

**DETEKSI SERANGAN MENGGUNAKAN *HONEYPOT*  
BERBASIS ANDROID**

**LAPORAN PROYEK AKHIR**



**Proyek Akhir ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat  
Kelulusan Progam Studi Diploma III Teknik Informatika dan Menciptakan  
Gelar Ahli Madya (A.Md.)**

**Oleh :**

**VERRANDY BAGUS PRASETYO**

**NIM. 361755401030**

**PROGAM STUDI DIPLOMA III  
TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BANYUWANGI  
2021**

*--Halaman ini sengaja dikosongkan--*

**DETEKSI SERANGAN MENGGUNAKAN *HONEYPOT*  
BERBASIS ANDROID**

**LAPORAN PROYEK AKHIR**



**Proyek Akhir ini Dibuat dan Diajukan untuk Memenuhi Salah Satu Syarat  
Kelulusan Progam Studi Diploma III Teknik Informatika dan Menciptakan  
Gelar Ahli Madya (A.Md.)**

**Oleh :**

**VERRANDY BAGUS PRASETYO**

**NIM. 361755401030**

**PROGAM STUDI DIPLOMA III  
TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BANYUWANGI  
2021**

*--Halaman ini sengaja dikosongkan--*

## **PERSEMBAHAN**

Alhamdulillah puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga Proyek Akhir ini dapat diselesaikan pada waktu yang telah ditentukan.

Proyek akhir ini saya persembahkan untuk:

1. Bapak Supadni dan Ibu Sri Sutilahwati selaku kedua orang tua yang telah senantiasa mendoakan serta memberikan dukungan dan semangat.
2. Kakak Verry Hadiwianto dan Saudari Virra Nur Cahayani yang selalu mendampingi saya.
3. Herman Yulianto, S.T., M.T. selaku Dosen pembimbing 1 dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom., selaku Dosen pembimbing 2. Terima kasih telah meluangkan waktu, memberikan serta membagikan ilmu dan pengalamannya, dan kesabaran untuk membimbing saya demi kelancaran Proyek Akhir ini.
4. Bapak Ibu Dosen dan staf Politeknik Negeri Banyuwangi khususnya Dosen dan staf Program Studi Teknik Informatika yang telah dengan ikhlas membimbing saya selama menjadi mahasiswa. Almamater Politeknik Negeri Banyuwangi yang saya banggakan.
5. Almamater Politeknik Negeri Banyuwangi yang saya banggakan.
6. Seluruh teman-teman mahasiswa Politeknik Negeri Banyuwangi khususnya Program Studi Teknik Informatika seperjuangan.

*--halaman ini sengaja dikosongkan--*

## **MOTTO**

“Cuma orang orang yang terus melangkah yang akan mengetahuinya”

*(Eren Jaeger)*

*--Halaman Ini Sengaja dikosongkan--*



## **PERNYATAAN**

Saya yang bertanda tangan dibawah ini :

Nama : Verrandy Bagus Prasetyo

NIM : 361755401030

Menyatakan dengan sesungguhnya bahwa proyek akhir yang berjudul “Deteksi Serangan Menggunakan Honeypot Berbasis Android” adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan atau plagiat. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Banyuwangi, .....

Yang menyatakan

Verrandy Bagus Prasetyo

NIM. 367155401030

*--Halaman ini sengaja dikosongkan--*

**DETEKSI SERANGAN MENGGUNAKAN *HONEYPOT*  
BERBASIS *ANDROID***

Proyek Akhir ini disusun untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Ahli Madya (A.Md)  
Politeknik Negeri Banyuwangi

Oleh :

**VERRANDY BAGUS PRASETYO**  
NIM. 361755401030

Tanggal Ujian : 17 Februari 2021

Menyetujui,

Pembimbing 1 : Herman Yulindoko, S.T., M.T.

(.....)

Pembimbing 2 : Alif Akbar Fitrawan, S.Pd., M.Kom.

(.....)

Penguji 1 : Junaedi Adi Prasetyo, S.T., M.Sc.

(.....)

Penguji 2 : Farizqi Panduardi, S.ST., M.T.

(.....)

Mengesahkan,  
Ketua Jurusan  
Teknik Informatika



**Eka Misfika Rini, S.Kom., M.Kom.**  
NIP. 198310202014042001

Mengetahui,  
Koordinator Program Studi  
D3 Teknik Informatika



**Moh. Dimyati A., S.T., M.Kom.**  
NIK. 2008.36.004

*--Halaman ini dikosongkan--*

## DETEKSI SERANGAN MENGGUNAKAN HONEYPOT BERBASIS ANDROID

Nama Mahasiswa : Verrandy Bagus Prasetyo  
NIM : 361755401030  
Dosen Pembimbing : 1. Herman Yuliandoko, S.T., M.T.  
2. Alif Akbar Fitrawan, S.Pd., M.Kom.

### ABSTRAK

Seiring dengan berkembangnya teknologi informasi, teknologi jaringan komputer juga ikut berkembang. Peranan teknologi jaringan komputer sebagai *resource* yang dibutuhkan untuk mengakses data dan bertukar informasi. Namun semakin berkembangnya jaringan komputer maka juga berkembang tindak kejahatan dalam bidang teknologi jaringan atau juga disebut *cyber attacking*. Serangan serangan tersebut dapat membuat kinerja sebuah *resource/server* tidak maksimal seperti contohnya *DDoS*, *DDoS* adalah sebuah serangan yang mengirimkan sebuah *request* ke dalam *server* secara menerus dengan transaksi data yang besar yang menyebabkan *server overload* hingga tidak mampu menampung semua *request*. adapun tipe serangan diperuntukan mengambil sebuah *resource* dengan cara mencoba segala kemungkinan *username* dan *password* yang dapat mengakses sebuah *user root resource* disebut juga *Bruteforce attacks*. Dalam permasalahan ini dibuatlah sistem dan aplikasi “Deteksi Serangan Menggunakan Honeypot Berbasis Android” merupakan sebuah aplikasi yang dapat memonitoring sebuah serangan terhadap server dengan cara memasang *tool honeypot* ke dalam server sebagai pendeteksi serangan kemudian log *honeypot* dijadikan *Rest Api* menggunakan *framework Django* dengan *database MongoDB* lalu mengaksesnya menggunakan *android*. Aplikasi ini dapat



mengetahui sebuah serangan yang terjadi ke dalam sebuah *server* lalu dapat memblokirnya. Aplikasi ini digunakan untuk mempermudah user dalam memantau sebuah serangan serta dapat melakukan pemblokiran secara mudah.

Kata kunci : *Django, Honeypot, Web, DDoS, Bruterforce, Android*





## **ATTACK DETECTION USING HONEYPOT BASED ON ANDROID**

Nama Mahasiswa : Verrandy Bagus Prasetyo  
NIM : 361755401030  
Dosen Pembimbing : 1. Herman Yuliandoko, S.T., M.T.  
2. Alif Akbar Fitrawan, S.Pd., M.Kom.

### **ABSTRACT**

Along with the development of information technology, computer network technology is also developing. The role of computer network technology as a resource needed to access data and exchange information. However, as computer networks develop, crime in the field of network technology also develop or also called cyber attacking. These attacks can make a resource/server performance not full performance called DDoS , DDoS is an attack that send a request to the server continuously with large data transactions which causes the server to overload so that it cannot accommodate all requests. As for the type of attack intended to takes a resource by trying all possible usernames and passwords which can access a user root resources are also called Brute force attacks. In this problem, the system and application “Attack Detection Using Honeypot Based on Android” is an application that is made can monitor attacks on the server by install honeypot tool into the server as an attack detector, then the honeypot log is used to Rest using the django framework with the mongoDB database and then accessing it using android. This application can detect an attack that has occurred to a server and then can block it. This application is used make easier for users to monitor attack and can easily block them.

keyword : *Django, Honeypot, Web, DDoS, Bruterforce, Android*

*--Halaman Ini Sengaja dikosongkan--*

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Proyek Akhir yang berjudul “Deteksi Serangan Menggunakan Honeypot Berbasis Android”. Proyek Akhir ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Diploma Tiga (D3) pada Program Studi Teknik Informatika Politeknik Negeri Banyuwangi.

Penyusunan Proyek Akhir ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terimakasih kepada :

1. Bapak Supandi dan Ibu Sri Sutilahwati selaku kedua orang tua saya, yang tidak putus mendoakan dan memberi dukungan selama ini.
2. Bapak Son Kuswadi, Dr. Eng. selaku direktur Politeknik Negeri Banyuwangi.
3. Bapak Herman Yuliandoko, S.T., M.T. dan Bapak Alif Akbar Fitrawan, S.Pd., M.Kom., selaku pembimbing yang telah memberikan banyak ilmu dan dukungan kepada penulis.
4. Serta semua pihak yang terkait yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa Proyek Akhir ini masih jauh dari kata sempurna, oleh karena itu penulis sangat mengharapkan segala kritik dan saran dari semua pihak yang bersifat membangun. Penulis berharap Proyek Akhir ini dapat bermanfaat khususnya bagi penulis dan umumnya bagi pembaca.

Banyuwangi,

Verrandy Bagus Prasetyo  
NIM. 361755401030

*--Halaman Ini Sengaja dikosongkan--*

## DAFTAR ISI

COVER LUAR .....	i
COVER DALAM .....	iii
PERSEMBAHAN .....	v
MOTTO .....	vii
PERNYATAAN.....	ix
PENGESAHAN .....	xi
ABSTRAK .....	xiii
ABSTRACT.....	xvii
KATA PENGANTAR .....	xix
DAFTAR ISI.....	xxi
DAFTAR GAMBAR .....	xxvii
DAFTAR TABEL.....	xxxix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
BAB 2 TINJAUAN PUSTAKA .....	5
2.1 Dasar Teori Pendukung.....	5
2.1.1 Politeknik Negeri Banyuwangi (Poliwangi) .....	5
2.1.2 Jaringan Komputer .....	5
2.1.3 Keamanan Jaringan .....	6
2.1.4 <i>Linux</i> .....	7
2.1.5 <i>Android</i> .....	7
2.1.6 <i>Honeypot</i> .....	7
2.1.7 <i>Dionaea</i> .....	8
2.1.8 <i>Kippo</i> .....	9
2.1.9 <i>Python</i> .....	10
2.1.10 <i>Django Framework</i> .....	11



2.1.11 <i>Android Studio IDE</i> .....	11
2.1.12 Serangan Terhadap <i>Server</i> .....	11
2.1.13 <i>Iptables</i> .....	15
2.1.14 Basis Data .....	15
2.1.15 Metode <i>Prototype</i> .....	16
2.1.16 <i>UML (Unified Modeling Language)</i> .....	17
2.3 Penelitian Terdahulu .....	19
<b>BAB 3 METODE PENELITIAN</b> .....	21
3.1 Waktu Pelaksanaan Penelitian .....	21
3.2 Tempat Pelaksanaan Penelitian.....	21
3.3 Jadwal Penelitian.....	21
3.4 Metode Pengembangan Sistem .....	21
3.4.1 Analisa Kebutuhan .....	22
3.4.2 Desain Sistem.....	22
3.4.3 Pembangunan/pembuatan <i>Prototype</i> .....	23
3.4.4 Evaluasi dan Perbaikan .....	23
3.5 Implementasi Honeypot Pada Jaringan Komputer.....	23
3.6 Gambaran Umum Sistem .....	23
3.6.1 Gambaran Sistem Saat Ini .....	24
3.6.2 Gambaran Sistem Yang Diusulkan .....	24
3.7 Pengujian Sistem.....	25
3.8 Spesifikasi Sistem .....	25
3.9 Desain Sistem.....	26
3.9.1 Desain Activity Diagram Aplikasi Android.....	26
3.9.2 <i>ERD Kippo</i> .....	27
3.9.3 ERD database .....	28
3.9.4 Desain Tampilan Aplikasi <i>Android</i> .....	28
<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	33
4.1 Hasil .....	33
4.1.1 Tampilan Halaman Login .....	33
4.1.2 Tampilan Halaman <i>Dashboard</i> .....	34
4.1.3 Tampilan Halaman <i>Kippo</i> .....	37





4.1.4	Tampilan Halaman Dionaea.....	39
4.1.5	Tampilan Halaman <i>Block List</i> .....	41
4.2	Pengujian.....	42
4.2.1	<b>Pengujian</b> Login.....	43
4.2.3	Pengujian <i>Port Scanning</i> Menggunakan <i>Nmap</i> .....	44
4.2.2	Pengujian Deteksi Serangan Pada <i>Honeypot Kippo</i> .....	45
4.2.3	Pengujian Deteksi Serangan Pada <i>Honeypot Dionaea</i> .....	46
4.2.4	Pengujian Fitur <i>Blocking Address</i> .....	49
4.2.5	Pengujian <i>Blackbox Testing</i> .....	50
BAB 5	PENUTUP .....	53
5.1	Kesimpulan .....	53
5.2	Saran.....	53
DAFTAR	PUSTAKA .....	55

*--Halaman ini sengaja dikosongkan--*

## DAFTAR GAMBAR

Gambar 2.1 Cara Kerja <i>Honeypot Dionaea</i> .....	9
Gambar 2.2 Cara Kerja <i>Honeypot Kippo</i> .....	10
Gambar 2.3 Metode Serangan <i>Brute-force</i> Menggunakan <i>Wordlist</i> .....	12
Gambar 2.4 Serangan <i>DDoS</i> .....	15
Gambar 2.5 Metode <i>Prototype</i> .....	16
Gambar 3.6 Langkah-langkah <i>prototyping</i> .....	22
Gambar 3.7 Implementasi <i>Honeypot Dionaea</i> dan <i>Kippo</i> .....	23
Gambar 3.8 Gambaran Sistem Saat ini .....	24
Gambar 3.9 Gambaran Umum Sistem yang Diusulkan .....	24
Gambar 3.10 Desain <i>Activity Diagram Android</i> .....	26
Gambar 3.11 Desain <i>ERD database Kippo</i> (Priya Rabadia, 2017).....	28
Gambar 3.12 Desain <i>ERD database untuk android</i> .....	28
Gambar 3.13 Desain <i>Honeypot Dionaea</i> pada Aplikasi <i>Android</i> .....	29
Gambar 3.14 Desain <i>Honeypot Kippo</i> pada Aplikasi <i>Android</i> .....	29
Gambar 3.15 Desain Daftar Blok pada Aplikasi <i>Android</i> .....	30
Gambar 3.16 Desain Menu .....	31
Gambar 17 Halaman <i>Login</i> .....	33
Gambar 18 Halaman <i>Dashboard</i> .....	34
Gambar 19 . Tampilan Halaman <i>Kippo</i> .....	37
Gambar 20 Tampilan <i>Kippo Command Line Input</i> .....	38
Gambar 21 <i>List Address Dionae</i> .....	39
Gambar 22 <i>Most Protocol Used</i> .....	40
Gambar 23 Tampilan Halaman <i>Block List</i> .....	41
Gambar 24 . Skenario Serangan.....	43
Gambar 25 . Tampilan Pesan <i>Error Login</i> .....	44
Gambar 26 . Pengujian <i>Port Scanning Menggunakan Nmap</i> .....	45
Gambar 27 . Penggunaan <i>Tool Hydra</i> .....	45
Gambar 28 . Hasil Serangan Menggunakan <i>Tools Hydra</i> .....	46
Gambar 29 . Tampilan Gambar <i>Tool Hping</i> .....	47
Gambar 30 . Hasil Aplikasi <i>Honeypot Dionaea</i> .....	48



Gambar 31 . Pengujian fitur <i>block address</i> .....	49
--	----



## DAFTAR TABEL

Tabel 2.1 <i>Activity Diagram</i> .....	18
Tabel 3.2 Waktu Pengerjaan Proyek Akhir.....	21
Tabel 3.3 Spesifikasi <i>Virtual Private Server</i> .....	26
Tabel 3.4 Spesifikasi Sistem Penyerang .....	26
Tabel 5. Pengujian <i>BlackBox</i> .....	50

*--Halaman ini sengaja dikosongkan--*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Teknologi informasi pada jaringan komputer selalu berkembang, namun selalu mempunyai masalah pada faktor keamanan. Faktor keamanan begitu penting dalam melindungi sebuah informasi data dikarenakan tidak semua informasi data bersifat umum dan tidak semua orang berhak mengaksesnya. Keamanan Jaringan berarti perlindungan jaringan dan data termasuk perangkat keras dan teknologi perangkat lunak dari ancaman-ancaman yang paling umum termasuk *worm*, *spyware*, *Trojan horse*, *zero hour*, *denial of service*, dan pencurian identitas (Sethi, 2017). Dalam penerapannya, *honeypot* adalah salah satu dari *tools* mencegah atau mendeteksi sebuah serangan.

