IMPLEMENTASI HONEYPOT DAN PORT KNOCKING DALAM MENDETEKSI SERANGAN *DISTRIBUTED DENIAL OF SERVIS ATTACK* (DDOS ATTACK*)* PADA SERVER JARINGAN

TESIS

Diajukan untuk memenuhi salah satu syarat dalam menyelesaikan program Pendidikan Magister (S-2) Pada Program Studi Sistem Komputer STMIK Handayani Makassar

Oleh :

SULIMAN
2018130019

PROGRAM STUDI SISTEM KOMPUTER

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

STMIK HANDAYANI MAKASSAR

MAKASSAR

2020

HALAMAN PENGESAHAN

Judul :

IMPLEMENTASI HONEYPOT DAN PORT KNOCKING DALAM MENDETEKSI SERANGAN DISTRIBUTED DENIAL OF SERVIS ATTACK (DDOS ATTACK) PADA SERVER JARINGAN

Disusun dan diajukan oleh :

SULIMAN
2018130019

Disetujui dan dinyatakan telah memenuhi syarat untuk diseminarkan :

Disahkan di : Makassar

Tanggal : 26 Agustus 2020

Menyetujui Komisi Penasehat

Pembimbing I	Pembimbing II
Prof. Dr. Ir. Andani Ahmad, M.T	**Adnan, ST., M.T., PhD**

Mengetahui
Ketua Program Studi Pasca Sarjana Sistem Komputer

Prof. Dr. Ir. Andani Ahmad, M.T

HALAMAN PERSETUJUAN

ABSTRAK

Sistem keamanan jaringan semakain hari kian makin berkembang seiring dengan kemajuan tegnologi komputer dan jaringan, begitu pula serangan pada sistem jaringan yang berbeda-beda metode dan perkembanganya, khususnya pada server yang menjadi pengendali utama dalam rangkaian sistem jaringan menjadi target utama dari orang yang hendak merusak atau mengambil keuntungan tertentu dari sistem jaringan yang dibangun, terlebih dalam bidang pendidikan perguruan tinggi yang hampir semua model sistemnya menggunakan jaringan. Sehingga dalam penelitian ini akan melakukan pengujian dengan serangan tertentu pada server jaringan dan akan menganalisis bagaimana sistem keamanan jaringan mendeteksi dan menggagalkan serangan.

Pada penelitian ini serangan yang digunakan adalah *distributed denial off service* (*ddos attack*) dengan berbagai jenis model serangan seperti *ddos attack request flooding*, *ddos attack traffick flooding*, *ddos brute force attack* dan *ddos attack SQL injection.* Sedangkan untuk mendeteksi dan menggagalkan serangan tersebut yaitu menggunakan metode *honeypot* dan *port knocking* yang akan berkolaborasi dalam mengamankan sistem server jaringan pada *sistem operasi windows*.

Didapatkan hasil bahwa *honeypot* dan *port knocking* mampu mengamankan server jaringan dari serangan *ddos attack*, dimanan *honeypot* 100% mampu mendeteksi dan menjebak serangan yang masuk sedangkan *port knocking* 100% memblokir dan menggagalkan serangan *ddos attack* yang dilakukan oleh penyerang.

Kata kunci : Server jaringan, *ddos attack*, *honeypot* dan *port knocking*.

ABSTRACT

Network security systems are increasingly developing along with advances in computer and network technology, as well as attacks on network systems of different methods and developments, especially on servers that are the main controllers in a series of network systems that are the main target of people who want to damage or take certain advantage of the network system built, especially in the field of higher education, where almost all system models use a network. So that in this study we will test with certain attacks on network servers and will analyze how the network security system detects and thwarts attacks.

In this study, the attack used is a distributed denial off service (ddos attack) with various types of attack models such as ddos attack request flooding, ddos attack traffick flooding, ddos brute force attack and ddos attack SQL injection. Meanwhile, to detect and thwart these attacks, namely using the honeypot and port knocking methods which will collaborate in securing network server systems on the Windows operating system.

The results show that honeypot and port knocking are able to secure network servers from ddos attacks, where honeypot is 100% capable of detecting and trapping incoming attacks while port knocking 100% blocks and thwarts ddos attacks carried out by attackers.

Keywords: network server, ddos attack, honeypot and port knocking.

KATA PENGANTAR

Dengan mengucapkan puji dan syukur atas kehadirat Allah *Subhana Wa Ta'ala* atas segala limpahan rahmat dan hidayahnya, shalawat serta salam penulis sanjungkan kepada Nabi Besar Muhammad *Shalallaahu Alayhi Wasallam*, yang telah membawa umat manusia dari alam jahiliyah ke alam yang berilmu pengetahuan, sehingga penulis dapat menyelesaikan penelitian tesis dengan judul "**Implementasi *Honeypot* dan *Port Knocking* Dalam Mendeteksi Serangan *Distributed Denial Of Servis Attack* (*Ddos Attack)* Pada Server Jaringan**" semoga karya ini bermanfaat bagi penulis, instansi, masyarakat dan ilmu pengetahuan itu sendiri khususnya dalam bidang tegnologi komputer dan jaringan, juga untuk memenuhi salah satu syarat program pendidikan Pascasarjana pada Program Studi Sistem Komputer STMIK Handayani Makassar.

Sebagai ungkapan terimakasih dan rasa hormat atas segala dukungan, saran, bimbingan serta jasa-jasa dalam membantu penulis menyelesaikan penelitian ini, penulis menyampaikan ucapan terimakasih yang sedalam-dalamnya kepada;

1. Bapak **Dr. Eng. Yuyun, MT** selaku Direktur Pascasarjana STMIK Handayani Makassar.
2. Bapak **Prof. Dr. Ir. Andani Ahmad, MT** selaku Ketua Program Studi Sistem Komputer Pascasarjana STMIK Handayani Makassar sekaligus Dosen Pembimbing satu, yang telah banyak membantu penulis dalam

memberikan kemudahan baik pengarahan maupun bimbingan selama penulisan penyusunan tesis ini.

3. Bapak **Adnan, ST, MT, Ph.D** selaku Dosen Pembimbing dua, yang telah banyak membantu penulis dalam memberikan ide, saran dan kritikanya.
4. Dosen-dosen Program Studi Magister Sistem Komputer yang telah memberikan bekal ilmu pengetahuian kepada penulis, semoga ilmunya menjadi amal *jariyah* didunia maupun diakhirat.
5. Rekan-rekan mahasiswa Program Studi Magister Sistem Komputer, yang telah banyak memberikan bantuan,semangat serta kebersamaan mulai awal perkuliahan hingga akhir selesainya perkuliahan di Pascasarjana STMIK Handayani Makassar.
6. Kampus STMIK Bina Bangsa Kendari sebagai tempat penelitian dalam menyelesaikan tugas akhir tesis ini, di ruangan Laboratorium yang digunakan sebagai sarana dalam pengujian penelitian semoga nantinya karya ini juga bermanfaat bagi institusi tersebut.
7. Teristimewa kepada kedua orang tua dan istri yang telah memberikan dukungan, semangat serta doa sehingga diberikanya kemudahan dalam menempuh program pendidikan Magister di STMIK Handayani Makassar.

Akhirnya penulis menyadari bahwa hasil karya ini masih banyak kekurangan dan kelemahan, untuk itu saran dan kritik yang konstruktif akan sangat membantu agar tesis ini dapat menjadi lebih baik.

Semoga kebaikan yang telah diberikan kepada penulis mendapatkan pahala yang melimpah dari Allah *Subhana Wa Ta'ala* T. Amin "Wabillahi Taufik Walhidayah Wassalamualaikum Warahmatullahi Wabarakatuh".

Makassar, Oktober 2020

Suliman

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknolologi jaringan terutama system keamanan jaringan yang semakin berkembang menuntut agar system keamanan jaringan untuk berkembang, terutama pada keamanan *server* yang merupakan salah satu tugas pokok dari system administrator jaringan. Hal ini didasarkan pada karakteristik umum dari sistem jaringan komputer yang pada dasarnya adalah tidak aman untuk diakses secara bebas oleh pengguna. Terbukanya *port* pada sistem jaringan untuk layanan yang sifatnya public maupun bersifat pribadi, memiliki kemungkinan resiko yang tinggi untuk diserang dan diganggu oleh hacker atau penyerang. Untuk mengatasi hal tersebut maka administrator atau pemilik jaringan membutuhkan sebuah keamanan yang baik dan handal untuk dapat menjaga jaringan *server* dari penyerang yang akan mengganggu server jaringan.

Sebelum adanya metode *honeypot* dan *port knocking* untuk keamanan *server* pada sistem operasi biasanya komputer server hanya menggunakan *firewall* untuk mengamankan lalulintas jaringan, namun firewallpun masih banyak memiliki kelemahan dan kekurangan. Untuk mengatasinya maka dibutuhkan pengembangan dari *firewall* yaitu dengan mengimplementasikan *honeypot* dan *port knocking* pada jaringan *server*. Dimana *port knocking* dapat mengontrol layanan *port* terbuka dan *port* tertutup[1]. Selain menggunakan metode *port knocking* dibutuhkan *honeypot* untuk mengalihkan *attacker* kedalam *server* tiruan dan mendeteksi serangan apa saja yang dilakukan oleh *attacker/intruder* pada

server sehingga penyerang terjebak dan tidak mengganggu server pada jaringan[2].

Serangan yang paling sering digunakan adalah *Port Scanning* dan *DDOS* (*Distributed Denial Of Service*) namun saat ini kita akan mengamati lebih jauh tentang serangan *DDOS Attack* dan mengatasinya dengan menggunakan *Honeypot* sebagai sistem jaringan tiruan untuk memonitor *attacker* untuk selanjutkan serangan *DDOS Attack* tersebut akan dihentikan dan diblokir dengan menggunakan metode *Prort Knocking* dengan membuka port dan terdeteksi user asing maka user tersebut akan di blok atau dihentikan.

Sampai saat ini, serangan *DDOS Attack* masih belum memiliki metode pencegahan yang dapat diterapkan pada semua jenis *DDOS Attack*. Hal ini kemungkinan disebabkan karena manajemen dan serangan *DDOS Attack* memiliki mekanisme yang berbeda-beda, para pengganggu sistem jaringan juga terus mengembangkan metode dalam melakukan serangan *DDOS Attack*, bahkan menggunakan metode baru untuk melakukan penyerangan. Saat ini, terdapat beberapa pendekatan untuk memerangi serangan *DDOS Attack*. Seperti metode honeypot dimana sistem kerjanya yaitu menjadikan salah satu komputer mudah atau rentan untuk dimasuki sehingga menjadi sasaran utama yang empuk untuk para attacker atau pengganggu jaringan masuk sehingga komputer lainya yang digunakan pada sistem jaringan nirkabel tetap aman menjalankan fungsinya dikolaborasikan juga dengan sistem keamanan jaringan dengan metode *Port Knocking* untuk melakukan ketukan kedalam port sehingga diizinkan masuk untuk mengakses server pada jaringan yang digunakan.

Pengujian penelitian ini akan dilakukan pada Laboratorium Kampus Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bina Bangsa Kendari yang dimana setiap proses ujian kompetensi mahasiswa dilakukan secara *online* maupun *offline* untuk lebih mempermudah dalam proses ujian dan akurat dalam memperoleh hasil nilai, namun pada pengelolaan jaringan hanya menggunakan satu server untuk melayani *user client* yang nantinya akan digunakan oleh mahasiswa dalam melakukan proses ujian, yang tentunya membutuhkan sistem keamanan jaringan yang baik dan handal untuk mengantisipasi jika terjadi adanya gangguan pada server jaringan dari serangan *user* yang tidak bertanggung jawab dengan tujuan mengganggu server jaringan dengan metode serangan *Ddos Attack* untuk melumpuhkan *client-server* agar tidak bisa terhubung pada jaringan atau jaringan menjadi *down, eksploitasi password* dan *username* pada *microtik router* dan menginjeksi database pada aplikasi web untuk bisa mendapatkan *password* dan *username* sehingga dapat *login* sebagai *admin*. Model sistem jaringan yang akan dibangun pada pengujian penelitian ini yang nantinya akan dijadikan sebagai target serangan dan pengujian dalam keamanan server jaringan adalah simulasi ujian *offline* jaminan internal mutu (JAMIN) dimana ujian ini biasanya digunakan pada mahasiswa semester VI (enam) dengan tujuan untuk menguji mahasiswa tentang kemampuanya dalam bidang *ekstrakurikuler* dasar komputer, aplikasi web *Computer Based Test* (*CBT*) digunakan pada calon mahasiswa baru untuk melakukan tes seleksi bersama masuk perguruan tinggi swasta (SBMPTS) dan *router microtik* yang nantinya akan digunakan untuk mengatur konvigurasi server pada jaringan. Sedangkan

untuk keamanan server jaringan yaitu menggunakan metode *Honeypot* dan *Port Knocking* dengan tujuan mengamankan server jaringan untuk menjebak dan memblokir serangan dari user yang tidak bertanggung jawab dari serangan *ddos attack.*

Pada umumnya H*oneypot* merupakan sebuah komputer atau situs jaringan yang terlihat seperti bagian dari jaringan yang sebenarnya terisolasi dan dimonitor. Jika dilihat dari kacamata *hacker* yang akan menyerang, *Honeypot* terlihat seperti layaknya sistem yang patut untuk diserang. Hingga saat ini serangan *Distributed Denial of Service* (Ddos) yang dilakukan oleh *attacker* pada server adalah untuk menyibukan lalulintas jaringan sehingga CPU server bekerja ekstra dalam mengola jaringan dan model penyeranganya pun bisa menggunakan *smart phone android* ataupun komputer selama bisa mengakses sistem jaringan tersebut.

Sedangkan *Port Knocking* merupakan metode atau cara untuk membuka akses ke *port* tertentu yang telah diblock oleh *Firewall* pada perangkat jaringan dengan cara melakukan ketukan yang sudah disetting bisa dengan tiga kali ketukan atau ping sehingga *port* terbuka. Jika koneksi yang dikirimkan oleh host tersebut sudah sesuai dengan *rule knocking* yang diterapkan, maka secara dinamis *firewall* akan memberikan akses ke *port* yang sudah diblock. Dengan cara ini, perangkat jaringan seperti *Router* atau *microtik* akan lebih aman, sebab admin jaringan bisa melakukan *blocking* terhadap *port-port* yang rentan terhadap serangan seperti *Distributed Denial of Service Attack* (*Ddos Attack*). Jika dilakukan *port scanning port-port* tersebut terlihat tertutup.

Berdasarkan uraian latar belakang di atas, penulis mempunyai gagasan untuk mengangkat permasalahan tersebut dalam penelitian yang berjudul **"implementasi *honeypot* dan *port knocking* dalam mendeteksi serangan *distributed denial of servis attack (ddos attack)* pada server jaringan".**

1.2 Rumusan Masalah

Dari latar belakang diatas dapat ditarik permasalahan untuk dijadikan perumusan masalah yaitu bagaimana analisis dari implementasi *honeypot* dan *prort knocking* dalam mendeteksi serangan *distributed denial of servis attack (ddos attack)* pada server jaringan.

1.3 Tujuan Penelitian

Berdasarkan pada masalah yang telah didefinisikan di atas maka tujuan penelitian ini adalah analisis dari implementasi *honeypot* dan *port knocking* dalam mendeteksi serangan *distributed denial of servis attack (ddos attack)* pada server jaringan.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Mengetahui model dari sistem serangan *Ddos Attack (Distributed Denial Of Servis Attack)* pada server jaringan.

2. Mengetahui implementasi kinerja *honeypot* dan *port knocking* dalam mendeteksi serangan *ddos attack (distributed denial of servis attack)* pada server jaringan.

1.5 Batasan Masalah

Batasan-batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Penelitian dilakukan untuk menganalisis kinerja *performance* pada CPU dan jaringan pada komputer server sebelum dan sesudah adanya serangan *ddos attack* dan memabandingkanya setelah terpasang *honeypot* dan *port knocking* pada server jaringan.
2. Melihat kinerja dan fungsi dari *honeypot* dalam mendeteksi serangan *Ddos Attack (Distributed Denial Of Servis Attack)* dan *port knocking* dalam mengamankan server jaringan dari serangan *ddos attack*.
3. Fokus penelitian ini hanya pada server jaringan, seperti pada jaringan *localhost* ujian jamin berbasis *offline*, *router microtik server* dan aplikasi *computer based tes (CBT)* berbasis web yang dapat diakses dengan jaringan *localhost*.

BAB II
KAJIAN PUSTAKA

2.1 *Honeypot*

Honeypot adalah sistem umpan untuk mengumpulkan informasi *attacker* dari penyerang dengan cara menunggu, memantau setiap aktivitas *attacker* yang memulai interaksi, mengumpulkan sebanyak – banyaknya data yang dikenali sebagai sebuah serangan untuk melindungi sistem dan jaringan [3].

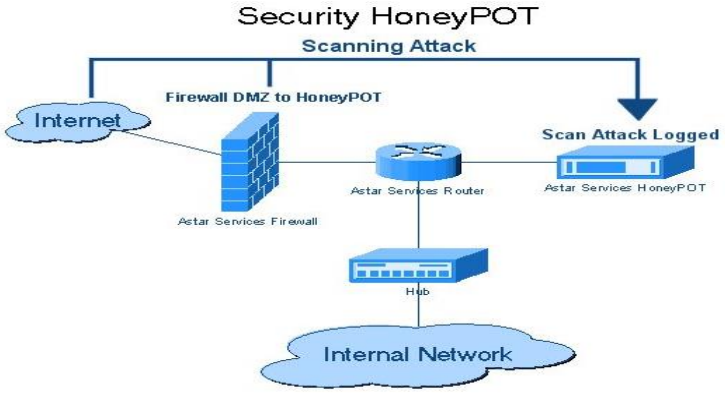
Saat ini untuk mengamankan jaringan komputer digunakan *Intrusion Detection System (IDS)* Namun *IDS* ini memiliki kelemahan, yaitu tidak mampu dalam menangani serangan baru yang belum diketahui sebelumnya. Laporan terbaru dari kasus *SSH Brute Force Attack* adalah serangan dengan tujuan memperoleh *credentials* dan *critical information*, seperti akses *administrator* dari target dan gambaran sistem yang digunakannya yang kemudian semua informasi tersebut akan di *internet* melalui *underground website.* *Honeypot* merupakan mekanisme baru dalam keamanan, membantu dalam memonitor dan mempelajari serangan. *Honeypot* adalah alat yang digunakan menjebak *attacker, honeypot* dapat memikat pengguna jahat dengan cara bertindak sebagai sistem yang mengandung data yang berharga atau layanan yang menarik, *honeypot* memungkinkan untuk dieksploitasi oleh *Attacker*. Hal inilah yang kemudian dapat membantu pakar keamanan profesional dan peneliti dalam proses pembelajaran terhadap teknik dan metode yang dilakukan oleh para *attacker*.

Honeypot tidak bisa mencegah serangan *cyber* terhadap jaringan sendiri, tetapi mereka dapat membantu dalam mengidentifikasi dan melakukan deteksi terhadap serangan ketika mereka digunakan bersama dengan perangkat pertahanan lainnya seperti *firewall*. *Honeypot* dapat menghasilkan sejumlah data yang memiliki nilai yang tinggi dan dapat juga menjadi tantangan bagi para pakar keamanan profesional [4].

1. Model Sistem *Honeypot*

Honeypot merupakan *security resource* yang yang sengaja dibuat untuk diselidiki, diserang, atau dikompromikan [5]. Pada umumnya *honeypot* berupa komputer, data, atau situs jaringan yang terlihat seperti bagian dari jaringan, tapi sebenarnya terisolasi dan dimonitor. Jika dilihat dari kacamata hacker yang akan menyerang, *honeypot* terlihat seperti layaknya sistem yang patut untuk diserang.

honeypot dapat diklasifikasikan berdasarkan pada tingkat interaksi yang dimilikinya. Tingkat interaksi dapat didefinisikan sebagai tingkat aktivitas penyerang didalam sistem yang diperbolehkan maka semakin tinggi pula tingkat interaksi *honeypot*.



Gambar 2.1 *Security Honeypot* [5]

2. *Low-interaction honeypot*

Low-interaction honeypot yaitu *honeypot* yang didesain untuk mengemulasikan *service* (layanan) seperti pada *server* yang asli, namun penyerang hanya mampu memeriksa dan terkoneksi ke satu atau beberapa *port*.

Kelebihan *low-interaction honeypot* yaitu:

1) Mudah di *install*, dikonfigurasi, *deployed*, dan *dimaintenance*.
2) Mampu mengemulasi suatu layanan seperti http, ftp, telnet dan sebagainya.
3) Difungsikan untuk deteksi serangan, khususnya pada proses *scanning* atau percobaan.

Kekurangan *low-interaction honeypot* ;

1) Layanan yang di berikan hanya berupa emulasi, sehingga penyerang tidak dapat berinteraksi secara penuh dengan layanan yang diberikan atau sistem operasinya secara langsung.
2) Informasi yang bisa kita dapatkan dari penyerang sangat minim. Apabila serangan dilakukan oleh "*real person*" bukan "*automated tools*" mungkin akan segera menyadari bahwa yang sedang dihadapi merupakan mesin *honeypot*, karena keterbatasan layanan yang bisa diakses.

3. *High-interaction honeypot*

High-interaction honeypot yaitu sistem operasi dimana penyerang dapat berinteraksi langsung dan tidak ada batasan yang membatasi interaksi tersebut. Namun dengan menghilangkan batasan-batasan tersebut akan menyebabkan tingkat risiko yang dihadapi semakin tinggi karena penyerang dapat memiliki akses *root*. Pada saat yang sama, kemungkinan pengumpulan informasi semakin meningkat dikarenakan kemungkinan serangan yang tinggi jika si penyerang telah mendapat akses *root*.

Kelebihan dari *high interaction honeypot* yaitu:

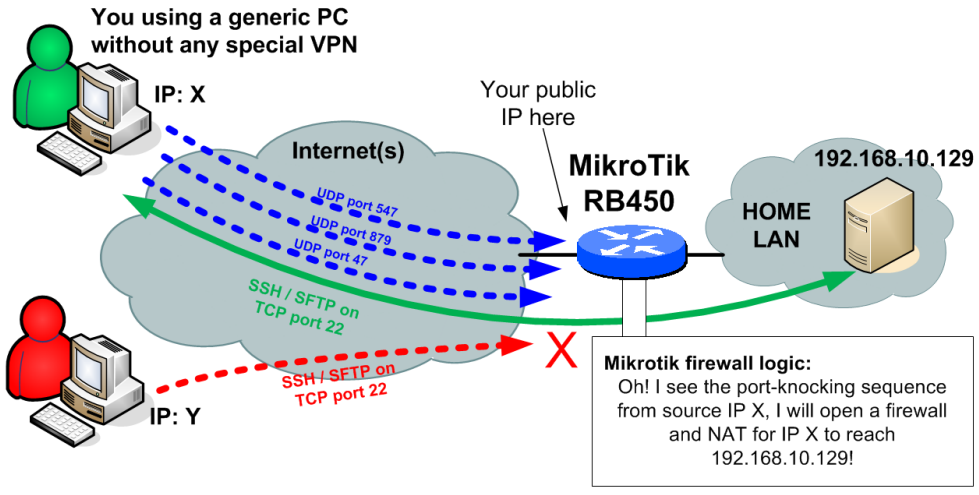
1) Penyerang berinteraksi langsung dengan sistem yang nyata termasuk diantaranya *sistem operasi*, *network*, hingga layanan yang diberikan *web service*, *ssh service*, *mail service* dan lain-lain.
2) Umumnya dibangun suatu sistem khusus dengan topologi yang telah dipersiapkan.
3) Sistem tersebut biasanya terdiri dari berbagai macam implementasi dari teknologi keamanan yang banyak digunakan untuk melindungi suatu sistem, seperti *firewall*, *IDS/IPS*, *router* dan lain-lain.
4) Target serangan berupa sistem operasi sebenarnya yang siap untuk berinteraksi secara langsung dengan penyerang.

Kekurangan *high interaction honeypot* :

1) Perencanaan dan implementasi sistem jauh lebih rumit dan dibutuhkan banyak pertimbangan.
2) *High-interaction honeypot* bersifat tidak efisien karena membutuhkan pengawasan berkala.
3) Apabila telah diambil alih oleh penyerang maka *honeypot* tersebut dapat menjadi ancaman bagi jaringan yang ada.

