IMPLEMENTASI *HONEYPOT* DENGAN *RASPBERRY* PI SEBAGAI ALAT BANTU PENDETEKSI KEAMANAN JARINGAN DAN PENANGKAP *MALWARE*

HONEYPOT IMPLEMENTATION WITH PI RASPBERRY AS A TOOL FOR SECURITY NETWORK DETECTION AND MALWARE CAPTURE

Refan Andros[1], Lukas[2]
Jurusan Teknik Elektro – Fakultas Teknik
Universitas Katolik Indonesia Atma Jaya – Jakarta
[1]refan_andros@hotmail.com, [2]lukas@atmajaya.ac.id

Abstrak

Saat ini *internet* sudah menjadi kebutuhan yang mendasar dalam hal komunikasi. Salah satu permasalahan keamanan dalam *internet* adalah *malware*. *Malware* adalah sebuah program atau perangkat lunak yang diciptakan untuk tujuan tidak baik. Untuk menangani masalah *malware*, diperlukan suatu alat yang dapat membantu mendeteksi *malware* dengan menangkapnya terlebih dahulu tanpa menjalankannya. Alat yang tepat untuk menangani masalah tersebut adalah *Honeypot Dionaea*. *Honeypot Dionaea* tidak memerlukan spesifikasi tinggi untuk dapat dijalankan sehingga *Raspberry Pi* cocok digunakan sebagai sensor *Honeypot Dionaea*. Data yang diperoleh *Dionaea* bukan hanya *malware*, tetapi juga data yang berkaitan dengan *malware* tersebut. Data tersebut akan diproses oleh *server* untuk ditampilkan dalam bentuk halaman *web*. *Server* tersebut akan dibangun menggunakan *Django* berbasis *Python* dan menggunakan basis data *SQLite*. Dari hasil pengujian, *Honeypot Dionaea* mampu menangkap *malware* yang kemudian dapat digunakan untuk membuat laporan dalam halaman *web*. Laporan tersebut dapat membantu analisis tentang frekuensi asal *malware*, frekuensi *port* yang diserang, dan frekuensi serangan rata-rata dalam sehari.

Kata kunci: *django, python, SQLite*3, *honeypot, raspberry pi*.

Abstract

Internet has become a fundamental need in Communications today. One of the internet security issues is malware, a program or software created to harm computers or computer system. A tool to detect and remove malware before it infects the computer is required to deal with this problem. The most appropriate tool to perform the task is Honeypot Dionaea. Since Dionaea Honeypot does not require high specifications to run, the Raspberry Pi is suitable for a Honeypot Dionaea sensor. The data obtained from the Honeypot Dionaea are not only malware but also other data related to the malware. The data are then processed by the analysis server and presented in a form of a web page. The server is built using the Phython-based Django and SQLite Database. The test results showed that Dionaea honeypot was able to capture the malware, of which the data can be used to create a report in the web page. The report could be used to analyze the original malware frequency, the frequency of attacked port, and the average frequency of attacks per day.

Keywords: *django, python, SQLite*3, *honeypot, raspberry Pi.*

Tanggal Terima Naskah : 19 November 2014
Tanggal Persetujuan Naskah : 01 Desember 2014

1. PENDAHULUAN

Saat ini *internet* sudah menjadi kebutuhan dalam bidang komunikasi. Jumlah pengguna *internet* yang semakin banyak, menjadikan *internet* sebagai alat penunjang dalam kebutuhan sehari-hari. Salah satu permasalahan keamanan dalam *internet* adalah *malware*. *Malware* beredar luas di dunia *internet*. Banyak informasi, seperti data pribadi, data akun bank, dan data sensitif lainnya, disimpan di banyak komputer yang terhubung dengan *internet*. Beberapa jenis *malware* dapat mencuri data, sehingga dapat merugikan pemiliknya. Oleh karena itu, faktor keamanan menjadi sangat penting dalam era komunikasi saat ini.

Malware adalah suatu perangkat lunak yang dibuat untuk menyusup atau merusak sistem komputer. *Malware* dapat mencuri data penting dari sistem komputer yang disusupinya. *Malware* terus berkembang sehingga perilaku *malware* ini menjadi sulit dideteksi oleh sistem keamanan komputer pada umumnya, hingga sistem tersebut telah disusupi. Untuk menangani masalah *malware* diperlukan suatu alat yang dapat mendeteksi *malware* dengan menangkapnya terlebih dahulu tanpa menjalankannya. Alat yang tepat untuk menangani masalah tersebut adalah *Honeypot Dionaea*. Data yang diperoleh *Dionaea* bukan hanya *malware*, tetapi juga data yang berkaitan dengan *malware* tersebut. Data tersebut memerlukan sebuah visualisasi agar dapat dianalisis dan dijadikan dalam bentuk laporan.

Honeypot Dionaea ini akan dipasang pada *Raspberry Pi* karena relatif murah dan hemat daya dibandingkan dengan komputer *desktop*. Untuk menjalankan *Honeypot Dionaea* tidak memerlukan spesifikasi tinggi, sehingga dapat dijalankan pada *Raspberry Pi.*

2. KONSEP DASAR

2.1 *Linux Raspbian Pi*

Linux Raspbian Pi adalah sistem operasi komputer yang digunakan pada alat *Raspberry Pi*. *Raspbian* merupakan *port* tidak resmi dari *Debian Wheezy Accorn Reduced Instrucion Set Computing Machine Hard Float* (ARMHF), dengan pengaturan kompilasi yang disesuaikan, untuk menghasilkan program *hard float* yang akan dijalankan pada *Raspberry Pi*. Hal ini memberikan kinerja lebih cepat secara signifikan untuk aplikasi yang banyak menggunakan operasi aritmatika *floating point*. Semua aplikasi lain juga akan mempunyai performa yang lebih tinggi dengan penggunaan instruksi lanjutan dari *Accorn Reduced Instrucion Set Computing Machine* (ARM)v6 *Central Processing Unit* (CPU) di *Raspberry Pi* [1].