*Honeypot* adalah sebuah sistem yang dibuat dengan tujuan utama untuk diserang, diakses atau diambil alih dengan cara yang tidak sah. *Honeypot* dibuat seakan akan mirip dengan sistem yang sebenarnya. *Honeypot* memiliki informasi data yang diperuntukkan untuk menjebak penyerang. Dengan adanya *honeypot*, *administrator* dapat menganalisa metode yang digunakan, beserta informasi celah celah yang digunakan oleh penyerang. Dari informasi tersebut *administrator* dapat memperbaiki celah celah dalam sebuah sistem yang menyebabkan kerentanan keamanan jaringan komputer.

Serangan-serangan yang ditujukan pada *honeypot* biasanya berupa *DDOS* (*Distributed Denial Of Service*) sebuah serangan yang digunakan untuk membanjiri lalu lintas jaringan internet, server atau web. Serangan *DDOS* dapat membuat suatu server mengalami *overload* atau ketidakmampuan server untuk *handle request-request* yang ada, salah satu cara *DDOS* adalah mengirimkan request secara terus menerus dengan transaksi data yang besar. Serangan yang kedua *bruteforce*, *bruteforce* adalah sebuah teknik penyerangan untuk meretas password dengan cara mencoba segala kemungkinan kombinasi yang ada untuk mendapatkan password tertentu. Umumnya serangan ini ditujukan pada layanan seperti *FTP*(*File Transfer Protocol*), *SSH*(*Secure Shell*), *Telnet*(*Telecommunications Network Protocol*). Serangan ini dapat menyebabkan

penyerang mendapatkan kata sandi sebuah layanan server biasanya digunakan untuk penyusupan secara tidak resmi. ketika si penyerang telah mendapatkan *username* beserta *password* mudah bagi penyerang untuk mendapatkan data atau informasi secara tidak resmi. Bahkan apabila si penyerang mendapatkan user dan password *root* sebuah server maka otomatis dapat diambil alih oleh si penyerang dan sangat berbahaya bagi perusahaan atau instansi tersebut.

Dalam implementasi untuk mendeteksi serangan dari sebuah jaringan dengan memadukan antara dua *honeypot* yaitu *honeypot diaonea* dan *honeypot Kippo*. *Honeypot diaonea* merupakan salah satu kategori *honeypot low interaction* sebagai penerus *nephentes*. *Dioanea* membuat emulasi layanan palsu yang akan dijadikan sebagai target utama serangan. *Diaonea* menggunakan bahasa pemrograman *python* sebagai bahasa *scripting*, *libemu* untuk mendeteksi *shellcode*, mendukung *IPv6* dan *TLS*. *Kippo* adalah salah satu jenis *honeypot medium interaction* yang didesain menggunakan bahasa pemrograman *python* untuk menyimpan informasi *bruteforce* dan menyimpan informasi penyusup didalam server (Tamminen, 2016).

Dengan notifikasi serangan yang dihubungkan dengan *android* mempermudah dalam memonitoring sebuah keamanan jaringan. Dikarenakan *Android* adalah sistem operasi berbasis linux yang dirancang untuk perangkat bergerak layar sentuh seperti telpon pintar dan komputer tablet. *Android* bersifat *open source* atau bebas digunakan, dimodifikasi, didistribusikan oleh para pengembang perangkat lunak.

## **1.2 Perumusan Masalah**

Dari latar belakang tersebut, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara mengimplementasikan *honeypot Kippo* dan *honeypot diaonea* pada sebuah jaringan komputer ?
2. Bagaimana cara kerja *honeypot Kippo* dan *honeypot diaonea* mendeteksi pada jaringan komputer ?
3. Bagaimana cara memantau aktifitas serangan komputer yang terjadi pada jaringan komputer menggunakan android ?

### 1.3 Tujuan

1. Mengetahui cara implementasi *honeypot Kippo* dan *honeypot diaonea* pada jaringan komputer
2. Mengetahui cara kerja *honeypot Kippo* dan *dionaea* dalam mendeteksi serangan pada jaringan komputer
3. Mengetahui cara memantau serangan yang terjadi pada sebuah jaringan komputer menggunakan android untuk mempermudah kinerja *administrator*.

### 1.4 Manfaat

1. Meningkatkan sebuah keamanan jaringan komputer
2. Mempermudah dalam memonitoring sebuah aktifitas serangan menggunakan android
3. Membantu administrator dalam mendapat informasi celah yang digunakan oleh penyerang sehingga keamanan dalam jaringan komputer dapat meningkat

### 1.5 Batasan Masalah

Adapun masalah-masalah dalam proyek akhir yang akan dibuat, adalah :

1. *Honeypot dionaea* digunakan untuk mendeteksi serangan *port scanning*, dan *DDoS*
2. *Honeypot Kippo* digunakan untuk mendeteksi *brute-force* pada *port SSH* dan merekam aktifitas penyusup
3. Membahas cara kerja *honeypot Kippo* dan *honeypot diaonea*
4. Aplikasi yang menginformasikan sebuah serangan yang terjadi pada jaringan komputer berjalan pada sistem operasi *android*
5. Menggunakan *python django* sebagai *web server*, *python django rest framework* sebagai *API* dan *sqllite* sebagai *database*
6. *Honeypot* diterapkan pada *VPS (Virtual Private Server)*
7. Penyerang menggunakan mesin *virtual*
8. Metode serangan yang akan diujikan *port scanning*, *DDoS* dan *brute-force*
9. Hanya mendeteksi serangan yang berasal dari luar

10. *Kippo* akan mendeteksi serangan *brute-force* apabila kesalahan *password/username* sebanyak 5 kali

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Dasar Teori Pendukung**

Pada dasar teori ini akan dibahas mengenai Jaringan Komputer, *honeypot Kippo*, *honeypot diaonea* dan lain lain yang berhubungan dengan dasar teori yang mendukung pembuatan tugas akhir.

##### **2.1.1 Politeknik Negeri Banyuwangi (Poliwangi)**

Politeknik Negeri Banyuwangi (Poliwangi) merupakan satu-satunya Politeknik Negeri yang ada di Kabupaten Banyuwangi. Poliwangi memiliki tujuh Program Studi yang terdiri dari tiga Program Studi D3 Teknik Informatika, Teknik Mesin, Teknik Sipil, dan empat Program Studi D4 Agribisnis, Manajemen Bisnis Pariwisata, Teknologi Pengolahan Hasil Ternak, Teknik Manufaktur Perkapalan. Seiring berjalannya waktu perkembangan kampus Politeknik Negeri Banyuwangi yang memiliki beberapa fasilitas, seperti Gedung 454 sebagai sarana pembelajaran semua mahasiswa, hotel Politeknik Negeri Banyuwangi jinggo sebagai sarana pembelajaran bagi Program Studi Manajemen Bisnis Pariwisata, laboratorium komputer, laboratorium jaringan, laboratorium hardware yang memadai untuk proses pembelajaran program studi Teknik Informatika, lahan persawahan yang memadai sebagai tempat pembelajaran Program Studi Agribisnis, memiliki perpustakaan yang merupakan salah satu sarana penting sebagai salah satu tempat belajar mahasiswa.

##### **2.1.2 Jaringan Komputer**

Jaringan komputer adalah sebuah jaringan telekomunikasi yang memungkinkan berkomunikasi antar komputer untuk saling menukar data. Tujuan dari jaringan komputer adalah dapat memberikan atau meminta layanan kepada komputer. Dalam jaringan komputer memiliki dua pihak yaitu pihak *client* dan *server*. Fungsi *server* adalah menyediakan sebuah layanan yang dapat digunakan oleh *client*. *Client* adalah pihak yang menerima atau meminta layanan kepada *server*.

Bentuk koneksi pada jaringan komputer tidak harus menggunakan kabel saja melainkan dapat menggunakan serat optik, *wireless*, atau gelombang mikro. Agar jaringan komputer dapat berfungsi, dibutuhkan sebuah layanan yang dapat mengatur pembagian sumber daya, dan juga dibutuhkan aturan-aturan (*protocols*) yang mengatur komunikasi dan layanan-layanan secara umum untuk seluruh sistem jaringan (Sopandi, 2010)

### 2.1.3 Keamanan Jaringan

Keamanan jaringan (*Network Security*) adalah proses pencegahan yang dilakukan oleh penyerang untuk terhubung ke dalam jaringan komputer melalui akses yang tidak sah atau penggunaan secara ilegal dari komputer dan jaringan (John D. Howard, 1997). Tugas keamanan jaringan dilakukan oleh seorang *administrator* jaringan. Berikut 5 ini adalah poin penting definisi dari keamanan jaringan antara lain:

a. *Confidentiality* (kerahasiaan)

Artinya mensyaratkan bahwa informasi atau data hanya bisa diakses oleh yang berwenang

b. *Integrity* (integritas)

artinya mensyaratkan bahwa informasi hanya dapat diubah oleh pihak yang memiliki wewenang.

c. *Availability* (ketersediaan)

artinya mensyaratkan informasi hanya dapat diubah oleh pihak yang berwenang

d. *Authentication* (autentikasi)

Artinya mensyaratkan bahwa pengirim suatu informasi dapat diidentifikasi dengan benar dan ada jaminan bahwa identitas tersedia untuk pihak yang memiliki wewenang ketika dibutuhkan

e. *Nonrepudiation*

Mensyaratkan bahwa baik pengirim maupun penerima informasi tidak dapat menyangkal pengiriman dan penerimaan. Sebagai contoh yang mengirimkan email untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirim email tersebut.

#### **2.1.4 Linux**

*Linux* Adalah sebuah sistem operasi *open source* dan bebas (*free*) dibawah lisensi *GNU (GNU is Not Unix) GPL (General Public License)*. arti dari *open source* adalah kode sumber (*source code*) diikutsertakan dalam program *linux* sehingga dapat dilihat oleh siapa saja tanpa harus menandatangani suatu perjanjian khusus seperti *NDA (Non Disclosure Agreement)*. *Linux* diciptakan oleh seorang mahasiswa Finlandia yang bernama Linus Torvalds. Sistem operasi *linux* adalah sistem buatan manusia yang canggih dan salah satu sistem operasi yang paling umum (Xiao, 2017).

#### **2.1.5 Android**

*android* adalah sistem operasi mobile yang berkembang saat ini dan berbasis ini dan berbasis *linux kernel* yang dirancang untuk perangkat seluler layar sentuh dan komputer tablet saat. *Android* yang disebarluaskan secara *open source* dan menggunakan bahasa pemrograman *java* berupa *java library* dengan lisensi *apache, free software* (Jubilee, 2015).

#### **2.1.6 Honeypot**

*honeypot* adalah sebuah sumber daya yang muncul sebagai *legitimate systems*. Telah lama terbukti sebagai efektif menangkap malware, membantu untuk melawan *spam* dan memberikan sinyal peringatan dini tentang ancaman yang akan datang. *SSH (Secure Shell), Telnet, HTTP* adalah fokus penelitian *honeypot* awal. Karena *SSH* adalah standar *De Facto* masuk server kedalam jaringan yang tidak aman, *SSH Honeypots* menjadi sangat berharga (Bertino E, 2017).

##### **a. Low Interaction Honeypot**

*Low interaction honeypot* seperti tersirat dari namanya memberikan peluang kecil untuk berinteraksi dengan penyerang. Mereka mudah diimplementasikan dan memiliki resiko rendah pada jaringan dan sistem. Tetapi, *low interaction honeypot* mengumpulkan informasi dalam jumlah terbatas seperti logging koneksi level rendah dan informasi level aliran jaringan (Solomon Z. Melese, 2016)

##### **b. Medium Interaction Honeypot**

*Medium interaction honeypot* dibandingkan dengan *low interaction honeypot*, jenis *honeypot* ini lebih banyak memberikan kesempatan dalam berinteraksi dengan penyerang untuk mengumpulkan informasi yang lebih rinci (Solomon Z. Melese, 2016)

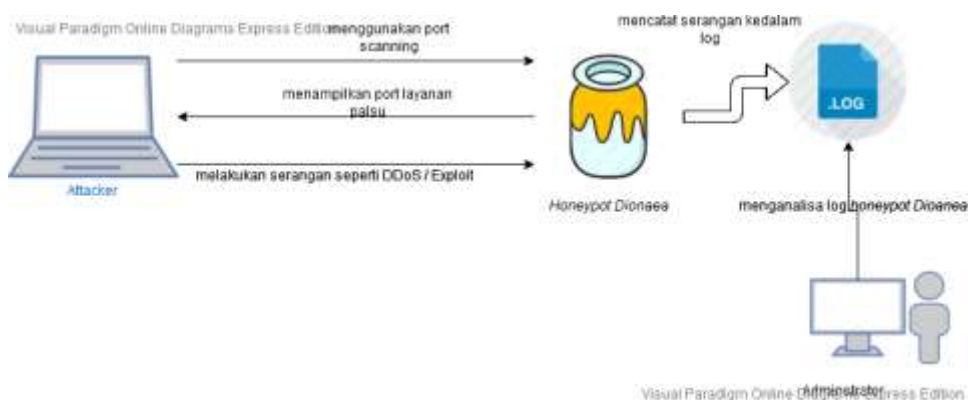
c. *High Interaction Honeypot*

*High interaction honeypot* jenis *honeypot* yang menawarkan layanan nyata dan sistem operasi untuk penyerang. Jenis-jenis *honeypot* memungkinkan penyerang untuk memiliki tingkat interaksi tertinggi dengan sistem nyata dan memungkinkan kita untuk mengumpulkan informasi sebanyak mungkin. Kelemahan dari *high interaction honeypot* adalah terlalu beresiko, dikarenakan penyerang dapat menggunakan *high interaction honeypot* untuk menyerang sistem lain. Untuk digunakan, memelihara, menkonfigurasi dan menganalisis, mereka memerlukan administrator jaringan berkemampuan tinggi. (Solomon Z. Melese, 2016)

### 2.1.7 Dionaee

*Dionaee* adalah *honeypot* yang bersifat *low interaction honeypot* yang diciptakan sebagai pengganti *nepenthes*. *Dionaee* digunakan untuk menjebak penyerang yang memanfaatkan kerentanan malware terhadap layanan atau layanan pada suatu jaringan. Pengembangan awal *dionaee* didanai oleh *Honeypot Project*, sebagai bagian dari *Honeynets Summer Of Code* pada tahun 2009. *Dionaee* menggunakan bahasa pemrograman *python* sebagai bahasa *scripting* dan *libemu* untuk mendeteksi *shellcodes*. Selain itu *dionaee* mendukung IPv6 dan *TPS* (*Transport Layer Security*)(Ion, 2015).