2.2 *Port Knocking*

Port Knocking adalah sebuah metode sederhana untuk memberikan akses *remote* tanpa meninggalkan *port* dalam keadaan selalu terbuka. Hal ini akan memberikan perlindungan kepada *server* dari *port scanning* dan serangan *scripts kiddies* [6].



Gambar 2.2 Metode *Port Knocking* [6]

Port Knocking memiliki metode buka *port* kepada suatu *klient* bila *klient* itu meminta, dan tutup kembali bila *klien*t telah selesai. Untuk menjalankan

metode ini, sebuah *server* haruslah memiliki *firewall* dan daemon untuk menjalankan metode *port knocking* yang berjalan di *server* tersebut. Dengan metode tersebut, *user* dituntut untuk memberikan *autentikasi* ke *server* agar *firewall* menulis ulang *rulenya* sehingga *user* diberi izin untuk mengakses *port* yang dimaksud. Dan setelah selesai, *user* mengirimkan *autentikasi* kembali untuk menutup *port* agar *firewall* menghapus *rulenya* yang ditulis sebelumnya untuk membuka *port*. Metode *port knocking* pada perancangan sistem usulan ini menggunakan protocol TCP untuk melakukan autentikasi. Sebelumnya port layanan yang terdapat pada *server* dikondisikan dalam keadaan *closed*/ tertutup sehingga layanan yang terdapat pada *server* tidak dapat diakses oleh siapapun. Ketika ingin mengakses layanan jaringan yang terdapat pada server harus melakukan autentikasi terlebih dahulu. Hal tersebut bertujuan untuk membuka *port* yang tertutup.

2.3 *Ddos* Attack (*Distributed denial-of-service Attack)*

Distributed denial-of-service attack (Ddos attack) merupakan jenis serangan yang telah ada sejak tahun 1990, dimana volume dan intensitas *Ddos* terus meningkat. Pada akhir tahun 2014, dilaporkan bahwa serangan *Ddos* merupakan teknik serangan yang paling populer [6]. Dengan demikian, *Ddos* merupakan salah satu ancaman utama dunia maya dan menjadi masalah utama keamanan cyber. *Ddos* disebut sebagai senjata pilihan hacker karena telah terbukti menjadi ancaman permanen bagi pengguna, organisasi dan *infrastruktur* di

Internet. Di sisi lain, serangan jaringan merupakan risiko untuk integritas, kerahasiaan dan ketersediaan sumber daya yang disediakan oleh organisasi [7].

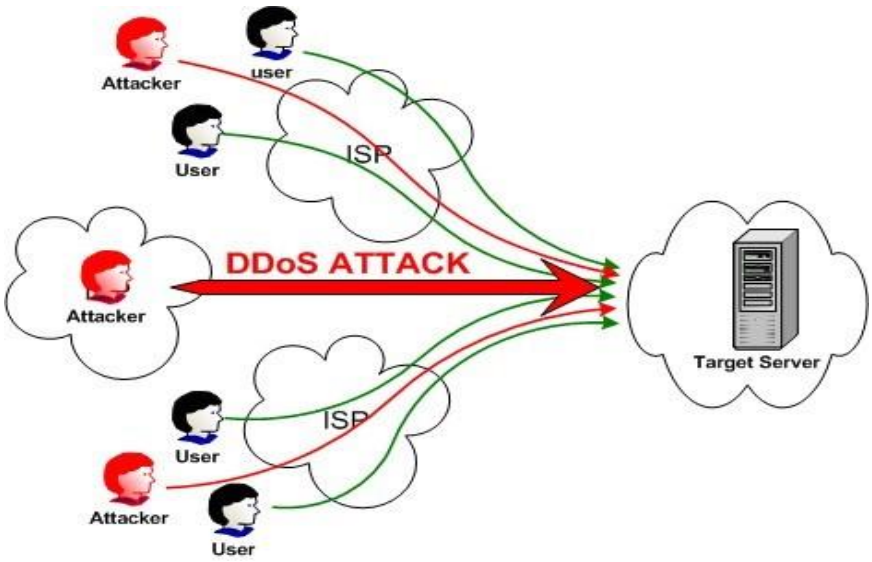
Pengenalan pola serangan *Ddos* pada *IDS* memiliki dua kelemahan. Pertama, karena *defisit* TCP/IP (B.Cid, 2015). Bagi *hacker*, serangan *Ddos* sangat mudah untuk dimulai, sementara korban sulit untuk menyadari. Selain itu, serangan *Ddos* mengalami perkembangan teknik yang mutakhir sebagai contoh adalah serangan *SYN-Flood*. Secara umum sebuah paket tunggal *SYN*, merupakan paket yang bersifat legal pada aktivitas jaringan sehingga sulit dideteksi sebagai artefak abnormal oleh IDS, sehingga IDS cukup sulit untuk membangkitkan alert apakah jaringan sedang diserang oleh *SYN-Flood*. Kedua, adanya masalah alert bersifat *false-positive* yang sering terjadi pada IDS yang berbasis signature, dimana pola jaringan normal dideteksi sebagai serangan *Ddos*, sehingga ketika benar-benar terjadi serangan *Ddos* waktu untuk menentukan dan melakukan tindakan mitigasi secara cepat untuk mengamankan jaringan tidak bisa dilaksanakan seefisien mungkin [8].

Ddos Attack atau Distributed Denial of Service attack adalah sebuah usaha serangan untuk membuat komputer atau server tidak dapat bekerja dengan baik. Dalam hal ini dapat menyebabkan performa server atau komputer menjadi sangat lambat. Selain itu, serangan *Ddos* juga mengganggu komunikasi antara sebuah *host* dan kliennya dengan berbagai cara. Selanjutnya memungkinkan pengubahan informasi yang dapat menyebabkan kerusakan pada sistem.

Berikut ini merupakan 3 tipe penggunaan teknik serangan *Ddos* yang umum digunakan saat ini;

1. *Request Flooding* adalah sebuah teknik serangan *Ddos* yang melakukan pengiriman *request*secara terus menerus sehingga membanjiri lalu lintas jaringan. Akibatnya pengguna lain yang juga meminta layanan tidak dapat dilayani oleh server.
2. *Traffic Flooding* merupakan teknik yang secara kerjasama seperti *request flooding*, pembedanya adalah paket yang dikirimkan. Jika dalam *request flooding* yang dikirim adalah *request* sedangkan dalam teknik *traffic flooding* yang dikirim adalah data sehingga pengguna lain tidak dapat dilayani.
3. *Crack Configuration*. Cara kerja pada teknik serangan *Ddos* adalah dengan mengubah konfigurasi dan merusak sistim bahkan komponen pada server sehingga server tidak dapat melakukan pelayanan kembali. Akan tetapi, teknik seperti ini sangat jarang digunakan karena cukup sulit untuk dilakukan kecuali untuk orang-orang tertentu.

Berikut adalah ilustrasi serangan *DDOS*



Gambar 2.3 Ilustrasi Serangan *Ddos* Attack [8]

Biasanya server jaringan yang mengalami serangan akan mengalami hal-hal berikut ini;

1. Kecepatan dan kinerja jaringan yang menurun.
2. Tidak berfungsinya bahkan hilangnya beberapa fitur.
3. Adanya sistem yang *crash* (rusak).
4. Peningkatan spams meningkat.
5. Website tidak dapat diakses.

2.4 Keamanan Jaringan Komputer

Keamanan jaringan komputer adalah suatu bentuk penanggulangan serangan yang dilakukan oleh *attacker* untuk masuk kedalam suatu jaringan komputer melalui lalu lintas jaringan yang tidak sah dari jaringan komputer luar[9].

Pada saat ini keamanan jaringan menjadi sangat penting dan patut untuk diperhatikan, jaringan yang terhubung dengan internet pada dasarnya tidak aman dan selalu dapat diekploitasi oleh para *hacker*, baik jaringan LAN maupun *Wireless*. Pada saat data dikirim akan melewati beberapa terminal untuk sampai tujuan berarti akan memberikan kesempatan kepada pengguna lain yang tidak bertanggung jawab untuk menyadap atau mengubah data tersebut. Dalam pembangunan perancangannya, sistem keamanan jaringan yang terhubung ke Internet harus direncanakan dan dipahami dengan baik agar dapat melindungi sumber daya yang berada dalam jaringan tersebut secara efektif dan meminimalisir terjadinya serangan oleh para *hacker*. Apabila ingin mengamankan

suatu jaringan maka harus ditentukan terlebih dahulu tingkat ancaman yang harus diatasi, dan resiko yang harus diambil maupun yang harus dihindari.

2.5 Penelitian terkait

Dalam penyusunan tesis ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian – penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada tesis ini. Adapun perbandingan penelitian yang terkait yaitu pada table 2.1 berikut;

Tabel 2.1 Tabel perbandingan penelitian terkait

No	Nama dan Tahun	Judul	Metode	Parameter	Hasil
1	Wilman[1], Iskandar Fitri[2], Novi Dian Nathasia3 Vol 3 No1 Maret 2018	Port Knocking Dan Honeypot Sebagai Keamanan Jaringan Pada Server Ubuntu Virtual	port knocking dan honeypot	Pengujian Keamanan Server, menggunakan aplikasi MobaXtreme dan *Putty*	implementasi *Port Knocking* dengan *Fitur Limit per-IP connection Rate* dan *honeypot* sebagai keamanan jaringan pada *Server* Ubuntu Virtual mampu mengamankan *server.*
2	Bagus Mardiyanto [1], Tutuk Indriyani[2], I Made Suartana[3] Vol 1, No 2,Septembe r 2016	Analisis Dan Implementasi Honeypot Dalam Mendeteksi Serangan Distributed Denial-Of-Services (*DDOS*) Pada Jaringan Wireless	Honeypot , honeyd dan iptables	*DDOS* (Distributed Denial of Service) pada server menggunaka n tools loic dengan jenis serangan Tcp Flood	Sebelum serangan beban cpu sebesar 15,25% dan setelah serangan beban cpu sebesar 45,98% dan setelah pembelokkan serangan beban cpu sebesar 30,83%.
3	Sutarti[1], Khairunnis a[2]	Perancangan Dan Analisis Keamanan Jaringan Nirkabel Dari Serangan	Honeypot	Dionaea adalah salah satu low interaction honeypot yang	Sistem honeypot telah berhasil meringankan tugas dari deteksi menjadi lebih sederhana, efektif dan murah.

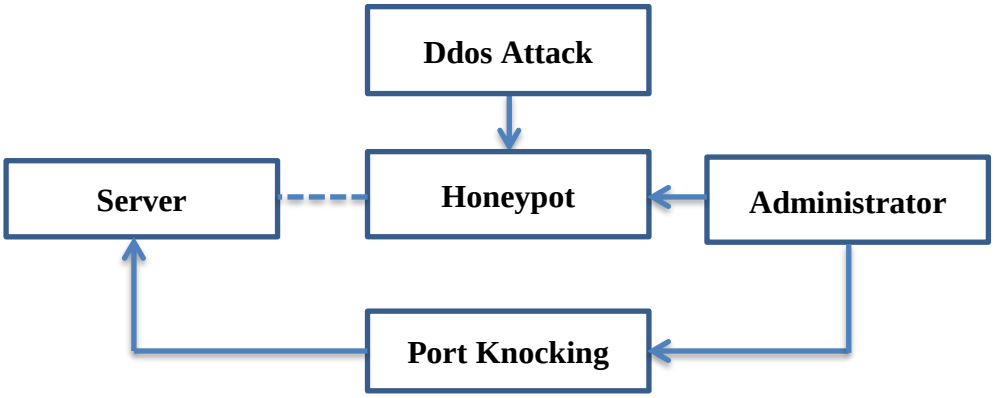
	Vol. 4 No. 2 September 2017	*Ddos* (*Distributed Denial Of Service*) Berbasis *Honeypot*		menawarkan layanan SMB, HTTP, FTP dan TFTP.	Konsepnya sendiri sangat mudah dipahami dan diimplementasikan. Honeypot sendiri ditujukan untuk mendeteksi serangan yang dilakukan oleh attacker dengan mengecoh attacker tersebut dengan fasilitas mirror server
4	Naufal Arkaana[1], Dolly Virgian Shaka Yudha Saktib[2] 19 September 2019	Implementasi Low Interaction Honeypot Untuk Peningkatan Keamanan Server dan Analisa Serangan Pada Protokol SSH	Honeypot	Honeypot ini berjenis low interaction menggunakan bahasa pemrograman python konsep network programming juga library paramiko untuk mengimplementasi protokol SSH yang digunakan	Hasil keseluruhan data penyerang berdasarkan negara yaitu 311 data serangan dengan jumlah 27 negara yang tercatat dan ditangkap oleh honeypot.
5	Devie Ryana Suchendra[1], Alfian Fitra Rahman[2], Setia Juli Irzal Ismail[3] Vol.10 No.2, Desember 2017	Penerapan Sistem Pengamanan Port Pada Layanan Jaringan Menggunakan Port Knocking	Port Knocking	Port knocking DNS, Port Knocking untuk Email dan Port knocking Untuk FTP	Hasil pengujian yang telah dilakukan menggunakan metode port knocking yang dikombinasikan dengan firewall di Mikrotik, dapat memberikan sistem keamanan autentikasi pada server layanan jaringan dan dapat mengamankan server dari 3 serangan yaitu Hydra, DoS, dan Telnet yang menggunakan ptotocol TCP
6	Rometdo Muzawi Vol. 2, No. 1, Juni	Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk	Utilitas Port Knocking	Serangan server dari sniffing acttack, Port Scanning	Hasil dari aplikasi pengendalian port dengan utilitas port knocking telah berhasil diuji dan menghasilkan kondisi port yang

	2016	Optimalisasi Sistem Keamanan Jaringan Komputer		pada port SSH yang terbuka	terbuka server berhasil melindungi layanan yang ada (disini berupa file server) dengan mengintegrasikan aturan firewall dengan program port knocking yang digunakan.
7	Rudi Hermawan Vol. 5 No. 1: 1-14	Analisis Konsep Dan Cara Kerja Serangan Komputer Distributed Denial Of Service (*Ddos*)	*Ddos* Attack (Distributed Denial Of Service Attack)	Serangan *Ddos* Attack pada website www.unindra.ac.id	Serangan *DDOS* menyebabkan bandwidth yang digunakan oleh korban akan habis yang mengakibatkan terputusnya koneksi antar server.
8	Syaifuddin[1], Diah Risqiwati[2], Eko Ari Irawan[3] *Techno.COM, Vol. 17, No. 4, November 2018*	*Realtime* Pencegahan Serangan *Brute Force* dan *DDOS* Pada Ubuntu *Server*	*Brute Force attack* dan *DDOS attack*	OpenSSH dan Apache. Sedangkan pada host penyerang terinstall sistem operasi Kali Linux dan program hydra, medusa, xerves dan browser.	implementasi fail2ban pada *Ubuntu server* versi 16 untuk mencegah serangan *bruteforce* dan *DDOS*
9	Sudiharyanto Lika[1], Roy Dwi Putra Halim[2] , Ihsan Verdian[3] Volume 4, No.2, 2018	Analisa serangan *sql injeksi* menggunakan *sqlmap*	*Sql injeksi sqlmap*	1. Mencari *database* yang terdapat pada situs web yang menjadi target. 2. Menginjeksi data-data yang terdapat pada *database* tersebut. 3. Mencari serta mendapatkan data yang cukup sensitif (user dan password) dari situs web tersebut.	aplikasi *SQLMAP* dari Kali Linux cukup handal untuk membobol kemanan dari situs web yang telah kami targetkan

2.6 Kerangka pikir

Pada penelitian ini peneliti akan mencoba melakukan analisis perbandingan pada *server* komputer jaringan yang sengaja untuk diserang dengan konsep metode *Ddos Attack (Distributed denial of service)* dengan membanjiri *server* dengan lalu lintas jaringan sehingga *server* pada jaringan ini sibuk sehingga tidak bisa untuk digunakan pengguna. Selanjutnya akan dilakukan serangan ulang namun pada server jaringan sudah dipasangkan satu komputer *koneypot* sehingga serangan yang tadinya mengarah pada server jaringan kini menuju ke komputer server *honeypot* sebagai jebakan untuk penyerang sehingga server jaringan dan pengguna pada jaringan tersebut tidak terganggu. Setelah diketahui adanya gangguan yang terdeteksi pada komputer *honeypot* barulah peneliti menggunakan metode *port knocking* untuk membuka *port* dengan metode ketukan tertentu sehingga dapat masuk kedalam server untuk memblokir penyusup yang mencoba masuk untuk mengganggu server jaringan.

Berdasarkan referensi dari peneliti-peneliti terdahulu dan permasalahan yang terkait dalam penelitian ini, maka berikut adalah kerangka pemikiran yang diusulkan dalam penelitian ini:



Gambar 2.4 Kerangka pikir

BAB III
METODE PENELITIAN

3.1 Waktu dan dan tempat penelitian

Penelitian ini dilakukan di Laboratorium STMIK Bina Bangsa Kendari dengan membuat konfigurasi server jaringan menggunakan *rotuer microtik*, model jaringan *client-server* untuk bisa menghubungkan aplikasi sistem ujian jamin berbasis *offline* dan aplikasi *web computer based test* dengan menggunakan jaringan *localhost*. Pengujian dalam penelitian ini menggunkan 11 unit komputer dimana 7 komputer digunakan untuk client, 1 komputer untuk *server honeypot*, 2 komputer digunakan untuk melakukan serangan *ddos attack* dan 1 komputer digunakan untuk server jaringan. Waktu penelitian dilakukan selama 3 bulan mulai dari bulan Mei, Juni dan Juli tahun 2020.

3.2 Pendekatan dan jenis penelitian

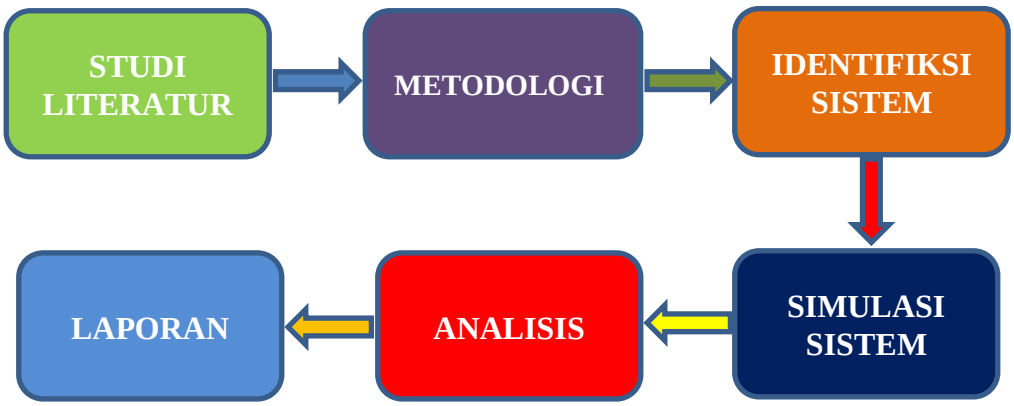
Pendekatan penelitian ini menggunakan tipe penelitian deskriptif [10].penelitian deskriptif adalah metode yang berfungsi untuk mendeskripsikan atau memberi gambaran terhadap obyek yang diteliti melalui data atau sampel yang telah terkumpul sebagaimana adanya, tanpa melakukan analisis dan membuat kesimpulan yang umum.

Jenis penelitian yang digunakan dalam penelitian ini adalah penelitian kuantitatif. Metode penelitian kuantitatif merupakan salah satu jenis penelitian yang spesifikasinya adalah sistematis, terencana dan terstruktur dengan jelas sejak awal hingga pembuatan desain penelitiannya. Metode penelitian kuantitatif, [10]

yaitu : “Metode penelitian yang berlandaskan pada filsafat positivisme, digunakan untuk meneliti pada populasi atau sampel tertentu, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif/statistik, dengan tujuan untuk menguji hipotesis yang telah ditetapkan”.

3.3 Tahapan penelitian

Pada tahapan penelitian menjelaskan bagaimana cara penelitian ini dilakukan, sehingga dapat memberikan rincian tentang alur atau langkah-langkah yang dibuat secara sistematis serta dapat digunakan dengan jelas dalam menyelesaikan masalah, membuat analisa terhadap hasil penelitian. Adapun tahapan penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur tahapan penelitian

3.4 Perancangan sistem honeypot dan port knocking

Perancangan sistem dalam penelitian ini dirancang untuk mengimplementasikan honeypot dan port knocking dalam keamanan sebuah server pada jaringan dari serangan *Ddos Attac (distributed denial of servic attack).*

Pada dasarnya *honeypot* sebagai pengalihan penyusup seolah – olah sudah masuk ke sever utama padahal penyerang atau *attacker* berada pada *ip* komputer diluar dari server sehingga jaringan yang sudah terbentuk tidak terganggu, *honeypot* bisa melihat log/aktivitas yang dikerjakan oleh *intruder* terhadap server.

Sedangkan *port knocking* berfungsi untuk membuka akses ke *port* tertentu yang telah diblock oleh Firewall pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu, koneksi bisa berupa protocol ICMP, TCP, dan UDP jika koneksi yang dikirimkan oleh *host* tersebut sudah sesuai dengan rule autentikasi yang diterapkan, maka secara dinamis *firewall* akan memberikan akses ke *port* yang sudah diblock sehingga jika terdapat user asing yang tidak dikenal *port* tersebut langsung ditutup untuk memblokir penyusup yang mencoba mengganggu server jaringan[11].

3.5 Perancangan serangan *distributed denial off service attack (ddos attack)*

Pada proses pengujian serangan dalam penelitian ini ada beberapa teknik ddos attack yang akan digunakan, yang nantinya akan dianalisis bagaimana keadaan komputer server pada jaringan sebelum dan sesudah adanya serangan dan bagaimana ketika sudah diterapkannya implementasi dari kemanan jaringan dengan menggunakan *honeypot* dan *port knocking* yaitu sebagai berikut;

1. Serangan *Ddos Attack Request Flooding*

   Request Flooding adalah sebuah teknik serangan *ddos* yang melakukan pengiriman *request* secara terus menerus sehingga membanjiri

lalu lintas jaringan. Akibatnya pengguna lain yang juga meminta layanan tidak dapat dilayani oleh server.

2. Serangan *Ddos Attack Traffic Flooding*

   Traffic Flooding merupakan teknik yang sistem kerjanya sama seperti *request flooding*. Pembedanya adalah paket yang dikirimkan, jika dalam *request flooding* yang dikirim adalah *request* sedangkan dalam teknik *traffic flooding* yang dikirim adalah byte data sehingga pengguna lain tidak dapat dilayani.

3. Serangan *Ddos Brute Force attack*

   Teknik *hacking Brute Force* adalah salah satu teknik penyerang untuk meretas *password* sebuah server, jaringan atau *host*, dengan cara mencoba semua kemungkinan kombinasi *password* yang ada pada *wordlist* atau "kamus *password*". Metode ini digunakan untuk menemukan *password* yang ingin diretas. Serangan *brute-force* merupakan metode coba-coba yang digunakan oleh peretas untuk menebak kredensial atau data terenkripsi seperti *login*, kata sandi, atau kunci enkripsi sebagai cara untuk bisa masuk kedalam server sebuah jaringan. Pada penelitian ini penyerang akan menggunaka metode *exploit microtik* dengan *winbox exploit* untuk mendapatkan *password microtik* agar bisa masuk kedalam server jaringan.

4. Serangan *Ddos attack SQL Injection*

   Structured Query Language (SQL) digunakan untuk melakukan *query*, mengoperasikan, dan mengelola sistem *database* seperti SQL

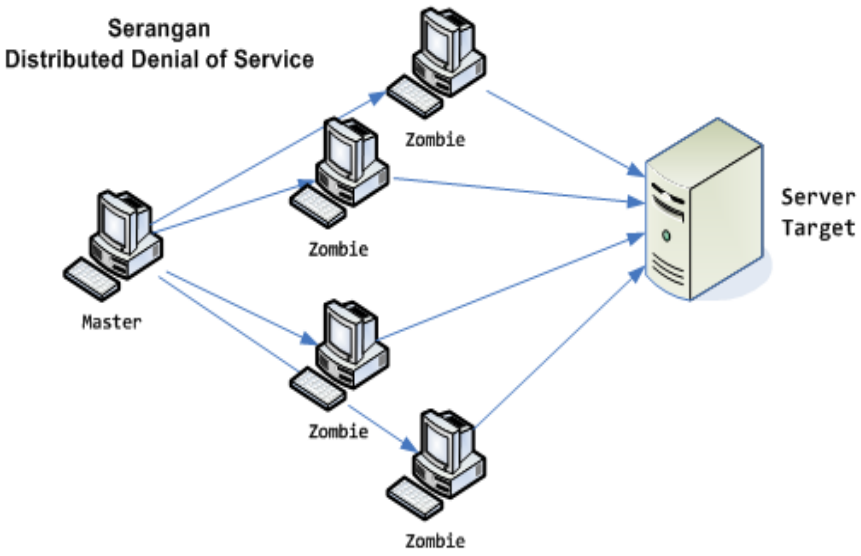
server, *oracle*, atau MySQL. Penggunaan umum SQL konsisten di semua sistem databatase, namun ada detail perbedaan tertentu yang khusus untuk setiap sistem.