2.2 *Raspberry Pi*

Raspberry Pi adalah komputer dengan ukuran sebesar kartu kredit yang disambungkan ke televisi dan *keyboard*. *Raspberry Pi* merupakan komputer kecil yang dapat digunakan seperti *desktop* PC untuk aplikasi *spreadsheets*, *word-processing*, dan permainan. *Raspberry Pi* dapat menjalankan *high-definition video*.

Raspberry Pi mempunyai dua model, yaitu, model A dan model B. *Raspberry Pi* model A memiliki soket *Ethernet*, satu soket *universal serial bus* (USB), dan memori sebesar 256 MB sedangkan model B memiliki soket *Ethernet*, dua soket USB, dan memori sebesar 512 MB [2].

2.3 *SQLite*

Dalam istilah sederhana, *SQLite* adalah paket perangkat lunak *domain* publik yang menyediakan sistem manajemen basis data relasional. Selain penyimpanan data dan manajemen, *SQLite* dapat memroses perintah *query* yang kompleks, yang menggabungkan data dari beberapa tabel untuk menghasilkan laporan dan ringkasan data. Suku kata *Lite* pada *SQLite* tidak mengacu pada kemampuannya. Sebaliknya, *SQLite* ringan pada kompleksitas pengaturan, administrasi *overhead*, dan penggunaan sumber daya. *SQLite* memiliki fitur berikut:

1. *Serverless*
   SQLite tidak memerlukan proses *server* yang terpisah atau sistem untuk beroperasi. *Library SQLite* mengakses *file storage* secara langsung.
2. *Zero Configuration*
   Tidak adanya *server* berarti tidak memerlukan konfigurasi. Hal ini membuat basis data *SQLite* semudah membuka *file*.
3. *Cross-Platform*
   Seluruh basis data berada dalam *file cross-platform* tunggal sehingga tidak memerlukan administrasi.
4. *Self-Contained*
   Sebuah perpustakaan tunggal berisi seluruh sistem basis data, yang terintegrasi langsung ke aplikasi *host.*
5. *Small Runtime Footprint*
   Sistem dibangun kurang dari satu *megabyte* kode dan hanya memerlukan beberapa *megabyte* memori. Dengan beberapa penyesuaian, ukuran *library* dan penggunaan memori dapat berkurang secara signifikan [3].

2.4 *Honeypot*

Menurut Lance Spitzner [4], *Honeypot* adalah sumber daya keamanan yang mempunyai nilai jika sistem disusupi atau diserang. Pada dasarnya *Honeypot* merupakan suatu alat untuk mendapatkan informasi dari penyerang. *Honeypot* merupakan sistem yang dirancang untuk diperiksa dan diserang.

Honeypot Dionaea merupakan salah satu *Honeypot* interaksi rendah yang bertujuan menangkap salinan *malware* berbahaya yang masuk ke dalam sistem. *Malware* tersebut biasanya ada pada layanan yang ditawarkan dalam jaringan. *Dionaea* menggunakan *Python* sebagai bahasa *script* dan *libemu* sebagai pemecah kode. *Dionaea* mendukung *Internet Protocol* v6 dan *Transport Layer Security* (TLS).

2.5 *Python*

Python adalah bahasa pemrograman yang dinamik, yang banyak digunakan secara luas dari banyak *domain* aplikasi, seperti pengembangan *website* dan *internet*, akses basis data, *Desktop Graphical User Interface*, ilmiah dan numerik, pendidikan, pemrograman jaringan, permainan, dan *Graphic 3D*. *Python* dapat berjalan pada semua sistem operasi, seperti *Linux*, *Windows*, *Mac*, *Omega*, dan lainnya. Bahasa pemrograman ini memiliki lisensi *open-source* yang dapat dengan gratis digunakan atau didistribusikan bahkan untuk penggunaan komersial [5].

2.6 *Malware*

Malware yang merupakan singkatan dari *malicious software*, adalah sebuah program atau perangkat lunak yang diciptakan untuk tujuan menyusup, mengganggu, atau bahkan merusak sistem operasi pada suatu perangkat komputer. Program ini dapat

menjalankan suatu perintah tertentu pada perangkat yang disusupinya. Jika salah satu perangkat sudah terinfeksi program *malware*, maka perangkat tersebut dapat menjalankan atau melakukan sesuatu tanpa sepengetahuan pemilik dengan tujuan tidak baik. Contoh dari *malware* adalah *Virus, Worm, Wabbit, Keylogger, Browser Hijacker, Trojan Horse, Spyware, Backdoor, Dialer, Exploit,* dan *Rootkit* [6].

2.7 *Secure Socket Shell*

Secure Socket Shell (SSH) adalah protokol program yang digunakan untuk *login* ke komputer lain melalui suatu jaringan, untuk mengeksekusi perintah-perintah pada mesin atau komputer lainnya. SSH menyediakan autentikasi kuat dan komunikasi yang aman lewat jaringan tidak aman [7].

