Coding : Jurnal Komputer dan Aplikasi
Volume 07, No. 03 (2019), hal 54-64 ISSN : 2338-493X

Implementasi Honeypot Kipo pada Sistem Keamanan *Server* Berbasis Web Monitoring dengan Notifikasi Otomatis menggunakan API Telegram

Fathuzzikri[1], Ikhwan Ruslianto[2], Uray Ristian[3]
[1,2,3]Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963
e-mail: [1]zikri2195@student.untan.ac.id, [2]Ikhwanruslianto@siskom.untan.ac.id,
[3]eristian@siskom.untan.ac.id

Abstrak

Serangan *port scanning* dan *bruteforce* merupakan ancaman yang umum terjadi pada *server*. Untuk menangani serangan tersebut, dilakukan sebuah penelitian dengan mengimplementasikan Honeypot Kippo yang dapat menyerupai sebuah sistem *server* yang asli. Selain itu lambatnya informasi mengenai terjadinya serangan terkadang berdampak besar terhadap keamanan *server*, maka pada penelitian ini dibangun sistem notifikasi otomatis menggunakan API Telegram yang dapat memberikan notifikasi apabila terjadi serangan. Integrasi antara Honeypot Kippo dan API Telegram menggunakan sebuah perantara berupa aplikasi *website* yang dibangun menggunakan *framework* Laravel. Fungsi dari aplikasi *website* selain sebagai perantara dalam mengintegrasikan Honeypot Kippo dengan API Telegram, adalah sebagai pengolah informasi yang telah dikumpulkan oleh Honeypot Kippo, untuk selanjutnya ditampilkan dalam bentuk infografis dan tabel agar memudahkan administrator *server* dalam mengetahui aktivitas serangan yang terjadi, aktivitas serangan akan dibagi menjadi beberapa tingkatan level sesuai rule yang telah dibuat. Dari pengujian yang telah dilakukan, didapatkan persentase keberhasilan penanganan serangan oleh sistem yang dibangun sebesar 99%. Waktu rata-rata yang dibutuhkan oleh aplikasi *website* dalam mengirim notifikasi otomatis adalah 2.8 detik. Kinerja *server* akan meningkat 7.6% untuk CPU dan 562607.2K untuk memori apabila serangan ditangani oleh sistem yang dibangun.

***Kata Kunci**: Honeypot Kippo, Bruteforce, Port Scanning,* Keamanan *Server,* Notifikasi Otomatis.

1. PENDAHULUAN

Server sebagai pusat tersimpannya berbagai informasi digital menyebabkan keamanan *server* harus selalu terjaga agar informasi-informasi tersebut tidak bocor ke pihak yang tidak bertanggung jawab. Serangan pada *server* umumya bersifat penyusupan yang bertujuan untuk mendapatkan akses penuh dalam mengelola isi *server*. Apabila penyusupan tersebut berhasil dilancarkan pada *server*, penyusup dapat mencuri, menghapus, mengedit isi *server* bahkan berpotensi mengambil alih *server*. Penyusupan dapat dilakukan diberbagai layanan yang tersedia di *server*, salah satunya adalah layanan SSH (*Secure Shell*) dengan cara *bruteforce*. *Bruteforce* adalah serangan dengan cara mencoba berbagai kemungkinan *username* dan *password* untuk *login* ke *server*, baik itu menggunakan *bot* maupun dengan *login* secara manual. Sebuah *server shared hosting* dapat mengalami percobaan serangan *brute force* sebanyak 1000-2000 atau lebih yang gagal, serangan ini umumya menargetkan layanan SSH dan FTP yang dilakukan oleh *host* asing tertentu, yang secara terus menerus melakukan percobaan login dengan menggunakan *username* dan *password* yang umum[1].

Salah satu teknologi yang dapat menjadi solusi permasalah ini adalah mengimplementasikan sebuah Honeypot ke dalam *server*, honeypot yang dibangun khusus untuk menangani serangan pada SSH adalah Kippo. Kippo dapat menangani dan mencegah serangan pada layanan SSH dengan menjebak

penyerang yang mencoba *login* ke *port* SSH yaitu *port* 22 lalu merekam semua aktivitas penyusupan didalam *server*. Sebuah penelitian terkait dengan judul Implementasi Sistem Keamanan *Server* menggunakan Honeypot dan Raspberry PI Terhadap Attacker[2], menggunakan Honeypot berjenis Kippo, dengan kesimpulan bahwa dengan memanfaatkan Raspberry PI untuk penggunaan Honeypot Kippo lebih menghemat biaya dan sumber daya yang digunakan.

Namun pada penelitian yang telah dilaksanakan tersebut, masih terjadi permasalahan mengenai tidak adanya notifikasi yang didapat oleh administrator *server* apabila terjadi serangan.

Berdasarkan permasalahan tersebut, dibuatlah sebuah penelitian yang didapat melengkapi penelitian sebelumnya, yaitu pengimplementasikan Honeypot Kippo pada sistem keamanan *server* berbasis *website* dengan notifikasi otomatis menggunakan API Telegram. Sebagai solusi dalam mencegah serangan langsung pada *server* dan alat pengumpulan data penyerangan, sistem yang dibangun dilengkapi dengan notifikasi melalui aplikasi *mobile* Telegram, yang akan didapat oleh administrator secara otomatis apabila terjadi serangan pada *server*.

2. LANDASAN TEORI

2.1 Honeypot

Honeypot adalah sebuah sumber daya keamanan yang memiliki nilai jika Honeypot tersebut diselidiki atau diserang[3]. Honeypot mampu menirukan *server* beserta sistem operasi dan layanannya, honeypot memiliki beberapa tingkatan berdasarkan tingkat interaksinya terhadap penyusup.