Cara kerja *honeypot dionaee* pada Gambar 2.1.





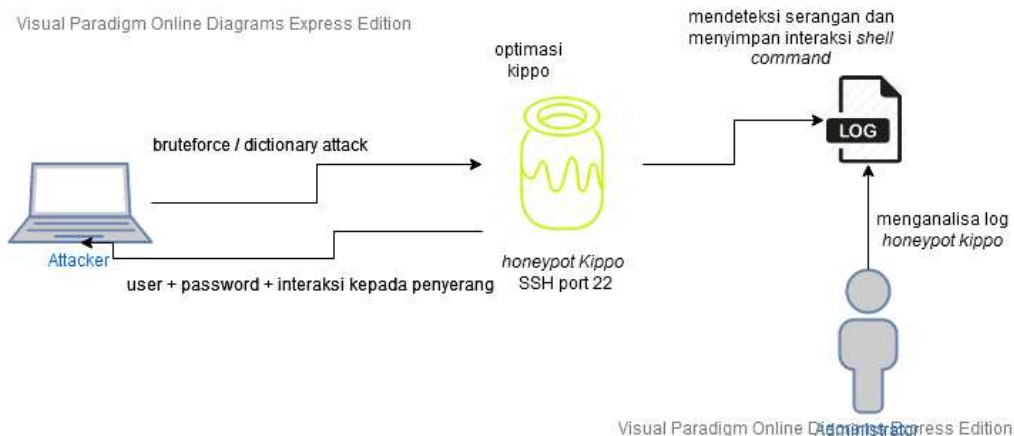
### **Gambar 2.1** Cara Kerja *Honeypot Diona*

Pada Gambar 2.1 dapat dijelaskan cara kerja *honeypot diona* dalam mendeteksi sebuah serangan. Komputer penyerang akan memulai skenario yang pertama adalah dengan *port scanning*, *port scanning* digunakan untuk mendapatkan informasi dari layanan-layanan *server* yang terbuka untuk digunakan dalam proses penyusupan. *Honeypot diona* akan merespon dengan sebuah layanan-layanan palsu. Penyerang akan menggunakan metode serangan *DDoS*. *Honeypot diona* mencatat sebuah serangan tersebut kedalam sebuah log yang digunakan untuk bahan analisa *administrator*. Dikarenakan *honeypot diona* sebuah *low-interaction honeypot* penyerang tidak dapat melakukan interaksi lebih jauh lagi.

#### **2.1.8 Kippo**

*Kippo* adalah *medium interaction honeypot* yang dibuat untuk mempelajari serangan *SSH*. Ini memiliki kemampuan mencatat semua upaya *username* dan *password* dari serangan *brute-force* dan *dictionary attacks*. Setelah berhasil masuk kedalam server *SSH*, *Kippo* juga merekam interaksi *shell* dengan penyerang. Sesi *SSH* yang tidak biasa, klien pertama akan membangun sebuah koneksi *tcp* dengan *SSH server* dan kemudian mereka bertukar informasi. Setelah tahap otentifikasi yang menggunakan *negotiating security algorithms*, klien akan mengirimkan permintaan login *SSH*. *Server SSH* akan memeriksa kombinasi *username* dan *password* untuk memutuskan klien berwenang atau tidak. Untuk sampai kepada *Kippo SSH honeypot* dengan melakukan langkah-langkah yang sama seperti sebelumnya kecuali klien sekarang adalah penyerang. *Username* dan *password* yang dimasukkan oleh penyerang dibandingkan dengan daftar *username* dan *password* yang telah dikonfigurasi sebelumnya yang disimpan dalam file *userdb*. Ketika penyerang menebak dengan *username* dan *password*, mereka diizinkan untuk masuk dan menjalankan beberapa perintah di *server Kippo honeypot*. *Kippo honeypot* memungkinkan mengeksekusi perintah baru seperti *ls* dan *wget*. Karena *honeypot* tidak mengetahui semua perintah *linux* yang sebenarnya, penyerang dapat mudah mengetahui apakah *server honeypot* atau sistem yang nyata. (Solomon Z. Melese, 2016).

Cara kerja *honeypot Kippo* akan ditunjukkan pada Gambar 2.2.



**Gambar 2.2** Cara Kerja *Honeypot Kippo*

Pada Gambar 2.2 dapat dijelaskan bahwa cara kerja *honeypot Kippo* adalah membuat layanan *SSH* palsu yang dapat berinteraksi. Komputer penyerang akan menggunakan metode *brute-force* untuk mendapatkan *user* dan *password* yang digunakan untuk masuk kedalam layanan *SSH*. *honeypot Kippo* akan mencatat sebuah serangan *brute-force* tersebut dengan cara menyimpan kedalam sebuah *log*. Penyerang akan mendapatkan sebuah *user* dan *password* yang akan digunakan masuk kedalam layanan *SSH honeypot Kippo*. ketika penyerang berhasil memasuki *honeypot Kippo* akan mengemulasikan sebuah fitur-fitur *SSH* seperti *server* yang asli. Dalam hal tersebut memungkinkan *honeypot Kippo* dapat berinteraksi dengan penyerang, ketika berinteraksi *honeypot Kippo* akan mencatat setiap interaksi *shell command* yang digunakan penyerang dengan cara menyimpan kedalam sebuah *log* yang akan dianalisa oleh *administrator*.

### 2.1.9 Python

*Python* adalah sebuah bahasa pemrograman tingkat tinggi yang berbasis *interpreted*, berorientasi objek, dengan semantik yang dinamis. Dengan tingkat tinggi yang dibangun dalam struktur data dan dikombinasikan dengan *dynamic typing* dan *dynamic binding*, membuatnya sangat menarik untuk *RAD* (*Rapid Application Development*), serta digunakan sebagai bahasa *scripting* untuk menghubungkan komponen yang ada bersama-sama. Sintaks dalam *python* sederhana dan sangat mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket,

yang mendorong modularitas program dan penggunaan kode kembali (Python Software Foundation, n.d.)

#### **2.1.10 Django Framework**

*Django* adalah *high-level python web framework* yang mendukung pembangunan aplikasi cepat dan desain pragmatis yang bersih. *Django* dibangun oleh pengembang yang berpengalaman, untuk mengurangi kerumitan dalam pengembangan *web*, sehingga user dapat fokus pada pembuatan aplikasi. *Django* gratis dan *open source*. (Django Software Foundation, n.d.)

#### **2.1.11 Android Studio IDE**

*Android Studio* adalah lingkungan pengembangan terpadu – *Integrated Development Enviroment (IDE)* untuk mengembangkan aplikasi *android* yang dikembangkan oleh *Google*. *Android studio* merupakan pengembangan dari *software eclipse IDE* (Wibisono, 2016). Yang dibuat berdasarkan *intelliJ IDEA*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *android studio* menawarkan fitur lebih banyak untuk meningkatkan produktifitas anda saat membuat aplikasi android, misalnya:

1. Sistem versi berbasis *grandle* yang fleksibel
2. *Emulator* yang cepat kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat *android*
4. Instan *run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat *APK* baru.
5. *Template* kode dan integrasi dengan *GitHub* membuat fitur aplikasi yang sama dan memasukkan kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Dukungan *c++* dan *NDK*.
8. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

#### **2.1.12 Serangan Terhadap Server**

Adapun scenario serangan yang digunakan untuk melakukan serangan terhadap server lain:

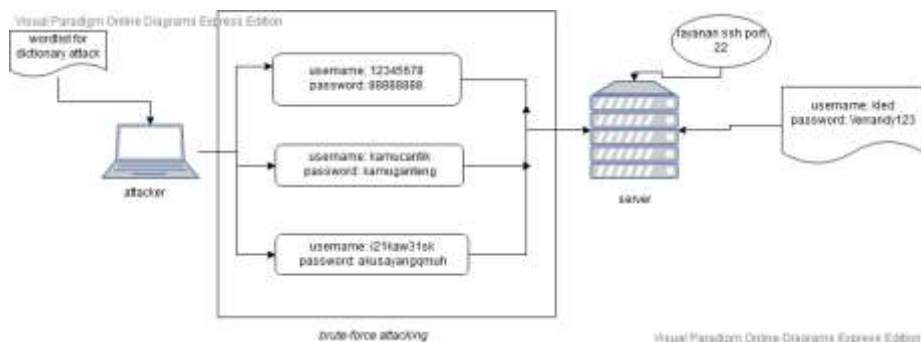
a. *Port Scanning*

*Port Scanning* adalah aktifitas yang dilakukan untuk memeriksa status *port TCP* dan *UDP* pada sebuah mesin. Atau proses untuk mencari dan melihat serta meneliti kemungkinan kelemahan dari suatu sistem yang terpasang pada suatu komputer atau perlengkapan dan peralatannya melalui *port* yang ditargetkan adalah melakukan pengintaian dengan melakukan “*port scanning*” untuk melihat layanan-layanan apa saja yang tersedia di *server* target dan menjadi ancaman yang serius bagi sistem. Contoh aplikasi *port scanning*: *Nmap*.

b. *Brute-force*

Serangan *brute force* adalah algoritma yang memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas. Penyelesaian permasalahan *password cracking* dengan menggunakan algoritma *brute force* akan menempatkan dan mencari semua kemungkinan *password* dengan masukan karakter dan panjang *password* tertentu tentunya dengan banyak sekali kombinasi *password* (Pratita, 2016).

Serangan *brute-force* ditujukan pada Gambar 2.3.



**Gambar 2.3** Metode Serangan *Brute-force* Menggunakan *Wordlist*

Pada Gambar 2.3 dapat dijelaskan cara kerja metode serangan *brute-force*. Serangan *brute-force* dengan *wordlist/dictionary* adalah sebuah serangan yang ditujukan untuk mendapatkan *username/password* dengan mencoba segala kemungkinan *username* dan *password* yang berada di file *wordlist*. Penyerang akan mencoba berbagai *username* dan *password* untuk memasuki kedalam layanan *ssh*, dalam serangan ini memungkinkan

untuk mendapatkan *user/password* apabila didalam *wordlist* terdapat kombinasi *username* dan *password* yang *login* pada layanan *SSH server* contoh aplikasi yang digunakan: *Hydra*. .

c. Serangan *DDoS*

Secara umum, paket data yang beredar di jaringan menggunakan TCP/IP untuk transmisinya. Paket ini sendiri tidak berbahaya, tetapi jika terlalu banyak paket yang abnormal, maka perangkat jaringan atau *server* akan mengalami kelebihan beban/*overload*. Kondisi ini akan dapat dengan cepat mengkonsumsi sumber daya sistem. Kasus lain adalah jika paket serangan memanfaatkan celah keamanan pada protokol tertentu (misalnya permintaan layanan yang tidak lengkap atau penyalahgunaan informasi protokol). Tindakan ini juga dapat menyebabkan kegagalan perangkat jaringan atau *server*. Kedua pendekatan serangan ini sama-sama mengakibatkan *DoS*. Kedua pendekatan ini merupakan pendekatan prinsip-prinsip serangan *DDoS*. Alasan sulit mengapa untuk mencegah serangan *DDoS* adalah karena pada suatu jaringan, lalu lintas yang sah dan ilegal tercampur. Identifikasi akan menjadi semakin sulit, ketika paket data serangan terlihat seperti paket data normal. Misalnya, dalam sistem *Intrusion Detection System* berbasis pencocokan pola *signature* yang khas, mungkin sulit untuk membedakan pesan ilegal dari pesan yang sah pada awal koneksi. Dalam banyak kasus, abnormalitas. Serangan *brute-force* ditujukan pada Gambar 2.3.

1. Serangan dengan basis *Bandwidth*

Serangan *DDoS* jenis ini mengirim pesan data sampah secara masal untuk menyebabkan *overload*, yang juga mengakibatkan berkurangnya *bandwidth* jaringan yang tersedia atau berkurangnya sumber daya perangkat jaringan. Seringkali *router*, *server* dan *firewall* yang diserang memiliki sumber daya yang terbatas. Serangan *overload* menyebabkan kegagalan perangkat jaringan untuk menangani akses yang normal, sehingga terjadi penurunan yang signifikan dalam kualitas layanan atau kelumpuhan total

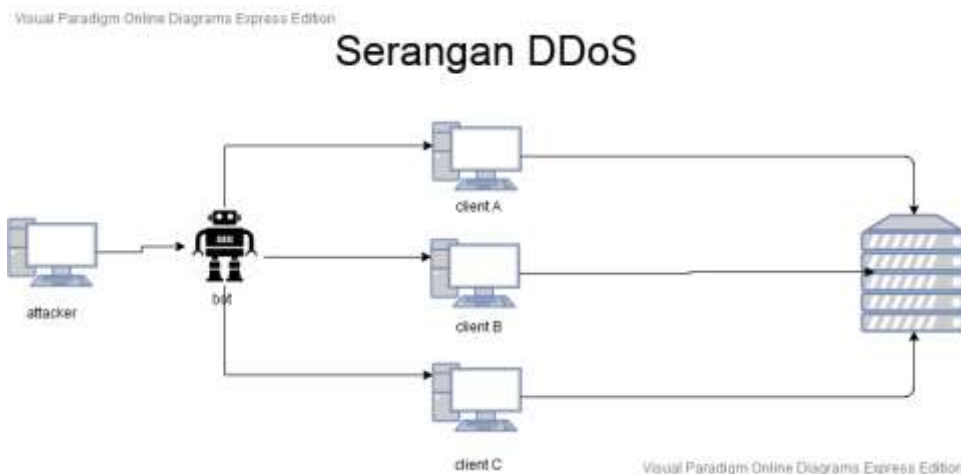
sistem (*DoS*). dalam kedua kasus itu berarti pengguna tidak dapat mengakses sistem yang mereka butuhkan.