SSH dapat digunakan untuk *login* ke komputer lain dan mengeksekusi berbagai perintah. SSH mendukung *tunneling, forwarding TCP port*, dan *X11*, sehingga dapat mengirim *file* melalui protokol *Secure File Transfer Protocol* (SFTP) dan *Secure Copy* (SCP).

2.8 *Hyper Text Markup Language*

Hyper Text Markup Language (HTML) merupakan bahasa standar pemrograman untuk membuat suatu halaman *web*, terdiri dari kode-kode yang akan memerintahkan *web browser* untuk menampilkan halaman *web* dengan berbagai macam format halaman, seperti teks, grafik, tautan, *audio*, dan *video* [8]. *Web browser* merupakan program yang dapat menerjemahkan kode program HTML sehingga dapat ditampilkan ke layar. Contoh *web browser* adalah *Internet Explorer, Mozzila Firefox*, dan *Google Chrome.*

2.9 *Django*

Django merupakan *web framework* yang bersifat gratis dan *open source* berbasis *Python*, menggunakan arsitektur *Model-View-Controller* (MVC). *Web framework* itu sendiri adalah sebuah alat yang digunakan untuk mempermudah dalam pembangunan sebuah situs *web*. *Django* menekankan pada pembuatan aplikasi yang dapat digunakan secara berulang-ulang dan mudah dikembangkan tanpa harus mengulang kode program yang sama. Model adalah lapisan yang digunakan untuk berinteraksi dengan basis data, *Template* adalah lapisan presentasi untuk HTML, sedangkan *View* adalah lapisan yang berisikan kode yang mengolah data dari *model* dan mengirimkannya ke dalam *Template*.

Django mempunyai beberapa kelebihan, antara lain [9]:

1. *Object-relational mapper*, yaitu mendefinisikan data model dalam *Python* dan menggunakan *Application Programming Interface* (API) untuk mengakses data tersebut.
2. *Automatic admin interface Django* menyediakan antarmuka *admin* secara otomatis.
3. *Elegant URL design*: pembuatan URL yang lebih mudah dan fleksibel.
4. *Template system*: pembuatan halaman dengan menggunakan *Template*
5. *Internationalization*: *Django* didesain untuk mempermudah dalam pembuatan *web* dalam berbagai bahasa.

2.10 *Javascript*

JavaScript [8] adalah bahasa yang digunakan untuk membuat program, yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja. *JavaScript* memberikan beberapa fungsionalitas ke

dalam halaman *web*, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antarmuka *web* [8].

JavaScript merupakan bahasa *script*, bahasa yang tidak memerlukan *compiler* untuk menjalankannya, cukup dengan *interpreter*. Tidak adanya proses kompilasi terlebih dahulu agar program dapat dijalankan. *Browser web Netscape Navigator* dan *Internet Explorer* adalah salah satu contoh *interpreter*, karena kedua *browser* ini telah dilengkapi dengan *interpreter JavaScript*. Namun, tidak semua *browser web* dapat menjadi *interpreter JavaScript* karena belum tentu *browser* tersebut dilengkapi dengan *interpreter JavaScript*.

JavaScript adalah bahasa *script* yang ringan dan mudah digunakan. Dengan adanya *JavaScript* ini, maka kini halaman *web* tidak hanya menjadi halaman data dan informasi saja, tetapi juga dapat menjadi suatu program aplikasi dengan antarmuka *web*. *JavaScript* merupakan bahasa pemrograman yang tidak membutuhkan lisensi. Jika *web browser* yang digunakan mendukung *JavaScript* maka aplikasi berbasis *web* dapat dibuat menggunakan *JavaScript*.

Pada umumnya program *JavaScript* adalah program yang disisipkan ke dalam halaman *web*, sehingga halaman *web* menjadi sebuah aplikasi yang berjalan di dalam *web browser*. Beberapa sistem operasi menggunakan *JavaScript* untuk membuat aplikasi *non-web*, seperti sistem operasi *MS Windows*, yang menggunakan istilah *Windows Scripting Host* (WSH) sebagai *interpreter JavaScript* dan *Vbscript*. Dengan demikian, program yang dibuat dengan *JavaScript* dan *VBScript* dapat langsung dijalankan di atas sistem operasi, tanpa harus menggunakan *browser web* terlebih dahulu.

2.11 *Virtual Machine*

Virtual machine (VM) [9] adalah suatu *environment*, umumnya sebuah program atau sistem operasi, yang tidak ada secara fisik tetapi dijalankan dalam *environment* lain. Dalam hal ini VM disebut *guest* dan *environment* yang menjalankannya disebut *host*.

3. PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Sistem yang akan dimplementasikan adalah *Honeypot Dionaea* menggunakan *Raspberry Pi* dan *server* analisis yang mengambil hasil dari *Honeypot Dionaea*. *Honeypot Dionaea* yang dipasang pada *Raspberry Pi*, berfungsi sebagai sensor untuk menangkap *malware*.