Honeypot memiliki beberapa tingkatan berdasarkan tingkat interaksinya terhadap penyusup, tingkatan tersebut adalah sebagai berikut:

1. *Low Interaction* Honeypot.
   Honeypot ini hanya sedikit berinteraksi dengan penyusup, dengan menirukan sistem operasi dan layanan secara spesifik. Kelebihan dari Honeypot jenis ini adalah mudah dikonfigurasikan dan informasi yang dikumpulkan lebih spesifik.
2. *Medium Interaction* Honeypot.
   Honeypot ini berinteraksi dengan penyusup pada tingkatan menengah. Honeypot ini dipasang pada server virtual namun tetap berada pada server utama, selain itu Honeypot jenis ini dapat mengemulsi layanan pada server utama secara menyeluruh, hal ini membuat Honeypot dengan tingkatan ini dapat mengumpulkan informasi lebih banyak dibanding *low interaction* Honeypot.
3. *High Interaction* Honeypot.
   Honeypot memiliki tingkat interaksi yang tertinggi, Honeypot ini memiliki sistem operasi asli dengan semua layanan yang sama persis dengan server asli. Dengan fitur honeypot jenis ini, penyusup akan benar-benar merasa bahwa mereka telah berhasil masuk ke server asli. Kekurangan dari Honeypot ini adalah rumit dalam mengkonfigurasinya, dan memiliki resiko tinggi jika penyusup berhasil lolos dari jebakan honeypot ini.

2.2 Honeypot Kippo

Kippo adalah salah satu jenis Honeypot dengan tingkat *medium interaction* yang didesain menggunakan bahasa python untuk menyimpan informasi *bruteforce* dan informasi aktivitas penyusup didalam *server*[4]. Honeypot jenis ini mudah dikonfigurasikan dan informasi yang dikumpulkan dapat lebih banyak. Beberapa informasi yang disimpan oleh Honeypot Kippo adalah sebagai berikut:

1. IP *Host* tempat disimpannya Honeypot.
2. IP Penyerang.
3. Waktu terjadinya serangan beserta rentang waktu penyerangan.
4. Perintah yang di*input*kan oleh penyerang saat terjebak didalam sistem Honeypot Kippo.
5. Waktu diinputkannya suatu perintah.
6. Aplikasi yang digunakan oleh penyerang saat melakukan penyerangan.
7. *Username* dan *Password* yang digunakan oleh penyerang.
8. Tingkat kesuksesan percobaan *login* dari pihak penyerang.
9. Menyimpan *file* yang dimasukkan oleh penyerang.

2.3 Laravel

Framework Laravel adalah salah satu

contoh *framework* yang dapat digunakan untuk membangun aplikasi *website*, yang membuat *framework* ini banyak digunakan adalah *syntax* dari *framework* ini yang ekspresif, rapi, dan mudah dipahami hingga mempercepat proses pembuatan *website*. Laravel adalah salah satu *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell. Sebagai sebuah *framework* PHP, Laravel hadir sebagai platform *web development* yang bersifat *open source*[5].

2.4 Mysql

Mysql merupakan salah satu *database server* yang berkembang dilingkungan open source dan didistribusikan secara gratis dibawah lisensi GPL[6]. MySQL adalah salah satu software sistem yang bersifat RDBMS (*Relational Database Management System*), RDBMS merupakan sebuah program yang memungkinkan pengguna untuk membuat dan mengatur *database* dengan model yang saling berhubungan. Mysql tergolong *software* sistem *database* yang dapat digunakan secara gratis dan tidak terbatas sistem operasi apapun. Pada penelitian Mysql akan digunakan sebagai tempat tersimpannya semua informasi yang diperoleh oleh Honeypot Kippo serta untuk memanajemen notifikasi otomatis.

2.5 Telegram

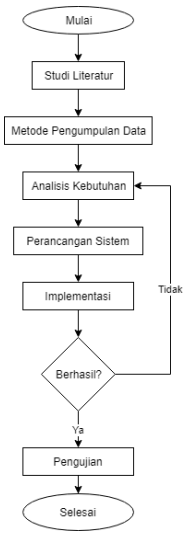
Telegram adalah *platform* olah pesan seluler yang menawarkan layanan pesan yang terenkripsi sehingga pengguna dapat mengirim pesan, foto, dan video ke kontak yang dipilih secara pribadi[7]. Telegram dapat digunakan secara gratis, dan menawarkan sebuah layanan yang memungkinkan pengembang *website* untuk mengintegrasikan *website* yang dibangun dengan aplikasi *mobile* Telegram. Untuk mengintegrasikan *website* dengan aplikasi *mobile* Telegram, digunakan sebuah fitur dari Telegram yaitu *bot* API dan *chat* Bot.

2.6 API Telegram

API ini memungkinkan anda untuk menghubungkan *bot* ke sistem anda. Telegram Bots adalah akun khusus yang tidak memerlukan nomor telepon tambahan untuk mengatur. Akun-akun ini berfungsi sebagai antarmuka untuk kode yang berjalan di suatu tempat di *server* anda[8]. Untuk menggunakan Telegram bot API, dibuat sebuah *bot* dan *channel* Telegram yang akan digunakan sebagai media yang akan mengirim notifikasi kepada pengguna.

3. METODE PENELITIAN

Diagram alir penelitian dapat dilihat pada Gambar 1.

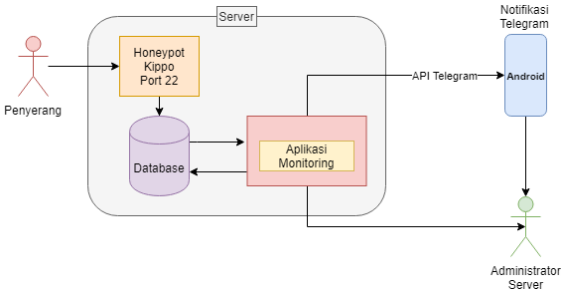


Gambar 1 Diagram Alir Penelitian

Penelitian dimulai dengan melakukan studi literatur dengan mengumpulkan referensi pendukung penelitian, dilanjutkan dengan menentukan metode pengumpulan data yang digunakan. Setelah langkah tersebut dilakukan, penelitian dilanjutkan dengan menganalisis kebutuhan penelitian, yang selanjutnya dilakukan perancangan sistem dan implementasi hasil perancangan. Apabila implementasi gagal dilaksanakan, penelitian akan kembali pada langkah analisis kebutuhan untuk memastikan bahwa seluruh kebutuhan penelitian dapat terpenuhi, namun apabila implementasi berhasil maka penelitian akan dilanjutkan dengan pengujian. Pengujian dilakukan dengan beberapa skenario.