## 2. Serangan dengan basis lalu lintas jaringan

Bentuk yang paling umum adalah serangan yang membanjiri lalu lintas jaringan. Serangan ini dilakukan dengan cara mengirimkan sejumlah besar paket *TCP*, paket *UDP*, paket *ICMP* yang tampaknya sah kepada *host/server* target. Beberapa serangan dengan basis ini juga dapat menghindari pemindaian sistem teknologi dengan kamuflase alamat asal. Permintaan yang sah pada akhirnya tidak terlayani karena begitu banyak paket serangan yang beredar di jaringan. Serangan ini juga dapat semakin merusak jika dikombinasikan dengan kegiatan ilegal lainnya, seperti eksploitasi menggunakan *malware* yang menyebabkan kebocoran informasi/pencurian data sensitif pada komputer target

## 3. Serangan dengan basis aplikasi

Serangan jenis ini biasanya mengirim pesan data tingkat layer aplikasi sesuai fitur bisnis yang spesifik, sehingga semakin berkurangnya sumber daya tertentu pada lapisan aplikasi (seperti jumlah pengguna dan koneksi aktif yang diperbolehkan ) dan layanan sistem tidak lagi tersedia. Serangan seperti ini biasanya tidak dilancarkan dalam volume yang terlalu besar, serangan dengan lalu lintas tingkat rendah pun dapat menyebabkan gangguan serius pada sistem atau bahkan kelumpuhan kinerja sistem bisnis (Septian Geges, 2015). Serangan *DDoS* ditunjukkan pada Gambar 2.4.



**Gambar 2.4** Serangan *DDoS*

Pada Gambar 2.4 dapat dijelaskan bahwa *attacker* menggunakan *bot*. untuk menyamar menjadi beberapa *client*. Dalam hal ini *client* meminta layanan kepada server. Dalam serangan *DDoS* *attacker* mampu membuat ratusan bahkan ribuan *client* yang dapat menyebabkan pelayanan server terhambat dikarenakan tidak mampu handle *request* yang terlalu banyak dan hingga paling parah dapat membuat *server* menjadi *hang*. Contoh aplikasi yang digunakan *PyDDoS*.

#### 2.1.13 *Iptables*

*Iptables* adalah suatu tools dalam sistem operasi *linux* yang berfungsi sebagai alat untuk melakukan *filter* (penyaringan) terhadap (*traffic*) lalu lintas data. Secara sederhana digambarkan sebagai pengatur lalu lintas data. Dengan *iptables* dapat mengatur semua lalu lintas jaringan dalam komputer, baik yang masuk ke komputer, keluar dari komputer, ataupun *traffic* yang sekedar melewati komputer.

#### 2.1.14 Basis Data

Menurut (Sucipto, 2017) *Database* adalah kumpulan data yang dihubungkan secara bersama-sama, dan gambaran dari data yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Secara umum *database* dapat diartikan sebagai sebuah tempat penyimpanan data sebagai pengganti dari sistem konvensional yang berupa dokumen file. Perancangan *database* sendiri terdapat

tiga fase utama yaitu perancangan *database* konseptual yang merupakan proses membangun model dari data yang digunakan dalam sebuah organisasi dan tidak tergantung pada pertimbangan fisik, perancangan *database logical*, merupakan proses membangun model dari informasi yang digunakan dalam perusahaan berdasarkan model data spesifikasi, dan terbebas dari DBMS (*Database Management System*) tertentu dan pertimbangan fisik lainnya.

Menurut (Rosa A.S., 2014) Sistem Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan

#### 2.1.15 Metode *Prototype*

Metode *Prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping*. Metode *prototype* ditunjukkan pada Gambar 2.5.



**Gambar 2.5** Metode *Prototype*

Tahap yang pertama adalah *listen* tahapan mendengarkan pelanggan, pada tahap ini proses menganalisa kasus dengan mengambil contoh pada bidang akademik yang menghadapi banyak complain dari mahasiswa meliputi proses belajar mengajar dan lain sebagainya. Permasalahan yang timbul dari complain mahasiswa ini tidak tertampung sehingga diperlukan suatu sistem yang dapat mengelola dan menyimpan semua keluhan yang dihadapi mahasiswa tersebut dan akademik dapat memberikan sebuah keputusan yang cepat dan tepat.



Tahapan yang kedua berupa tahapan membuat dan memperbaiki prototype. pada tahapan ini berusaha mendesain secara cepat dan kemudian membuat aplikasi atau software sesuai dengan analisis kebutuhan yang sudah dilakukan yang disesuaikan dengan konsumen atau *user*.

Tahap mencoba aplikasi dan evaluasi *prototype* dengan cara menguji dengan studi kasus yang sudah dianalisis bersama-sama dengan pakar. Jika pada tahapan *costumer test user* atau pakar merasa software belum sesuai dengan yang diinginkan dapat dilakukan perbaikan software aplikasi dengan kembali ke tahapan yang pertama (Siti, 2015)

#### **2.1.16 UML (*Unified Modeling Language*)**

*Unified Modelling Language (UML)* adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* mendefinisikan notasi dan *syntax/semantic* seperti bahasa-bahasa lainnya. Membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun yaitu dengan menggunakan *UML*. *UML* mempunyai beberapa diagram diantaranya adalah *Use Case Diagram* dan *Activity Diagram*. (Rembulan, 2015).

##### **A. *Activity Diagram***



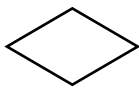


Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang. Struktur diagram mirip seperti flowchart atau Data Flow Diagram pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Activity diagram berfungsi untuk menggambarkan workflow / aliran kerja dari suatu proses bisnis. Suatu aliran kerja bisa saja dituangkan dalam bentuk narasi / teks, akan tetapi jika aliran kerjanya sudah kompleks maka kita akan kesulitan untuk membayangkan bagaimana proses itu terjadi. Oleh karena itu, dibuatlah activity diagram sebagai salah satu cara untuk menggambarkan

aliran kerja tersebut. (Harisantyo, Nugraha, Prasetyawan, Nugraha, & Sulaiman, 2015).

Menurut (Rembulan, 2015) Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Pada *Activity* diagram terdapat komponen/symbol-simbol yang ditunjukkan pada Tabel 2.1. (Yusmiarti, 2016)

**Tabel 2.1 Activity Diagram**

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan/ JoinAsosiasi 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

6.	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
	Nama swimlane	

### 2.3 Penelitian Terdahulu

Sebagai bahan acuan untuk penyusunan proyek akhir ini, penulis juga menjadikan beberapa penelitian terdahulu sebagai referensi diantaranya:

Fitrotul Amalia, (2018), dalam “Rancang Bangun Sistem Keamanan Berbasis *Honeypot Dionaea* Pada Layanan *Cloud Server* Dengan *Report Web*”, telah melakukan penelitian merancang dan membangun *honeypot dionaea* pada layanan *cloud server* kemudian *honeypot dionaea* yang telah dibangun diuji dengan dua macam serangan yaitu *scanning* dan *exploit*.

Fathuzzikri, Ikhwan Ruslianto, Uray Ristian(2019), dalam “Implementasi *Honeypot Kippo* pada Sistem Keamanan *Server* Berbasis *Web Monitoring* dengan Notifikasi Otomatis Menggunakan Api *Telegram*”, telah melakukan penelitian implementasi *honeypot Kippo* pada keamanan *server*, kemudian *honeypot Kippo* diuji dengan *tool nmap*. Hasil dari pengujian yang dilakukan dari penyerang menghasilkan analisa kinerja *CPU*, *memory*, jumlah penyerang dalam bentuk table.

Sutarti, Khairunnisa (2017), dalam “Perancangan dan Analisis Keamanan Jaringan *Nirkabel* Dari Serangan *DDoS (Distributed Denial of Service)* berbasis *honeypot*” telah melakukan penelitian merancang dan menganalisis kemaman jaringan *nirkabel* dari serangan *DDoS*. Kemudian *honeypot* dengan *tool honeyD* dirancang dan dikonfigurasi kedalam jaringan *nirkabel*. *Honeypot* yang telah dikonfigurasi diuji menggunakan *tool nmap* dan *DDoS*. dari hasil pengujian mendapatkan perbedaan pengujian jaringan awal sistem dengan pengujian akhir sistem keamanan *honeypot*, jaringan yang dipasang dengan *honeypot* mendapatkan hasil tidak ada serangan yang masuk kedalam sebuah sistem.

Sedangkan penelitian yang penulis lakukan, menggunakan dua buah *honeypot* yaitu *Honeypot Dionaea* dan *Honeypot Kippo*. *Honeypot Dionaea* sebagai pendeteksi serangan pada *server* berupa *DDoS* dan *port scanning* dengan cara menyediakan *port-port* palsu sebagai layanan, sedangkan *Honeypot Kippo* digunakan sebagai pendeteksi serangan *Brute-force* dan mengumpulkan informasi aktifitas penyerang yang menyerang *SSH honeypot Kippo*. Dalam hal ini *android* digunakan untuk memonitoring keamanan jaringan komputer.

## BAB 3

### METODE PENELITIAN

#### 3.1 Waktu Pelaksanaan Penelitian

Penelitian ini dijadwalkan akan dilaksanakan selama kurang lebih 6 bulan (enam bulan) terhitung mulai bulan Februari s/d Juni 2020.

#### 3.2 Tempat Pelaksanaan Penelitian

Tempat penelitian Proyek Akhir ini akan dilakukan di Politeknik Negeri Banyuwangi Jl. Raya Jember KM. 13 Kabat, Labanasem.

#### 3.3 Jadwal Penelitian

Berikut adalah jadwal penelitian yang akan dilakukan sesuai waktu pelaksanaan dan mencakup tahap perencanaan, desain sistem, pengujian, evaluasi, dan penyusunan laporan, berikut rinciannya ditunjukkan pada Table 3.2.

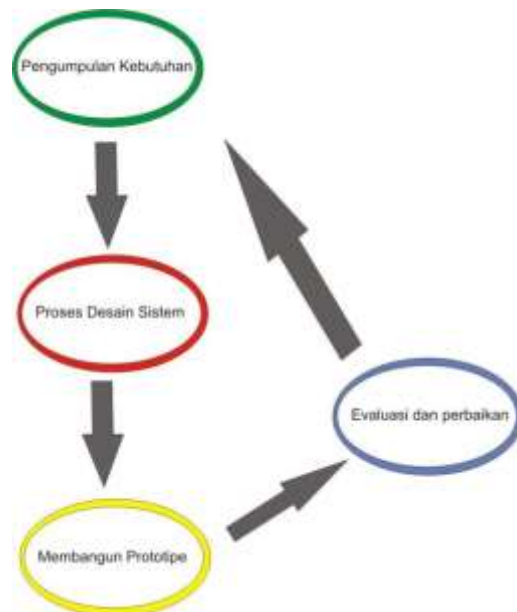
**Tabel 3.2 Waktu Pengerjaan Proyek Akhir**

No	Kegiatan	2020																	
		Februari			Maret			April			Mei			Juni					
1	Analisa Kebutuhan																		
2	Desain Sistem																		
3	Membangun prototype																		
4	Pengujian Sistem																		
5	Implementasi Sistem																		
6	Pembuatan Laporan																		

#### 3.4 Metode Pengembangan Sistem

Metode yang digunakan dalam pengerjaan tugas akhir yaitu metode *prototype*. Metode *prototype* digunakan karena dapat menghasilkan *prototyping* yang dapat diterapkan pada sistem kecil maupun besar dengan harapan agar proses pengembangan dapat berjalan dengan baik, tertata serta dapat selesai

dengan waktu yang tepat. Secara umum metode *prototype* memiliki 4 tahap untuk membuat *prototyping* ditunjukkan pada Gambar 3.4.



**Gambar 3.6** Langkah-langkah *prototyping*

#### **3.4.1 Analisa Kebutuhan**

Dalam rangka melakukan pengembangan sistem diperlukan penilaian kebutuhan awal dan analisa tentang ide atau gagasan untuk membangun ataupun mengembangkan sistem. Analisis dilakukan untuk mengetahui komponen apa saja pada sistem yang sedang berjalan, dapat berupa hardware, software, jaringan dan pemakai sistem sebagai level pengguna akhir sistem. Langkah selanjutnya adalah mengumpulkan informasi yang dibutuhkan pengguna akhir yang meliputi biaya dan manfaat sistem yang dibangun ataupun dikembangkan.

Analisa kebutuhan sistem mendefinisikan kebutuhan sistem yang berupa:

1. *input* sistem
2. *output* sistem
3. proses yang berjalan dalam sistem
4. basisdata yang digunakan (Purnomo, 2017).

#### **3.4.2 Desain Sistem**

Tahapan yang dilakukan dengan perancangan sistem dari analisa kebutuhan yang ada dengan menggunakan perangkat permodelan sistem seperti *activity*

diagram, *ERD* (*Entity Relationship Diagram*). Untuk menunjang dalam pembuatan *prototyping* sistem/

### 3.4.3 Pembangunan/pembuatan *Prototype*

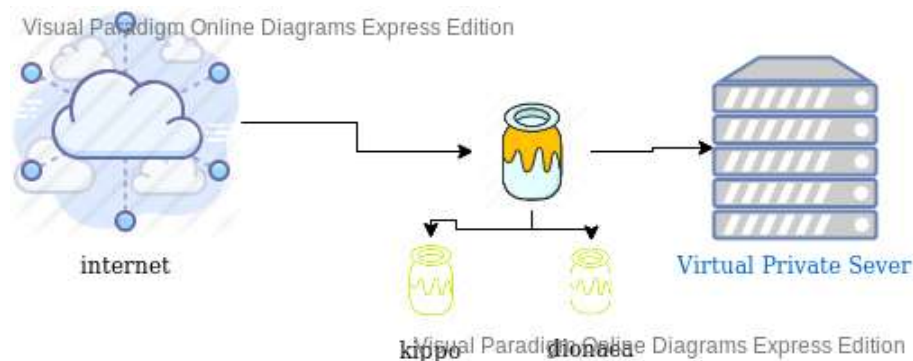
Pada tahap ini membangun sebuah *prototype* dengan membuat perancangan sementara yang berfokus pada penyajian yaitu membuat *input* dan *output* (Siti, 2015).

### 3.4.4 Evaluasi dan Perbaikan

Melakukan evaluasi terhadap sistem yang telah dibangun, apakah sistem tersebut sudah berjalan dengan keinginan, jika iya maka akan melanjutkan pada tahap implementasi jika tidak akan kembali pada tahap analisa kebutuhan.

## 3.5 Implementasi Honeypot Pada Jaringan Komputer

Dalam suatu jaringan komputer yang berhubungan dengan internet memiliki resiko lebih tinggi dalam keamanan jaringan. Dalam pengamanan jaringan dari luar honeypot diimplementasikan didepan server untuk mencegah penyerangan langsung masuk kedalam server. Implementasi ditujukan pada Gambar 3.7.