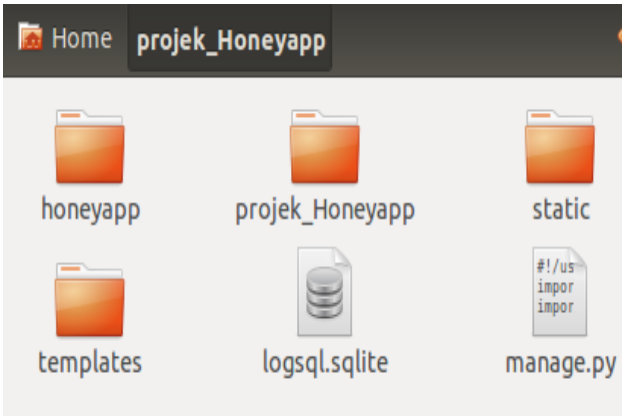
Bila terjadi serangan, maka sistem akan mengemulasikan *Honeypot Dionaea* untuk mengambil data dari penyerang dan membuat serangan tersebut seolah-olah berhasil. Data yang diambil akan disimpan ke dalam *log* dalam basis data. Basis data yang digunakan adalah *SQLite*. Basis data ini akan diunduh melalui *protocol* ssh ke dalam *server* analisis tanpa mengganggu proses kerja *Honeypot Dionaea*. *Server* analisis ini akan memroses data dari *Honeypot Dionaea* untuk ditampilkan dalam bentuk halaman *web*. Pada *server* analisis akan digunakan *Django* sebagai *web framework*.

3.2 Konfigurasi *File* pada *Server* Analisis

Server analisis yang telah dipasang *Django* versi 1.6 ini akan dikonfigurasi dengan susunan *file* pada beberapa *folder* seperti pada Gambar 1, dengan susunan *file project* pada Gambar 2.

```
/projek_Honeyapp
   /honeyapp
            /admin.py
            /__init__.py
            /models.py
            /views.py
            /tables.py
            /test.py
      /static
            /js
      /templates
      /projek_Honeyapp
            /__init__.py
            /settings.py
            /wsgi.py
            /urls.py
      logsql.sqlite
```

Gambar 1. Hierarki *file* pada *project django*

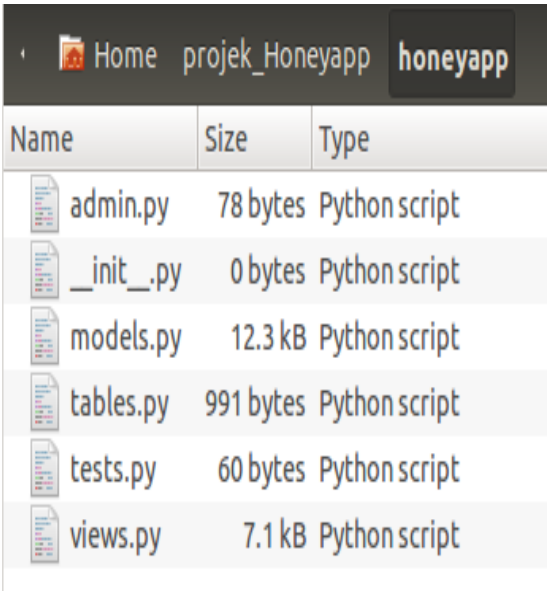


Gambar 2. Susunan *file project django*

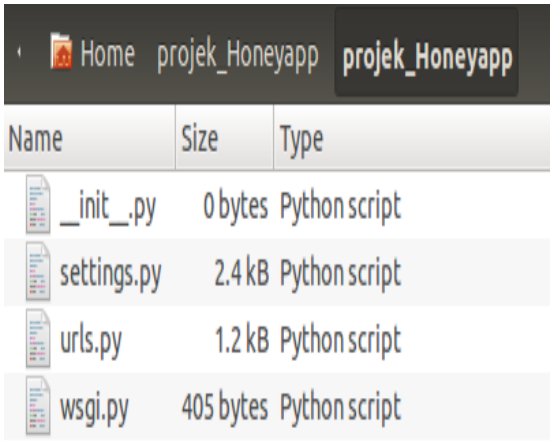
Folder templates berisi *file* berekstensi .html untuk halaman situs. *Folder static* berisi *script javascript* yang diperlukan untuk membuat *chart*. *File* logsql.sqlite merupakan basisdata dari *server* yang diperoleh dari *Honeypot Dionaea* pada *Raspberry Pi*. *File* manage.py merupakan *file* yang dibentuk secara *default* saat membuat *project* dengan *Django*.

Isi *folder honeyapp* adalah admin.py, models.py, views.py, init.py dan tables.py. Admin.py merupakan *file* untuk menentukan bagian dari model basis data yang masuk ke dalam situs *admin*. Models.py merupakan *file* yang berisi kode model basis data. Views.py merupakan *file* yang berisi kode halaman yang akan dijalankan. Tables.py merupakan *file* untuk membuat halaman tabel pada *library Django*-tables2. Isi *folder honeyapp* dapat dilihat pada Gambar 3.

Folder projek_Honeyapp berisi *file* init.py, settings.py, urls.py, dan wsgi.py. *File* settings.py merupakan konfigurasi dari projek *Django*. Urls.py merupakan *file* untuk mengkonfigurasi *uniform resource locator* (URL) yang terdapat pada *server* analisis. Wsgi.py adalah *file* untuk konfigurasi *Django* pada saat pemasangan. Isi *folder* projek_Honeyapp/projek_Honeyapp dapat dilihat pada Gambar 4.