SQL injection adalah jenis aksi *hacking* pada keamanan komputer di mana seorang penyerang bisa mendapatkan akses ke basis data di dalam sistem dengan memanfaatkan sebuah celah keamanan yang terjadi dalam lapisan basis data untuk mendapatkan *password* atau *username* pada sebuah aplikasi web yang tidak diproteksi dengan baik.

Sebelum serangan ddos attack dilakukan terlebih dahulu penyerang melakukan *scanning ip addres* dan *port* yang digunakan oleh server jaringan dengan menggunakan *software advanced ip scanner* dan *Scaning port nmap* setelah *ip* dan *port* ditemukan barulah pengujian serangan *ddos attack* dilakukan.

3.6 Model sistem dan analasis

Model sistem dalam penelitian ini yaitu mencoba untuk melakukan serangan *Ddos Attack* pada perangkat server jaringan yang bertujuan untuk melakukan pengujian *Ddos Attack* yang nantinya akan mendeteksi serangan dengan *honeypo*t dan menghentikanya atau memblok serangan dengan *port knocking*. Berikut gambar 3.2 gambaran umum serangan *Ddos Attack.*

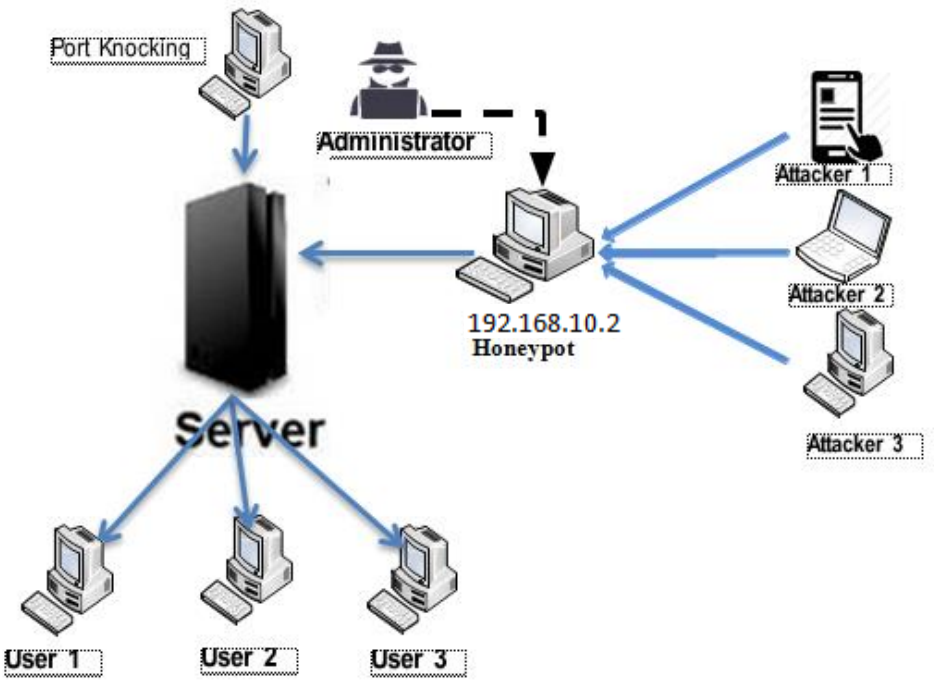


Gambra 3.2 gambaran umum serangan *Ddos Attack*

Serangan *ddos attack* bertujuan untuk mengganggu sistem pada server jaringan agar tidak dapat melayani komputer *klient*, yang juga pada serangan ini akan mencoba melakukan *eksploitasi* pada *server microtik* untuk mendapatkan *username* dan *password* untuk dapat masuk menguasai server pada jaringan tersebut, serangan selanjutnya yaitu mencoba untuk menginjeksi *struktur database* agar bisa membuka halaman web sebagai pengguna admin, namun pada pengujian ini serangan tersebut akan dideteksi dan digagalkan dengan imlplementasi *honeypot* dan *port knocking*.

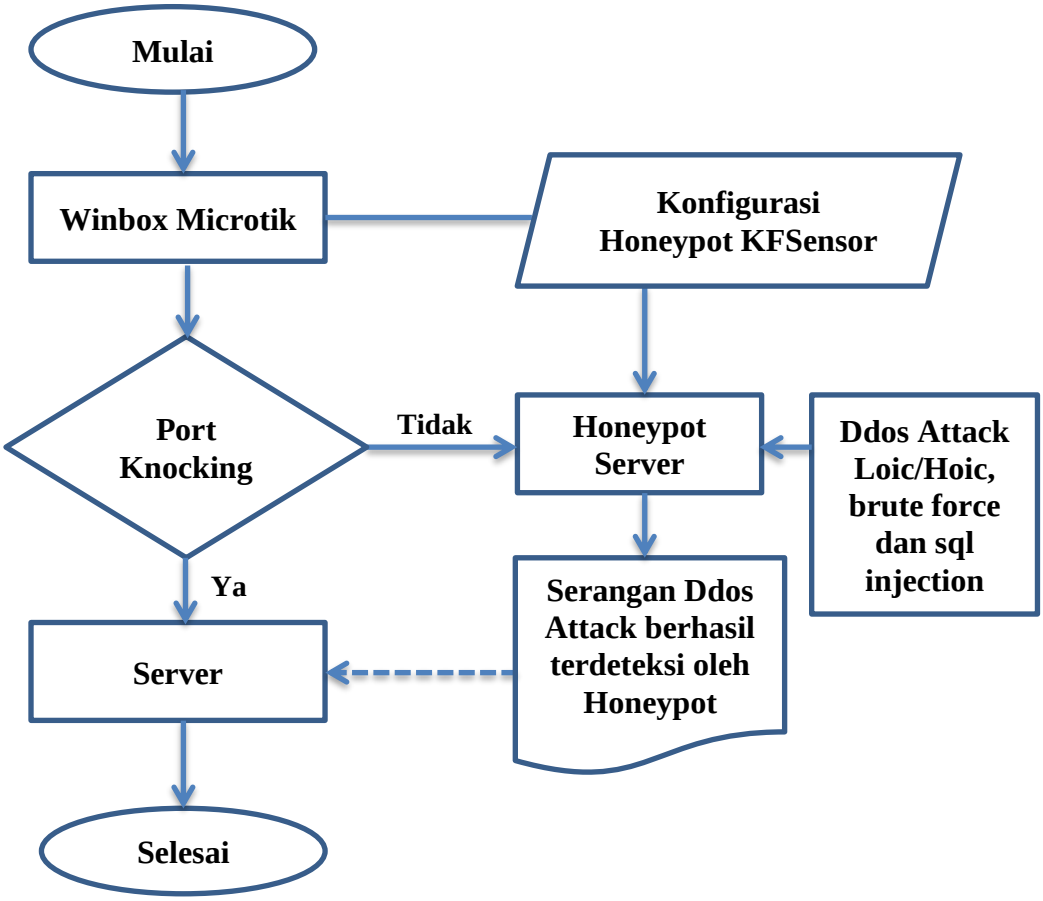
Pada dasarnya *port knocking* dapat didefinisikan sebagai suatu metode komunikasi antara dua computer, sedangkan *honeypot* sebagai pengalihan penyusup seolah – olah sudah masuk ke sever utama. Berikut gambaran umum pada *hardware* dalam proses perancangan implementasi *Honeypot* dan *Port*

Konocking sebagai keamanan jaringan pada server dari serangan *Dddos Attack* seperti gambaran umum sistem pada gambar 3.3 dibawah:



Gambar 3.3 Gambaran umum sistem *Honeypot* dan *Prot Knocking*

Pada gambaran umum sistem *honeypot* dan *port knocking* gambar 3.3 diatas server jaringan menjadi target dari serangan *ddos attack*, namun serangan tersebut telah dialihkan dan masuk kedalam server *honeypot* yang dimonitor langsung oleh administrator, yang selanjutnya administrator akan mengambil langkah-langkah tindakan pemblokiran *user* serangan *ddos attack* dengan metode *port knocking*. Setelah mengetahui bagaimana konsep perancangan *hardware* pada gambaran umum sistem gambar 3.3 diatas, berikut adalah model *flowchart* dari gambaran umum *software* seperti pada gambar 3.4 dibawah:



Gambar 3.4 Flowchart gambaran umum *software*

3.7 Kebutuhan *Hardware* dan *Software*

Berikut adalah kebutuhan *hardware* dan *software* yang digunakan dalam proses pengujian penelitian, dalam merancang server jaringan dan mengimplementasikan *Honeypot* dan *Port Knocking* dalam mendeteksi serta mengatasi serangan *Distributed Denial of Service Attack* (*Ddos Attack*) untuk keamanan server jaringan, pada table 3.1 berikut:

Tabel 3.1 Kebutuhan *hardware* dan *software*

KEBUTUHAN PERANGKAT	
HARDWARE	11*Unit* Personal Komputer
	MIKROTIK Router Wireless HAP-Lite2 RB941-2nD-TC
	Kabel UTP
	Konektor RJ-45
SOFTWARE	*Sistem Operasi Windows* 10-64 *bit* dan *windows* 7-64 *bit*
	Xampp
	Web Browser : *US Browser, Google Chrome*
	*Microtik Router/Winbox (*untuk *server* dan *port knocking)*
	Advanced ip scanner dan *Scaning port nmap*
	Loic, hoic, phyton 3.8.5-27, script winbox exploit dan *script sqlmap*
	Honeypot kfsens40
	Ujian jamin offline dan *computer based test (CBT)*

BAB IV
HASIL DAN PEMBAHASAN

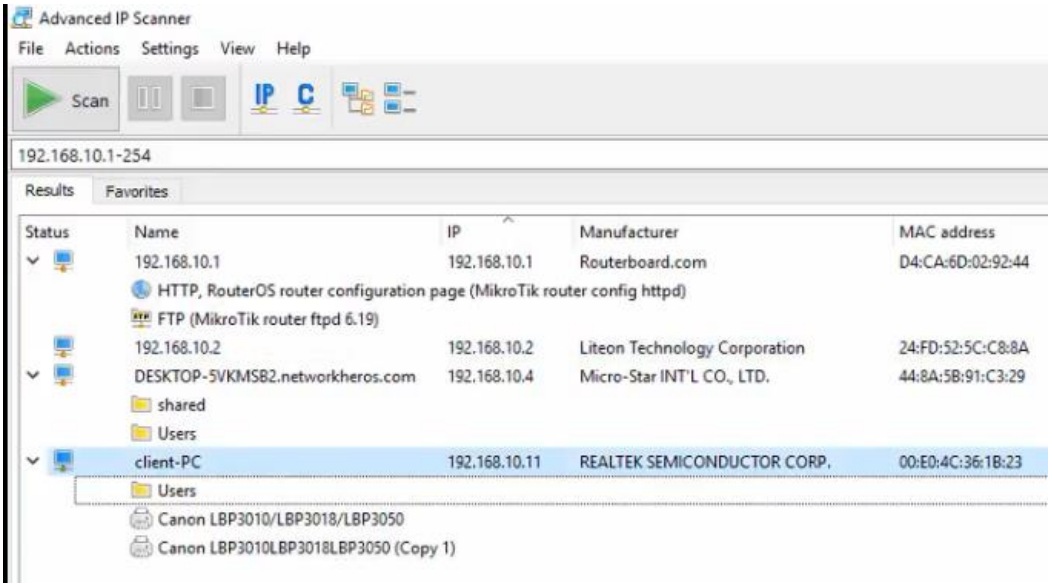
4.1 Hasil penelitian

Hasil penelitian yang didapatkan dari pengujian terhadap sistem keamanan server jaringan menggunakan *honeypot* dan *port knocking* dari serangan *Distributed Denial Of Servis Attack (Ddos Attack)* adalah sebagai berikut;

4.1.1 Simulasi serangan *Distributed Denial Of Servis Attack (Ddos Attack)*

Pada proses simulasi serangan *ddos Attack* akan digunakan beberapa teknik dan analisis serangan dilakukan sebelum menggunakan keamanan jaringan *honeypot* dan *port knocking* serangan *ddos attack* yang dilakukan pada pengujian penelitian ini menggunakan beberapa target seperti ujian jamin *offline* dengan menggunakan jaringan *local warless network, eksploit microtik* dan *SQL injection* yaitu untuk mendapatkan struktur database dari sebuah aplikasi web agar bisa *login* sebagai admin.

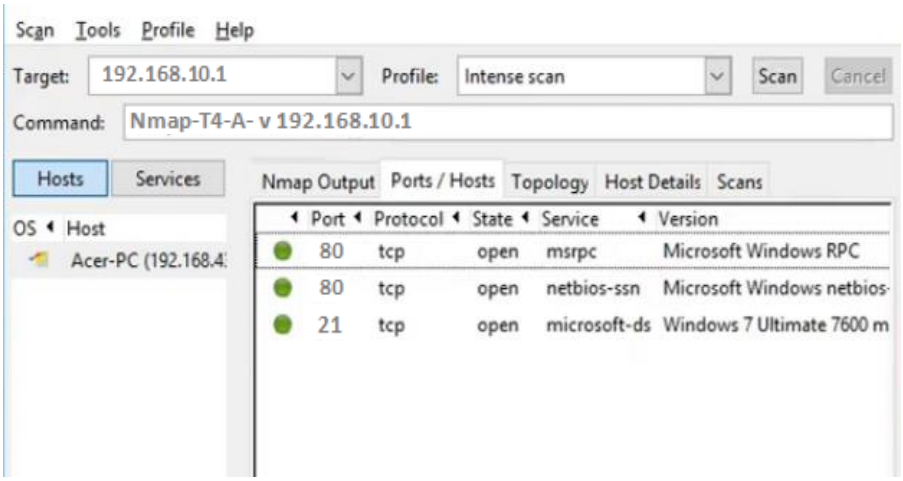
Sebelum serangan dilakukan penyerang harus mendapatkan ip target terlebih dahulu agar bisa melakukan serangan ke sistem server jaringan yang digunakan untuk bisa melakukan *ddos attack*, *tools* yang akan digunakan yaitu *advanced ip scanner*, berikut hasil *scanner ip* pada gambar 4.1 Hasil *advanced ip scanner*.



Gambar 4.1 Hasil *advanced ip scanner*

Didapatkan hasil *scanner ip* target sasaran yaitu 192.168.10.1, 192.168.10.2, 192.168.10.4 dan 192.168.10.11 namun yang digunakan oleh komputer pada server jaringan adalah 192.168.10.1 *ip* inilah yang nantinya dalam pengujian pada penelitian ini akan dilakukan serangan sebelum terpasang sistem keamanan jaringan *honeypot* dan *port knocking* pada server jaringan.

Setelah didapatkan *ip* target selanjutnya yaitu proses *scanning port* untuk memastikan serangan *ddos attack* pada *port* berapa dan menggunakan *protocol* apa, berikut hasil *scaning port* target menggunakan *software nmap* pada *windows* seperti pada gambar 4.2 *Scaning port nmap*.



Gambar 4.2 *Scaning port nmap*

Didapatkan hasil *scaning nmap* pada *ip* 192.168.10.1 yaitu ternyata menggunakan *port* 80 dan *port* 21 dengan *protocol tcp*, *ip address* dan *port* inilah yang nantinya akan menjadi target serangan *ddos attack*.

Tabel 4.1 Hasil scanning *ip address* dan p*ort*

No	Jenis Scanning	Hasil ip address	Hasil port
1	*Advanced ip scanner*	192.168.10.1 192.168.10.2 192.168.10.4 192.168.10.11	-
2	*Scaning port nmap*	-	80 21

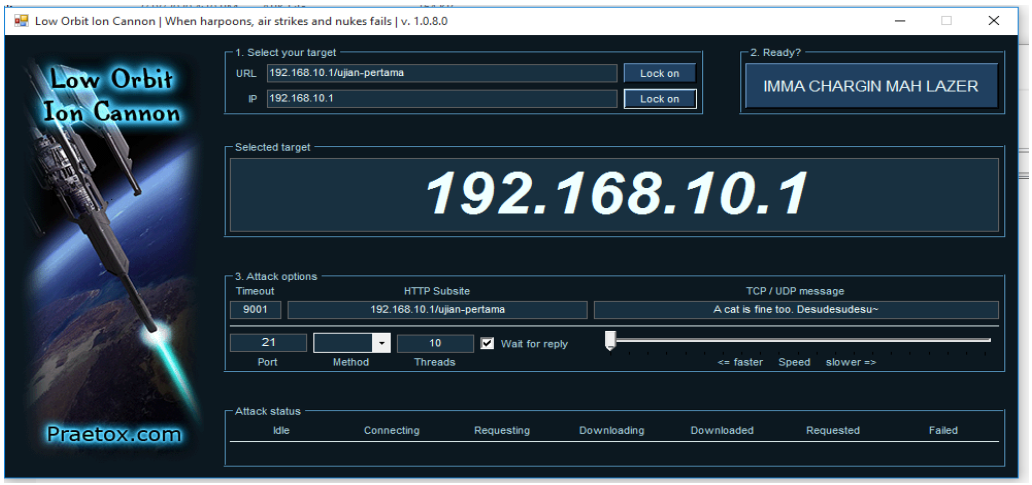
Setelah didapatkan hasil *scanning ip address* dan *port* target,dimana *ip* yang digunakan server adalah 192.168.10.1 terdapat keterangan pada *manufactur* iyalah *routerboard* dan *port* 21 terdapat keterangan pada *service* yaitu *microsoft-ds*. Tahapan selanjutnya adalah melakukan serangan *ddos attack* dimana pada server jaringan ini belum dikonfigurasi dengan *honeypot* dan *port knocking*, berikut tahapan-tahapan dan hasil serangan *distributed denial of servis attack (ddos attack)*;

4.1.1.1 Serangan *ddos attack* pada ujian jamin berbasis *offline* menggunakan jaringan *wireless*

Pada tahap ini pengujian penelitian yang dilakukan yaitu menggunakan jaringan *local area network* dengan simulasi ujian berbasis *offline* dengan percobaan pengujian serangan menggunakan 2 *software ddos* yaitu *hoic* untuk *request flooding* pada jaringan dan *loic* untuk *traffic flooding* pada jaringan secara bersamaan untuk mengirim *request* dan paket data dalam jumlah banyak sehingga CPU pada komputer server bekerja semakin keras dan lalulintas jaringan menjadi sibuk dan padat sehingga jaringan yang terhubung ke server sangat lambat dan bahkan sebagian tidak bisa untuk diakses.

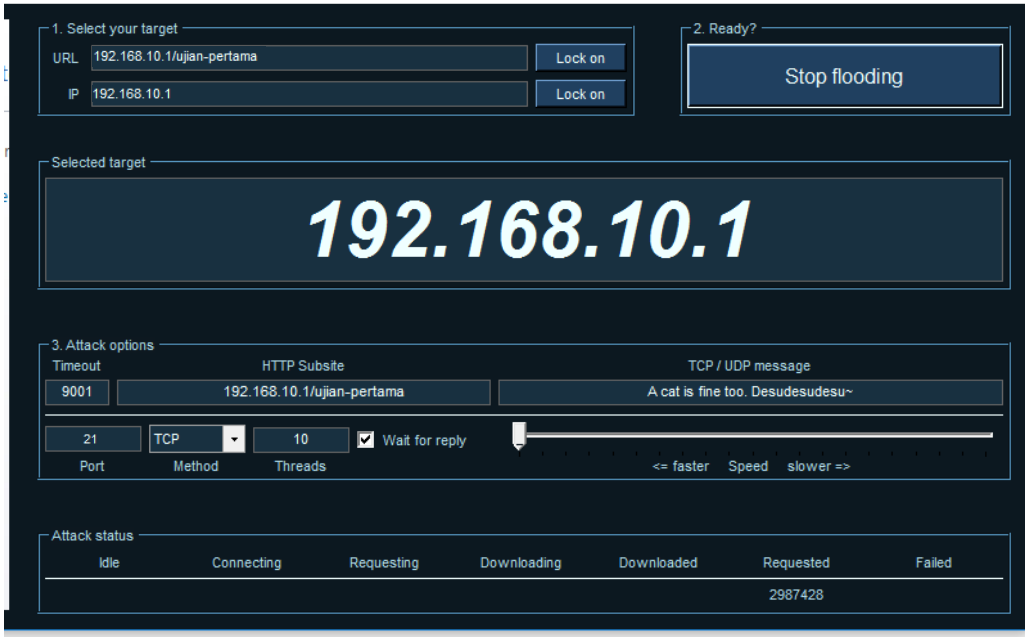
1. Serangan *Ddos Attack Request Flooding*

   Pada pengujian *Ddos Attack Request Flooding* untuk mengirimkan *reques*t pada jaringan secara terus menerus penyerang menggunakan *software loic* tujuannya adalah untuk membanjiri lalu lintas jaringan sehingga server jaringan tidak bisa melayani pengguna lain, seperti pada gambar 4.3 *Ddos attack loic*;



Gambar 4.3 *Ddos attack loic*

Setelah *software* berjalan masukan alamat URL target 192.168.10.1/ujian-pertama dan *ip* target 192.168.10.1 yang sudah terdeteksi yang selanjutnya penyerang akan melumpuhkan dengan *riquest* secara terus menerus dengan klik *IMMA CHARGIN MAH LAZER* secara otomatis *request* terkirim dan berjalan seperti gambar 4.4 *Ddos attack request loic.*



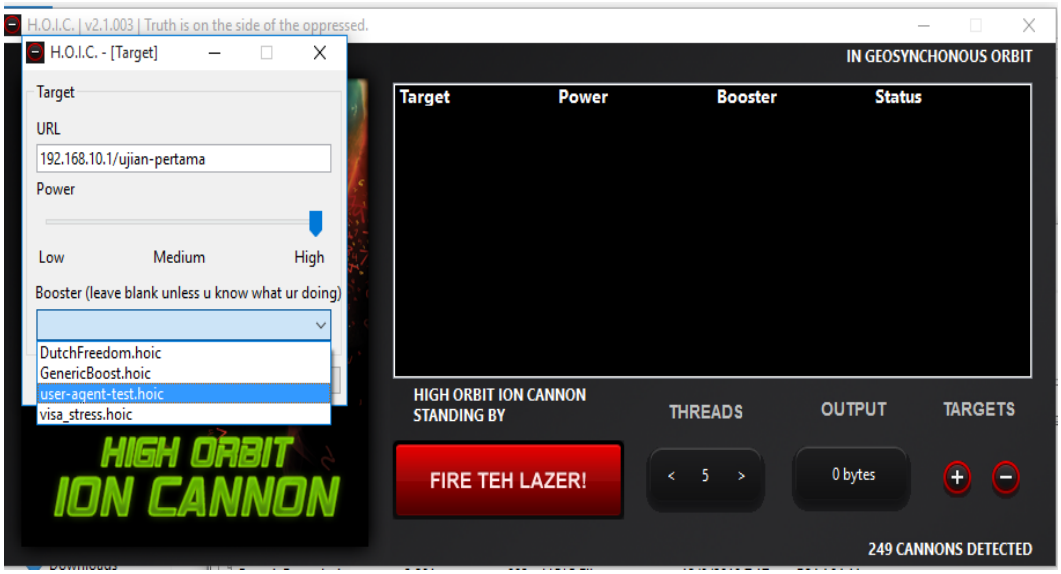
Gambar 4.4 *Ddos attack request loict*

Request flooding sedang berjalan pada *port* 21, *metdhod* TCP dan *threads* 10 dengan level kecepatan *faster* atau kecepatan tertinggi dalam level pengujian ini, *name* serangan *desudesudesu* pengujian serangan akan dihentikan jika sistem jaringan sudah *down*.