**Gambar 3.7** Implementasi *Honeypot Dionaea* dan *Kippo*

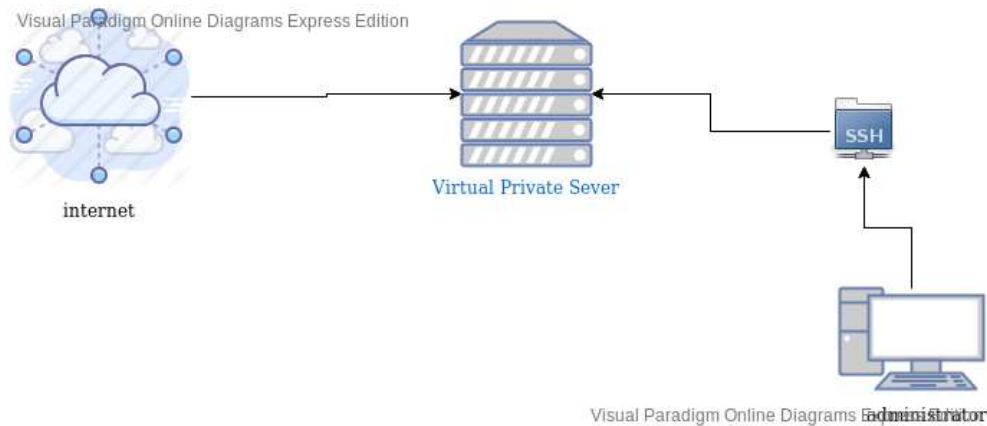
Pada Gambar 3.7 dapat dijelaskan bahwa *honeypot dionaea* dan *Kippo* diterapkan didepan *virtual private server* yang langsung terhubung pada *internet*. Kedua *honeypot* tersebut digunakan sebagai pendeteksian dini serangan jaringan komputer.

## 3.6 Gambaran Umum Sistem

Gambaran umum sistem saat ini sangat diperlukan dalam hal pembuatan ataupun pengembangan suatu sistem. Pada gambaran umum sistem terdapat dua

pokok bahasan yang akan dijelaskan yaitu, gambaran umum sistem yang berjalan dan gambaran sistem yang diusulkan. Tujuan dari pembahasan gambaran umum sistem ini adalah untuk mengetahui pembaharuan atau pengembangan yang akan dilakukan terhadap sistem yang telah berjalan sebelumnya.

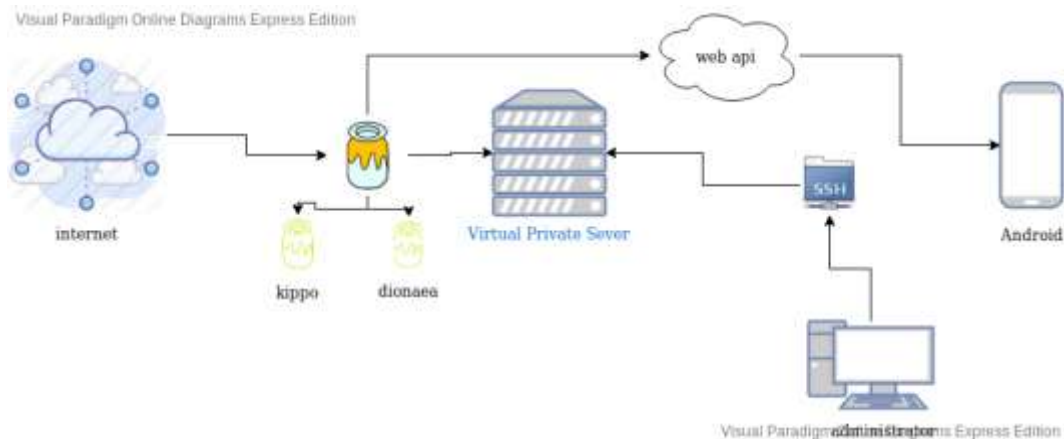
### 3.6.1 Gambaran Sistem Saat Ini



**Gambar 3.8** Gambaran Sistem Saat ini

Gambar 3.8 dapat dijelaskan bahwa *virtual private server* langsung berhubungan langsung dengan *internet*, dalam konfigurasinya menggunakan layanan *SSH*. Karena kurangnya sistem keamanan, maka serangan yang berasal dari internet akan berdampak langsung terhadap *virtual private server*.

### 3.6.2 Gambaran Sistem Yang Diusulkan



**Gambar 3.9** Gambaran Umum Sistem yang Diusulkan



Dari Gambar 3.9 dari sistem yang sedang berjalan sebelumnya, maka diterapkan implementasi sistem keamanan *honeypot dionaea* dan *honeypot Kippo*. Dalam penerapannya kedua *honeypot* tersebut menggunakan fitur *SSH* yang telah disediakan oleh *virtual private server*, dalam hal ini fungsi dari *honeypot dionaea* mengemulasikan layanan palsu yang digunakan menjebak serangan yang berasal dari *internet*. Fungsi dari *honeypot Kippo* mendeteksi sebuah serangan yang diperuntukkan terhadap layanan *SSH*, kemudian hasil dari serangan tersebut diintegrasikan dengan *android* sehingga memudahkan *administrator* dalam memonitoring serangan jaringan komputer. Bagan pada *honeypot* area antara *internet* dan *server* yang berarti *honeypot* diletakkan didalam *DMZ (Demilitary Zone)* atau didepan server agar serangan *attacker* diterima *honeypot* sehingga dapat melindungi *server* dari *attacker*.

### 3.7 Pengujian Sistem

Pengujian menggunakan *Black Box* dilakukan untuk menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi dari aplikasi berjalan sesuai dengan kebutuhan atau belum, fungsi yang dimaksud adalah penggunaan button dalam aplikasi dan pembacaan data dalam database menggunakan *django REST framework* untuk menampilkan sebuah deteksi serangan pada *android*, serta pengujian juga dilakukan pada sistem keamanan jaringan apakah sesuai dengan skenario yang diharapkan atau tidak.

### 3.8 Spesifikasi Sistem

Deteksi serangan menggunakan *honeypot* berbasis *android*. Akan menggunakan dua buah komputer dengan sistem operasi *linux* dengan *distro* yang berbeda:

- a. Sistem operasi yang digunakan sebagai penerapan *honeypot* ditujukan pada Tabel 3.3

**Tabel 3.3** Spesifikasi *Virtual Private Server*

Spesifikasi	Keterangan
Sistem Operasi	<i>Ubuntu Server</i>
<i>RAM</i>	1GB

- b. Mesin *virtual guest OS* sebagai penyerang Tabel 3.4 Spesifikasi *Kali Linux* sebagai *guest OS*

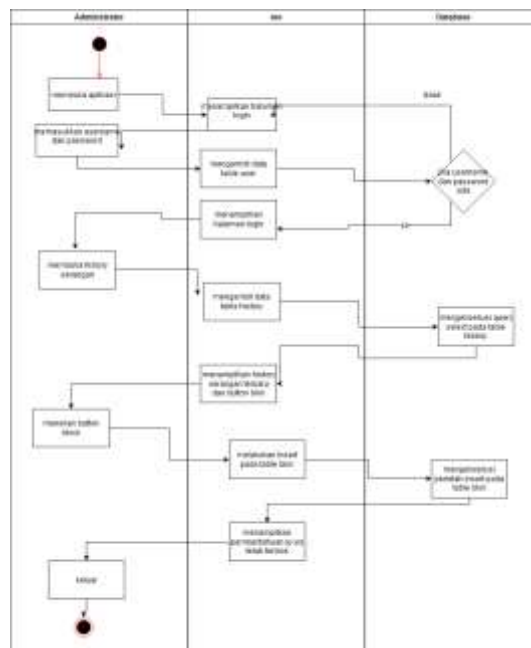
**Tabel 3.4** Spesifikasi Sistem Penyerang

Spesifikasi	Keterangan
Sistem Operasi	<i>Kali Linux</i> 64-bit
Posisi	<i>Guest OS / penyerang</i>
<i>RAM</i>	1 GB
<i>Network Adapter</i>	<i>Atheros AR9271</i>

### 3.9 Desain Sistem

### 3.9.1 Desain Activity Diagram Aplikasi Android

Desain *activity diagram* pada aplikasi *android* ditujukan pada Gambar 3.10.

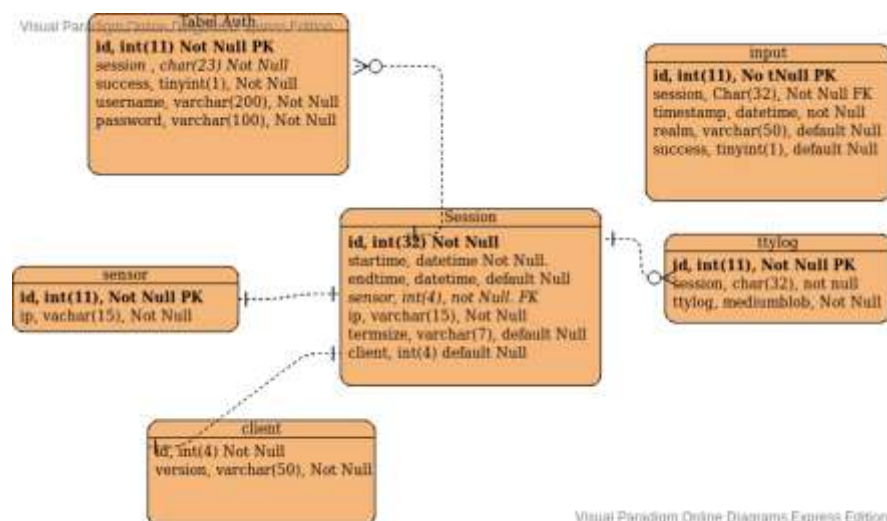


**Gambar 3.10** Desain *Activity Diagram* Android

Pada Gambar 3.8 menunjukkan alur proses kerja aplikasi *android* dalam melakukan blok *ip address* pada sistem keamanan jaringan yang menggunakan *honeypot*. Pada proses awal *administrator* membuka aplikasi *android*. Lalu pada sistem aplikasi *android* menampilkan sebuah halaman login. *Administrator* memasukkan *username* dan *password* untuk login. Pada sistem aplikasi *android* melakukan validasi *username* dan *password* apakah *username* dan *password* benar. Apabila *username* dan *password* salah, akan mengembalikan pada halaman login. Apabila *username* dan *password* benar aplikasi *android* akan menampilkan sebuah tampilan halaman menu. Pada tahap ini *administrator* membuka sebuah halaman *history*, aplikasi *android* mengambil data dari tabel *history* yang berisikan serangan serangan terbaru. Database akan memproses *query* yang untuk menampilkan data yang diminta oleh aplikasi *android*. Lalu aplikasi *android* akan menampilkan data tersebut dan button blok kedalam *layout history*. *Administrator* mengambil langkah dengan menekan tombol blok. Aplikasi *android* akan memproses dengan cara *insert* kedalam table blok dengan ip yang tercantum pada *layout history*. Database akan mengeksekusi perintah tersebut ketika selesai, aplikasi *android* akan menampilkan bahwa ip telah di blok. *administrator* keluar dari aplikasi.

### 3.9.2 ERD Kippo

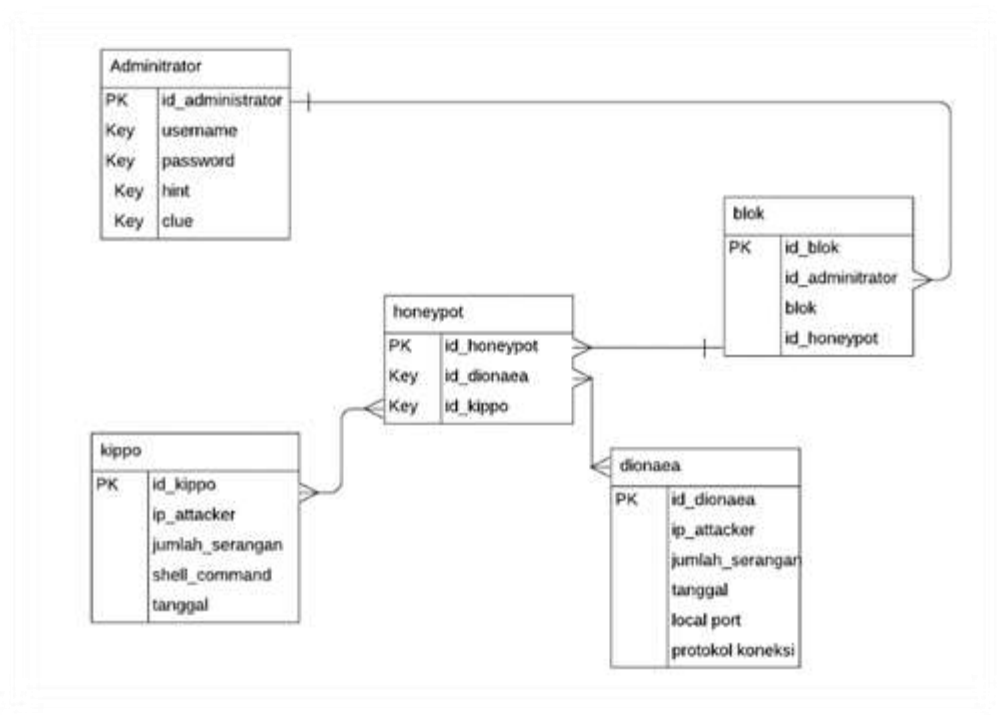
Pada *honeypot Kippo* telah memiliki sebuah desain *ERD* yang digunakan untuk menyimpan sebuah log dan informasi *shell command* yang memonitoring aktifitas penyusupan yang terjadi terhadap port *SSH*. *Honeypot Kippo* memiliki sebuah penyimpanan log yang diintegrasikan kedalam database. Desain *ERD* ditujukan pada Gambar 3.9.



**Gambar 3.11** Desain *ERD database Kippo* (Priya Rabadia, 2017)

### 3.9.3 ERD database

Untuk menampilkan sebuah *output* serangan pada *android* membutuhkan sebuah database untuk yang dihubungkan dengan *log honeypot Kippo* dan *dionaea*. Dalam hal ini desain database akan ditunjukan pada Gambar 3.10



**Gambar 3.12** Desain *ERD database untuk android*

### 3.9.4 Desain Tampilan Aplikasi *Android*

Desain tampilan pada aplikasi *android* mempermudah *administrator* dalam memonitoring sistem keamanan jaringan komputer

#### A. Desain Tampilan *Honeypot Dionaea* Aplikasi *Android*

Pada tampilan *honeypot dionaea android* hasil dari sebuah serangan. Pada tampilan tersebut ip penyerang, protokol yang digunakan, port yang diserang, jumlah serangan pada jaringan beserta sebuah aksi yang dilakukan untuk memblokir ip address. Tampilan *honeypot dionaea* ditunjukkan pada Gambar 3.13.