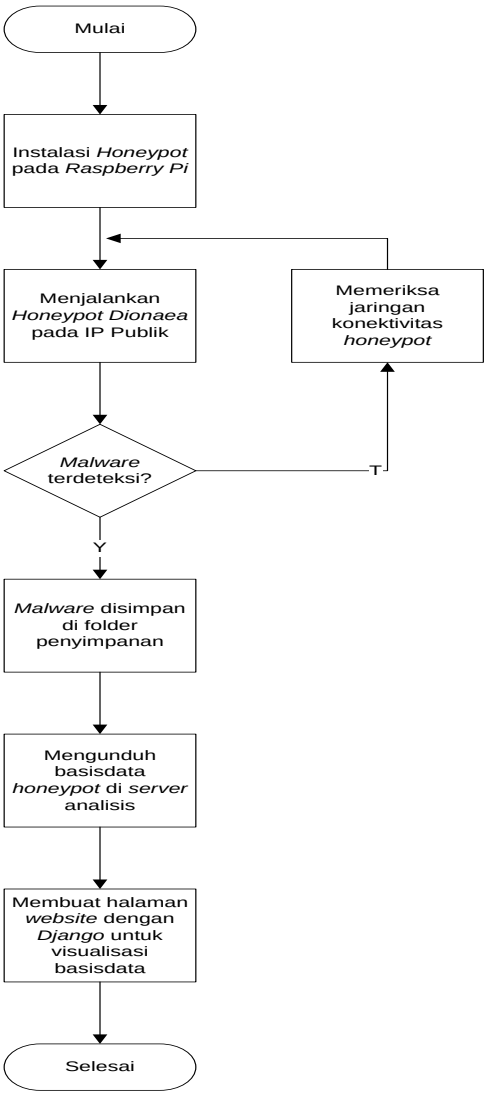
Gambar 3. Isi *folder honeyapp*



Gambar 4. Isi *folder* projek_honeyapp/ projek_honeyapp

3.3 Implementasi *Honeypot Dionaea*

Impementasi *Honeypot Dionaea* dilakukan dengan tahapan: instalasi *Honeypot* pada *Raspberry Pi*, jalankan *honeypot* pada IP Publik, jika terdeteksi, *malware* akan disimpan pada *folder* penyimpanan, unduh data tentang *malware* untuk ditampilkan pada halaman *web* sebagai laporan. Proses ini ditunjukkan pada Gambar 5.



Gambar 5. Diagram alir implementasi *honeypot dionaea*

3.4 Konfigurasi *Honeypot Dionaea* pada *Raspberry Pi*

Honeypot Dionaea dipasang pada *Raspberry Pi* dengan sistem operasi *Raspbian*. Tahap yang dilakukan untuk pemasangan *Honeypot Dionaea* adalah sebagai berikut:

1. Tambahkan *repository* pada *source list raspbian* (/etc/apt/sources.list)

```
deb http://packages.s7t.de/raspbian wheezy main
```

2. *Update source list raspbian*

```
sudo apt-get update
```

3. *Install Dionaea* dan *dependency*-nya

```
sudo apt-get install libglib2.0-dev libssl-dev
libcurl4-openssl-dev libreadline-dev libsqlite3-dev
libtool automake autoconf  build-essential subversion
git-core flex bison pkg-config libnl-3-dev libnl-genl-3-dev
libnl-nf-3-dev libnl-route-3-dev liblcfg libemu libev
dionaea-Python dionaea-cython libpcap udns dionaea
```

4. Salin konfigurasi standar *dionaea*

```
sudo cp /opt/dionaea/etc/dionaea/dionaea.conf.dist
/opt/dionaea/etc/dionaea/dionaea.conf
```

5. Ubah *file* /etc/rc.local agar dionaea dapat berjalan secara otomatis saat awal *boot* dengan menambahkan:
```
_IP=$(hostname –I) || true
    If [ “$_IP”]; then      Printf “My IP address is    %s\n” “$_IP”
    fi
/opt/dionaea/bin/dionaea –D
    Exit 0
```
6. Ganti konfigurasi sshd agar *port honeypot* tidak *conflict* dengan akses ssh dari luar. Caranya dengan mengganti konfigurasi pada *file* /etc/ssh/sshd_config, mengganti *port* 22 dengan *port* 22888
7. *Restart Raspberry Pi*
   Honeypot ini akan dipasang pada sebuah *Internet Protocol* (IP) publik dan berjalan selama 24 jam.

3.5 Pembuatan Basis Data pada *Server* Analisis Berdasarkan Basis Data *Honeypot Dionaea*

Basis data pada *Honeypot Dionaea* merupakan basis data *SQLite*. Basis data *SQLite* tidak dapat diakses dari luar sehingga untuk memperoleh data dari basis data tersebut, basis data tersebut perlu diunduh ke *server* analisis. *Server* analisis mampu *memroses* data pada basis data tersebut tanpa mengganggu *Honeypot Dionaea* pada *Raspberry Pi*. Basis data pada *server* analisis diperoleh dari hasil unduhan dengan menggunakan protokol SCP.

Cara mengunduh basis data *Honeypot Dionaea* pada *Raspberry Pi* dengan menggunakan perintah baris pada terminal *server* analisis, seperti berikut:
```
scp -P 22888 pi@110.35.83.17:
/opt/dionaea/var/dionaea/logsl.sqlite home/nama_komputer/projek_Honeyapp
```

Dengan menggunakan fitur *inspectdb* dari *Django*, yaitu membuat model dari basis data yang sudah ada maka basis data *Honeypot Dionaea* dapat digunakan pada *server* analisis.

Setelah itu jalankan perintah baris pada terminal:
```
python manage.py inspectdb >    models.py
```

Perintah tersebut befungsi untuk membuat model basis data berdasarkan basis data *Honeypot Dionaea Raspberry Pi*. Hasil model tersebut akan digunakan pada aplikasi sebagai basis data yang terpisah dari *honeypot* sehingga proses pengolahan basis data dapat dilakukan di *server* analisis tanpa mengganggu kerja *Honeypot Dionaea* pada *Raspberry Pi*.