2. Serangan *Ddos Attack Traffic Flooding*

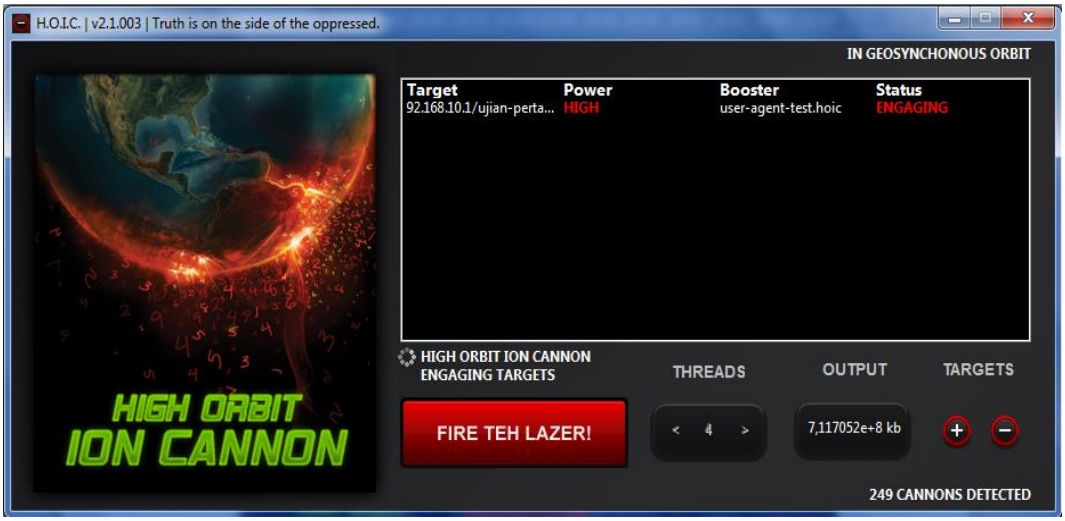
Traffic flooding pada *ddos attack* ini pada dasarnya yaitu mengirim paket *byte* berupa data sebanyak-banyaknya dengan menggunakan *software*

hoic agar server jaringan tidak bisa melayani pengguna lain hampir sama dengan *riquest flooding* namun perbedaanya *traffic flooding* berupa byte data yang dikirimkan keluar kedalam ip target sasaran, seperti pada gambar 4.5 URL target *ddos attack traffick hoic*;



Gambar 4.5 URL target *ddos attack traffick hoic*

Pada serangan *ddos attack hoic* masukan alamat URL dari halaman *website* yang terhubung ke jaringan untuk selanjutnya menjadi target penyerang memasukan *bytes* data dalam jumlah banyak pada pengujianya menggunakan URL 192.168.10.1/ujian-pertama sebagai target server, sedangkan untuk *power* pilihanya ada *low*, *medium* dan *high* peneliti menggunakan *high* untuk kecepatan yang tinggi selanjutnya *booster* ada beberapa pilihan tergantung keinginan penyerang akan melakukan kelumpuhanya sampai ditingkat mana namun dalam penelitian ini penyerang akan menggunakan *user-agent-test.hoic* untuk mengirim paket data ke server jaringan, seperti pada gambar 4.6 *ddos attac traffic hoic*.



Gambar 4. 6 *Ddos attac traffic hoic*

Status pada *software hoic ENGAGING* yang artinya sedang berjalan untuk memasukan paket *traffic* berupa byte data kedalam *ip server* sebagai target untuk mengganggu sistem jaringan agar tidak bisa digunakan antara *user* komputer *client-server*.

Dari hasil pengujian serangan pada aplikasi web ujian jamin berbasis *offline* dengan model jaringan *localhost* pada server jaringan menggunakan *ddos attack request flooding* dan *traffick flooding* serangan berhasil masuk pada komputer server jaringan berupa *request* paket dan byte data. Berikut pada tabel 4.2 Hasil serangan *ddos attack request flooding* dan *ddos attack traffick flooding* sebelum terpasang *honeypot* dan *port knocking*.

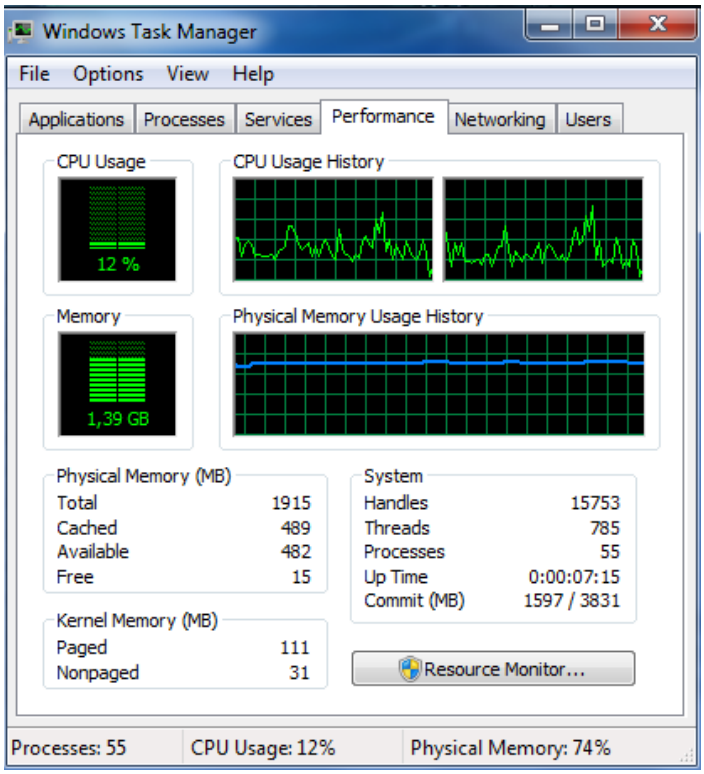
Tabel 4.2 Hasil serangan *ddos attack request flooding* dan *ddos attack traffick flooding*

Pengujian	**Jenis serangan**	**Ip target/ Port**	**Waktu**	**Level serangan**	**Hasil serangan**
Uji 1	*Ddos attack request flooding*	192.168.10.1/ 21	31 menit	*Faster*	2987428 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	31 menit	*High*	7.117052e+8 kb
Uji 2	*Ddos attack request flooding*	192.168.10.1/ 21	25 Menit	*Faster*	2735891 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	25 menit	*High*	5.363787e+8 kb
Uji 3	*Ddos attack request flooding*	192.168.10.1/ 21	22 menit	*Faster*	2583947 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	22 menit	*High*	7.023877e+7 kb
Uji 4	*Ddos attack request flooding*	192.168.10.1/ 21	37 menit	*Faster*	3310978 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	37 menit	*High*	2.116352e+9 kb
Uji 5	*Ddos attack request flooding*	192.168.10.1/ 21	29 menit	*Faster*	2872984 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	29 menit	*High*	1.080268e+9 kb

Pada pengujian serangan *ddos attack request flooding* dan *ddos attack traffick flooding* dilakukan 5 kali tahapan pengujian sebelum digunakanya sistem keamanan *honeypot* dan *port knocking*. Lama waktu serangan untuk pengujian pertama 31 menit, pengujian kedua 25, pengujian ketiga 22 menit, pengujian keempat 37 dan pengujian kelima 29 menit atau dengan rata-rata waktu sekitar 28 menit, serangan akan dihentikan jika server jaringan target sudah tidak bisa lagi digunakan atau *lost connection* pada komputer *user client* sudah tidak bisa lagi terhubung ke server, dimana serangan *ddos attack request flooding* menggunakan *ip* target yaitu 192.168.10.1 dengan *port* 21 dan TCP koneksi pada level serangan *faster*, *name* serangan *desudesudesu* dengan jumlah *request time out* masuk pada server jaringan setiap *and off Connection* pada pengujian pertama 2987428 *seconds*, kedua 2735891 *seconds*, ketiga 2583947 *seconds*, keempat 3310978 *seconds* dan kelima 2872984 *seconds*. Sedangkan serangan *ddos attack traffick flooding* juga menggunakan *ip* yang sama yaitu 192.168.10.1 karena dalam pengujian serangan ini targetnya 1 server pada *port* 21 dengan level *power* serangan *high* dan *booster* tingkat gangguan *user-agent-test.hoic* jumlah paket berupa *byte* data yang masuk pada server jaringan pada pengujian serangan pertama 7.117052e+8 kb, kedua 5.363787e+8 kb, ketiga 7.023877e+7 kb, keempat 2.116352e+9 kb dan kelima 1.080268e+9 kb, berdasarkan hasil serangan *ddos attack* tersebut server jaringan menjadi *down* dimana *web browser client* tidak bisa lagi melakukan ujian jamin.

3. Analisis hasil komputer server setelah adanya serangan *ddos attack request loic* dan *ddos attack traffic hoic*

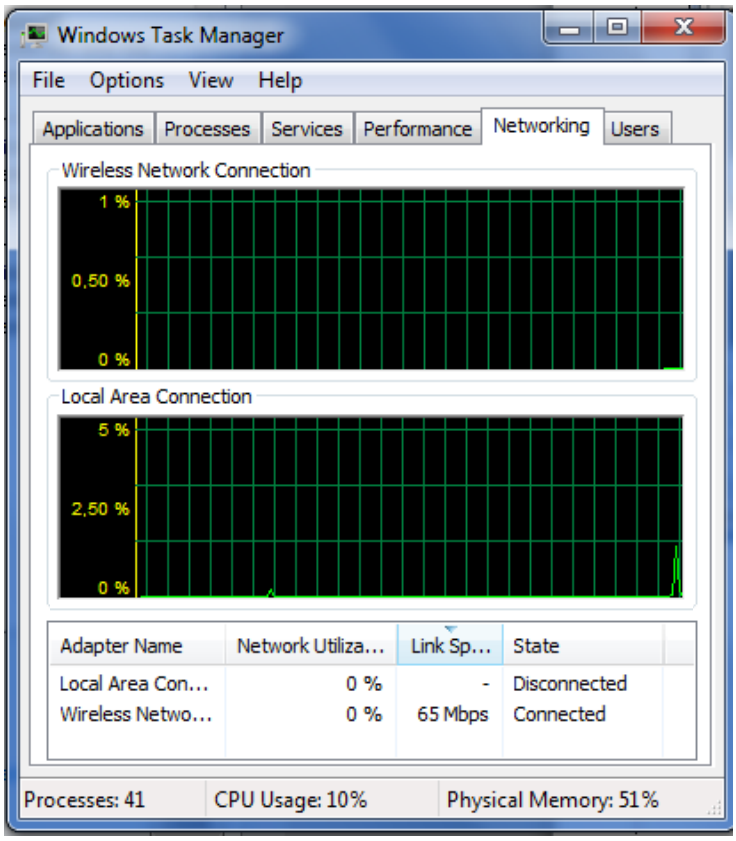
setelah serangan *ddos attack* dilakukan ternyata hasilnya berhasil mengganggu server jaringan dan dapat dilihat pada *performance* CPU komputer server kinerjanya semakin berat karena banyaknya lalulintas jaringan yang masuk kedalam sistem jaringan, berikut gambar 4.7 hasil sebelum terjadinya serangan *ddos attack.*



Gambar 4.7 CPU *usage* sebelum serangan *ddos attack*

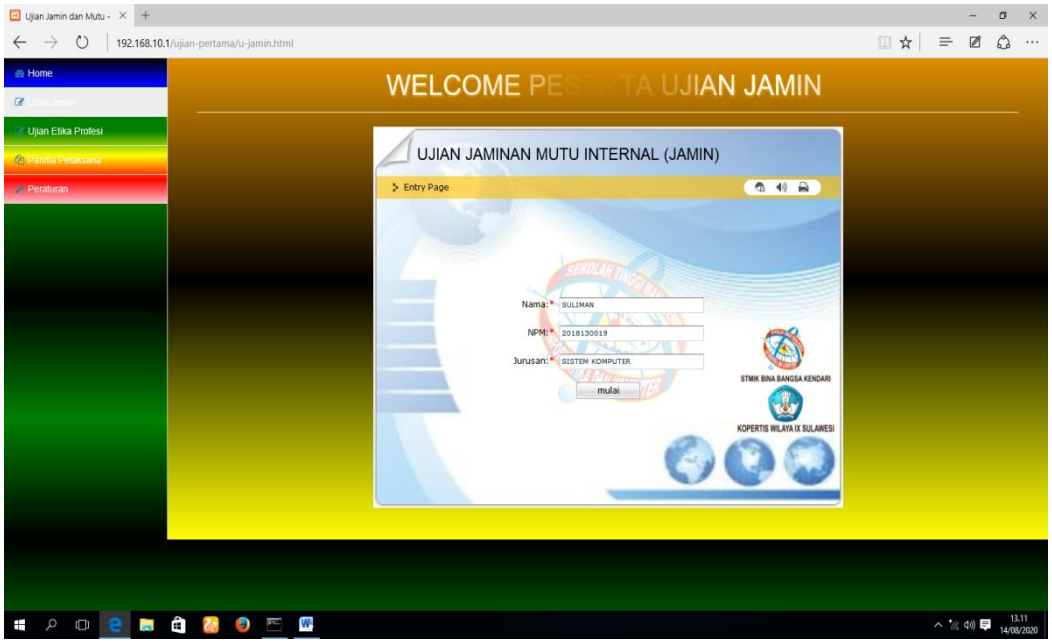
Sebelum terjadinya serangan *performance* kinerja CPU *usage* pada *windows task manager* menunjukan 5% sampai 12% yang artinya CPU masih bekerja secara normal dan juga pada *networking* untuk kerja pada jaringan *local area connection* masih 1% sampai 5% menunjukan bahwa

masih bekerja secara baik dan beban kerjanya ringan karena memang hanya sedikit pengguna yang terhubung ke jaringan karena dalam pengujian ini peneliti hanya menggunakan 7 komputer *user*, 1 komputer server *honeypot*, 2 komputer untuk melakukan serangan *ddos attack* dan 1 komputer *server*, seperti pada gambar 4.8 kinerja jaringan *local area connection.*



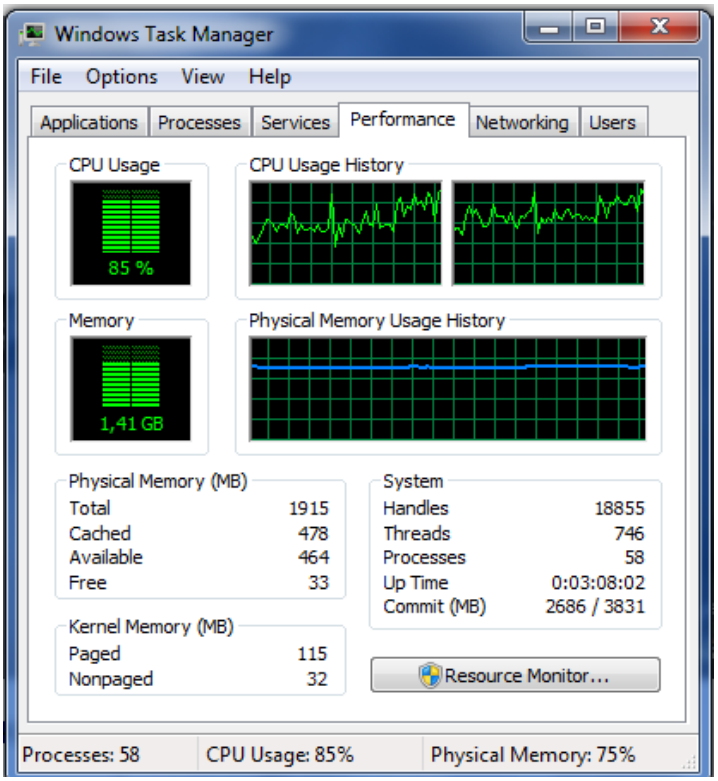
Gambar 4.8 *Network connection*

Sistem jaringan pada pengguna yang terhubung ke jaringan server ujian berbasis *offline* masih bisa terhubung sehingga pengguna lainya bisa masuk untuk melakukan ujian jamin berbasis *offline* dengan membuka *browser* dengan alamat ip 192.168.10.1/ujian-pertama, seperti pada gambar 4.9 Ujian jamin *offline*.



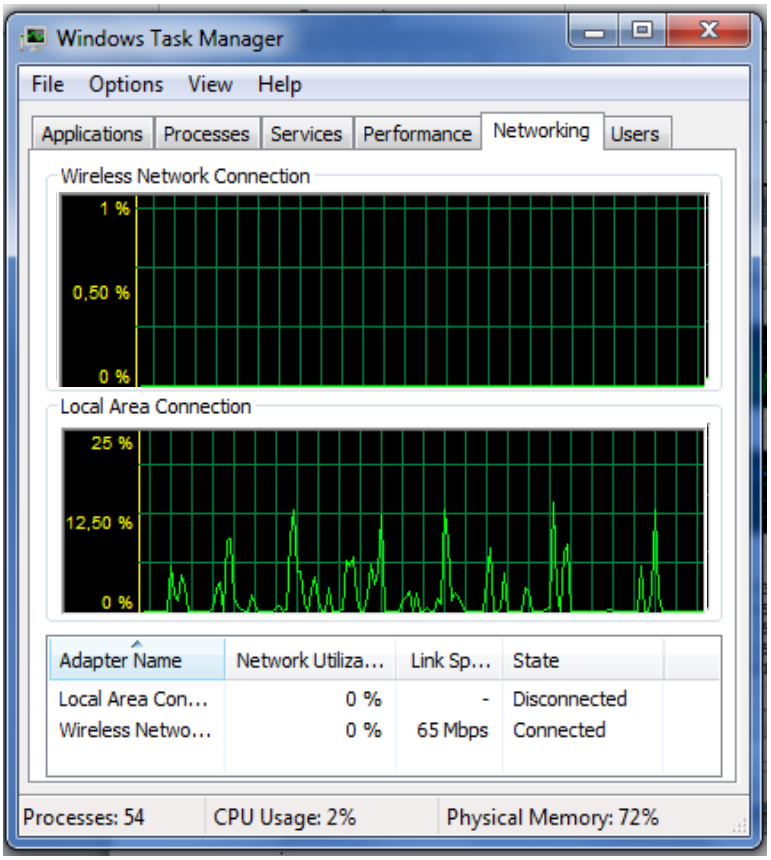
Gambar 4.9 Ujian jamin *offline*

Setelah terjadi serangan *ddos attack* dengan menggunakan *ddos attack request loic* dan *ddos attack traffic hoic* kinerja CPU semakin meningkat dapat dilihat pada CPU *usage history traffic* kinerja CPU mencapai 71% hingga 85% yang artinya kinerjanya semakin berat dikarenakan lalulintas jaringan yang semakin sibuk dan padat setelah adanya terjadi serangan *ddos attack*, seperti pada gambar 4.10 *CPU usage* setelah ada serangan *ddos attack.*



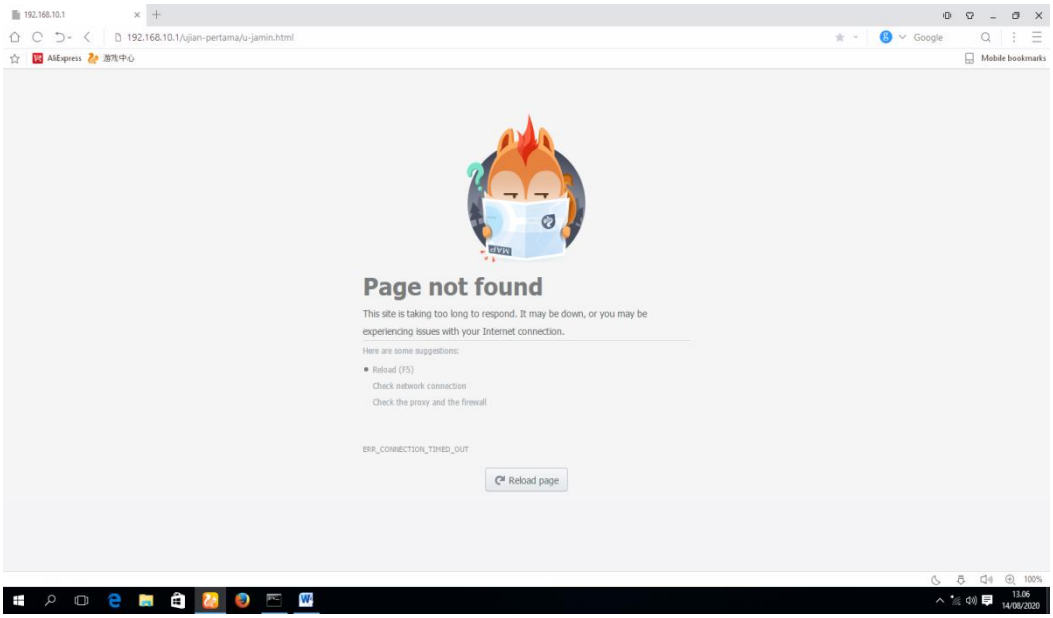
Gambar 4.10 CPU *usage* setelah ada serangan *ddos attack*

Beban kerja CPU semakin meningkat setalah adanya serangan *ddos attack* dan juga pada *local area connection* presentasi kinerja jaringan meningkat *trafficknya* mencapai 25% hingga 50% dikarenakan setelah dilakukan serangan *ddos attack* lalulintas jaringan semakin sibuk karena dibanjiri *request time out* yang masuk dan banyaknya paket data yang masuk. Seperti pada gambar 4.11 performa *networking* setelah adanya serangan *ddos attack*.



Gambar 4.11 *Performance networking* setelah adanya serangan *ddos attack*

Setelah serangan *ddos attack* berhasil masuk kedalam sistem jaringan serverpun menjadi sibuk dengan terbukanya *port-port* secara *public* pada server menjadikan bebasnya serangan masuk sehingga pengguna lain *client* yang ingin masuk kejaringan untuk melakukan ujian jamin tidak bisa terhubung ke server, selain itu pengguna yang sudah terhubung pun *web browsernya* menjadi lambat dan tidak bisa terhubung, seperti pada gambar 4.12 halaman web browser tidak bisa terhubung.



Gambar 4. 1 Halaman web browser tidak bisa terhubung

Pada hasil pengujian serangan jaringan sebelum digunakannya *honeypot* dan *port knocking* didapatkan sistem jaringan menjadi tidak stabil halaman web yang terhubung ke server jaringan pun menjadi *down* akibat banyaknya paket *byte* data dan *request* yang masuk pada server jaringan, berikut tabel 4.3 Perbandingan kinerja CPU dan jaringan pada saat adanya serangan *ddos attack request flooding* dan *ddos attack traffic flooding.*

Tabel 4.3 Hasil *performance* CPU dan jaringan saat adanya serangan *ddos attack request flooding* dan *ddos attack traffic flooding*

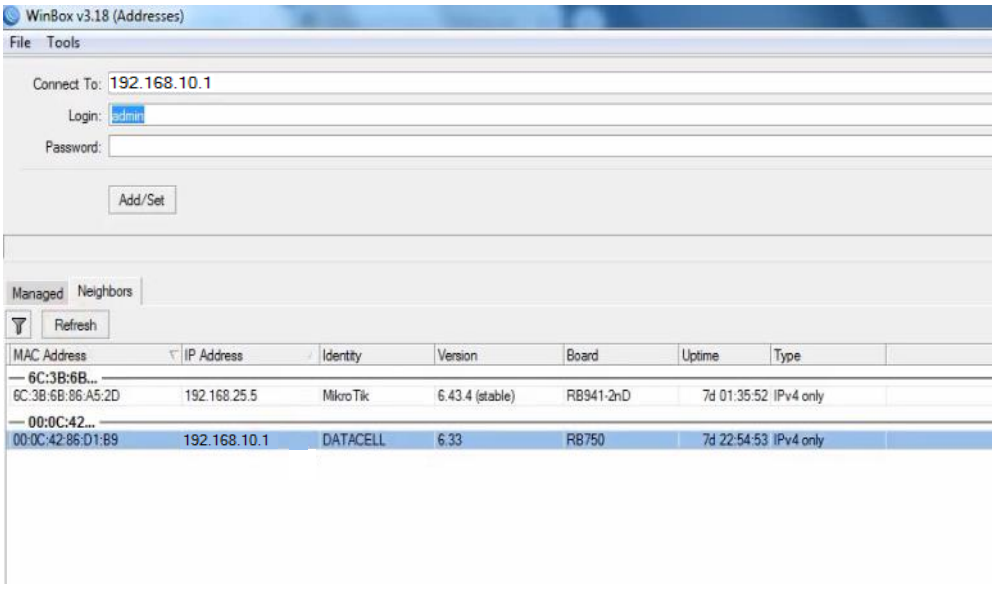
Pengujian	***Rata-rata performance***	**Sebelum serangan**	**Setelah serangan**
Uji 1	*CPU Usage History*	5%-12%	71%-85%
	Networking History LAC	1%-5%	20%-25%
Uji 2	*CPU Usage History*	5%-23%	75%-90%
	Networking History LAC	1%-5%	20%-30%
Uji 3	*CPU Usage History*	5%-16%	70%-89%
	Networking History LAC	1%-5%	20%-25%
Uji 4	*CPU Usage History*	5%-21%	74%-96%
	Networking History LAC	1%-5%	25%-45%
Uji 5	*CPU Usage History*	5%-18%	70%-92%
	Networking History LAC	1%-5%	25%-35%

Dari hasil analisis tabel 4.3 terlihat bahwa adanya perbedaan yang sangat signifikan dari hasil kinerja *performance CPU usage history* dan *networking* sebelum serangan dan setalah adanya serangan *ddos attack request loic* dan *ddos attack traffic hoic* selama 5 kali pengujian, terlihat pada pengujian pertama hasil rata-rata *performance CPU usage history* sebelum serangan 5%-12% setelah adanya serangan *CPU usage history* 71%-85%, pengujian kedua *performance CPU usage history* sebelum serangan 5%-23% setelah serangan 75%-90%, pengujian ketiga *performance CPU usage history* sebelum serangan 5%-16% setelah serangan 70%-89%, pengujian keempat *performance CPU usage history* sebelum serangan 5%-21% setelah serangan 74%-96% dan pengujian kelima *performance CPU usage history* sebelum serangan 5%-18% setelah serangan 70%-92%. Sedangkan pada hasil rata-rata *networking history local area connection (LAC)* sebelum adanya serangan hanya mencapai 1%-5% dari lima kali pengujian semua user *client* masih terhubung pada komputer server jaringan untuk melakukan ujian jamin tidak ada kenaikan yang signifikan pada kinerja sistem jaringan namun setelah adanya serangan *ddos attack request flooding* dan *ddos attack traffic flooding* kinerja *performance networking history LAC* meningkat pada pengujian serangan pertama rata-rata *performance networking history LAC* mencapai 20%-25%, pengujian kedua rata-rata *performance networking history LAC* mencapai 20%-30%, pengujian ketiga rata-rata *performance networking history LAC* mencapai 20%-25%, pengujian keempat rata-rata *performance networking history LAC* mencapai 25%-45% dan pengujian kelima rata-rata *performance networking history LAC* mencapai 25%-35%.