Ip	Protokol	Port yang	Jumlah	Waktu	Aksi
100.20.111	TCP	80	10000	2 Agustus 2019	<input checked="" type="checkbox"/>
100.22.110	TCP	443	2000	3 Agustus 2019	<input checked="" type="checkbox"/>
200.113.x	TCP	80	31	4 Agustus 2019	<input checked="" type="checkbox"/>

**Gambar 3.13** Desain *HoneyPot Dionaea* pada Aplikasi *Android*

#### B. Desain Tampilan *HoneyPot Kippo* pada Aplikasi *Android*

Pada tampilan *honeypot Kippo* hasil dari sebuah deteksi serangan yang dari *honeypot Kippo* itu sendiri. Pada tampilan tersebut berisi ip penyerang, protokol yang digunakan, interaksi *shell* saat terjadi penyusupan dan tombol aksi untuk melakukan pemblokiran. Desain tampilan tersebut ditujukan pada Gambar 3.14.

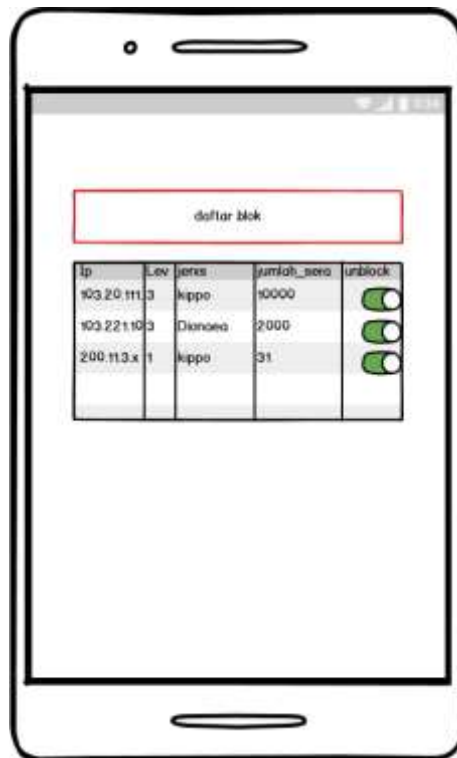


Ip	Protokol	Interaksi	Jumlah	Waktu	Aksi
100.20.111	TCP	ls	10000	2 Agustus 2019	<input checked="" type="checkbox"/>
100.22.110	TCP	pwd	2000	3 Agustus 2019	<input checked="" type="checkbox"/>
200.113.x	TCP	mkdir	31	4 Agustus 2019	<input checked="" type="checkbox"/>

**Gambar 3.14** Desain *HoneyPot Kippo* pada Aplikasi *Android*

### C. Desain Tampilan Daftar Blok pada Aplikasi Android

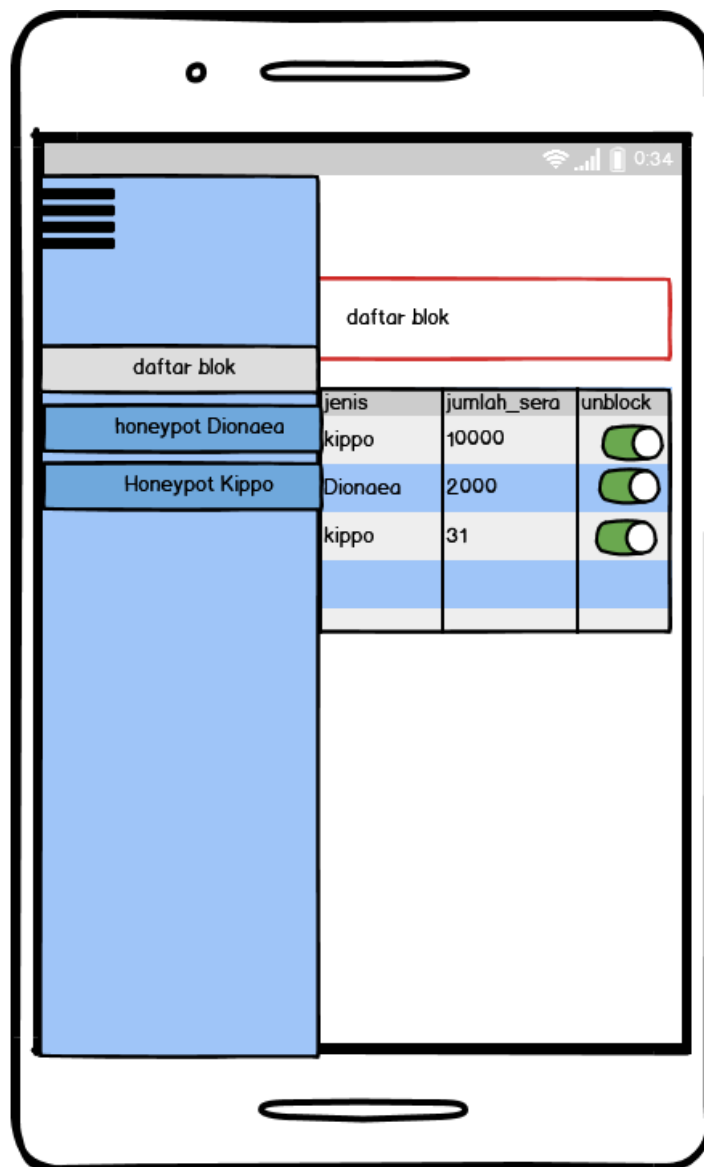
Pada tampilan daftar blok pada aplikasi *android* merupakan dari hasil aksi pemblokiran terhadap ip penyerang. pada tampilan tersebut berisi ip penyerang, level serangan, jumlah serangan dan aksi *unblok*. Tampilan daftar blok ditujukan pada Gambar 3.15.



**Gambar 3.15** Desain Daftar Blok pada Aplikasi *Android*

### D. Desain Tampilan Menu pada Aplikasi *Android*

Tampilan menu pada aplikasi *android* berisi tampilan-tampilan untuk memilih sebuah *layout* yang ingin dibuka seperti daftar blok, *honeypot dionaea*, *honeypot Kippo*. Tampilan menu ditunjukkan pada Gambar 3.16.



**Gambar 3.16** Desain Menu





## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

Aplikasi Deteksi Serangan Menggunakan Honeypot Berbasis *Android* adalah sebuah aplikasi *android* yang berfungsi memonitoring sebuah serangan yang berada pada *VPS* serta dapat melakukan sebuah aksi yaitu pemblokiran *Ip Address*

##### 4.1.1 Tampilan Halaman Login

Pada gambar 4.1 merupakan halaman login, halaman ini digunakan untuk login ke dalam aplikasi *honeypot*



**Gambar 17** Halaman *Login*

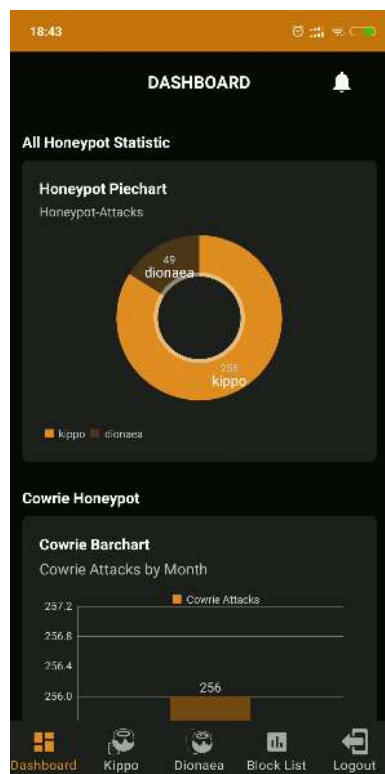
Pada Gambar 17 menunjukkan tampilan halaman *login* dengan penjelasan fitur sebagai berikut:

1. Kolom *username* digunakan untuk mengisi akun pengguna

2. Kolom *password* digunakan untuk mengisi sandi yang sesuai dengan akun
3. Tombol *login* digunakan untuk memverifikasi *username* dan *password* apabila benar maka otomatis akan masuk kedalam aplikasi *honeypot*

#### 4.1.2 Tampilan Halaman *Dashboard*

Gambar 18 merupakan halaman *dashboard* yang berisikan sebuah informasi penyerangan dalam bentuk grafik yang dapat digunakan sebagai analisa penyerangan dari waktu ke waktu



**Gambar 18** Halaman *Dashboard*

Pada Gambar 18 dengan penjelasan fitur sebagai berikut :

1. *Honeypot Piechart*

*Honeypot piechart* adalah sebuah grafik yang berbentuk kue pie berisikan data dari semua serangan yang berada didalam VPS dengan *Honeypot Kippo* dan *honeypot dionaea* digunakan sebagai pembanding serangan yang berada pada kedua *honeypot* tersebut

2. *Kippo Barchart*

*Kippo barchart* adalah sebuah grafik yang berbentuk bar digunakan untuk menampilkan sebuah serangan yang berada pada *honeypot Kippo* per bulan.

3. *Kippo Attacks*

*Kippo Attacks* adalah tampilan semua jumlah semua serangan yang berada pada *honeypot Kippo*

4. *Kippo Attacks by Days*

*Kippo attacks by days* adalah tampilan jumlah serangan per hari pada *honeypot Kippo*

5. *Top 10 Kippo address*

*Top 10 Kippo address* adalah sebuah 10 *ip address* yang paling banyak melakukan serangan terhadap *honeypot Kippo*

6. *Kippo Attacks By username*

*Kippo attacks by username* adalah sebuah grafik *pie chart* yang digunakan untuk menampilkan data *username* yang paling banyak digunakan pada *honeypot Kippo*

7. *Kippo Attacks By password*

*Kippo attack by password Kippo attacks by username* adalah sebuah grafik *pie chart* yang digunakan untuk

menampilkan data *password* yang paling banyak digunakan pada *honeypot Kippo*

8. *Dionaea Barchart*

*Dionaea barchart* adalah sebuah grafik bar yang menampilkan jumlah serangan perbulan

9. *Dionaea Attacks*

*Dionaea attacks* adalah tampilan total semua serangan yang berhasil ditangkap *honeypot dionaea*

10. *Dionaea attacks by day*

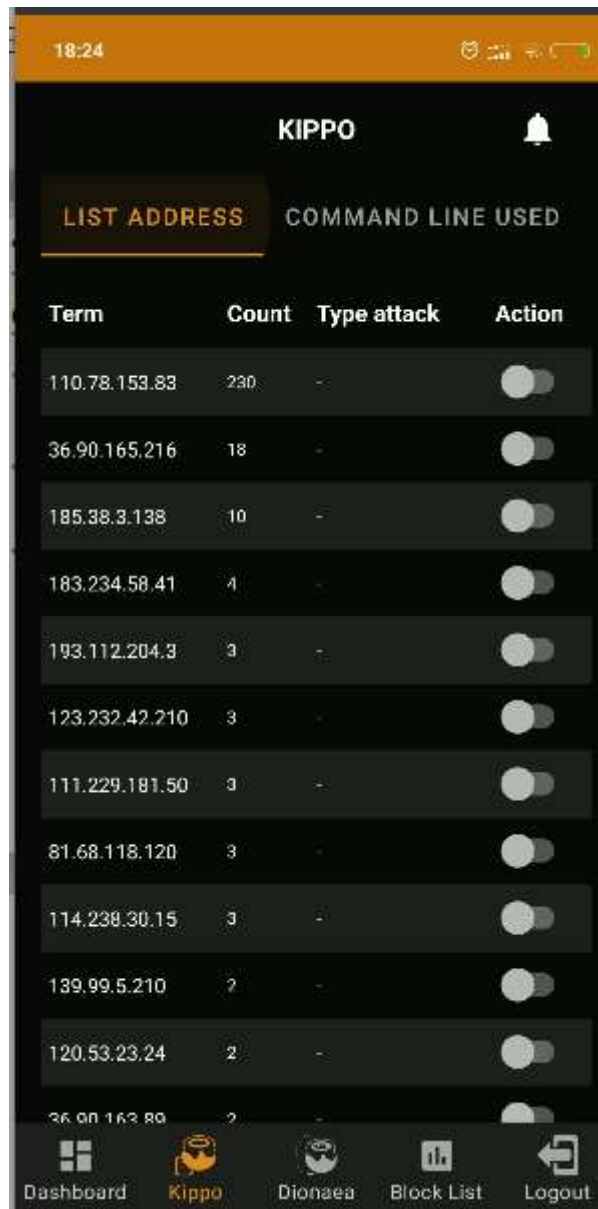
*Dionaea attacks by day* adalah tampilan total serangan yang berada pada dionae per hari

11. *Top 10 dionaea address*

*Top 10 dionaea address* adalah 10 *ip address* yang paling banyak menyerang *honeypot dionaea*

#### 4.1.3 Tampilan Halaman *Kippo*

Gambar 19 merupakan tampilan halaman *Kippo* yang berisikan fitur list address dan command line input



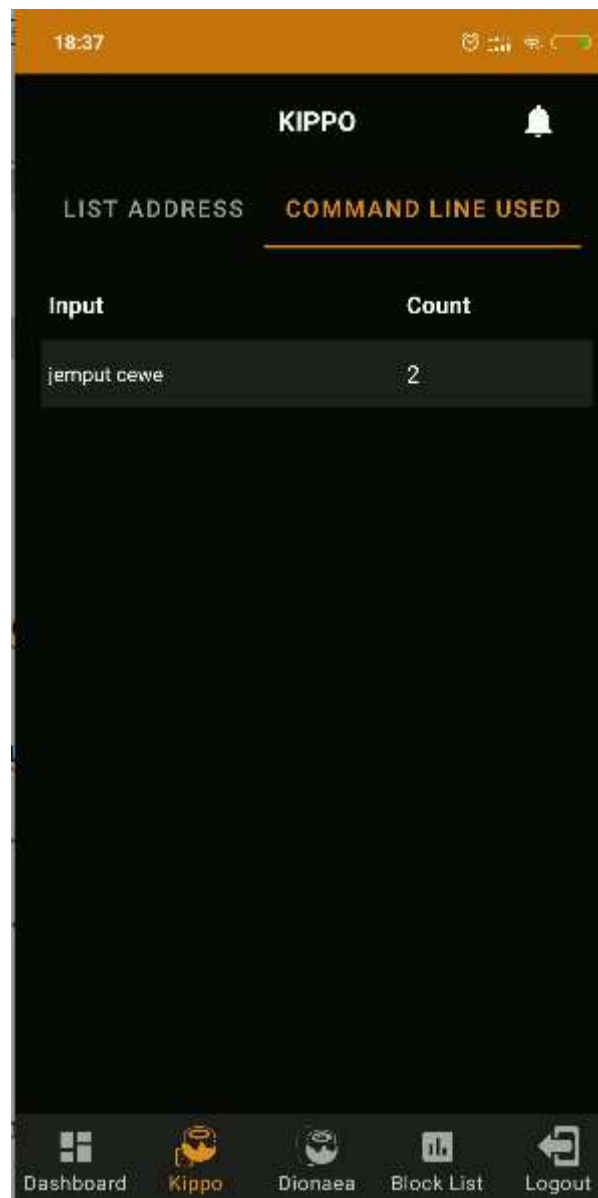
**Gambar 19.** Tampilan Halaman Kippo

Pada Gambar 19 terdapat tampilan *Kippo List Address* yang memiliki fitur :

1. Menampilkan *ip address* dari sebuah serangan

2. Menampilkan total jumlah serangan pada setiap *ip address*
3. Memiliki fitur action yang digunakan untuk memblokir *ip address*

Gambar 20 merupakan tampilan *command line input* yang berisikan sebuah perintah perintah pada linux ketika sebuah serangan *bruteforce* menemukan kombinasi *username* dan *password* dengan benar.