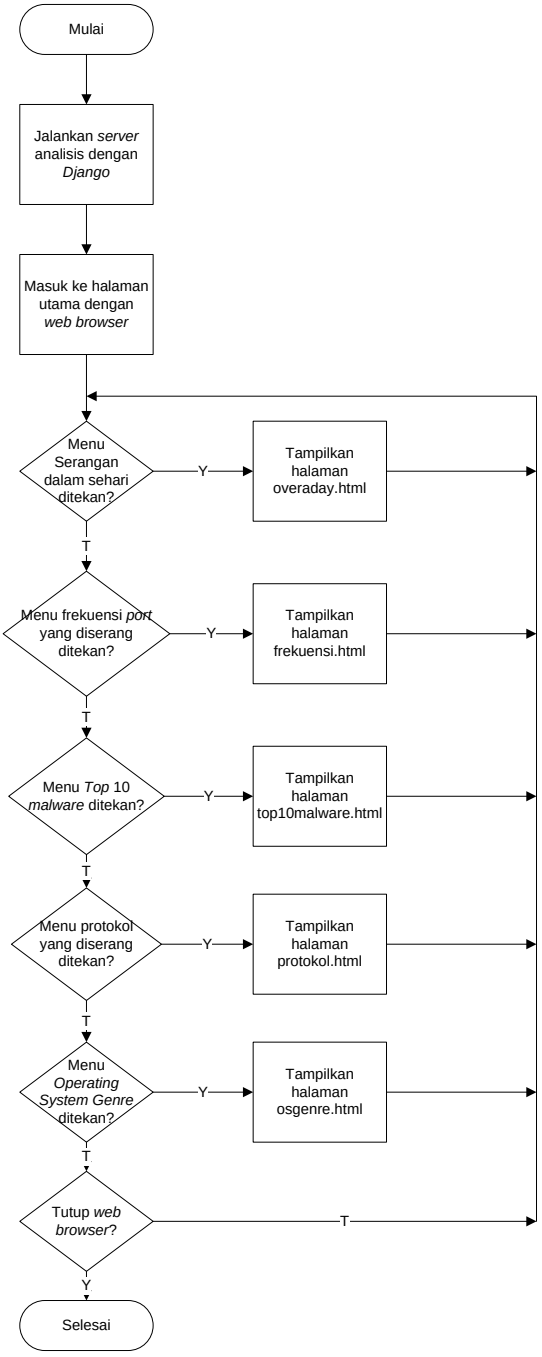
3.6 Perancangan Aplikasi *Web* untuk Visualisasi Basis Data *Honeypot Dionaea*

Aplikasi *web* memiliki fasilitas untuk menampilkan laporan tentang serangan *malware* dalam sehari, frekuensi *port* yang diserang, 10 jenis *malware* yang paling sering menyerang, protokol yang diserang, dan sistem operasi yang diserang. Gambar 6 menunjukkan diagram alir tentang fasilitas pada aplikasi *web*.

Aplikasi ini merupakan sebuah antarmuka pengguna yang memudahkan seorang administrator jaringan untuk memperoleh hasil dari basis data *dionaea* yang telah diperoleh dari *Raspberry Pi*. Aplikasi *web* ini akan dibuat dengan bahasa pemrograman *Python* dan dipasang pada sebuah *Virtual Machine*. Aplikasi ini bertujuan untuk memudahkan *administrator* jaringan dalam mengamati *malware* yang menyerang *honeypot*. Untuk pembuatan aplikasi ini diperlukan beberapa aplikasi pendukung. Berikut ini adalah aplikasi yang dipasang untuk membuat aplikasi *web* tersebut:

1. *Django* versi 1.6
2. *SQLite* 3
3. *Python*
4. *Django-tables2*

Pada halaman utama aplikasi ini akan ditampilkan pada sebuah base.html yang berisi beberapa menu untuk menampilkan hasil dari basisdata *dionaea*, dapat dilihat pada Gambar 7. Pada saat menu tersebut ditekan maka akan menampilkan data yang sesuai dengan hasil *query*.



Gambar 6. Diagram alir aplikasi *web*

Pada halaman utama *website* terdapat pilihan:

1. Serangan dalam sehari
   Menampilkan *chart* serangan yang terjadi dalam sehari.
2. Frekuensi *port* yang diserang
   Menampilkan *chart* frekuensi *port* yang diserang.
3. *Top* 10 *Malware*
   Menampilkan *malware* yang paling sering diunduh oleh *honeypot*.
4. *Remote host attacker*
   Menampilkan alamat *Internet* Protocol (IP) yang sering menyerang *honeypot*.
5. *Operating system genre*
   Menampilkan jumlah serangan *malware* pada sistem operasi tertentu.

Gambar 7. Halaman utama

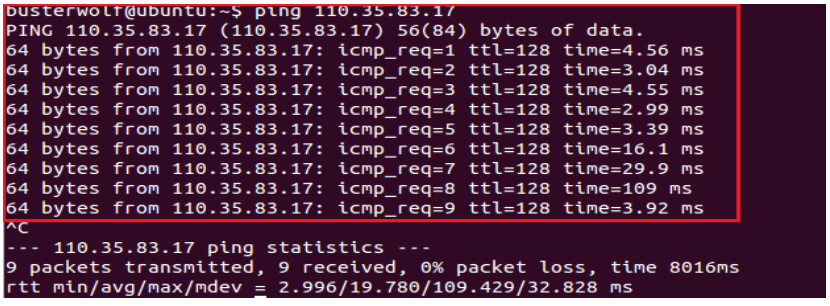
Untuk menampilkan data dengan *chart* digunakan *script Javascript* yang bernama *highchart*. *Javascript* ini akan membantu menampilkan data dalam bentuk *chart* yang memudahkan dalam analisis.

