mulai penyerangan hingga notifikasi terkirim.

Tabel 3.1 Analisis Pengujian Serangan

No	Tipe Serangan	Tingkat Akurasi Waktu (*timestamp*)		
		Awal Serangan	Terdeteksi	Terkirim

Pada Tabel 3.1 merupakan tabel yang digunakan untuk menganalisa serangan yang dilakukan terhadap sistem dimana terdapat aturan yang telah

ditetapkan. Skenario pengujian ini bertujuan bertujuan untuk mengetahui apakah sistem mampu mendeteksi adanya serangan hingga memberikan notifikasi.

3.7.1 Kuisioner Pengujian

Pengujian sistem ini diharapkan sistem yang dibuat mempermudah administrator untuk mengetahui keadaan *server* yanng dilindungi. Pengujian dilakukan dengan wawancara dan simulasi pengujian serangan kepada pihak yang berkompeten dalam masalah keamanan. Beberapa pertanyaan yang akan diajukan kepada pihak-pihak terkait sebagai berikut:

a. Apakah Snort sebagai IDS sudah berjalan dengan baik dalam pendeteksian serangan ?
b. Apakah aturan (*rule*) dapat mendeteksi serangan sesuai pengujian ?
c. Apakah IDS menyimpan intrusi dalam *log file* dan *database* ?
d. Apakah *web based* untuk *monitoring* dapat memberikan informasi yang sesuai ?
e. Apakah notifikasi pada Telegram mengenai serangan sesuai dengan pengujian ?
f. Apakah IDS mengirimkan laporan dokumen PDF mengenai serangan yang terdeteksi dalam jangka waktu tertentu?

Pengujian sistem diperlukan untuk memastikan apakah sistem yang telah dibuat sudah berjalan sesuai dengan apa yang diharapkan. Pada Tabel 3.2 di bawah dapat ditarik kesimpulan dapakah hasil pengujian sistem sesuai dengan tahap pegujian sesuai dengan skenario yang telah ditetapkan.

Tabel 3.2 Analisis Pengujian Serangan

No	Skenario Pengujian Sistem	Uji Coba	Hasil yang diharapkan	Hasil pegujian sistem	Kesimpulan