Dari hasil analisis *performance CPU usage* dan *networking* setelah terjadi adanya serangan *ddos attack* komputer server bekerja semakin berat dalam melayani *client* pengguna jaringan lain, sistem jaringan menjadi *down* bahkan *browser web* tidak bisa dibuka untuk mengakses aplikasi web yang terhubung ke server sehingga ujian jamin berbasis *offline* tidak bisa untuk dilaksanakan.

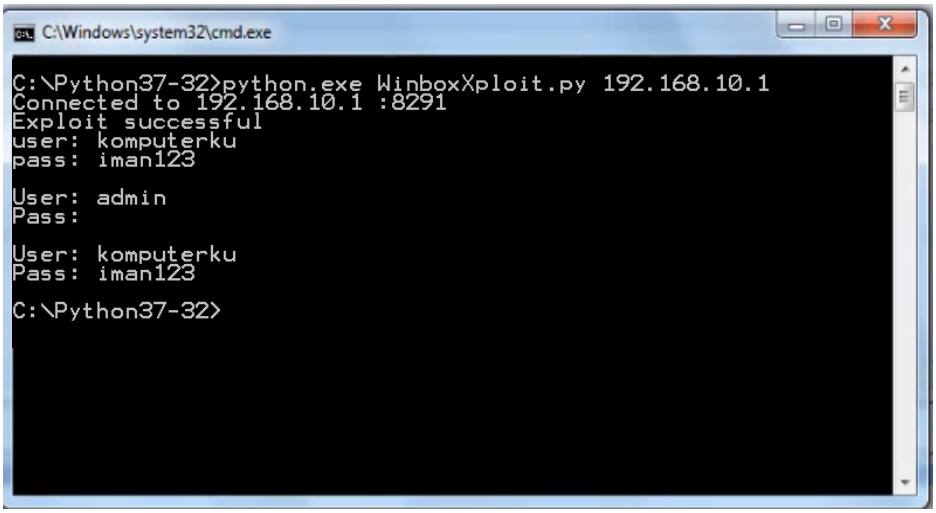
4.1.1.2 Serangan *Ddos Brute Force Attack*

Serangan *ddos brute force attack* merupakan teknik penyerang untuk dapat masuk kedalam sistem jaringan dengan cara mendapatkan *password* pada *microtik* yang digunakan untuk mengatur server jaringan, pada penelitian ini *soffware* yang dugunakan adalah *phyton* versi 3.8.5 dan dikonvigurasikan dengan *script winbox exploit*. Setelah selesai menginstal *phyton* selanjutnya *ekstrak* file *script winbox* kedalam folder *phyton* lalu buka *software winbox* dan masukan *ip address microtik* target serangan, seperti pada gambar 4.13 *ip address microtik* target.



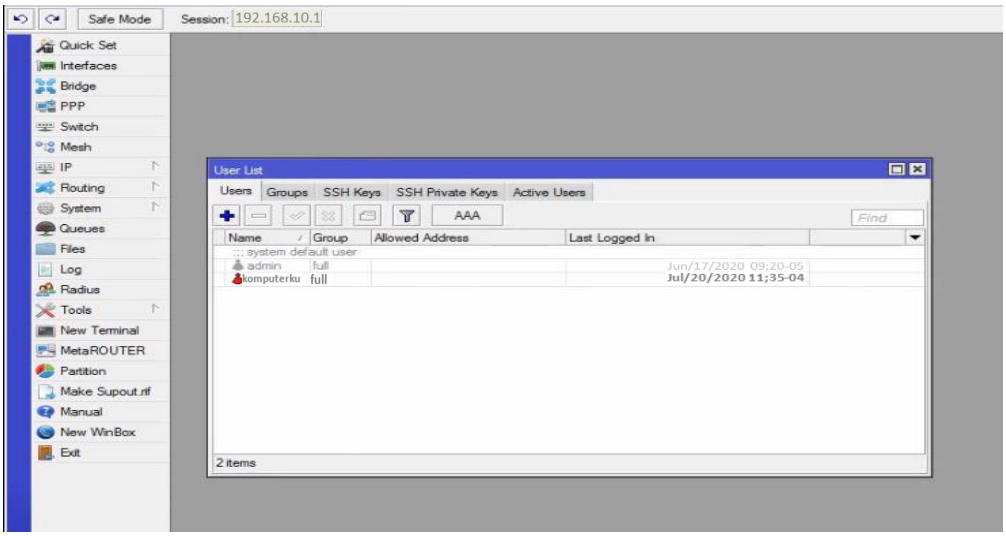
Gambar 4.13 *ip address microtik* target

Setelah didapatkan *ip address* yang akan menjadi target serangan selanjutnya proses *scanning password* dan *user login* pada *command prompt* dengan kode *python.exe winboxexploit.py 192.168.10.1* dan hasilnya *user* dan *password* dapat terdeteksi, seperti pada gambar 4.14 hasil *exploit password microtik.*



Gambar 4.14 Hasil *exploit password microtik*

Pada proses *exploit password* dan *user* didapatkan *password* = iman123 sedangkan *user* = komputerku berhasil terdeteksi yang selanjutnya akan kita coba *login* pada *microtik* menggunakan *winbox*, seperti pada gambar 4.15 *login* hasil *exploit winbox*.



Gambar 4.15 *Login* hasil *exploit password winbox*

User dan *password* ternyata bisa digunakan untuk masuk ke *microtik* melalui *winbox*, setelah berhasil masuk maka penyerang bisa mengatur server jaringan sesuai dengan keinginanya, karena *microtik* pada saat ini belum dilakukan *setting* untuk keamanan dari serangan *ddos attack*, berikut tabel 4.4 hasil serangan *ddos brute force attack.*

Tabel 4.4 Hasil serangan *ddos brute force attack*

NO	Target serangan	Status serangan	Hasil serangan
Uji 1	FTP *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
Uji 2	FTP *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
Uji 3	FTP *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
	SSH *brute force*	Router	Sukses menampilkan

	attack	memproduksi log	*username* dan *password router microtik*
Uji 4	FTP *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
Uji 5	FTP *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	Router memproduksi log	Sukses menampilkan *username* dan *password router microtik*

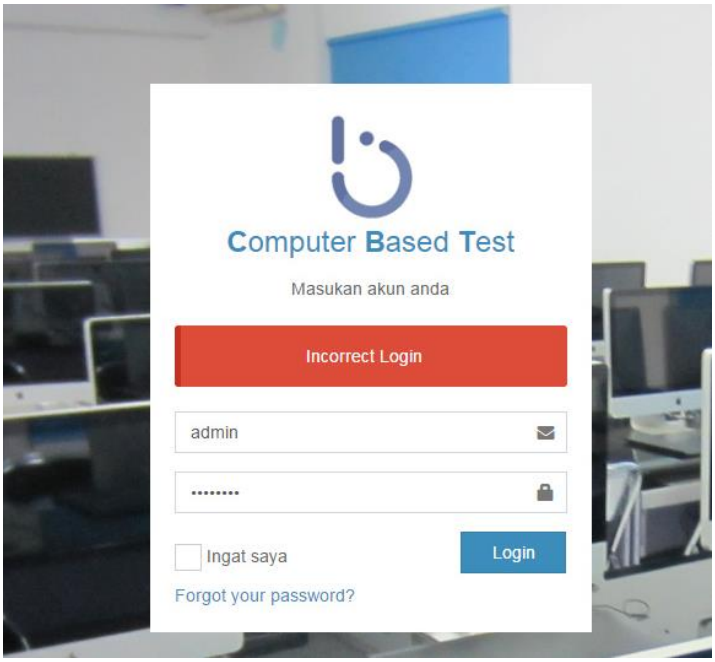
Pada pengujian serangan *ddos brute force attack* dilakukan 5 kali pengujian serangan pada server jaringan hasilnya sama, pada *microtik* yang menggunakan *protocol* FTP ataupun SSH dalam melayani *client-server* pada *router microtik* terdapat *port* yang masih terbuka secara publik sehingga *username* dan *password* pada *microtik* dapat ditampilkan pada saat proses *script winbox exploit*, *router microtik* pun mendeteksi adanya log aktifitas *user* pada saat *exploitasi microtik* berlangsung namun *admin* tidak dapat melakukan pencegahan secara langsung jika penyerang sudah mendapatkan *username* dan *password login* pada *microtik*.

4.1.1.3 Serangan *Ddos Attack SQL Injection*

Ddos attack SQL Injection merupakan teknik bagaimana seorang penyerang bisa mengakses *database* beserta dengan tabelnya, ketika sudah mengetahui table databasenya maka penyerang dapat mengetahui struktur isi table untuk mendapatkan *username admin* atau *password* agar bisa

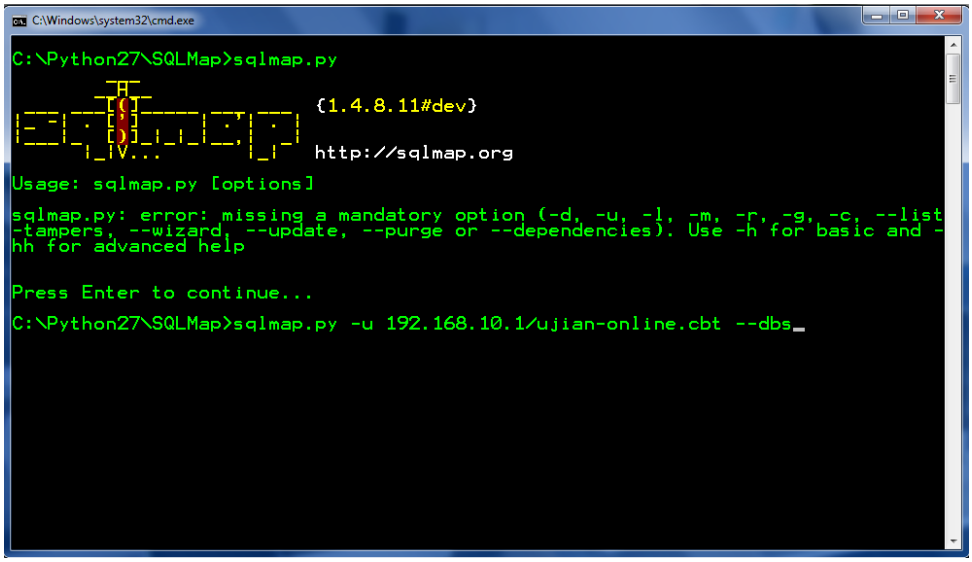
masuk kedalam aplikasi web untuk menguasai akses aplikasi tersebut seluruhnya.

Perangkat *software* yang dibutuhkan dalam *SQL injection* ini yaitu *python27* dan *skript sqlmap* setelah proses *instalasi* selesai selanjutnya peneliti akan mencoba melakukan penyerangan pada aplikasi berbasis web *computer based test (CBT)* seperti pada gambar 4.16 halaman *login* aplikasi.



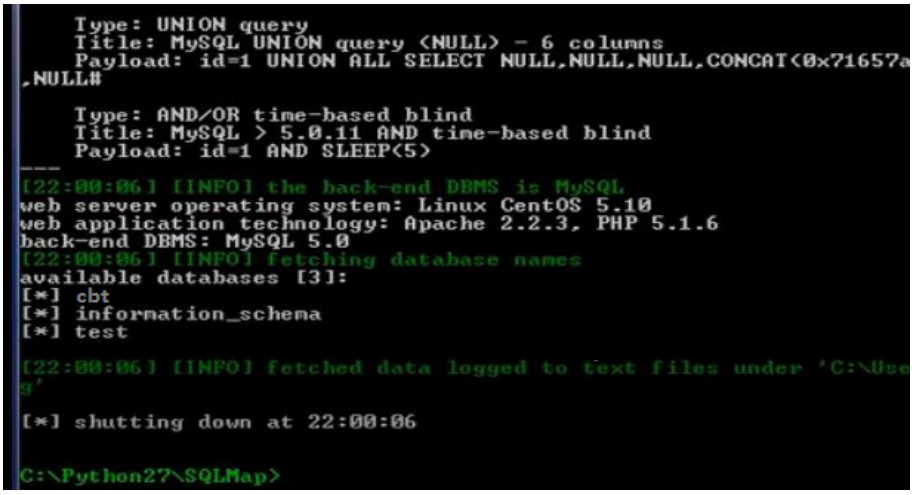
Gambar 4.16 Halaman *login* aplikasi

Pada gambar diatas penyerang tidak bisa akses untuk masuk kedalam aplikasi tersebut status *incorrect login* karena *username* dan *password* yang dimasukan tidak sesuai untuk itu dilakukan *SQL injection* untuk dapat mengatahui *username* dan *password* yang digunakan pada aplikasi web tersesbut, seperti pada gambar 4.17 proses *SQL injection sqlmap.*



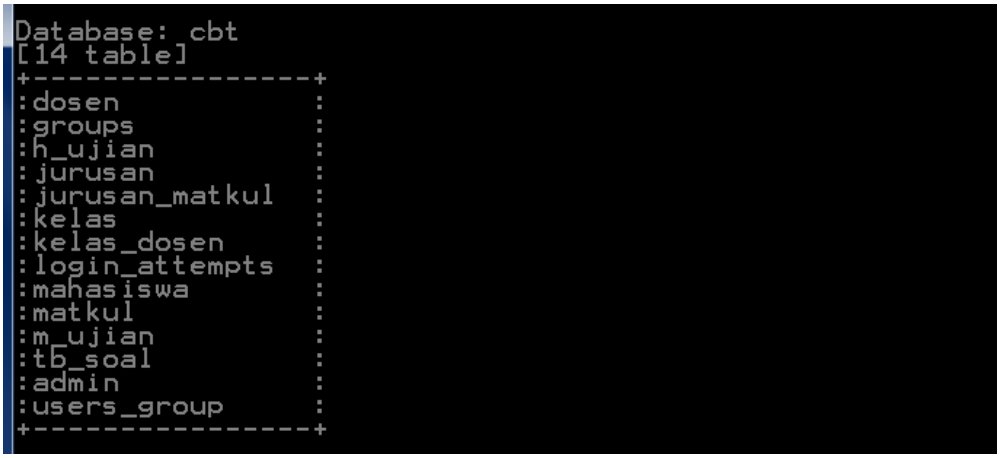
Gambar 4.17 Proses *SQL injection sqlmap*

Pada tahap ini penyerang akan melakukan *exploitasi database* untuk menampilkan struktur *database* yang digunakan pada aplikasi web ini dengan alamat server URL 192.168.10.1/ujian-online-cbt, selanjutnya tambahkan "sqlmap.py –u" sebelum alamat url dan "--dbs" pada akhir alamat url untuk mengetahui daftar basis data yang digunakan pada aplikasi web tersebut, seperti pada gambar 4.18 hasil *exploitasi database*



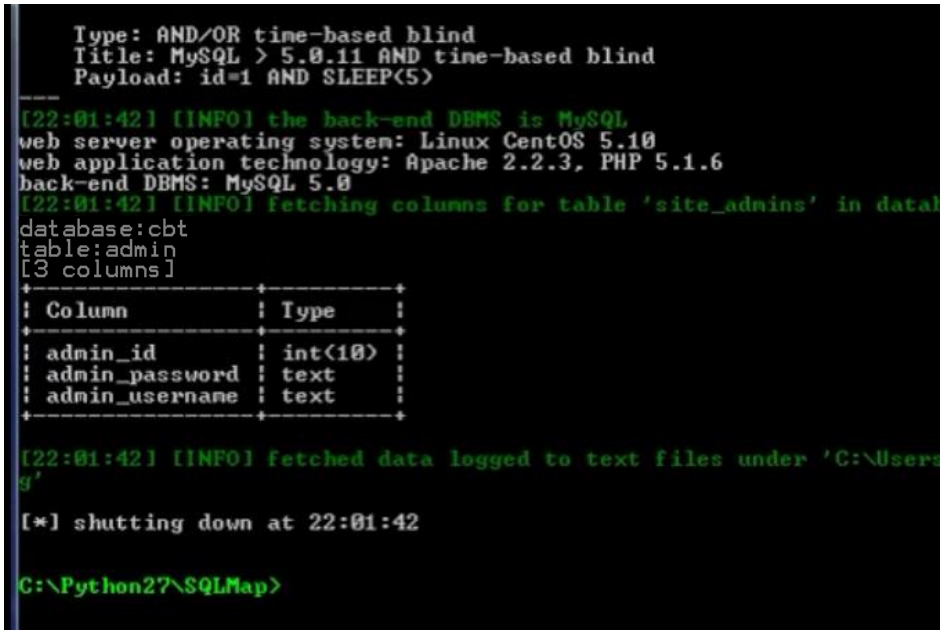
Gambar 4.18 Hasil *eksploitasi database* dengan *sqlmap*

Setelah *eksploitasi database* selesai ditemukan ada 3 *database* yaitu *cbt*, *information schema* dan *test* untuk melihat *table admin* selalu berada pada *database* yang pertama yaitu *cbt*, selanjutnya ketikan "sqlmap.py –u 192.168.10.1/ujian-online-cbt –D cbt --table" untuk melihat table berikut hasilnya seperti pada gambar 4.19 *struktur table cbt*.



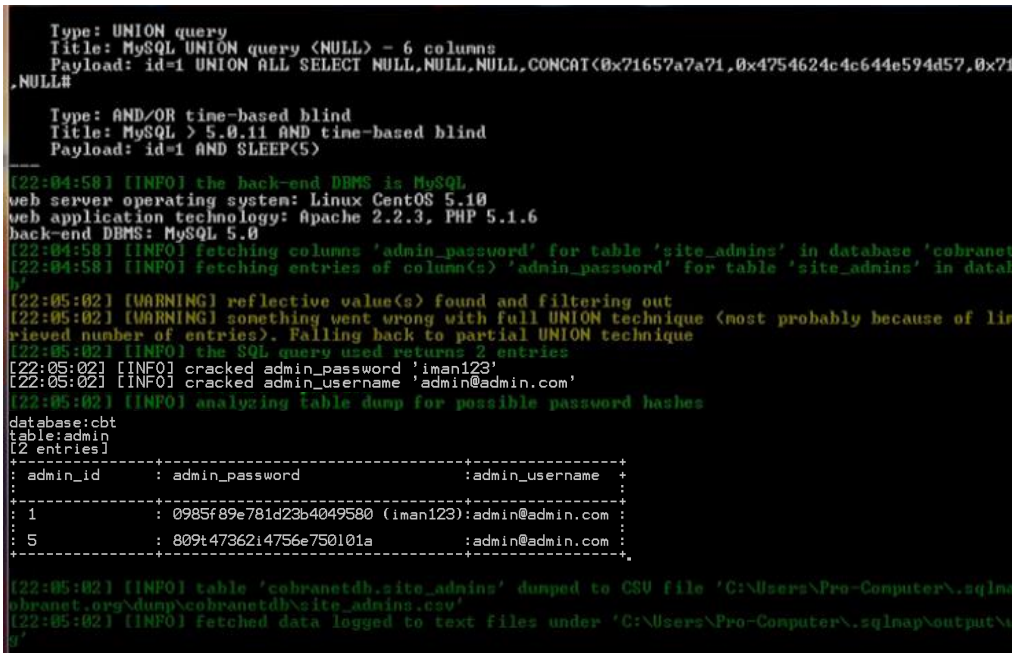
Gambar 4.19 *Struktur table cbt*

Ditemukan ada 14 table pada database cbt selanjutnya penyerang akan melihat struktur *columns* dari table admin agar bisa melihat *username* dan *password* ketikan "sqlmap.py –u 192.168.10.1/ujian-online-cbt –D cbt –T admin --columns" maka akan muncul struktur *columns* dari table admin seperti pada gambar 4.20 struktur *columns admin*.



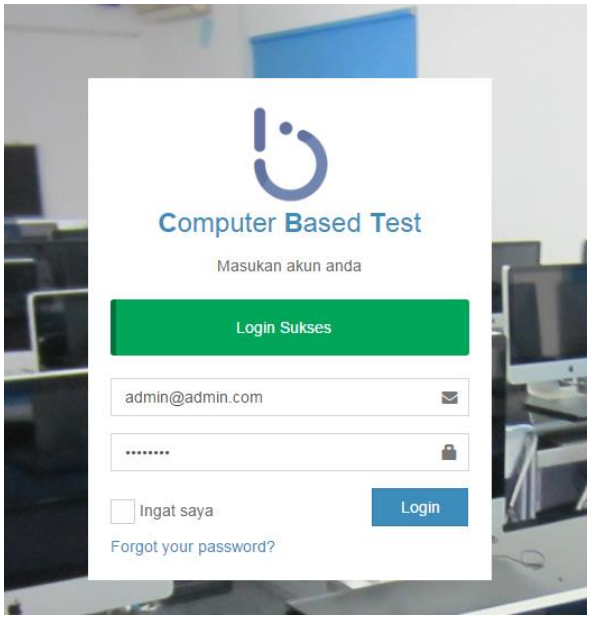
Gambar 4.20 *Struktur columns admin*

Pada struktur table *admin* terdapat 3 *column* yaitu admin_id dengan type *integer*, *admin_password* type *text* dan *admin username* type teks. Setelah ditemukan *column* selanjutnya adalah melihat isi dari *admin_username* masukan pada *command prompt* "sqlmap.py –u 192.168.10.1/ujian-online-cbt –D cbt –T admin –C admin_password, admin_username --dump" maka akan muncul *password* dan *usename* dari aplikasi web tersebut, seperti pada gambar 4.21 *password* dan *username admin*.