4. PERANCANGAN

Untuk memperjelas alur sistem yang dibangun, maka dibuatlah sebuah racangan sistem. Rancangan sistem keseluruhan pada penelitian ini secara dapat dilihat pada Gambar 2.



Gambar 2 Rancangan Sistem

Service Secure Shell (SSH) yang secara *default* terpasang pada *port* 22 di dalam sistem *server*, akan dipindahkan ke *port* lain yang pada penelitian ini menggunakan *port* lain. *Port* 22 yang telah kosong akan dipasang sistem Honeypot Kippo, Honeypot Kippo akan menjebak penyerang yang melakukan penyerangan melalui *service* SSH pada *server*. Honeypot Kippo menyimpan seluruh data serangan yang ditangani kedalam sebuah *database*, adapun data yang dikumpulkan adalah berupa kombinasi *username* dan *password*, waktu percobaan dilakukan, keberhasilan *login*, IP penyerang, dan seluruh aktivitas yang dilakukan penyerang. Apabila penyerang berhasil *login* pada sistem Honeypot Kippo, Honeypot Kippo akan menyimpan semua informasi mengenai aktivitas yang dilakukan oleh penyerang saat berada didalam *server* seperti *command* yang dimasukkan, keberhasilan eksekusi, dan waktu pengeksekusian. Semua informasi yang telah dikumpulkan kedalam *database* akan ditampilkan pada aplikasi *monitoring* keamanan *server* dan sebagai informasi yang akan didapat oleh administrator *server* melalui notifikasi pada aplikasi Telegram.

Server yang digunakan pada penelitian ini akan dipasang beberapa *tools* dan komponen lain yang dibutuhkan, *server* yang digunakan akan menggunakan sistem operasi Centos 6.10 (*final*), *client* yang akan disimulasikan sebagai penyerang akan menggunakan sistem operasi Parrot, karena sistem operasi ini mendukung kebutuhan pengujian sistem, yaitu memiliki *tools* Nmap, SSH, dan Hydra yang dapat digunakan untuk pengujian.

4.1 Perancangan Perangkat Keras

Perancangan perangkat keras bertujuan untuk menghubungkan setiap perangkat keras yang dibutuhkan dalam penelitian. Setiap perangkat yang dibutuhkan akan dihubungkan dalam sebuah jaringan lokal yang dapat diakses secara *wireless*. Perangkat keras yang secara umum adalah sebuah *server* dan laptop akan dihubungkan dalam sebuah jaringan yang sama. Perangkat keras yang akan digunakan pada penelitian ini antara lain:

1. *Server*

   Penelitian ini menggunakan sebuah *server* yang akan menjadi target serangan. *Server* yang digunakan telah dikonfigurasi sesuai kebutuhan penelitian, konfigurasi yang dilakukan antara lain konfigurasi SSH, konfigurasi Honeypot Kippo, dan konfigurasi *database*. Selain itu *server* akan digunakan sebagai lokasi penyimpanan aplikasi *website monitoring* keamanan *server*. Sistem operasi yang digunakan pada *server* ini adalah Centos 6.10 (*final*).
2. Laptop

   Laptop yang digunakan adalah laptop dengan sistem operasi berbasis debian, laptop ini akan digunakan sebagai alat penyerangan pada skenario pengujian serangan. Selain itu laptop yang menjadi *client* juga akan digunakan sebagai untuk mengakses aplikasi *website monitoring* keamanan *server*.
3. *Smartphone*

   Smartphone yang dibutuhkan dalam penelitian ini adalah smartphone dengan sistem operasi Android. Pada *smartphone* telah terpasang sebuah aplikasi *mobile* Telegram, Telegram akan digunakan untuk menerima notifikasi saat terjadi serangan pada *server*.
4. Perangkat keras jaringan

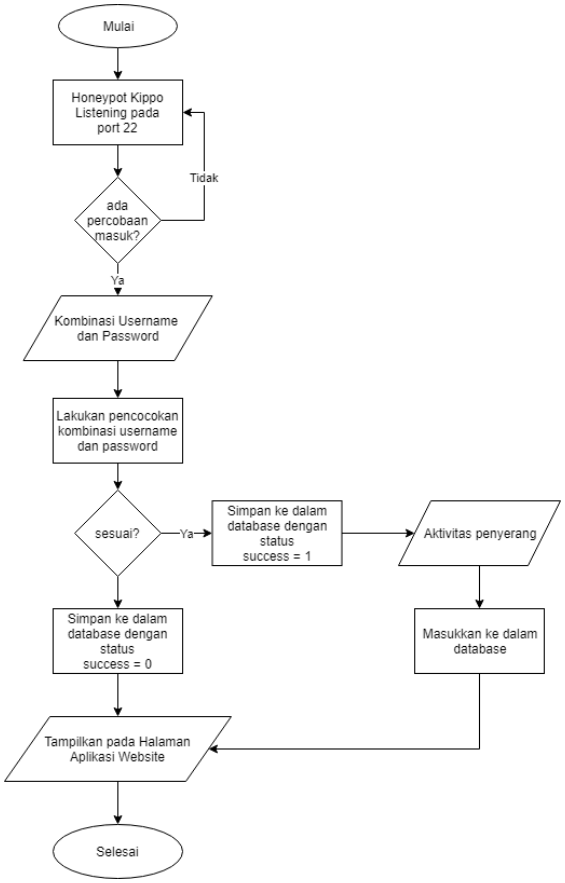
   Pada perancangan perangkat keras, akan digunakan jaringan lokal yang menghubungkan seluruh perangkat keras yang dibutuhkan. Adapun perangkat keras untuk kebutuhan jaringan antara lain *switch*, kabel UTP, konektor RJ-45 CAT5, dan *access point*.

4.2 Perancangan Konfigurasi SSH dan Honeypot Kippo

Perancangan sistem yang dibangun akan dimulai dari mengkonfigurasi SSH, tujuan dari konfigurasi ini adalah memindahkan *port* untuk layanan SSH. SSH secara default menggunakan *port* 22, akan dikonfigurasi agar menggunakan *port* lain. Pemindahan *port* dilakukan karena *port* 22 akan digunakan untuk sistem Honeypot Kippo, agar penyamaran sistem Honeypot Kippo sebagai *server* asli dengan layanan SSH

dapat berjalan baik. Setelah konfigurasi *port* SSH dan Honeypot Kippo berhasil, selanjutnya adalah mengkonfigurasi Honeypot Kippo sebagai alat penanganan serangan dan pengumpulan informasi penyerangan.