**Gambar 20** Tampilan *Kippo Command Line Input*

Pada Gambar 20 terdapat fitur sebagai berikut:

#### 1. Input

Input merupakan sebuah *command line* yang berasal dari *honeypot Kippo* ketika sebuah serangan mampu menemukan *username* dan *password* secara benar.

#### 2. Count

*Count* merupakan kalkulasi total dari sebuah *input* yang berasal dari *honeypot Kippo*

### 4.1.4 Tampilan Halaman Dionaea

tampilan halaman *dionaea* merupakan tampilan yang memiliki 2 fitur yaitu *list address* dan *most protocol used*

#### a. List address dioneae

List address dionaea merupakan tampilan ip address dari sebuah serangan *honeypot dionaea*. *List address* ditampilkan pada Gambar 21.



Term	Count	Action
96.68.206.252	5	<input checked="" type="checkbox"/>
186.91.120.146	6	<input type="checkbox"/>
14.180.108.177	4	<input type="checkbox"/>
103.99.203.42	4	<input type="checkbox"/>
209.17.97.16	3	<input type="checkbox"/>
188.165.50.197	3	<input type="checkbox"/>
118.70.52.7	3	<input type="checkbox"/>
36.90.86.142	3	<input type="checkbox"/>
187.252.253.182	3	<input type="checkbox"/>
201.102.99.39	3	<input type="checkbox"/>
113.182.237.114	2	<input type="checkbox"/>
156.207.247.9	2	<input type="checkbox"/>

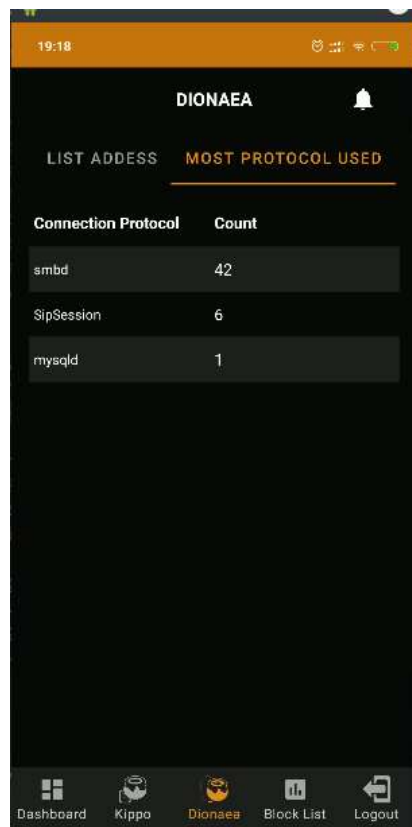
**Gambar 21** *List Address Dionae*

Pada Gambar 21 merupakan tampilan *List Address Dionaea* yang memiliki fitur sebagai berikut:

1. Menampilkan *ip address* sebuah serangan.
2. Menampilkan jumlah serangan berdasarkan *ip address*.
3. Memiliki fitur blokir *ip address* serangan pada *action*

b. *Most Protocol Used*

*Most protocol Used* sebuah tampilan *service* yang paling banyak diserang pada *honeypot dionaea*. *Most protocol used* ditampilkan pada Gambar 22



Connection Protocol	Count
smbd	42
SipSession	6
mysqld	1

**Gambar 22** *Most Protocol Used*

Pada Gambar 22 merupakan tampilan *most protocol used* yang memiliki fitur sebagai berikut:

1. *Connection Protocol*



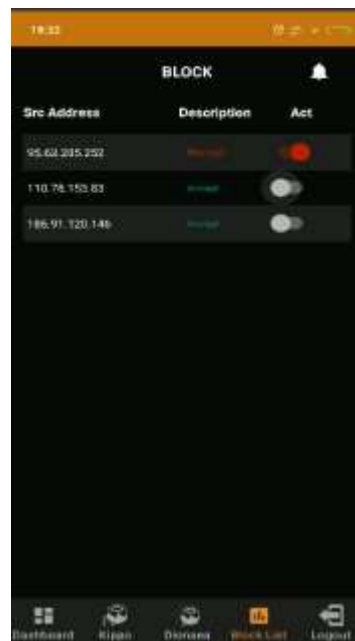
*Connetcion protocol* adalah sebuah service sengaja dibuat untuk diserang yang disediakan oleh *honeypot dionaea*

## 2. Count

*Count* adalah total dari sebuah serangan pada *connection protocol* tersebut

### 4.1.5 Tampilan Halaman *Block List*

Halaman *block list* merupakan tampilan sebuah rentetan *ip address* yang telah diblokir oleh *user*. Ditampilkan pada Gambar 23



**Gambar 23** Tampilan Halaman *Block List*

Pada Gambar 23 merupakan tampilan Halaman *Block List* yang memiliki fitur sebagai berikut :

#### 1. *Src Address*

*Src Address* merupakan *ip address* serangan yang berasal dari *honeypot Kippo* dan *honeypot dionaea*

#### 2. *Description*

*Description* merupakan keterangan apakah *ip address* tersebut telah diblokir atau diijinkan

### 3. *Act*

*Act* merupakan sebuah fitur bloking ip address

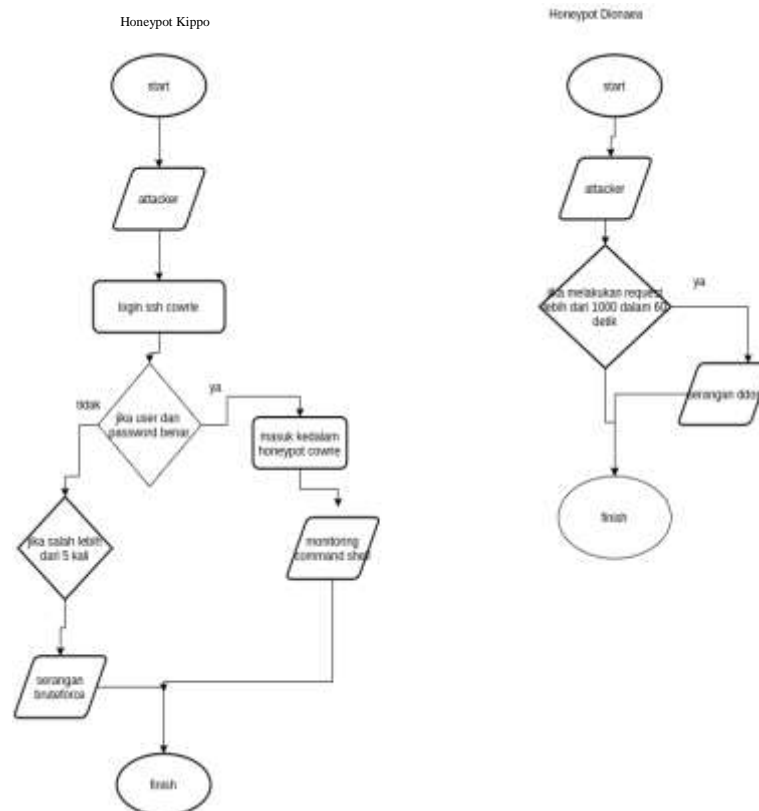
## 4.2 Pengujian

Metode yang digunakan dalam pengujian ini yaitu metode *Blackbox testing*. Tahap pengujian dilakukan untuk menjamin kualitas dan fungsional sehingga dapat berjalan dengan lancar sesuai dengan yang diharapkan. Dilakukan pengujian dengan memasukkan data yang sesuai dan tidak sesuai untuk melihat respon sistem. Spesifikasi *Ip address* dapat ditampilkan pada Tabel 5.

**Tabel 5. *Ip Address***

Hardware	<i>Ip Address</i>
<i>Server target</i>	34.66.225.217
<i>Attacker</i>	36.90.162.248

Skenario serangan dapat ditampilkan pada gambar 24.



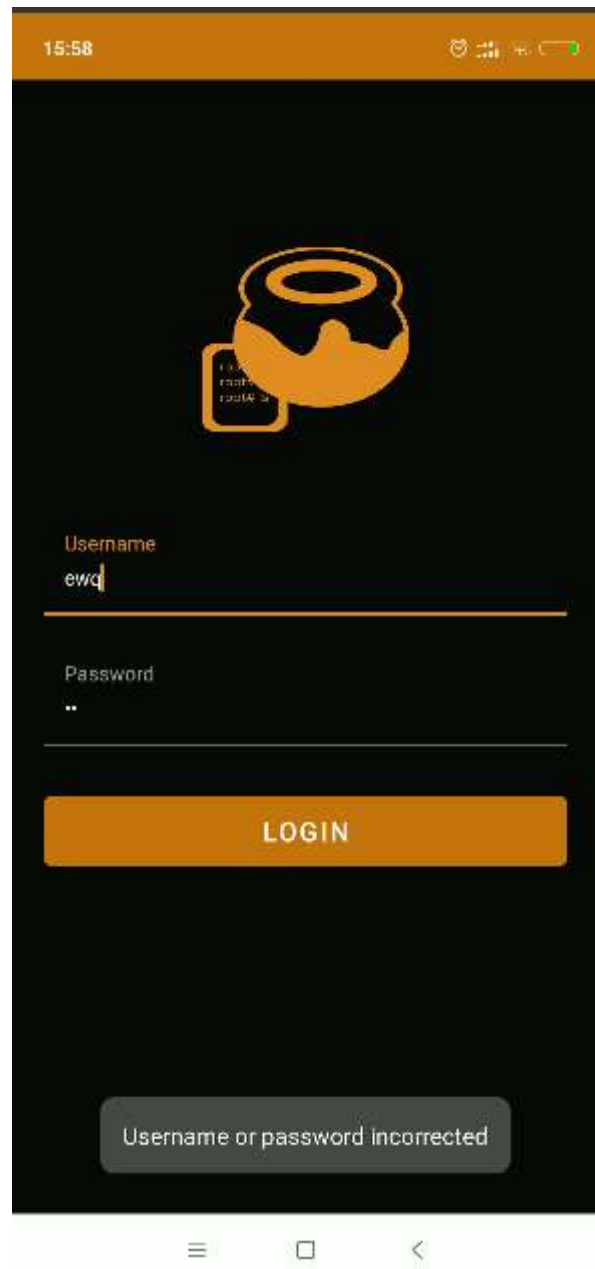
**Gambar 24.** Skenario Serangan

1. HoneyPot Kippo
  - a) Attacker akan mencoba memasuki ssh *kippo/cowrie*
  - b) Jika kombinasi user dan password benar maka otomatis masuk kedalam *command shell honeypot kippo/cowrie*
  - c) Jika kombinasi *user* dan *password* salah sebanyak 5 kali akan terdeteksi sebagai serangan *bruteforce*
2. HoneyPot Dionaea
  - a) Attacker akan melakukan *request* pada *port* layanan *honeypot dionaea*
  - b) Jika melakukan *request* lebih dari 1000 dalam waktu 60 detik maka dianggap serangan *DDoS*

#### 4.2.1 Pengujian Login

Pengguna akan memasukan username dan password yang terdaftar pada *database*, apabila pengguna salah memasukkan *username* atau *password* maka

akan memunculkan pesan error '*username or password incorrected*'. pesan error akan ditunjukkan pada Gambar 24.



**Gambar 25.** Tampilan Pesan *Error Login*

#### **4.2.3 Pengujian *Port Scanning* Menggunakan *Nmap***

Pengujian *port scanning* pada *server* diperuntukkan untuk melihat *port* yang terbuka menggunakan *tool nmap*. Pengujian ditampilkan pada Gambar 25.

```
~: bash — Konsole
File Edit View Bookmarks Settings Help

verrandy@pop-os:~$ nmap 34.66.225.217
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-05 17:38 WIB
Nmap scan report for 217.225.66.34.bc.googleusercontent.com (34.66.225.217)
Host is up (9.22s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
88/tcp    open  http
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 19.95 seconds
verrandy@pop-os:~$
```

**Gambar 26.** Pengujian Port Scanning Menggunakan Nmap

#### 4.2.2 Pengujian Deteksi Serangan Pada *Honeypot Kippo*

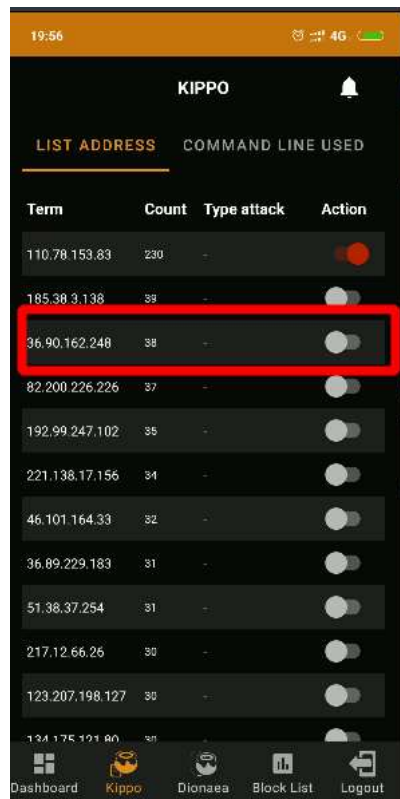
Pengujian deteksi serangan pada *honeypot Kippo* menggunakan tool yang bernama *hydra* untuk menggunakan serangan *bruteforce attack*. Penggunaan dapat ditunjukkan sebagai berikut:

1. Tahap kedua penggunaan tool *hydra*. Dapat ditampilkan pada Gambar 27.

```
verrandy@pop-os:~$ hydra -V -l root -P 00-indonesian-wordlist.lst ssh://34.66.225.217 -s 22
hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-01-02 16:27:26
WARNING! Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 79898 login tries (l:1/p:79898), ~4994 tries per task
[DATA] attacking ssh://34.66.225.217:
[ATTEMPT] target 34.66.225.217 - login "root" - pass "a" - 1 of 79898 [child 0] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "aa" - 2 of 79898 [child 1] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "ab" - 3 of 79898 [child 2] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "aba" - 4 of 79898 [child 3] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "aba-aba" - 5 of 79898 [child 4] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abad" - 6 of 79898 [child 5] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abadi" - 7 of 79898 [child 6] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abadiah" - 8 of 79898 [child 7] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abadiah" - 9 of 79898 [child 8] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abadikan" - 10 of 79898 [child 9] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abab" - 11 of 79898 [child 10] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abal" - 12 of 79898 [child 11] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abalkan" - 13 of 79898 [child 12] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abainana" - 14 of 79898 [child 13] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abak" - 15 of 79898 [child 14] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abaka" - 16 of 79898 [child 15] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abaktinal" - 17 of 79898 [child 16] (0/0)
[ATTEMPT] target 34.66.225.217 - login "root" - pass "abakus" - 18 of 79898 [child 1] (0/0)
```

**Gambar 27.** Penggunaan Tool Hydra

Hasil dari deteksi sebuah serangan *honeypot Kippo* ditampilkan pada Gambar 28.