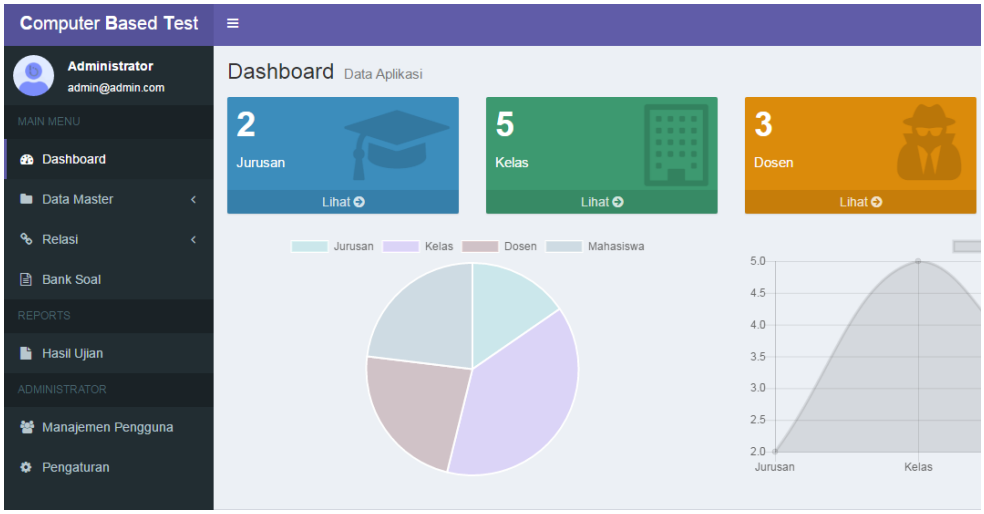
Gambar 4.21 *password* dan *username admin*

Tahap terakhir didapatkan hasil *password* dan *username* dari aplikasi web tersebut dengan *password* ‘iman123’ sedangkan *username* ‘admin@admin.com’ yang selanjutnya akan digunakan penyerang untuk dapat masuk kedalam aplikasi, seperti pada gambar 4.22 *login* aplikasi cbt.



Gambar 4.22 *login* aplikasi cbt

Pada pengujian penelitian ini penyerang sudah berhasil masuk kedalam aplikasi web tersebut dengan status berhasil masuk sebagai *admin* dan mempunyai akses penuh terhadap aplikasi tersebut, berikut pada gambar 4.23 tampilan halaman *admin*.



Gambar 4.23 Tampilan halaman *admin*

Kini penyerang bebas untuk mengakses data dan lain sebagainya dalam aplikasi web ini karena memang belum ada pengamanan khusus terhadap jaringan server, sehingga penyerang bebas dan berhasil melakukan *ddos attack sql injection* pada aplikasi web ujian *online* berbasis *offline* ini, berikut tabel 4.5 analisis hasil serangan *SQL Injection*.

Tabel 4.5 Hasil *SQL Injection*

NO	Jenis *eksploitasi*	Hasil *SQL Injection*		Keterangan
		Berhasil	Gagal	
1	Database	√	-	Terdapat 3 database yaitu cbt, information_schema dan test
2	Table	√	-	Terdapat 14 table yaitu table dosen, groups, h_ujian, jurusan, jurusan_matkul, kelas, kelas_dosen, login_attempts, mahasiswa, matkul, m_ujian, tb_soal, admin dan users_group
3	Column	√	-	Terdapat ada 3 yaitu admin_id, admin_passwword dan admin_username
4	Entries	√	-	Terdapat 2 entries yaitu *password* (iman123) dan *username* (admin@admin.com)

Dari hasil *exploitasi database* yang dilakukan oleh penyerang menggunakan teknik *SQL Injection sqlmap* terdapat beberapa struktur *database*, *table* dan *coloumn* yang ditampilkan yang selanjutnya didapatkan *password* dan *username admin* yang akan digunakan untuk masuk kedalam aplikasi web tersebut setelah tervalidasi dan dicoba masuk ternyata penyerang berhasil masuk kedalam aplikasi web dengan menggunakan *username admin*.

Setelah semua serangan berhasil dilakukan karena dalam pengujian penelitian ini pada tahap awal yaitu server jaringan belum menggunakan sistem

keamanan jaringan sehingga sangat rentan untuk dilakukan serangan oleh pengguna yang tidak bertanggung jawab. Berikut daftar serangan *ddos attack* yang berhasil dilakukan oleh penyerang pada tabel 4.6 serangan *ddos attack* sebelum menggunakan *honeypot* dan *port knocking*.

Tabel 4.6 Hasil pengujian jaringan sebelum menggunakan *honeypot* dan *port knocking*

NO	Jenis serangan	Proses *runing*		Keterangan
		Berhasil	Gagal	
1	*Ddos Attack request Flooding*	√	-	*Request* paket masuk setiap *and off conection* 2987428 *seconds*, 2735891 *seconds*, 2583947 *seconds*, 3310978 *seconds* dan 2872984 *seconds.*
2	*Ddos Attack Traffic Flooding*	√	-	*Traffick* paket data yang masuk 7.117052e+8 kb, 5.363787e+8 kb, 7.023877e+7 kb, 2.116352e+9 kb dan 1.080268e+9.
3	*Ddos Brute Force Attack*	√	-	Berhasil menampilkan *username* dan *password router microtik*
4	*Ddos Attack SQL injection*	√	-	Berhasil menampilkan *password* (iman123) dan *username* (admin@admin.com) pada aplikasi web

Semua hasil pengujian berhasil dilakukan serangan pertama yaitu pada ujian berbasis *ofline* dan serangan dilakukan bersamaan dengan url yang sama dengan alamat ip yang sama yaitu serangan *ddos attack request flooding* paket *request time out* yang masuk untuk melakukan *and off conection* 2987428 *seconds*, 2735891 *seconds*, 2583947 *seconds*, 3310978 *seconds* dan 2872984 *seconds* sedangkan pengujian serangan *ddos attack traffic flooding* paket berupa *byte* data yang masuk 7.117052e+8 kb, 5.363787e+8 kb, 7.023877e+7 kb, 2.116352e+9 kb dan 1.080268e+9, serangan ini berhasil melumpuhkan server

jaringan yang sedang berjalan melayani komputer *client* pada ujian jamin *offline* dengan jaringan *localhost*. Selanjutnya serangan *ddos brute force attack* dari hasil *exploit* yang dilakukan penyerang berhasil menampilkan *username* dan *password router microtik* sedangkan pada pengujian serangan yang terakhir yaitu *ddos attack SQL injection* dimana penyerang mencoba masuk pada aplikasi web menggunakan *user admin* dan hasilnya penyerang berhasil menampilkan *password* dan *username* untuk *login* ke dalam aplikasi tersebut.

4.1.2 Konfigurasi *Honeypot* dan *Port Knocking*

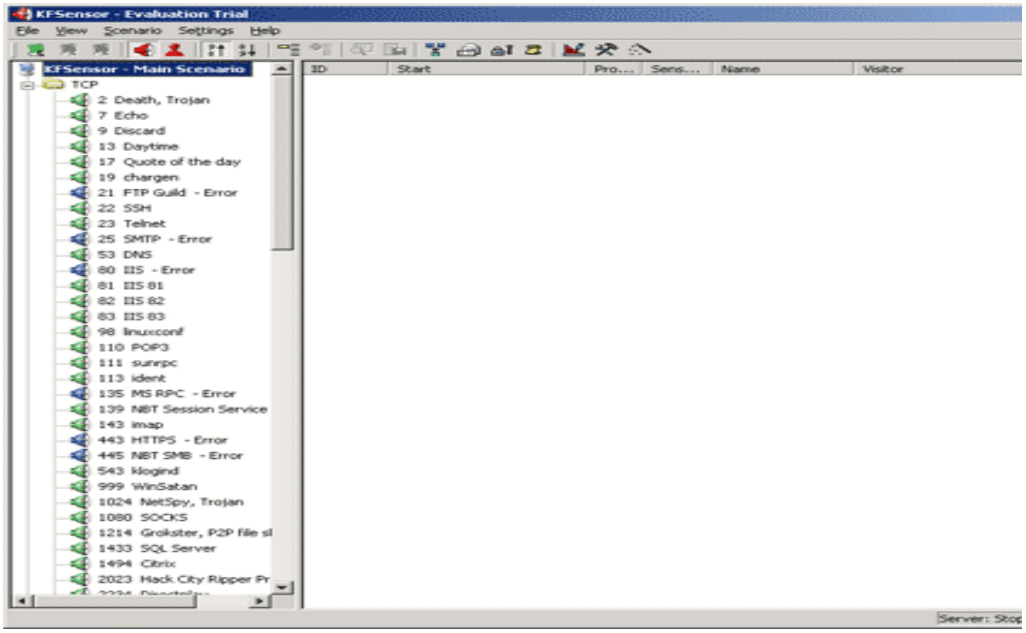
4.1.2.1 Konfigurasi *Honeypot* pada keamanan server jaringan

Dalam penelitian ini *software honeypot* yang akan digunakan adalah *KFSensor*, setelah *software* ini terinstal dan terpasang selanjutnya yaitu proses pengaturan konfigurasi pada sistem jaringan server. *KFSensor* memiliki beberapa persyaratan perangkat keras sederhana. Minimal membutuhkan *prosesor* 1 GHz, ruang *hard disk* 30 MB, dan RAM 128 MB. Pabrikan merekomendasikan prosesor 1,5 GHz, ruang *hard disk* 500 MB, RAM 512 MB, dan *database SQL*.

Setelah aplikasi terpasang dan menunjukkan bahwa *Setup* sudah selesai, selanjutnya adalah proses *konfigurasi KFSensor* pada server jaringan, sebagai berikut;

1. Menggunakan *KFSensor*

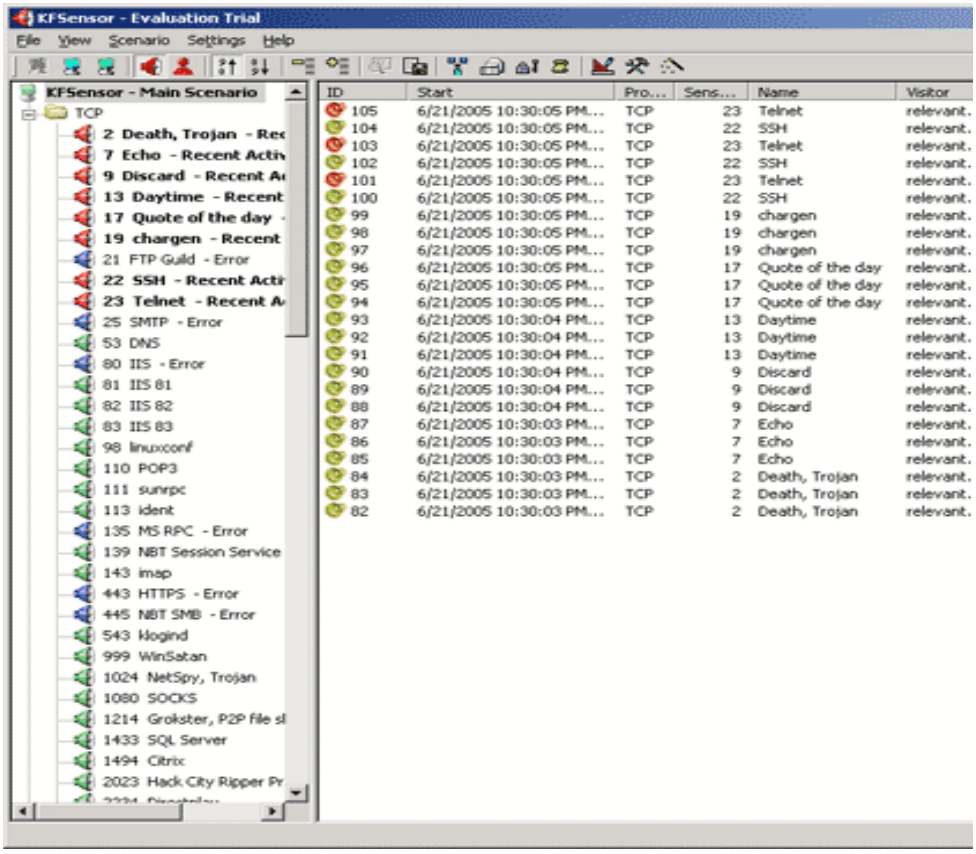
Saat *Setup wizard* ditutup, berikut tampilan *KFSensor* utama yang ditunjukkan pada Gambar 4.24 layar utama *KFSensor*.



Gambar 4.24 Layar utama *KFSensor*

Pada kolom di sebelah kiri berisi daftar nomor *port*, jika ikon di sebelah kiri daftar *port* berwarna hijau, itu berarti *KFSensor* secara aktif memantau port tersebut untuk mencari serangan. Jika ikon berwarna biru, itu berarti telah terjadi kesalahan dan *KFSensor* tidak mengawasi *eksploitasi* yang ditujukan ke *port* tertentu tersebut.

Setelah perangkat lunak aktif dan berjalan selanjutnya menguji perangkat lunak tersebut dengan meluncurkan pemindaian port terhadap mesin yang menjalankan *KFSensor*. Untuk pemindaian port menggunakan *utilitas shareware*. Ini hanya memindai blok alamat *ip*, mencari *port* terbuka. Pada gambar 4.25 menunjukkan bagaimana *KFSensor* bereaksi terhadap pemindaian port parsial.



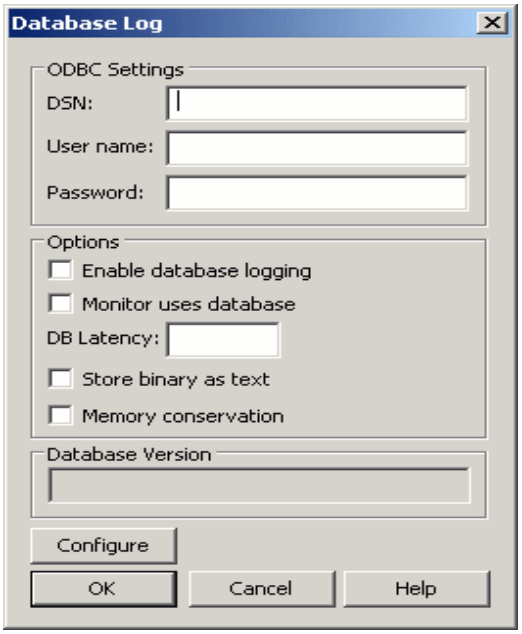
Gambar 4.25 *KFSensor* bereaksi terhadap pemindaian *port* parsial

Jika ikon sebelah kiri pada *port* yang dipindai berubah menjadi merah menunjukkan aktivitas terbaru. Ada ringkasan dari semua aktivitas yang terdeteksi di kolom sebelah kanan atau apliksi yang sedang berjalan pada komputer kita. Ikon di sebelah setiap entri diberi kode warna merah atau kuning sesuai dengan tingkat keseriusan aplikasi berjalan. Untuk melihat informasi yang lebih detail kita bisa klik pada aplikasi yang terdeteksi waktu mulai dan berakhir peristiwa aplikasi berjalan pada komputer, alamat *ip* mesin tempat aktivitas berasal, dan bahkan domain tempat mesin penyerang berada.

2. Membangun *log aktivitas*

Setelah *honeypot* dapat mendeteksi aktivitas dan dapat mencatat beberapa aktivitas selanjutnya yaitu *KFSensor* mencatat aktivitas ke *database*. Dengan begitu dapat mencatat aktivitas yang lebih permanen agar bisa mendeteksi serangan *DDOS attack* sehingga bisa memberikan keamanan pada server jaringan yang digunakan.

KFSensor melakukan aktivitas log secara *default* pada file teks dari pada file *database*, penting untuk menggunakan *database* adalah karena database umumnya memiliki kapasitas yang lebih tinggi dan lebih mudah untuk mencari apa yang dibutuhkan pada file teks. *Database* pilihan untuk *KFSensor* adalah SQL, pilih perintah *Log Database* dari menu Pengaturan konsol *KFSensor*. Setelah itu akan muncul kotak dialog yang ditunjukkan pada Gambar 4.26. Cukup isi bagian yang kosong untuk menghubungkan *KFSensor* ke *database*.

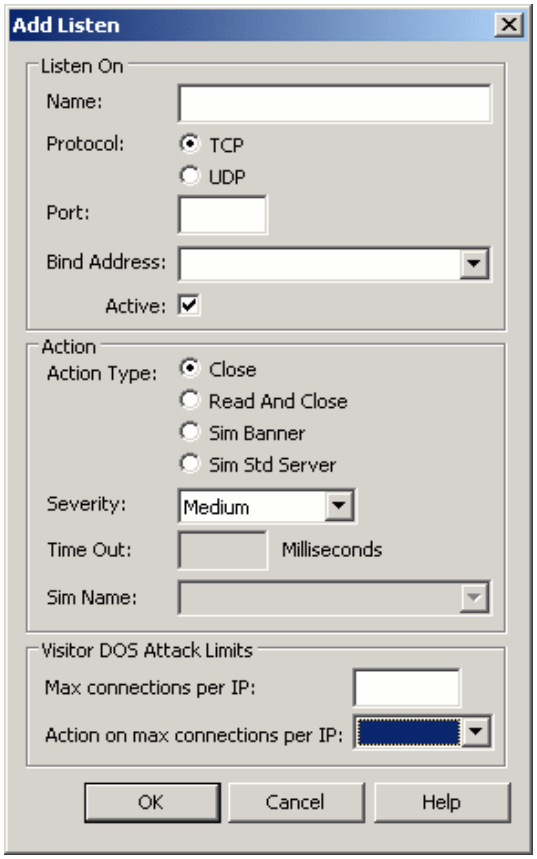


Gambar 4.26 Kotak dialog *Log Database*

3. Mengubah konfigurasi tindakan pada *honeypot KFSensor*

Untuk mengubah aturan, pilih perintah edit skenario aktif dari menu skenario. Setelah itu pilih kotak dialog yang berisi ringkasan dari semua aturan yang ada, klik edit untuk mengedit aturan atau klik tombol tambah untuk membuat aturan baru. Kedua prosedur tersebut bekerja dengan cara yang sama.

Klik tombol *Add* akan muncul kotak dialog *Add Listen*, ditunjukkan pada gambar 4.27 kotak dialog *add listen*.



Gambar 4.27 kotak dialog *add listen*

Beberapa bidang pilihan seperti *protocol*, *port*, dan *Bind Address*. Bidang ini memungkinkan kita memilih untuk aturan listen/mendengarkan.

Misalnya, kita dapat mengkonfigurasi aturan untuk *listen port TCP* pada alamat *ip* 192.168.10.2 bagian alamat mengikat dari aturan pada *Bind Address*.

Sekarang setelah kita menentukan listen, saatnya untuk mengkonfigurasi tindakan yang diambil aturan saat lalu lintas yang terdeteksi pada port yang ditentukan. Pilihan adalah *close, read and close, Sim Banner*, dan *Sim Std Server*. Opsi tutup memberi tahu aturan untuk hanya mengakhiri koneksi. *Opsi Sim Std Server* dan *Sim Banner* berkaitan dengan emulasi server. *Opsi Sim STD* server memungkinkan mengemulasi server yang lebih kompleks, seperti server IIS. Jika memilih untuk opsi sim kita harus mengisi nama simulator tepat di bawah pada *time out*.

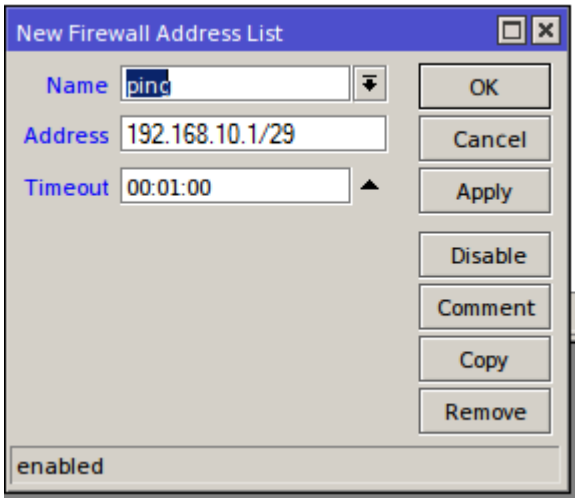
Bagian terakhir dari kotak dialog *add listen* adalah bagian serangan *DOS attack*. Bagian ini memungkinkan mencegah serangan *denial of service* terhadap *KFSensor*. dapat menentukan jumlah maksimum koneksi ke mesin per alamat *ip*, jika ambang terlampaui, kita dapat memilih untuk mengabaikan koneksi yang berlebihan atau dapat mengunci alamat *ip* yang menyerang.

4.1.2.2 Konfigurasi *port knocking* pada keamanan server jaringan

Pada konfigurasi *port knocking* perangkat yang digunakan adalah microtik setelah pengaturan jaringan pada server selesai selanjutnya iyalah pengaturan *port knocking*, sebagai berikut;

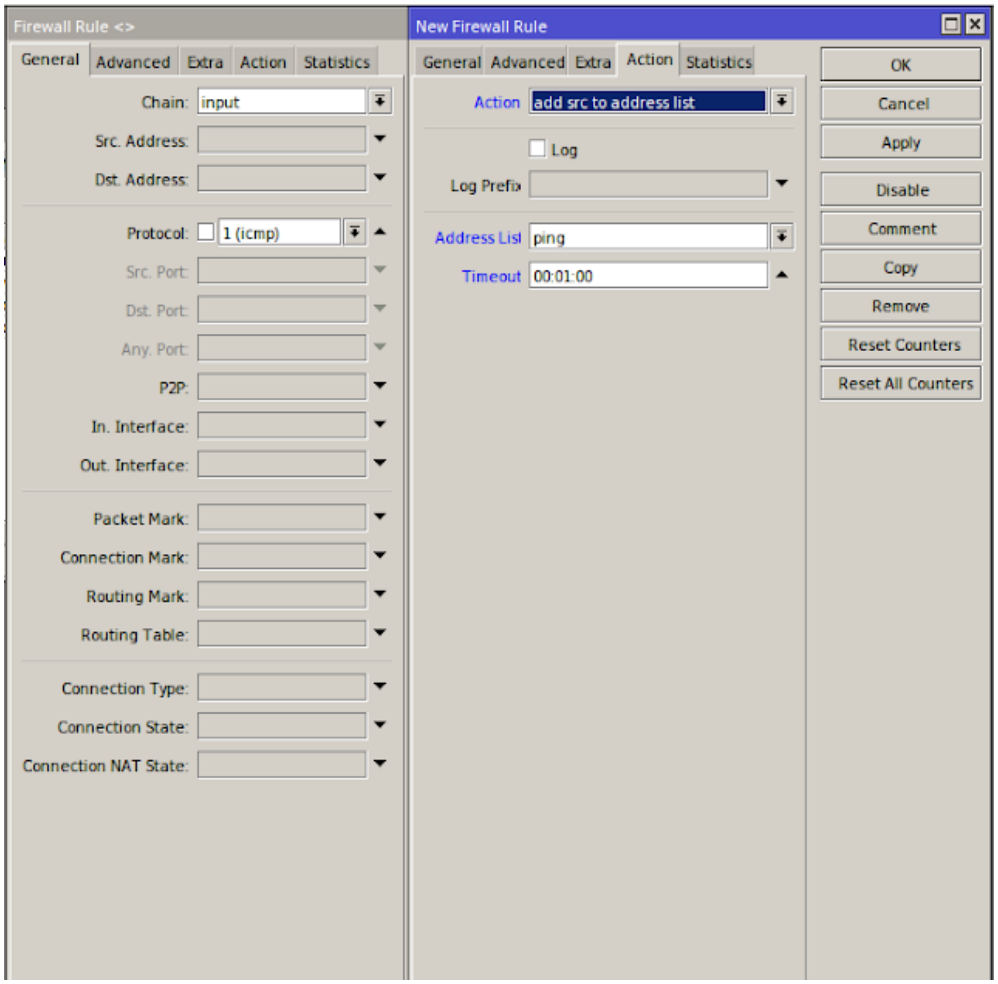
1. Memberi alamat pada *ip* yang di gunakan untuk *login* pada *address list* secara manual. Isikan pada kolom *name* nama group sesuai dengan

keinginan, sedangkan *addres* isikan sesuai dengan alamat yang akan diizinkan masuk oleh pengguna dan pada *timeout* adalah pengaturan lamanya pengguna atau alamat yang akan masuk. Seperti pada gambar 4.28 *New firewal addres list*.