Sistem Honeypot Kippo merupakan sistem yang dirancang untuk menyerupai sebuah *server* asli, khususnya pada sisi layanan SSH. Tujuan dari sistem Honeypot Kippo ini adalah mengelabui penyerang agar tidak masuk ke dalam sistem *server* yang asli. Diagram alir sistem Honeypot Kippo dapat dilihat pada Gambar 3.



Gambar 3 Diagram Alir Honeypot Kippo

Sistem Honeypot Kippo bekerja dengan cara megelabui penyerang yang berusaha masuk ke dalam *server* melalui layanan SSH, Honeypot Kippo dapat dikonfigurasi agar serangan yang masuk pada sistemnya dapat login dengan mudah sebagaimana login pada *server* asli. Setelah penyerang berhasil dijebak ke dalam sistem Honeypot Kippo, setiap aktivitas yang dilakukan oleh penyerang akan di respon sebagaimana *server* asli merespon setiap perintah yang masuk, selain itu Honeypot Kippo akan menyimpan data penyerangan dan aktivitas tersebut ke dalam sebuah *database* yang telah disiapkan.

Pada saat penyerang melakukan percobaan penyusupan dengan cara *login* pada *port* 22 yang merupakan sistem Honeypot Kippo, kombinasi *username* dan *password* yang digunakan penyerang akan dicocokkan dengan konfigurasi kombinasi *username* dan *password* pada Honeypot Kippo. Kombinasi *username* dan *password* pada Honeypot Kippo akan diatur agar proses *login* dapat dilakukan dengan mudah. Pada umumnya penyerang akan melakukan penyerangan dengan metode *bruteforce* untuk mendapatkan kombinasi *username* dan *password* yang cocok, apabila serangan ini ditangani oleh *server* secara langsung, *server* dapat mengalami crash dikarenakan jumlah percobaan *login* yang umumnya sangat banyak, sehingga membuat *server* kesulitan dalam melayani setiap *request* yang masuk.

Setelah penyerang berhasil masuk dan terjebak ke dalam sistem Honeypot Kippo, penyerang dapat beraktivitas dengan meng*input*kan *command* sebagaimana aktivitas yang dilakukan pada *server* asli. Honeypot Kippo akan merespon setiap perintah yang masuk sebagaimana *server* asli. Beberapa contoh *command* yang umum dan dapat direspon oleh Honeypot Kippo adalah *adduser, ls, cd, cat, nano, apt-get, dan wget*. Setiap perintah yang di inputkan oleh penyerang akan disimpan ke dalam *database* Mysql yang telah disiapkan. Honeypot Kippo akan menangani dan mengumpulkan informasi serangan yang terjadi, mulai dari percobaan *login* hingga perintah yang di*input*kan penyerang.

Hasil pengumpulan informasi yang tersimpan dalam *database*, akan ditampilkan dalam bentuk grafik dan tabel pada aplikasi *website*. Aplikasi *website* monitoring keamanan *server* akan menampilkan informasi mengenai jumlah serangan, IP penyerang, direktori yang banyak dikunjungi, seluruh aktivitas penyerang, dan seluruh kombinasi *username* dan *password* yang digunakan oleh penyerang. Selain itu, aplikasi *website* yang dibangun menggunakan *framework* Laravel ini berfungsi sebagai pengolah data untuk menentukan level serangan dan pengiriman notifikasi otomatis apabila terjadi serangan. Level serangan dibagi menjadi tiga, ketentuan level dapat dilihat pada Tabel 1.

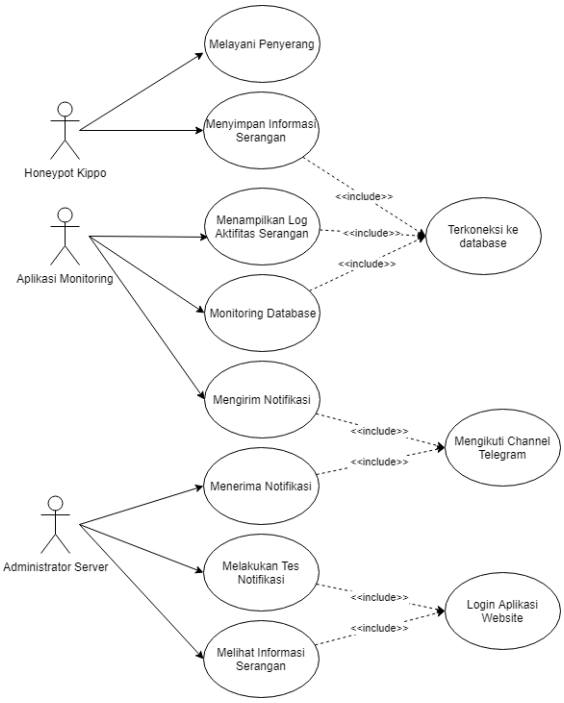
Tabel 1 Ketentuan Level Serangan

Level 1	Level 2	Level 3
Melakukan percobaan login Login gagal	Melakukan percobaan login Login berhasil Hanya melihat-lihat isi *server*.	Melakukan percobaan login Login berhasil Melakukan aktivitas penambahan atau pengeditan isi *server*.

Tujuan dari pengelompokkan serangan adalah untuk mengetahui tingkat bahaya aktivitas yang dilakukan penyerang terhadap *server*, tingkat bahaya yang dimaksud adalah bagaimana pengaruh aktivitas serangan terhadap integritas data dan peforma *server*. Setiap terjadi penyerangan terhadap *server* dan perubahan level pada penyerang, aplikasi *website* akan mengirim notifikasi kepada *administrator server*. Notifikasi diterima *administrator server* berupa notifikasi pada aplikasi *mobile* Telegram, dengan pesan berupa informasi mengenai penyerang dan level serangan yang terjadi.