The screenshot shows the KIPPO application interface. At the top, there's a status bar with the time 19:56 and 4G signal. Below it, the app title 'KIPPO' is centered. There are two tabs: 'LIST ADDRESS' (selected) and 'COMMAND LINE USED'. The main content is a table with four columns: 'Term', 'Count', 'Type attack', and 'Action'. The table lists several IP addresses and their associated counts. The entry for '36.90.162.248' with a count of '38' is highlighted with a red rectangular box. The 'Action' column for each entry contains a toggle switch.

Term	Count	Type attack	Action
110.78.153.83	230	-	<input checked="" type="checkbox"/>
185.38.3.138	38	-	<input type="checkbox"/>
36.90.162.248	38	-	<input type="checkbox"/>
82.200.226.226	37	-	<input type="checkbox"/>
192.99.247.102	35	-	<input type="checkbox"/>
221.138.17.156	34	-	<input type="checkbox"/>
46.101.164.33	32	-	<input type="checkbox"/>
36.89.229.183	31	-	<input type="checkbox"/>
51.38.37.254	31	-	<input type="checkbox"/>
217.12.66.26	30	-	<input type="checkbox"/>
123.207.198.127	30	-	<input type="checkbox"/>
134.175.191.80	28	-	<input type="checkbox"/>

At the bottom, there's a navigation bar with five icons and labels: 'Dashboard', 'Kippo', 'Dionaea', 'Block List', and 'Logout'.

**Gambar 28.** Hasil Serangan Menggunakan *Tools Hydra*

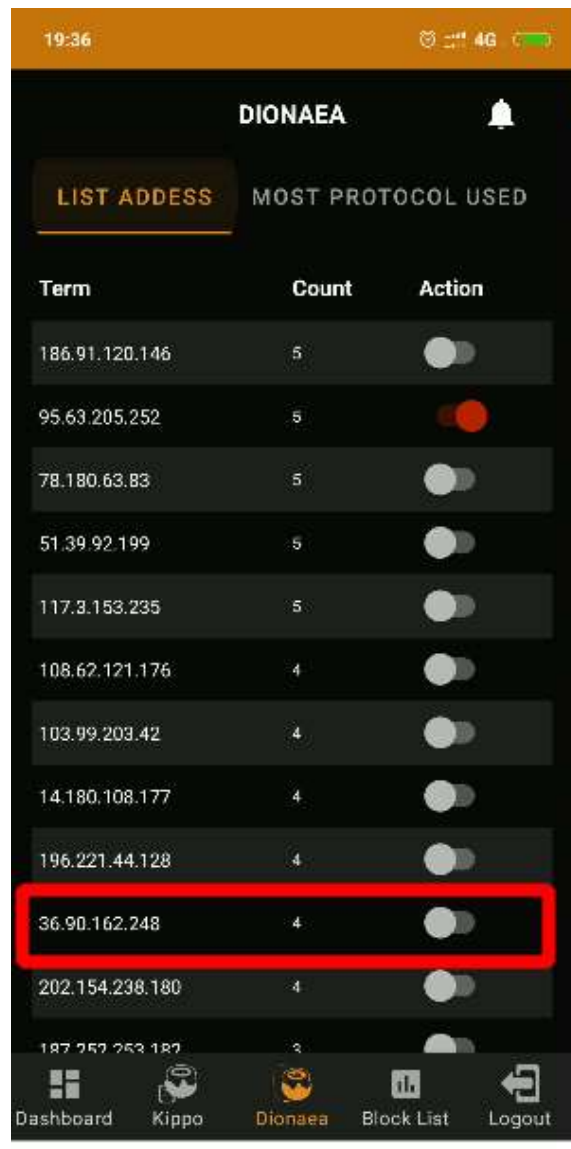
#### 4.2.3 Pengujian Deteksi Serangan Pada *Honeypot Dionaea*

Pengujian deteksi serangan pada *honeypot dionaea* menggunakan *tool* simple yang bernama *hping3* untuk menggunakan serangan *DDoS attacks*. Penggunaan ditampilkan pada gambar.

```
$ sudo hping3 -i u1 -S -p 445 34.66.225.217
[sudo] password for kali:
HPING 34.66.225.217 (eth0 34.66.225.217): S set, 40 headers + 0 data bytes
len=46 ip=34.66.225.217 ttl=64 id=14247 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14248 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14249 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14250 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14251 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14252 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14253 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14254 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
len=46 ip=34.66.225.217 ttl=64 id=14255 sport=445 flags=SA seq=0 win=65535
rtt=0.0 ms
```

**Gambar 29.** Tampilan Gambar *Tool Hping*

Hasil dapat ditampilkan pada Gambar 30.



The screenshot shows the DIONAEA mobile application interface. At the top, the status bar displays the time 19:36, signal strength, 4G network, and battery level. The app header is orange with the title 'DIONAEA' and a notification bell icon. Below the header, there are two tabs: 'LIST ADDRESS' (selected) and 'MOST PROTOCOL USED'. The main content area is a table with three columns: 'Term', 'Count', and 'Action'. The table lists several IP addresses with their respective counts and toggle switches. The row for IP 36.90.162.248 is highlighted with a red rectangle. The bottom navigation bar contains five icons: Dashboard, Kippo, Dionaee (selected), Block List, and Logout.

Term	Count	Action
186.91.120.146	5	<input type="checkbox"/>
95.63.205.252	5	<input checked="" type="checkbox"/>
78.180.63.83	5	<input type="checkbox"/>
51.39.92.199	5	<input type="checkbox"/>
117.3.153.235	5	<input type="checkbox"/>
108.62.121.176	4	<input type="checkbox"/>
103.99.203.42	4	<input type="checkbox"/>
14.180.108.177	4	<input type="checkbox"/>
196.221.44.128	4	<input type="checkbox"/>
36.90.162.248	4	<input type="checkbox"/>
202.154.238.180	4	<input type="checkbox"/>
187.252.253.182	3	<input type="checkbox"/>

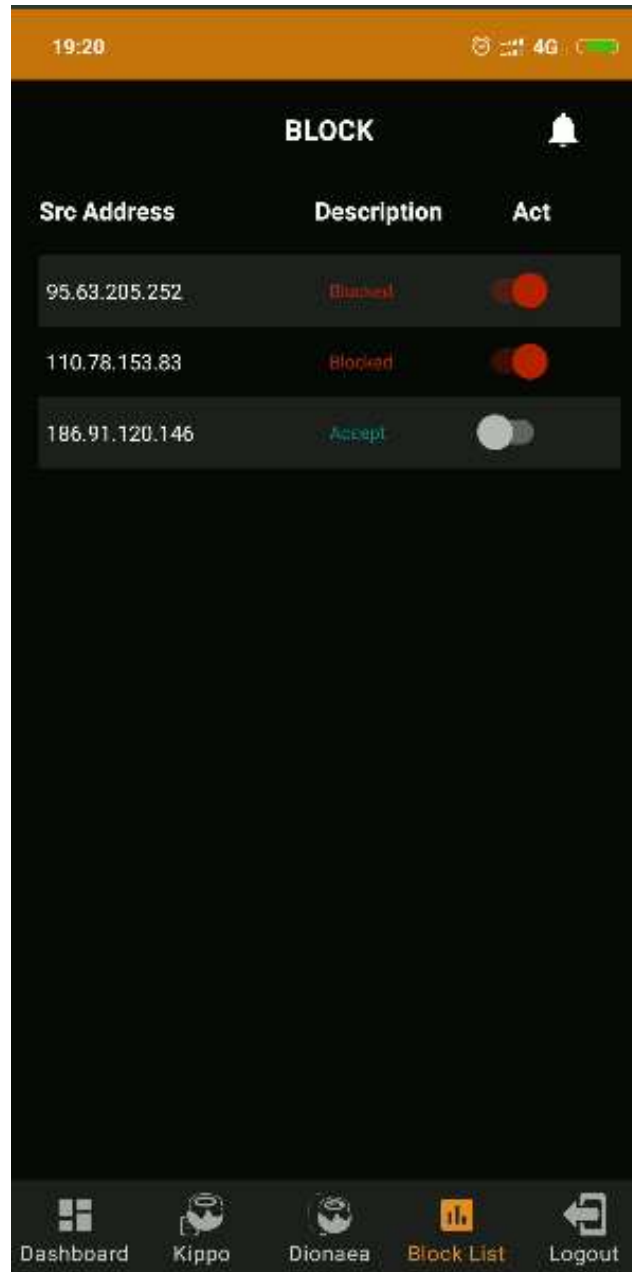
**Gambar 30.** Hasil Aplikasi *Honeypot Dionaee*

Pada Gambar 30 merupakan hasil serangan yang berasal dari tools *Hping*.



#### 4.2.4 Pengujian Fitur *Blocking Address*

Pengujian ini diperuntukkan memblokir *ip address* penyerang. Fitur blokir *address* ditampilkan pada Gambar 31.



**Gambar 31.** Pengujian fitur *block address*

#### 4.2.5 Pengujian *Blackbox Testing*

Berdasarkan hasil pengujian yang dilakukan untuk melihat kesesuaian pada sistem, apakah sistem berjalan sesuai dengan yang diharapkan atau tidak, maka hasil pengujian menghasilkan data yang diperoleh ditunjukkan pada Tabel 5

**Tabel 6. Pengujian *BlackBox***

No.	Features	Skenario Pengujian	Test Case	Hasil Pengujian	Status	Bukti Pengujian
1.	Login	Login menggunakan akun yang sudah terdaftar	Username: verrandy Password : verrandy123	Pengguna akan masuk ke dalam dashboard	Valid	
		Login menggunakan <i>username</i> dan <i>password</i> tidak terdaftar	Username: poliwangi Password : polinema	Sistem menampilkan pesan pemberitahuan <i>username</i> dan <i>password</i> salah	Valid	Gambar 24
2.	Kippo	Melakukan serangan <i>honeypot Kippo</i>	Menggunakan tool <i>hydra</i> dengan <i>user: root</i> dan <i>password</i> menggunakan <i>wordlist</i>	Menampilkan jalannya serang pada tool <i>hydra</i>	Valid	Gambar 27
		Mendeteksi Serangan <i>honeypot Kippo</i>	Memilih <i>menu honeypot Kippo</i>	Menampilkan data serangan <i>honeypot Kippo</i>	valid	Gambar 28
3.	Dionaea	Melakukan serangan <i>honeypot dionaea</i>	Menggunakan tools <i>PyDDoS</i>	Menampilkan tools <i>PyDDoS</i>	Valid	Gambar 29
		Mendeteksi Serangan	Memilih menu	Menampilkan list serangan	Valid	Gambar 29

		<i>honeypot dionaea</i>	<i>honeypot dionaea</i>	yang terdeteksi		
5.	Block	Melakukan action block pada <i>address</i> terpilih yang terdapat <i>honeypot dionaea</i> dan Kippo	<i>Click</i> action	Aplikasi akan menampilkan <i>action merah</i> apabila ip telah ter block	Valid	Gambar 31,

.



## BAB 5 PENUTUP

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya mengenai Aplikasi Deteksi Serangan Menggunakan *Honeypot* Berbasis *Android*:

1. Aplikasi *Honeypot* ini dibuat dengan bahasa pemrograman Java, dengan menggunakan Backend web server framework *django* sebagai rest api dan *MongoDB* sebagai basisdata. Dalam penerimaan sebuah serangan DDoS dan bruteforce menggunakan honeypot *dionaea* dan *Kippo*
2. Aplikasi *Honeypot* hanya dapat digunakan oleh *admin* yang pemilik sebuah *VPS(Virtual Private Server)*.
3. Aplikasi *Honeypot* mengambil data yang berasal dari rest api backend web server *django* yang telah terintegrasi dengan log yang berasal dari honeypot *Kippo* dan *dionaea*.

Aplikasi Deteksi serangan honeypot ini masih belum bisa dikatakan sempurna dan masih ada perlunya pengembangan sistem untuk membangun sebuah security lebih baik

#### 5.2 Saran

Aplikasi Deteksi serangan honeypot ini masih belum bisa dikatakan sempurna dan masih ada perlunya pengembangan sistem untuk membangun sebuah security lebih baik. Beberapa saran untuk mengembangkan aplikasi sebagai berikut:

1. Melakukan manajemen *caching manajemen web server* agar dapat menerima data *rest api* secara *low latency*.
2. Menambahkan tool *fail2ban* untuk memblokir serangan secara otomatis .
3. Menambahkan fitur push notification apabila ada sebuah serangan baru yang dilakukan secara masif.
4. Membuat manajemen server agar lebih optimal dalam menghandle sebuah serangan.



## DAFTAR PUSTAKA

- Bertino, E., dan Islam, N. (2017). *Botnets and Internet Of Things Security Computer*.
- Django Software Foundation. (n.d.). Retrieved 2020, from [www.djangoproject.com](https://www.djangoproject.com/): <https://www.djangoproject.com/>
- Harisantyo, B., Nugraha, L. S., Prasetyawan, N., Nugraha, S. N., & Sulaiman. (2015). *Makalah Pemrograman Berbasis Objek Diagram Activity*. Depok: Universitas Gunadrama Fakultas Teknologi Industri Teknik Informatika.
- Jubilee Enterprise. (2015). *Mengenal Dasar Dasar Pemrograman Android*.
- Kosasi, S., & Yuliani, I. A. (2015). Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online. *Jurnal Simetris*, 6(1), 28-29.
- Pratita, H. S. (2016). Analisa Brute Force Attack menggunakan Scanning Aplikasi pada HTTP Attack.
- Python Software Foundation. (n.d.). Retrieved 2020, from [www.python.org](https://www.python.org/doc/essays/blurb/): <https://www.python.org/doc/essays/blurb/>
- Rembulan, A. D. (2015). *Makalah Unified Modeling Language (UML)*. Cirebon: Stikom Poltek.
- Rosa A.S., M. S. (2014). *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung.
- Rukmana, A., & Desiyani, I. D. (2017). *Metodologi Dan Metode Rapid Application Development (RAD)*. Sumedang: Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Sumedang.





Septian Geges, W. W. (2015). *Pengembangan Pencegahan Serangan Distributed Denial Of Service (DDoS) Pada Sumber Daya Jaringan Dengan Integrasi Network Behaviour Analysis dan Client Puzzle*.

Solomon Z. Melese, P. A. (2016). *Honeypot System for Attacks on SSH Protocol*.

Sucipto. (2017). Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar. *Jurnal Intensif*, 1(1), 35-36.

Tamminen, U. (2016, September 30). *SSH Honeypot*. Retrieved from <https://github.com/desaster/Kippo>

Wibisono, L. A. (2016). *PENGENDALIAN “ROLLBOT” MENGGUNAKAN ANDROID MELALUI BLUETOOTH DAN ARDUINO [ TUGAS AKHIR ]*.

Yusmiarti, K. (2016). Perancangan Sistem Distribusi Produk Teh Hitam Berbasis Web Pada PTPN VII Gunung Dempo Pagar Alam. *Jurnal Informatika*, 4(2), 3-4.