Gambar 4.28 *New firewal addres list*

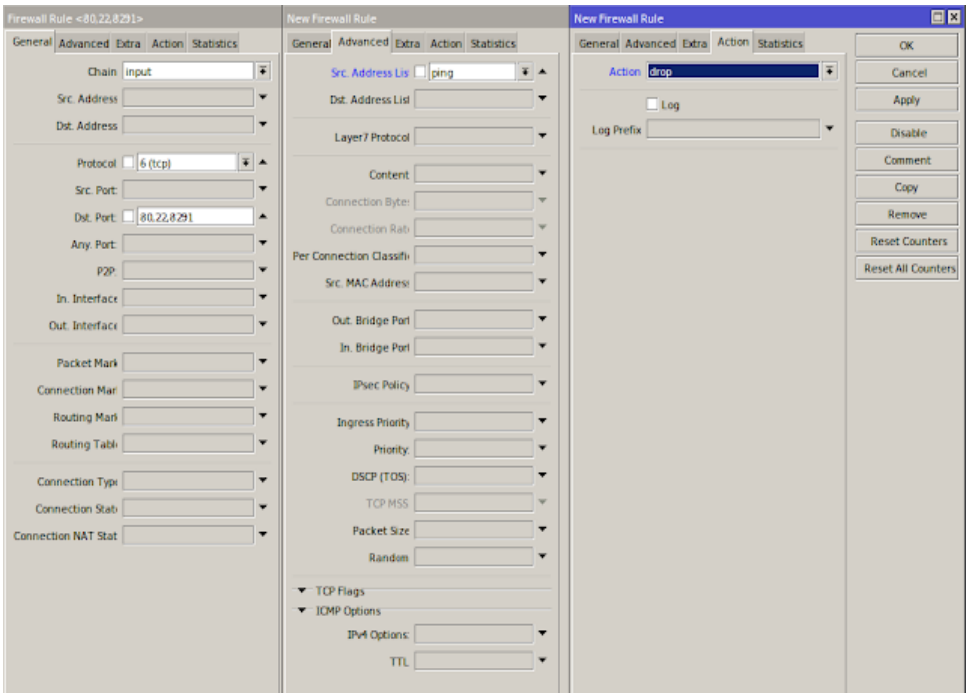
2. Selanjutnya menambahkan *rule* yaitu dengan memasukkan *ip address host* yang mengirimkan paket ICMP (ping) ke router ke dalam sebuah *address-list* secara otomatis yang sudah terdaftar pada *address-list* yang bisa akses winbox ke router dan lama waktu yang telah di tentukan. Isikan pada *general chain* yaitu *input, protokol* isikan *icmp, Action* isikan *add src to address list, Address list* isikan nama yang di buat di address list yaitu ping sedangkan timeout adalah lama waktu yang di yang di berikan jika nama dari ping tadi masuk, seperti pada gambar 4.29 *Firewall rule general*.



Gambar 4.29 *Firewall rule general*

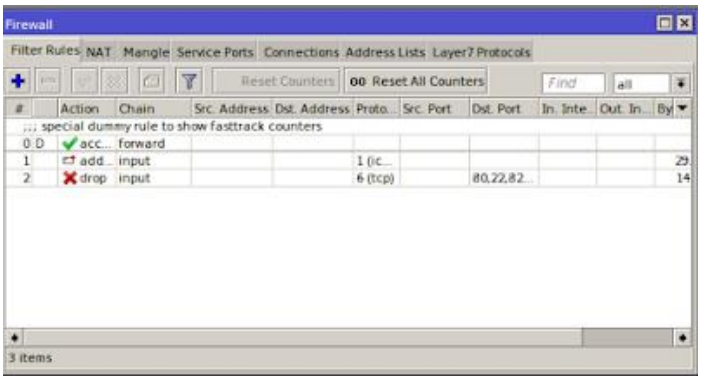
3. Membuat *rule firewall filter* untuk melakukan *blocking* akses winbox ke router dari sumber (*src-address*) selain dari *ip address* yang sudah terdaftar dalam *address-list* yaitu untuk melakukan aksi blokir jika ada serangan *ddos attack* dari alamat asing yang hendak masuk pada server jaringan. Berikut daftar isian pada *general*, *chain* isikan *input*, protokol isikan *icmp*, dst *port* isikan 80,22,21,8291 *port* ini adalah *port http,SSH FTP* dan *Winbox advanced.* Pada *src address list* isikan ping yaitu nama group yang di buat di *address list* sebelumnya pada gambar 4.28 sedangkan pada *action* isikan drop atau menutup *port* 80,22,21,8291

yang di gunakan *login* kecuali *ip* yang sudah terdaftar di *address list*, seperti pada gambar 4.30 *Firewall rule advanced*;



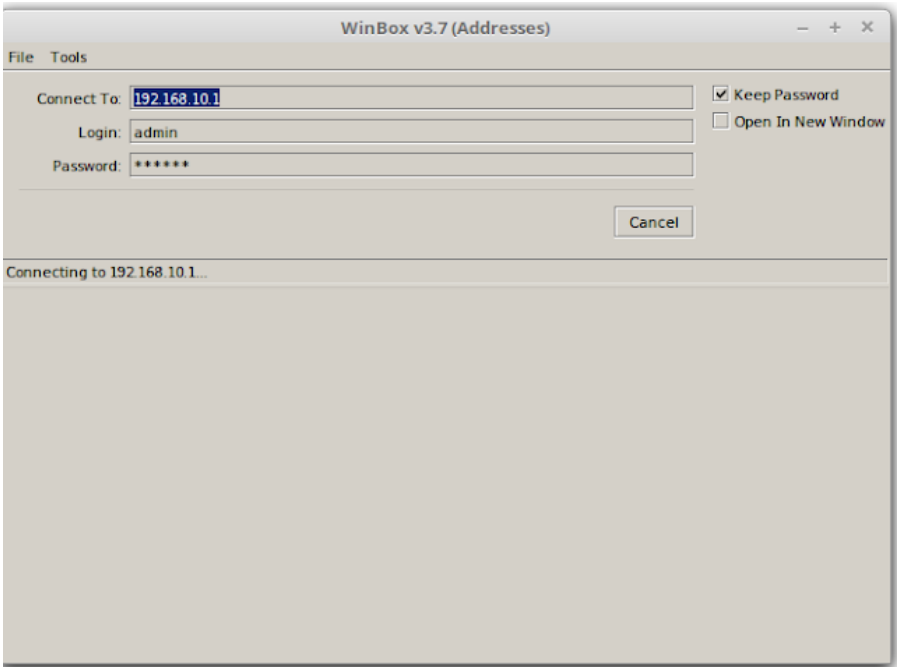
Gambar 4.30 *Firewall rule advanced*

4. Kemudian akan muncul tabel *firewall* seperti gambar 4.31 di bawah, dimana ada *ip* yang status pada *action* yaitu *acc* sedangkan status *chain forward* artinya ada trafik paket yang melewati router, sedangkan status *action* pada *add* dan *drop* yaitu *chainya input* yang artinya ada trafik paket yang masuk ke router, seperti pada gambar 4.31 *Tabel firewall*.



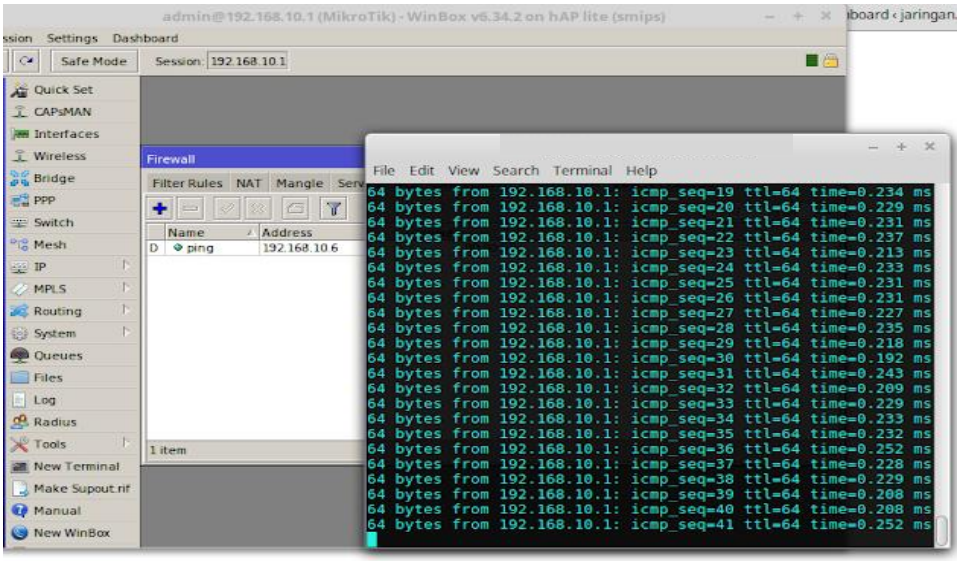
Gambar 4.31 *Tabel firewall*

5. Selanjutnya coba keluar dari *winbox* lalu masuk lagi dan hasilnya tidak bisa *connect* atau terhubung, jika ingin *login* harus melakukan ping terlebih dahulu dengan alamat *ip address* yang sudah dibuat.



Gambar 4.32 Menu *login* pada *winbox*

6. Kemudian *ping ip* yang telah terdaftar di *address list*, lalu coba *login* lagi dan hasilnya admin berhasilnya masuk pada *winbox mikrotik*.



Gambar 4. 2 Hasil *login winbox microtik*

Setelah admin berhasil masuk dengan menggunakan metode *port knocking* selanjutnya admin bisa melihat lalulintas jaringan yang dilewati user jika ada *ip* asing yang hendak menyusup admin bisa melakukan tindakan pencegahan dengan memblokir *ip* yang tidak dikenal yang sudah terdeteksi terlebih dahulu pada *honeypot KFSensor*.

Tabel 4.7 Hasil konfigurasi *honeypot* dan *port knocking* pada server jaringan

Jenis keamanan	**Hasil konfigurasi**		**Rule konfigurasi**	**Keterangan**
	Berhasil	**Gagal**		
Honeypot	√	-	Protocol TCP, port FTP 21 *ip* 192.168.10.2	Mengalihkan/menjebak serangan *ddos attack* dan merekam log aktifitas pada server jaringan
Port Knocking	√	-	Port webfig (tcp 80), SSH (tcp 22), FTP (tcp 21), winbox (tcp 8291) *ip* 192.168.10.1	Mengamankan port pada server jaringan

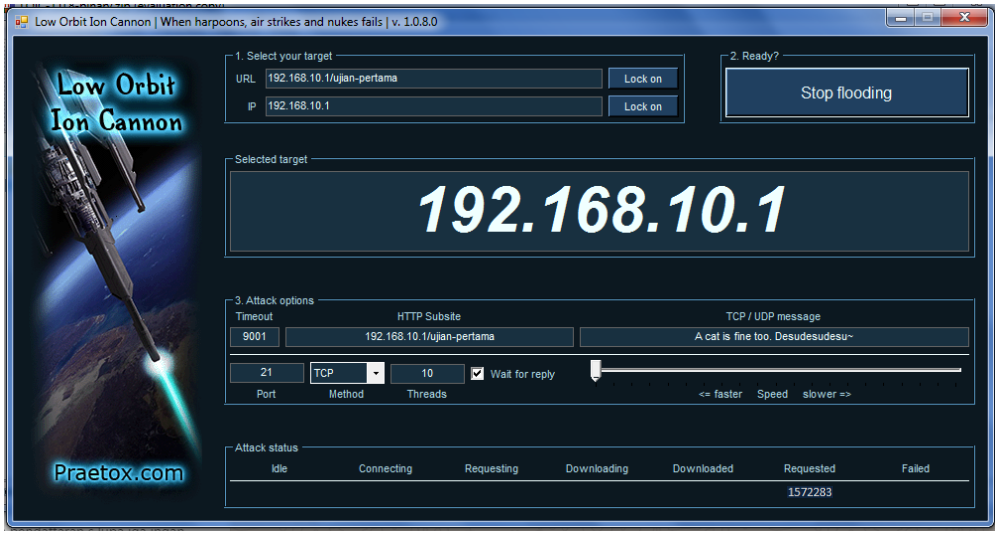
Fungsi dari *honeypot* diterapkanya pada keamanan server jaringan adalah untuk mendeteksi adanya serangan sekaligus memantau setiap aktivitas *user client* yang terhubung pada server jaringan, konfigurasi pada *honeypot* menggunakan *ip* 192.168.10.2 sebagai alamat untuk menjebak serangan *ddos attack, port* yang digunakan adalah FTP 21 protocol TCP karena sistem jaringan ini menggunakan *localhost area conenection,* sedangkan *port knocking* bertugas mengamankan *port-port* pada jaringan agar tidak mudah bagi *user* asing masuk kedalam sistem jaringan dan

untuk memblokir *user* yang tidak dikenal jika ingin mengganggu sistem jaringan, *port-port* yang diamankan adalah port 80,22,21 dan 8291.

4.1.3 Analisis serangan setelah menggunakan kemanan jaringan *Honeypot* dan *Port Knocking*

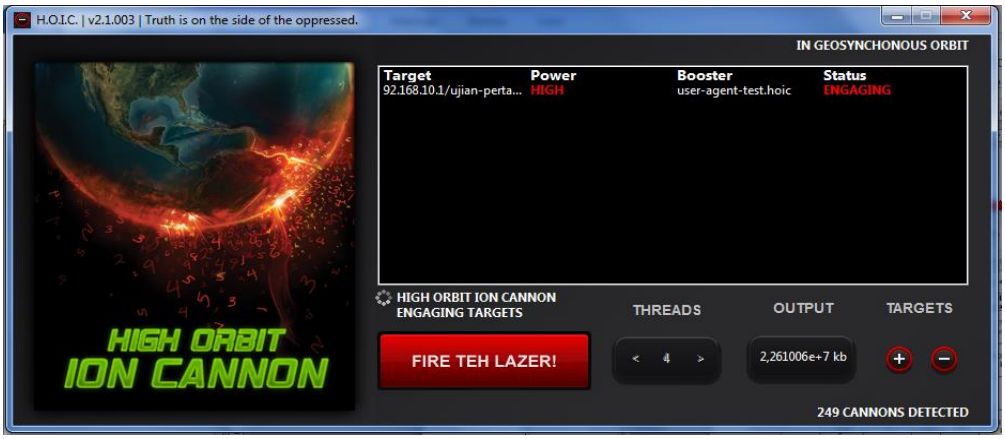
4.1.3.1 Mendeteksi serangan *ddos attack request flooding* dan *traffick flooding*

Setelah *honeypot* terpasang pada sever jaringan dan *port knocking* telah diatur konfigurasinya pada *microtik* untuk keamanan server jaringan, selanjutnya akan dilakukan penyerangan ulang agar dapat menganalisis hasil dari serangan setelah *honeypot* dan *port knocking* terpasang pada sistem jaringan server, berikut hasil serangan *ddos attack request flooding* pada gambar 4.34 Serangan setelah server jaringan terpasang *honeypot* dan *port knocking*.



Gambar 4.34 Serangan *ddos attack request flooding* setelah server jaringan terpasang *honeypot* dan *port knocking*

Dapat dilihat bahwa pada pengujian serangan setelah terpasangnya *honeypot* dan *port knocking request* berhasil masuk pada jaringan namun telah dialihkan dan masuk pada server *honeypot KFSensor* jadi serangan ini tidak sampai mengganggu server jaringan sehingga tidak mengganggu layanan *client-server* jaringan yang sedang melaksanakan ujian jamin. Bersamaan dengan serangan *request flooding* dilakukan juga pengujian serangan *ddos attack traffick flooding* dengan hasil seperti pada gambar 4.35 Serangan *ddos attack traffick flooding* setelah server jaringan terpasang *honeypot* dan *port knocking.*



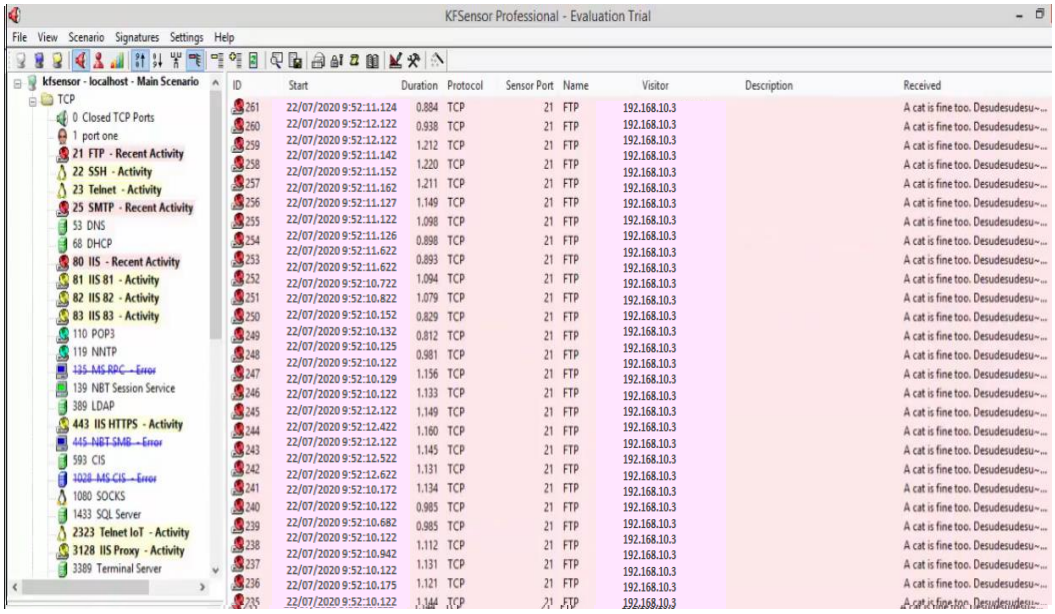
Gambar 4.35 Serangan *ddos attack traffick flooding* setelah server jaringan terpasang *honeypot* dan *port knocking*

Pada pengujian serangan *ddos attack traffick flooding* juga berhasil masuk pada jaringan berupa *byte* data yang juga terdeteksi dan dialihkan oleh *honeypot*, pengujian serangan *ddos attack request flooding* dan *traffick flooding* setelah terpasangnya *honeypot* dan *port knocking* dilakukan 5 kali pengujian dengan hasil serangan sebagai berikut, pada tabel 4.8 Hasil serangan *ddos attack request flooding* dan *traffick flooding* setelah terpasangnya *honeypot* dan *port knocking* pada server jaringan.

Tabel 4.8 Hasil serangan *ddos attack* setelah terpasangnya *honeypot* dan *port knocking*

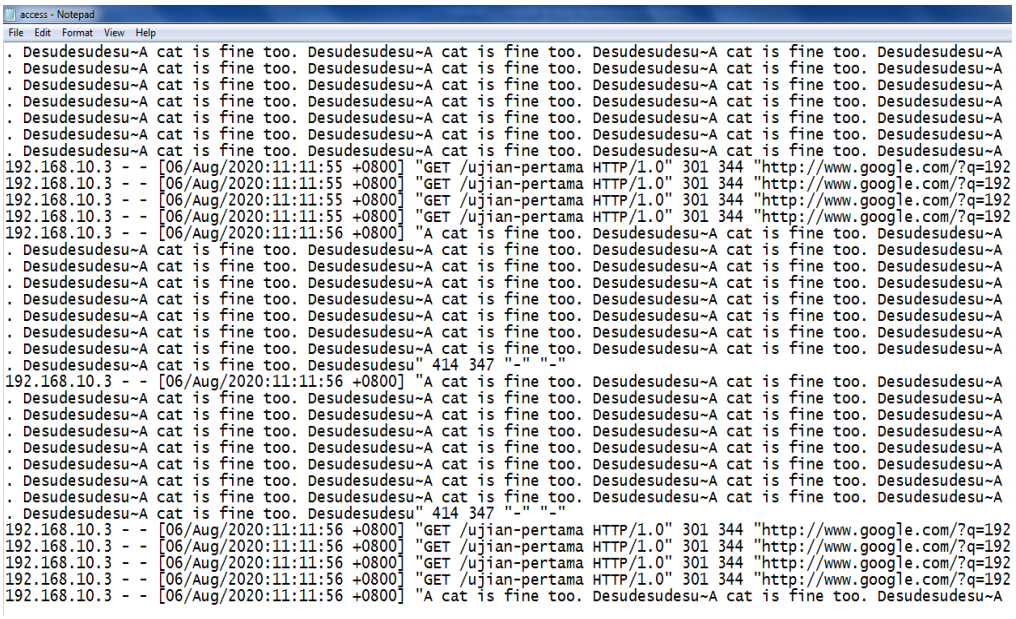
Pengujian	**Jenis serangan**	**Ip target/ port**	**Waktu**	**Level serangan**	**Hasil serangan**
Uji 1	*Ddos attack request flooding*	192.168.10.1/ 21	10 menit	*Faster*	1572283 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	10 menit	*High*	2.261006e+7 kb
Uji 2	*Ddos attack request flooding*	192.168.10.1/ 21	10 menit	*Faster*	1094759 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	10 menit	*High*	2.063770e+7 kb
Uji 3	*Ddos attack request flooding*	192.168.10.1/ 21	10 menit	*Faster*	1948750 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	10 menit	*High*	3.055146e+7 kb
Uji 4	*Ddos attack request flooding*	192.168.10.1/ 21	10 menit	*Faster*	1749371 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	10 menit	*High*	2.780476e+7 kb
Uji 5	*Ddos attack request flooding*	192.168.10.1/ 21	10 menit	*Faster*	1430156 *seconds*
	Ddos attack traffick flooding	192.168.10.1/ 21	10 menit	*High*	2.509194e+7 kb

Pengujian serangan dilakukan dengan lama waktu rata-rata 10 menit karena setelah serangan *ddos attack request flooding* dan *traffick flooding* terdeteksi oleh *honeypot* selanjutnya dilakukan pemblokiran *user* dengan *port knocking.* Hasil serangan *ddos attack request flooding* yang berhasil masuk pada server *honeypot* pengujian serangan pertama 1572283 *seconds,* serangan kedua 1094759 *seconds,* serangan ketiga 1948750 *seconds,* serangan keempat 1749371 *seconds* dan serangan kelima 1430156 *seconds.* Sedangkan pada pengujian serangan *ddos attack traffick flooding* paket *byte* data yang masuk ke server *honeypot* pada serangan pertama 2.910889e+6 kb, serangan kedua 2.063770e+7 kb, serangan ketiga 3.055146e+7 kb, serangan keempat 2.780476e+7 kb dan serangan kelima 2.509194e+7 kb, setiap serangan hasil data *request timout* dan *byte* data berbeda-beda jumlahnya dikarenakan tergantung pada kecepatan akses jaringan pada saat serangan dan juga kesiapan ruang pada *port* jaringan yang berbeda ketika serangan *ddos* masuk walaupun waktu pada saat serangan dilakukan sama. Serangan yang masuk pada server *honeypot* tidak sampai masuk pada komputer server jaringan sehingga tidak mengganggu proses ujian yang sedang berlangsung. Dalam penelitian ini penyerang menggunakan *ip* 192.168.10.3 untuk melakukan serangan ke *ip* server jaringan, berikut hasil serangan seperti pada gambar 4.36 *Honeypot KFSensor* deteksi serangan *ddos attack request flooding* dan *traffick flooding.*



Gambar 4.36 *Honeypot KFSensor* deteksi serangan *ddos attack request flooding* dan *traffick flooding*

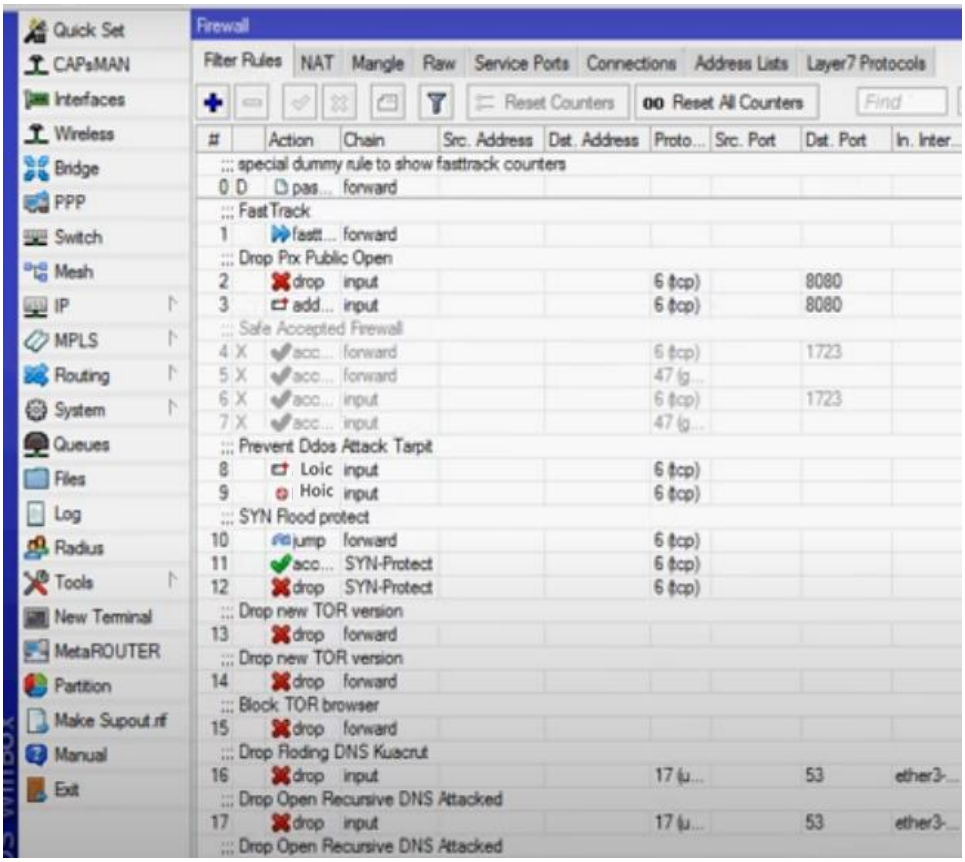
Serangan *ddos attack* yang masuk pada sistem jaringan berhasil dialihkan oleh *honeypot KFSensor* yang bekerja secara *realtime* sehingga serangan tidak sempat merusak pengguna lain yang sedang terhubung ke jaringan server untuk melakukan ujian jamin *offline*, analisa serangan yang masuk yaitu dari *port* 21 FTP dengan *ip address* 192.168.10.3 jenis serangan *ddos attack request flooding* dan *traffick flooding* dengan nama paket data *traffick* dan *request* yaitu *desudesudesu*, berikut gambar 4.37 hasil dari aktivitas paket yang masuk pada *logs acsess* server *xampp control panel.*



Gambar 4.37 *Logs acsess* server *xampp control panel*

Didapatkan hasil bahwa serangan yang masuk terdapat dua jenis serangan yaitu *ddos attack request flooding* dan *traffick flooding* yang mencoba masuk secara bergantian dengan tujuan ingin merusak server jaringan agar tidak bisa digunakan oleh pengguna *user* lain.