4.3 Use Case Diagram Sistem

Dalam tahap merancang sistem yang dibuat digunakan *use case diagram* sebagai gambaran fungsi didalam sistem yang dapat dilakukan oleh aktor didalam sistem. Terdapat tiga aktor pada sistem ini, yaitu *administrator server*, Honeypot Kippo, dan Aplikasi Monitoring. Honeypot Kippo bertugas sebagai yang menangani dan menyimpan informasi serangan. Aplikasi *Website* bertugas sebagai yang menampilkan hasil penanganan serangan oleh Honeypot Kippo. *Administrator server* dapat menjadi pengguna sistem ini dengan menjadi anggota *channel* Telegram yang telah dikonfigurasi sebagai *channel* penerima notifikasi kemanan *server* dan memiliki akun didalam aplikasi *monitoring* keamanan *server*. Use case diagram sistem dapat dilihat pada Gambar 4.



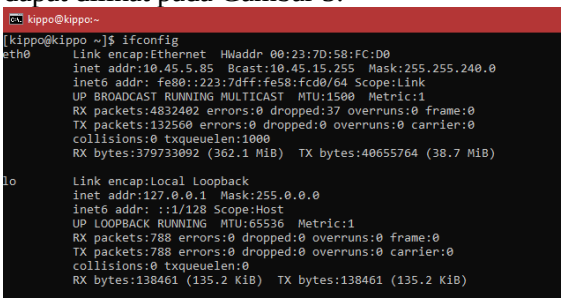
Gambar 4 Use Case Diagram Sistem

Administrator server dapat menggunakan aplikasi *monitoring* keamanan *server* jika telah memiliki akun pada aplikasi tersebut, untuk mendapatkan notifikasi otomatis saat terjadi serangan, *administrator server* harus bergabung ke dalam *channel* khusus pada aplikasi Telegram, pada *channel* tersebut sudah terdapat *chatbot* yang akan memberikan notifikasi.

5. HASIL DAN PEMBAHASAN

5.1 Implementasi Arsitektur Perangkat Keras

Implementasi perangkat keras yang dilakukan meliputi proses merakit komponen perangkat keras yang telah direncanakan. Hasil implementasi perancangan perangkat keras dapat dilihat pada Gambar 5.



Gambar 5 IP *Server*

IP yang digunakan oleh *server*, *server* menggunakan jaringan lokal Universitas Tanjungpura. Untuk mengakses *server*, digunakan laptop yang terhubung ke jaringan

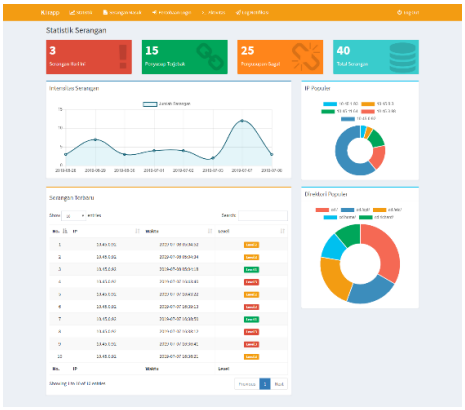
lokal Universitas Tanjungpura. *Server* dapat diakses secara remote pada IP 10.45.5.85, IP ini diterapkan secara statis agar IP *server* tidak berubah apabila terjadi gangguan pada jaringan. *Server* dapat diakses melalui layanan SSH, layanan SSH akan dikonfigurasi menyesuaikan kebutuhan sistem pada bab perancangan. Untuk mengetahui keberhasilan implementasi perangkat keras, dilakukan pengecekan koneksi antara *server* dengan laptop administrator, pengecekan dilakukan dengan cara ping.

5.2 Implementasi Aplikasi Monitoring

Hasil perancangan aplikasi monitoring keamanan *server* dapat diakses melalui *browser* dengan alamat 10.45.5.85 pada jaringan lokal Universitas Tanjungpura. Aplikasi monitoring keamanan *server* dibangun berbasis *website* menggunakan *framework* Laravel. Aplikasi yang telah diakses akan menampilkan halaman depan dengan penjelasan singkat mengenai aplikasi monitoring keamanan *server*. Antarmuka halaman depan dapat dilihat pada Gambar 6.

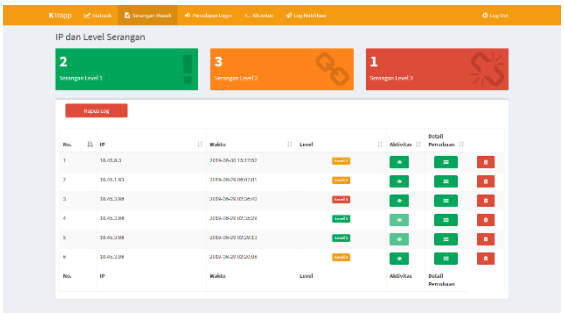
Gambar 6 Halaman Depan Aplikasi

Aplikasi *website* monitoring keamanan *server* memiliki beberapa halaman utama, diantaranya halaman statistik, serangan masuk, aktivitas serangan, percobaan login, dan notifikasi. Halaman statistik menampilkan beberapa informasi umum mengenai penyusupan yang telah ditangani oleh Honeypot Kippo. Tampilan halaman dashboard dapat dilihat pada gambar 7.



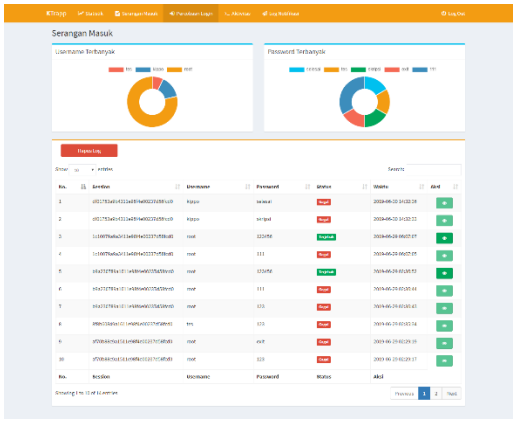
Gambar 7 Halaman Dashboard

Antarmuka halaman utama *dashboard* aplikasi menampilkan informasi umum mengenai penyerangan yang telah ditangani oleh Honeypot Kippo. Komponen halaman ini terdiri lima menu utama, empat *cardbox* berisi jumlah percobaan penyusupan, grafik intensitas penyusupan pada 10 hari terakhir, Tabel IP penyerang beserta waktu terjadinya serangan dan level serangan, serta Pie Chart yang menampilkan informasi mengenai direktori yang banyak diakses oleh penyusup dan IP terbanyak yang melakukan penyerangan.