4. PENGUJIAN SISTEM

Pengujian sistem ini dilakukan melalui dua tahap, yaitu, pengujian *Honeypot Dionaea* pada *Raspberry Pi* kemudian pengujian *server* analisis dalam mengolah data

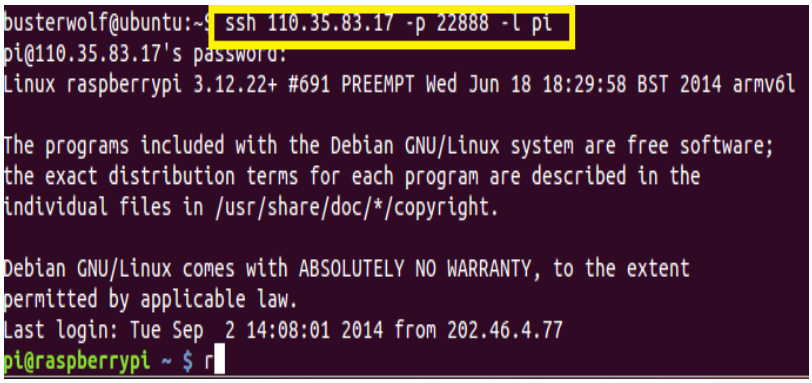
4.1 Pengujian *Honeypot Dionaea*

Honeypot Dionaea yang telah dipasang pada *Raspberry Pi* pada IP Publik ini berfungsi sebagai *sensor* yang menerima *malware* dan sebagai sumber basis data untuk *server* analisis. Untuk menguji *Honeypot Dionaea* terhubung dengan *Internet*, hal pertama yang dilakukan adalah melakukan tes *ping*. Dengan menggunakan perintah baris *ping* pada komputer dengan tujuan alamat IP *Honeypot Dionaea*, dapat dilihat pada Gambar 8.



Gambar 8. Tes *ping*

Bila *honeypot* berhasil di-*ping*, maka kemudian dilakukan tes untuk akses ssh. Perintah baris ssh dengan *username* dan *password* untuk masuk ke *shell* dalam *honeypot*, dapat dilihat pada Gambar 9.



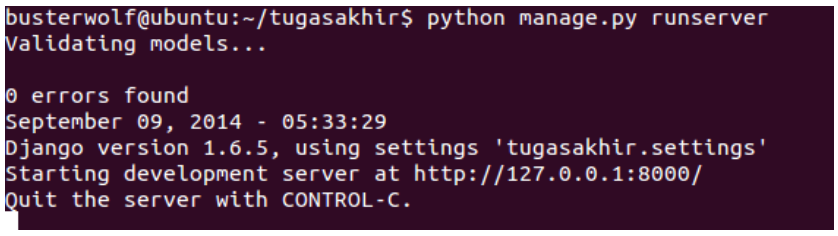
Gambar 9. Tes dengan SSH

Bila berhasil masuk SSH, maka akan diperiksa *folder binaries* untuk mengetahui apakah *honeypot* dapat mengambil *malware* dari *internet*, dapat dilihat pada Gambar 10. Bila terdapat *file* dengan nama yang dienkripsi md5, maka *Honeypot Dionaea* berhasil hidup dan mampu mengambil *malware* dari *internet*.

Gambar 10. Isi *folder binaries*

4.2 Pengujian *Server* Analisis

Server analisis yang sudah memiliki basis data dari *Honeypot Dionaea* dilakukan dengan menjalankan *server project Django* yang telah dibuat, dapat dilihat pada Gambar 11. Yang diuji dari *server* ini adalah saat proses menampilkan data pada *web browser* saat tautan tertentu dipilih. Halaman utama saat *server Django* diakses melalui *web browser* dapat dilihat pada Gambar 7.



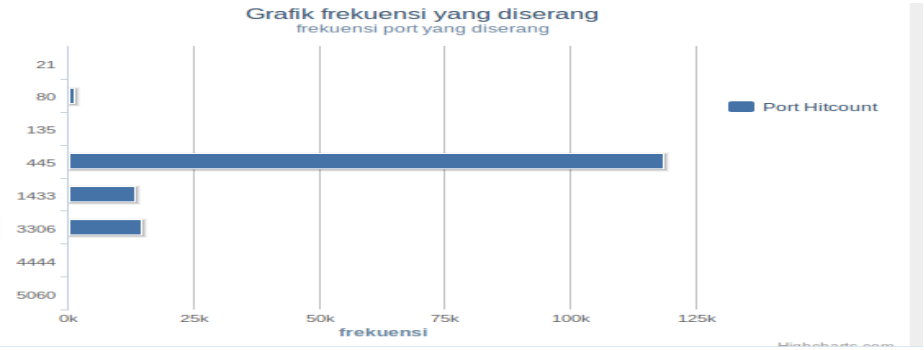
Gambar 11. *Run server django*

Tautan yang akan diujikan adalah sebagai berikut:
1. Serangan dalam sehari
2. Frekuensi *Port* yang diserang
3. *Top* 10 *Malware*
4. *Remote Host Attacker*
5. *Operating System Genre*

Hasil dari setiap tautan yang dipilih dapat dilihat pada Gambar 12 sampai dengan Gambar 16. Hasil tautan serangan dalam sehari pada Gambar 12 merupakan grafik dari serangan rata-rata yang diterima oleh *honeypot* dalam satuan jam.