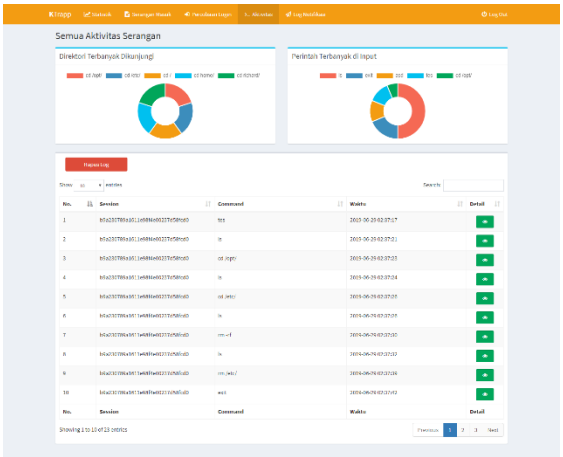
Gambar 8 Halaman Serangan Masuk

Halaman serangan masuk memiliki dua komponen, komponen pertama adalah 3 buah card box yang menampilkan jumlah serangan yang telah dikelompokkan berdasarkan level serangan. Penyerang yang berstatus level 1 hanya dapat dilihat seluruh percobaan login yang dilakukannya, karena pada level 1 penyerang belum berhasil masuk ke dalam sistem Honeypot Kippo. Penyerang yang berstatus level 2 dan level 3 dapat dilihat seluruh percobaan login yang telah dilakukannya, dan seluruh aktivitas yang dilakukan penyerang saat serangan ditangani oleh sistem Honeypot Kippo.



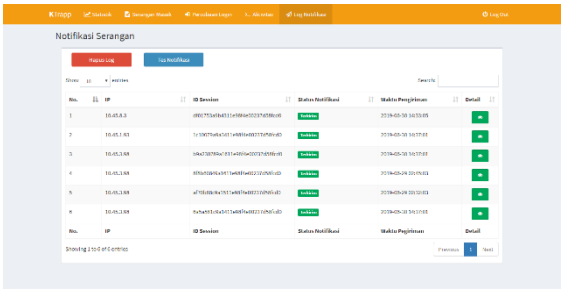
Gambar 8 Halaman Percobaan Login

Gambar 8 menampilkan halaman percobaan login, halaman ini memiliki tiga komponen, dua komponen berbentuk pie chart dan satu tabel. Pie chart pertama menampilkan *username* terbanyak yang digunakan oleh penyerang, dan pie chart kedua menampilkan *password* terbanyak yang digunakan oleh penyerang. Tabel percobaan login menampilkan informasi mengenai kombinasi *username* dan *password* penyerang, apabila status login berhasil, maka tombol aksi akan bersifat enable. Tombol aksi ini akan mengarahkan administrator ke halaman aktivitas serangan pada percobaan serangan yang dipilih.



Gambar 9 Halaman Aktivitas Serangan

Gambar 9 diperlihatkan antarmuka halaman aktivitas serangan yang telah ditangani oleh Honeypot Kippo. Pada komponen pie chart, dapat diketahui command apa yang sering digunakan oleh penyerang dan direktori terbanyak yang dikunjungi. Selain itu terdapat sebuah tabel dengan lima kolom yang menampilkan informasi aktivitas serangan. Pada kolom *Session* ditampilkan *session* yang digunakan oleh penyusup saat berhasil dijebak kedalam Honeypot Kippo. Setiap satu kali percobaan penyusupan, penyusup akan memiliki sebuah *session* yang dibuat oleh Honeypot Kippo untuk membedakan setiap peyerang yang melakukan serangan. Pada kolom Detail terdapat tombol yang dapat yang akan mengarahkan administrator ke halaman aktivitas serangan berdasarkan *session* penyerang. Tombol pada kolom detail akan menampilkan halaman yang berisi sebuah tabel, tabel pada halaman ini menampilkan informasi command, status, dan waktu diinputkannya command.

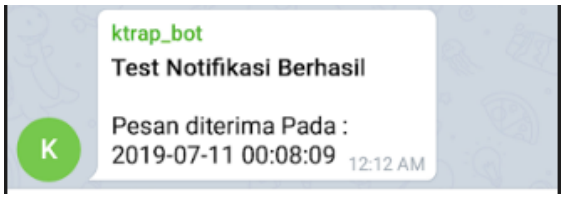


Gambar 10 Halaman Notifikasi

Gambar 10 diperlihatkan antar muka halaman notifikasi, yang merupakan log setiap notifikasi yang dikirim. Setiap notifikasi yang dikirim dikelompokkan berdasarkan *session* penyerang, sehingga setiap satu penyerang hanya akan dikirim maksimal tiga notifikasi, yaitu notifikasi serangan level 1, notifikasi serangan level 2, dan notifikasi serangan level 3. Pada kolom detail terdapat sebuah tombol yang akan menampilkan informasi kombinasi *username* dan *password* yang digunakan oleh penyerang pada *session* yang dipilih.

5.3 Implementasi API Telegram

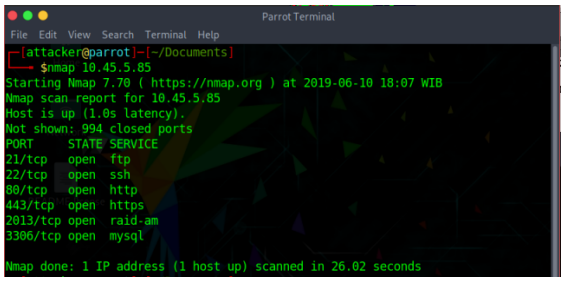
Sistem yang dibangun untuk *rule* level serangan dan notifikasi bersifat otomatis, sistem akan menjalankan sebuah command pada sisi *server* untuk memicu pemantauan *database*. Sistem otomatis yang dibangun memanfaatkan fitur task scheduler pada Laravel dan *cronjob* pada sisi *server*. Pada sisi aplikasi dilakukan pemanggilan fungsi oleh *task scheduler* Laravel, yang selanjutnya fitur *task scheduler* Laravel akan dipanggil oleh *cronjob* pada sisi *server*. Hasill implementasi API Telegram dapat dilihat pada Gambar 11.



Gambar 11 Notifikasi Telegram

5.4 Pengujian Sistem Honeypot Kippo

Pengujian ini dilakukan menggunakan tools bernama Nmap, hasil dari scanning Nmap akan dibandingkan dengan konfigurasi yang telah dilakukan.



Gambar 12 Hasil Scanning Nmap

Dari Gambar 12 dapat dilihat bahwa hasil dari scanning menggunakan Nmap menghasilkan output yang sama dengan konfigurasi yang telah dilakukan. Pada gambar diatas menunjukkan bahwa konfigurasi berhasil dilakukan, dan sistem Honeypot Kippo yang menyerupai SSH telah berjalan pada *port* 22.

Pengujian *rule* serangan bertujuan untuk mengetahui keberhasilan implementasi kode program *rule* level serangan yang telah dibuat. Selain itu pengujian rule serangan dilakukan untuk mengetahui kecepatan proses yang dilakukan oleh sistem terhadap serangan yang terjadi. Pengujian Level serangan yang diimplementasikan antara lain yaitu level 1, level 2, dan level 3.

Tabel 1 Pengujian Rule Level 2

No	IP	Level Serangan	Waktu (/dtk)
1	10.45.0.1	Level 2	1
2	10.45.0.1	Level 2	1
3	10.45.0.1	Level 2	3
4	10.45.3.218	Level 2	1
5	10.45.3.218	Level 2	2
6.	10.45.9.210	Level 2	1
7.	10.45.9.210	Level 2	2
8.	10.45.9.210	Level 2	1
9.	10.45.9.210	Level 2	2
10.	10.45.4.215	Level 2	1
11.	10.45.4.215	Level 2	2
12.	10.45.4.215	Level 2	2
13.	10.45.4.215	Level 2	1
14.	10.45.4.215	Level 2	2
15.	10.45.4.215	Level 2	2
16.	10.45.4.215	Level 2	1
17.	10.45.5.211	Level 2	1
18.	10.45.5.211	Level 2	2
19.	10.45.5.211	Level 2	1
20.	10.45.5.211	Level 2	1

Pada Tabel 1 dapat dilihat bahwa sistem melakukan aksi perubahan ke level 2 sebagai respon aktivitas penyerang adalah rata rata sebesar 1.5 detik. Administrator *server* akan mendapat notifikasi setiap terjadi perubahan level pada penyerang.

Tabel 2 Pengujian Rule Level 3

No	IP	Level Serangan	Waktu (/dtk)
1	10.45.0.1	Level 3	3
2	10.45.0.1	Level 3	3
3	10.45.0.1	Level 3	1
4	10.45.3.218	Level 3	1
5	10.45.3.218	Level 3	1
6.	10.45.9.210	Level 3	2
7.	10.45.9.210	Level 3	1
8.	10.45.9.210	Level 3	2
9.	10.45.9.210	Level 3	1
10.	10.45.4.215	Level 3	1
11.	10.45.4.215	Level 3	1
12.	10.45.4.215	Level 3	1
13.	10.45.4.215	Level 3	2
14.	10.45.4.215	Level 3	1
15.	10.45.4.215	Level 3	2
16.	10.45.4.215	Level 3	1
17.	10.45.5.211	Level 3	2
18.	10.45.5.211	Level 3	1
19.	10.45.5.211	Level 3	1
20.	10.45.5.211	Level 3	2

Pada Tabel 2 dapat dilihat bahwa sistem melakukan aksi perubahan level sebagai respon aktivitas penyerang adalah rata rata sebesar 1.5 detik. Sistem akan meng-*update database* pada kolom notif dengan menyamakan nilainya dengan nilai pada kolom level, agar sistem dapat mengetahui jumlah notifikasi yang terkirim berdasarkan tingkat level. Sistem akan mengirim notifikasi secara otomatis setiap terjadi perubahan level pada penyerang.

Tabel 3 Pengujian Notifikasi Otomatis

No	Pengujian Notifikasi	Selisih Waktu (/dtk)

1	Pengujian ke 1	2
2	Pengujian ke 2	3
3	Pengujian ke 3	3
4	Pengujian ke 4	2
5	Pengujian ke 5	2
6	Pengujian ke 6	4
7	Pengujian ke 7	2
8	Pengujian ke 8	2
9	Pengujian ke 9	3
10	Pengujian ke 10	3
11	Pengujian ke 11	2
12	Pengujian ke 12	3
13	Pengujian ke 13	3
14	Pengujian ke 14	3
15	Pengujian ke 15	3
16	Pengujian ke 16	4
17	Pengujian ke 17	4
18	Pengujian ke 18	3
19	Pengujian ke 19	3
20	Pengujian ke 20	2

Pada Tabel 3 dapat diambil nilai rata rata bagi sistem untuk mengirim notifikasi kepada administrator adalah 2.8 detik setiap notifikasi.

Tabel 4 Pengujian Serangan *Bruteforce*

No	Skenario	Jumlah Data	Peforma CPU (%)		Peforma Memori (k)	
			SSH	Honeypot	SSH	Honeypot
1		Idle	0,1		307468	
2		100	17,1	1,6	374428	254585
3		200	19,4	1,7	419264	255700
4		300	17,4	1,7	384732	261396
5		400	17,8	1,8	387632	262760
6		500	21,1	1,7	388444	264496
7	Brute Force	600	16,6	1,7	390468	266480
8		700	17,4	1,7	392104	270200
9		800	17,5	1,7	393568	272804
10		900	17,5	1,9	394808	274292
11		1000	17,8	1,7	397024	305168
12		1100	21,0	1,6	409516	310996
13		1200	17,0	1,8	413456	312732
14		1300	18,3	2,3	487876	315088
15		1400	20,4	1,7	442216	329100
16		1500	19,3	1,8	446124	379816
17		1600	17,0	1,7	450716	387132
18		1700	16,7	3,3	451484	396308
19		1800	17,1	1,7	456800	402632
20		1900	17,0	1,7	461032	405732
21		2000	20,0	1,7	484072	411436

Dari Tabel 4 dapat dilihat bahwa kinerja *server* tetap stabil yaitu dengan nilai rata-rata CPU = 1.8% dan memori = 316942,65K. Sedangkan apabila serangan ditujukan langsung pada SSH maka kinerja *server* akan mengalami kenaikan rata-rata pada CPU sebesar 18.2% dan pada memori sebesar 421288,2K. Dengan menggunakan Honeypot Kippo sebagai alat penanganan serangan, kinerja *server* tidak mengalami perubahan signifikan saat terjadi serangan.