Gambar 12. Serangan rata-rata dalam sehari

Hasil tautan frekuensi *port* yang diserang pada Gambar 13 merupakan grafik *bar* untuk menunjukkan jumlah *hitcount port* yang diserang untuk memperoleh *malware.*



Gambar 13. Frekuensi *port* yang diserang

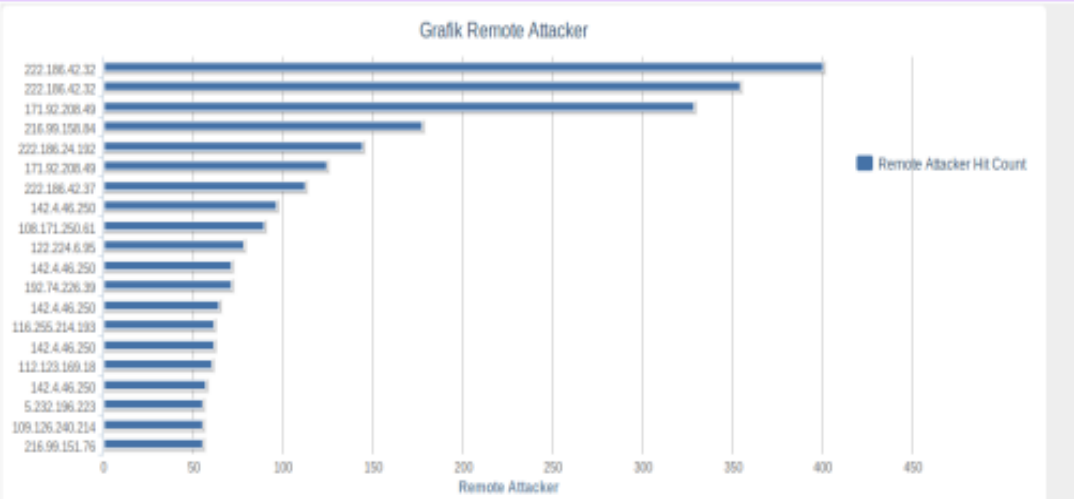
Data *top* 10 *malware* dapat dilihat pada Gambar 14. Hasil tautan *remote host attacker* pada Gambar 15 adalah grafik alamat IP dan jumlah *hitcount*-nya dalam memberikan *malware* ke *honeypot.* Data *operating system genre* dapat dilihat pada Gambar 16.

Download Md5 Hash	Download Md5 Hash Hitcount
cae4b7963f5e43033664299a4d5bd176	5305
26156811dacf6bf756cecfff692cd8b4	3784
78c9042bbcefd65beaa0d40386da9f89	3518
2e8da5a55865a091864a4338ef4d2e44	3509
060722ac0e512e73f6c16ebe87229bea	3262
0c059b0d1d5a03f69a21185987c17d5c	2687
72a6e8eb9b6f6d19cacd6d9c41dbdea6	2408
908f7f11efb709acac525c03839dc9e5	1447
87136c488903474630369e232704fa4d	893
4d37c4497728bcbc5f5f8ffdd171f482	526

- Page 1 of 53
- Next
- 10 of 525 items

Malware md5 download table
Virus Total Table
Virus Total Scans table

Gambar 14 *Top* 10 *malware* dalam *md5 hash*



Gambar 15 *Remote attacker*

OS Genre	OS Genre Hitcount
Windows	5162500
—	415941
Linux	21781
FreeBSD	227
Novell	101
HP-UX	94
SunOS	65
Solaris	19
ExtremeWare	5
Redline	2
Cacheflow	1
Cisco	1
NetBSD	1
OpenBSD	1

Gambar 16 *Operating system genre*

5. KESIMPULAN

Dari hasil pembahasan, dapat disimpulkan beberapa hal sebagai berikut:

1. *Honeypot Dionaea* pada *Raspberry Pi* mampu memperoleh *malware* yang ada di *internet.*
2. Basis data *Honeypot Dionaea* dapat digunakan *server* Analisis untuk ditampilkan dalam bentuk halaman *web.*
3. *Malware* yang diperoleh *Honeypot* dapat diteliti lebih lanjut.

REFERENSI

[1]. Raspbian. "*Raspbian FAQ".* http://www.raspbian.org/RaspbianFAQ. (diakses 5 Oktober 2013).

[2]. Raspberry Pi. "*Faqs Raspberry Pi".* http://www.raspberrypi.org/faqs. (diakses 5 Oktober 2013).

[3]. Kreibich, J.A. 2010. Using SQLite. Sebastopol: O'Reilly Media

[4]. Spitzner, L. 2002. Honeypots Tracking Hacker. Boston: Pearson Education, Inc.

[5]. Python. "*About Python*". http://*Python*.org/about (diakses 5 Oktober 2013).

[6]. Ariyus, D. 2007. Kamus Hacker. Yogyakarta: Andi Publisher.

[7]. Info Komputer. 2003. "Konektivitas Linux dan Windows". Info Komputer No 12, hlm. 128.

[8]. Veer, E.V. 2004. JavaScripts for Dummies. New York: IDG Books Worldwide,Inc.

[9]. VMware, Inc. "*Virtualization Overview".* http://www.vmware.com/pdf/virtualization.pdf. (diakses 27 Agustus 2014).