Tabel 5 Skenario Penyusupan

No	Nama Skenario	Jumlah Penyerang	Peforma CPU (%)	Memori digunakan (k)
1		1	7,1	878264
2		2	7,0	876528
3		3	7,8	876652
4		4	5,6	876652
5		5	7,1	875288
6		6	8,0	874544
7		7	7,0	873676
8		8	6,1	873800
9		9	8,4	872436
10		10	7,1	871940
11	Penyusupan	11	6,6	870452
12		12	8,2	869328
13		13	8,6	869460
14		14	6,9	868096
15		15	8,6	867104
16		16	9,1	866608
17		17	9,7	843332
18		18	9,1	866732
19		19	7,1	865368
20		20	8,9	865244

Dari Tabel 5 dapat dilihat bahwa penyusupan menyebabkan kenaikan kinerja *server* dari kondisi normal. Saat menangani serangan, Honeypot Kippo menyebabkan kenaikan kinerja *server* rata-rata dengan CPU=7.7% dan memori=870075.2K. Kenaikan kinerja *server* dari kondisi awal adalah CPU = 7.6% dan memori = 530975.2K.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian pada penggunaan Honeypot Kippo dan API Telegram dalam pengamanan *server*, didapatkan kesimpulan sebagai berikut:

1. Pada pengujian hasil konfigurasi Honeypot Kippo, Honeypot Kippo berhasil berjalan pada *port* 22. Pada saat dilakukan scanning *port*, Honeypot Kippo terdeteksi sebagai layanan SSH.
2. Honeypot Kippo dapat menangani lebih dari satu serangan secara bersamaan, bahkan lebih dari satu penyusup dalam waktu yang sama.
3. Penanganan serangan *bruteforce* oleh Honeypot menggunakan *resource server* lebih sedikit dibanding serangan langsung kepada layanan SSH. Pengaruh penyusupan terhadap kinerja *server* rata-rata sebesar 7.7% pada CPU dan menggunakan memori rata-rata sebanyak 870075.2K.

4. Honeypot Kippo menggunakan *resource* CPU sebesar 1.8% dan memori sebesar 316942,65K untuk menangani penyusupan.
5. Penelitian ini berhasil mengintegrasikan sistem Honeypot Kippo, Aplikasi *website*, dan API Telegram.
6. Dalam menangani serangan, sistem yang dibangun akan mengelompokkan tingkat serangan dengan waktu rata-rata 1.5 detik, dan mengirim notifikasi dengan waktu penerimaan rata-rata 2.8 detik.
7. Hal yang menyulitkan pada penelitian ini adalah pada saat pengujian *bruteforce*. *Tools* Hydra yang digunakan untuk melakukan *bruteforce* akan mengirim *request login* sebagai penyerang yang berbeda-beda, sehingga Honeypot Kippo membuat banyak *session*. Hal ini menyebabkan notifikasi yang terkirim menjadi sangat banyak bahkan terkadang terjadi *error* pada *database*, dan *load time* pada aplikasi *website* menjadi lama.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan maka diperoleh saran bagi penelitian lebih lanjut sebagai berikut:

1. Penggunaan *tool* IDS seperti Snort agar tidak perlu mengkonfigurasi ulang SSH, dan langsung mengarahkan penyerang ke sistem Honeypot.
2. Penelitian selanjutnya diharapkan menambah fitur tambahan yang dapat menganalisa aktivitas penyerang secara umum, dan memberikan notifikasi mengenai direktori yang rentan dikunjungi oleh penyerang.
3. Penelitian selanjutnya diharapkan melakukan pengembangan dengan menggunakan *server* yang dapat diakses publik, atau mengamankan *server* yang digunakan oleh suatu instansi. Agar objek penelitian selanjutnya berfokus pada analisa aktivitas penyerang.
4. Mencoba menggunakan tipe Honeypot yang berbeda, sehingga dapat dibandingkan dengan penelitian ini.

DAFTAR PUSTAKA

[1] Resita. (2017, April 11). *Memblokir Usaha Login Brute Force*. Diambil kembali dari Basis Pengetahuan Layanan Hosting MWN: https://kb.masterweb.com/artikelmemblokir-usaha-login-brute-force.html

[2] Utomo, A. N. (2018). Implementasi Sistem Keamanan *Server* menggunakan Honeypot dan Raspberry PI Terhadap Attacker. *Jurnal Rekayasa Informasi*.

[3] Spitzner, L. (2002, 9 13). *Honeypots : Definitions and Value of Honeypots*. Diambil kembali dari Honeypots : Definitions and Value of Honeypots: www.tacking-hackers.com

[4] Tamminen, U. (2016, September 30). *SSH Honeypot*. Dipetik 2019, dari Github: https://github.com/desaster/kippo

[5] Sandi, A. (2017, 2017 2017). *Alasan Mengapa Kamu Harus Menggunakan Framework Laravel*. Dipetik 2018, dari Codepolitan: https://www.codepolitan.com/alasan-mengapa-kamu-harus-menggunakan-framework-laravel-5a08d435ddcfb

[6] Prasetyo, D. D. (2005). Mengelola *Database* Dengan Visual Basic.NET dan Mysql. *PT.Elex Media Komputindo*.

[7] *Telegram*. (2018). Dipetik 2019, dari ProgrammableWeb: https://www.programmableweb.com/api/telegram

[8] *Telegram APIs*. (2018). Dipetik 2019, dari Telegram: https://core.telegram.org