Serangan yang sudah terdeteksi oleh *honeypot kfsensor* selanjutnya dilakukan tindakan pemblokiran serangan pada server jaringan agar tidak terus menerus akan memberatkan komputer CPU *server* bekerja untuk melayani *client* yang sedang terhubung ke jaringan dengan menggunkan *microtik* dengan metode *port knocking* untuk mengakses *server microtik* dengan meode ketukan tertentu untuk bisa masuk ke *port* server jaringan yang sudah dilakukan konfigurasi, seperti pada gambar 4.38 Memblokir serangan *ddos attack* dengan *port knocking*.

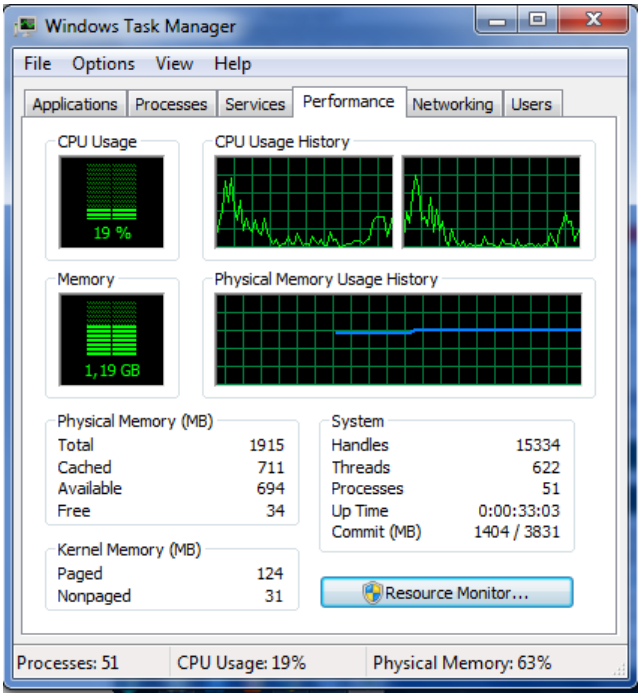


Gambar 4.38 Memblokir serangan *ddos attack* pada *port knocking*

Setelah masuk pada *microtik* dengan metode *port knocking* didapatkan telah masuk serangan *ddos attack request flooding* dengan menggunakan *software loic* dan *traffick flooding* menggunakan *software hoic* yang selanjutnya akan dilakukan pemblokiran pada *address list* agar serangan benar-benar dihentikan sehingga tidak lagi mengganggu sistem pada server jaringan.

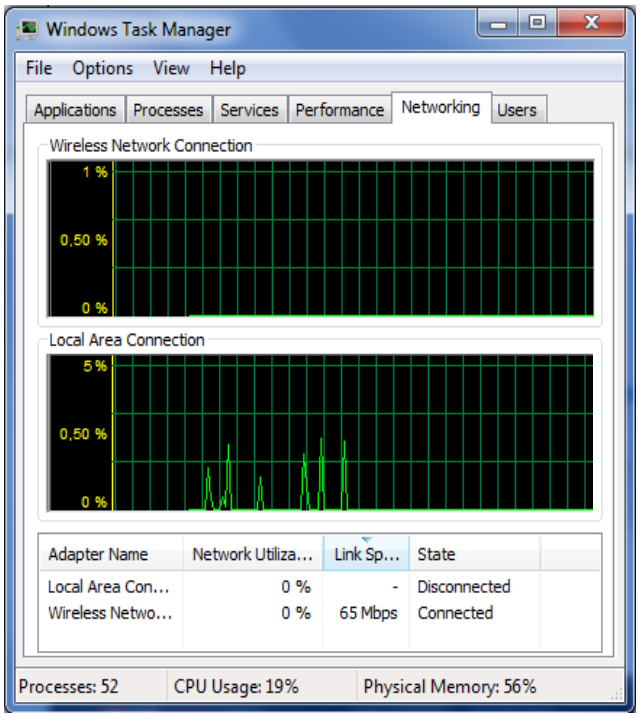
Dengan digunakannya *honeypot* untuk menjebak serangan *ddos attack* sebagai target maka CPU pada komputer serverpun bekerja normal seperti biasa karena semua *request* dan *traffick* paket data yang masuk pada jaringan tidak sampai kepada komputer server, namun telah disaring pada

komputer server *honeypot* yang bekerja terus menerus secara *realtime* selama sistem jaringan digunakan untuk melayani pengguna lain pada saat ujian *offline* berjalan. Berikut gambar 4.39 Hasil kerja CPU komputer server setelah adanya serangan *ddos attack* dan setelah adanya pengamanan server jaringan dengan *honeypo*t dan *port knocking*.



Gambar 4.39 *CPU usage history* seletah penerapan *honeypot* dan *port knocing*

Pada analisa kinerja komputer server *CPU usage history* presentasinya hanya mencapai 5%-19% yang artinya komputer server bekerja stabil walaupun adanya terjadi serangan *ddos attack* secara terus menerus selama sistem jaringan berjalan, berikut juga hasil dari analis pada *performance traffick* jaringan pada gambar 4.40 *Performance traffick local area connection.*



Gambar 4.40 *Performance trafick local area connection*

Dari hasil analisis kinerja jaringan *local area connection* setalah adanya serangan *ddos attack* didapatkan persentasinya mencapai 1%-5% kinerja sistem pada jaringan stabil dibandingkan pada saat sebelum digunakanya sistem keamanan jaringan *honeypot* dan *port knocking* traffiknya meningkat mencapai 20%-45%, berikut tabel 4.9 Analisis *performance CPU usage history* dan *networking history LAC* serangan setelah terpasang *honeypot* dan *port knocking*.

Tabel 4.9 Analisis hasil komputer server setelah terpasang *honeypot* dan *port knocking*

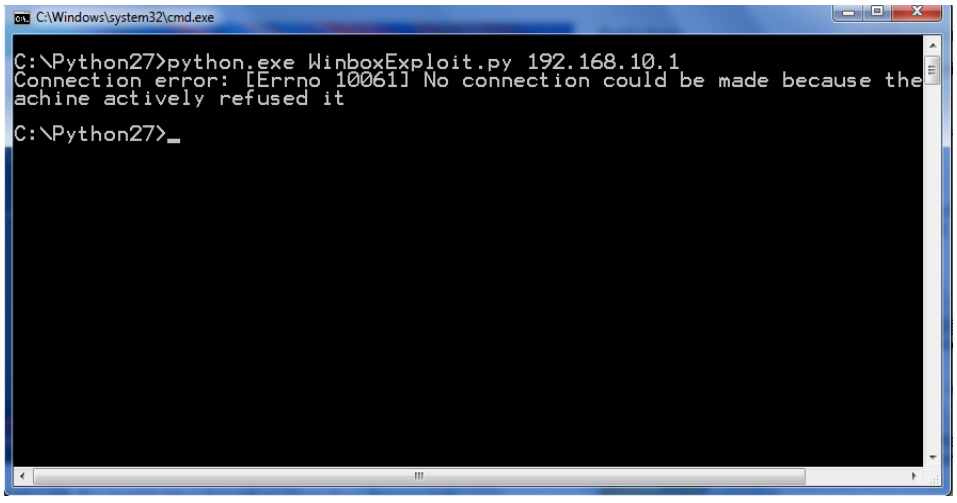
Pengujian	**Rata-rata performance**	**Sebelum serangan**	**Setelah serangan**	**Setelah terpasang *honeypot* dan *port knocking***
Uji 1	*CPU Usage History*	5%-12%	71%-85%	5%-19%
	Networking History LAC	1%-5%	20%-25%	1%-5%
Uji 2	*CPU Usage History*	5%-23%	75%-90%	5%-22%
	Networking History LAC	1%-5%	20%-30%	1%-7%
Uji 3	*CPU Usage History*	5%-16%	70%-89%	5%-25%
	Networking History LAC	1%-5%	20%-25%	1%-5%
Uji 4	*CPU Usage History*	5%-21%	74%-96%	5%-24%
	Networking History LAC	1%-5%	25%-45%	1%-8%
Uji 5	*CPU Usage History*	5%-18%	70%-92%	5%-17%
	Networking History LAC	1%-5%	25%-35%	1%-5%

Setelah *honeypot* dan *port knocking* terpasang pada server jaringan serangan *ddos attack request flooding* dan *ddos attack traffick flooding* yang masuk ke dalam server *honeypot kfsensor* kinerja komputer *server* dan *client* pun tidak terganggu dimana kinerja *performance CPU usage* menjadi lebih stabil pada pengujian pertama rata-rata mencapai 5%-19%, pengujian kedua 5%-22%, pengujian ketiga 5%-25%, pengujian keempat 5%-24% dan pengujian kelima 5%-17% sedangkan kinerja *performance* pengujian pertama pada jaringan rata-rata mencapai 1%-5%, pengujian kedua 1%-7%,

pengujian ketiga 1%-5%, pengujian ke eempat 1%-8%, dan pengujian kelima 1%-5%.

4.1.3.2 Mendeteksi serangan *Ddos Brute Force attack*

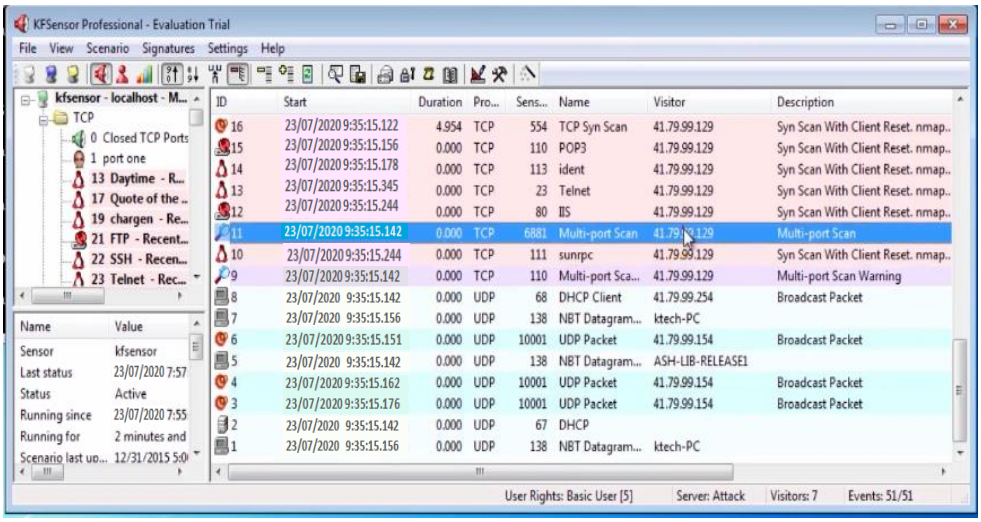
Pada saat melakukan *ddos brute force attack* hal yang paling utama yang harus penyerang ketahui adalah mendapatkan *ip address* target untuk bisa dilakukan *eksploit microtik* server target dengan *software advanced ip scanner*, setelah didapatkan *ip* target 192.168.10.1 selanjutnya proses *exploit microtik* server dilakukan namun hasilnya gagal karena *microtik* sudah menggunakan *port kn*ocking untuk mengamankan server jaringan sehingga penyerang tidak bisa menembus port-port yang sudah ditutup pada *microtik* yang sudah diprotek keamanan jaringannya dengan metode *port knocking* dengan status *connection error*, seperti pada gambar 4.41 *eksploit microtik* setelah pengaturan *port knocking*.



Gambar 4.41 *Eksploit microtik* setelah pengaturan *port knocking*

Dalam pengujian penyerangan diatas dapat dilihat bahwa serangan yang dilakukan tidak berhasil, namun dalam proses *scanning* mendapatkan

ip address dengan metode nmap terdeteksi oleh *honeypot* pada *protocol TCP*, seperti pada gambar 4.42 *Honeypot* deteksi serangan *brute force.*



Gambar 4. 424 *Honeypot* deteksi serangan *brute force*

Dari hasil *ddos brute force attack* dan terdeteksi pada *honeypot* yaitu *multy port scan* dari log aktivitas yang masuk pada jaringan, walaupun secara langsung serangan ini tidak mengganggu sistem jaringan dan juga gagal dalam *eksploit microtik* karena sudah terproteksi oleh *port knocking* namun tetap saja semua aktifitas jaringan akan terekam di *honeypot* sebagai keamanan server jaringan, berikut tabel 4.10 Analisis serangan *brute force* setelah terpasang *honeypot* dan *port knocking*.

Tabel 4.10 Analisis hasil serangan *brute force* setelah terpasang *honeypot* dan *port knocking*

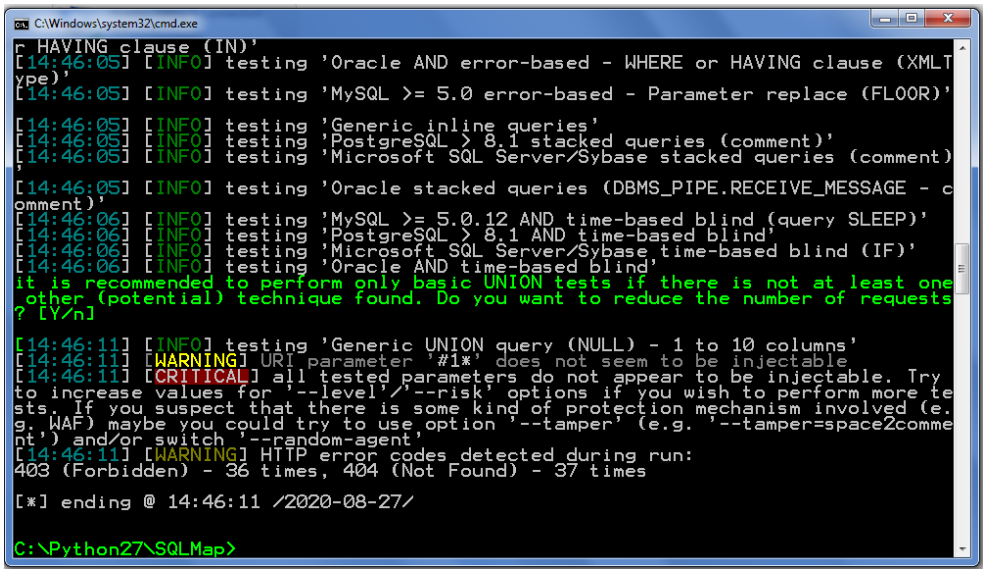
Pengujian	**Target serangan**	**Status keamanan**		**Hasil serangan**
		Honeypot	***Port Knocking***	
Uji 1	FTP *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	*Honeypot* mendeteksi	Router tidak memproduksi	Gagal menampilkan

		serangan	log	*username* dan *password router microtik*
Uji 2	FTP *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
Uji 3	FTP *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
Uji 4	FTP *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
Uji 5	FTP *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*
	SSH *brute force attack*	*Honeypot* mendeteksi serangan	Router tidak memproduksi log	Gagal menampilkan *username* dan *password router microtik*

Hasil analisis dari pengujian serangan *ddos brute force attack* penyerang tidak dapat menembus *port* FTP dan SSH dalam melakukan *exploitasi password* dan *username* pada microtik dikarenakan *port* tersebut sudah tertutup dan sudah dikonfigurasi dengan metode *port knocking* dan aktivitas log serangan pun terdekteksi pada *honeypot* karena adanya aktivitas pada jaringan.

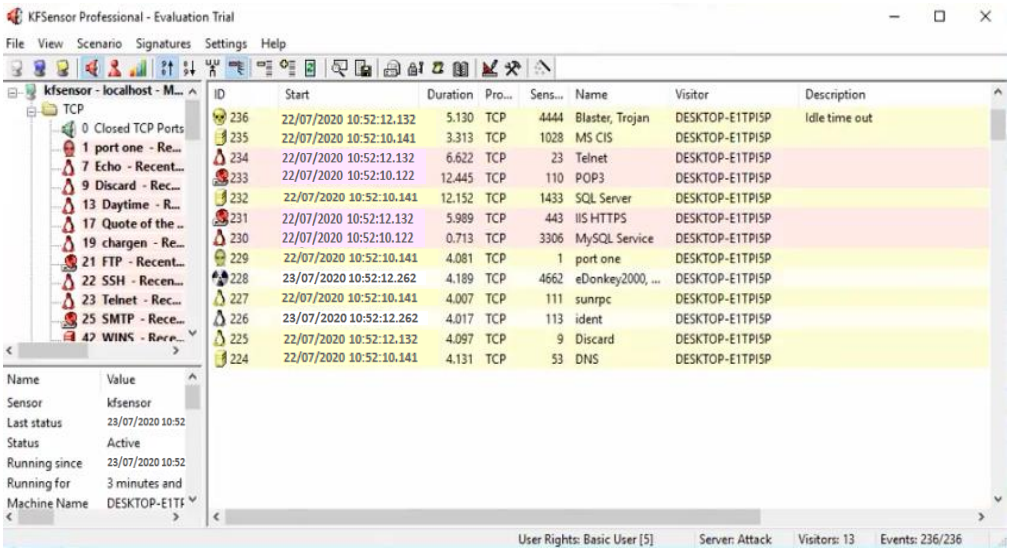
4.1.3.3 Mendeteksi Serangan *ddos attack SQL injection*

Pada serangan *SQL injection user* hendak masuk ke halaman *admin* dengan cara melihat *database, tabel* dan *coloumn* agar bisa melihat struktur *database* untuk bisa mendapatkan *username* dan *password* yang digunakan admin sebagai pengguna yang berperan pernuh terhadap aplikasi web tersebut namun pada penelitian percobaan serangan ini gagal dilakukan karena server pada jaringan sudah terproteksi dengan baik dengan *port knocking* dan *honeypot*, seperti pada gambar 4.43 Serangan *SQL Injection*.



Gambar 4.43 Serangan *SQL Injection*

Setelah serangan *SQL injection* dilakukan dan hasilnya tidak berhasil dapat juga dilihat pada lalulintas aktivitas pada keamanan jaringan server *honeypot* yang mengidentifikasi adanya *user* dari luar yang hendak mengeksploitasi pada aplikasi *computer based tes* berbasis web yang dijadikan sebagai target sasaran serangan, seperti pada gambar 4.44 *Honeypot* mendeteksi *ddos SQL injection attack*.



Gambar 4. 44 *Honeypot* mendeteksi *Ddos SQL injection attack*

Dari hasil yang terdeteksi oleh *honeypot* ada lalu lintas *user* yang hendak masuk pada jaringan ternyata penyerang akan melakukan *SQL Injection* pada *SQL server* sehingga terdeteksi pada *honeypot server*, berikut tabel 4.11 Analisis hasil serangan *SQL injection* setalah terpasang *honeypot* dan *port knocking*.

Tabel 4.11 Analisis hasil serangan *SQL injection* setalah terpasang *honeypot* dan *port knocking*

NO	Jenis *eksploitasi*	Hasil *SQL Injection*		Keterangan
		Berhasil	Gagal	
1	Database	-	√	Tidak menampilkan *database*
2	Table	-	√	Tidak menampilkan *table*
3	Column	-	√	Tidak menampilkan *column*
4	Entries	-	√	Tidak menampilkan *username* dan *password*

Dari hasil analisis serangan pada tabel 4.11 diatas serangan *SQL Injection* tidak berhasil mengexploitasi *database, table* dan *query* pada aplikasi web dan dilakukan pengujian sebanyak 5 kali dengan hasil yang sama yaitu gagal menampilkan *username* dan *password* yang digunakan *admin* pada aplikasi web *CBT*, dikarenakan konfigurasi pada pengaturan *port knocking* mampu membatasi *user-user* asing dan memblok *user* berbahaya yang akan masuk kedalam sistem jaringan melalui *port* tertentu dibandingkan jika komputer server hanya menggunakan sistem *firewall* biasa.

Sistem yang dibangun dengan menggunakan metode *port knocking* lebih aman karena disertakan juga *virtualhost* sebagai *port* bayangan agar admin bisa mengakses *database* tanpa melalui port 80 sehingga penyerang tidak akan mengetahui *port* mana yang terbuka dan semua serangan yang terdeteksi di sever *honeypot* langsung dilakukan pemblokiran pada *microtik* melalui metode *port knocking*, sehingga komunikasi jaringan pada komputer *server* dan *client* menjadi aman dari serangan *ddos attack*, berikut tabel 4.12 Hasil *recovery* deteksi serangan *ddos attack*.

Tabel 4.12 Hasil *recovery* deteksi serangan *ddos attack*

Jenis serangan	**Deteksi serangan**		**Protocol**	**Keterangan**
	Honeypot	**Port Knocking**		
Ddos attack request flooding	*Desudesudesu-A*	*Loic*	*TCP port* 21	Paket *request timeout and off connection*
Ddos attack traffick flooding	192.168.10.3	*Hoic*	*TCP port* 21	Paket *byte* data
Ddos brute force attack	*multy port scan*	Tidak ada log aktivitas	TCP Port FTP 21 Port SSH 22	*Eksploitasi username* dan *password microtik*
Ddos attack SQL injection	*SQL server*	Tidak ada log aktivitas	*TCP*	Menginjeksi *struktur database*

Didapatkan hasil deteksi sistem kemanan jaringan *honeypot* dan *port knocking* dari serangan *ddos attack request flooding* dengan nama *desudesudesu-A* menggunakan *software loic* pada *protocol TCP* dan *port FTP* 21 dengan jenis serangan berupa *request timeout and off connection*, sedangkan pada serangan *ddos attack traffick flooding* ditemukan alamat *ip* penyerang yaitu 192.168.10.3 dengan menggunakan *software hoic* pada *protocol TCP* dan *port FTP* 21 jenis serangan berupa paket *byte* data. Pada serangan *ddos brute force attack* terdeteksi serangan oleh *honeypot* dengan nama *multy port scan* sedangkan pada *port knocking* tidak ada log aktivitas karena *exploitasi* awal *microtik* gagal dilakukan pada *protocol TCP Port FTP 21* dan *port SSH* 22, selanjutnya untuk serangan *ddos attack SQL injection* terdeteksi oleh *server honeypot* berupa *name SQL server* yang mencoba menginjeksi struktur *database MSQL* dan pada *port knocking* tidak ada log aktivitas pada server *microtik* sehingga serangan tersebut gagal dilakukan untuk menembus *port* target. Berikut hasil analisis dari semua

pengujian serangan *ddos attack* pada tabel 4.13 analisis hasil semua serangan *ddos attack* setelah terpasang *honeypot* dan *port knocking* pada server jaringan.

Tabel 4.13 Analisis hasil serangan *ddos attack* setelah terpasang *honeypot* dan *port knocking* pada server jaringan

NO	Jenis serangan	Proses *runing*		Keterangan
		Berhasil	Gagal	
1	*Ddos Attack Request Flooding*	-	√	*Request* paket gagal masuk ke server jaringan namun masuk ke *server honeypot*
2	*Ddos Attack Traffic Flooding*	-	√	*Traffick* paket data gagal masuk ke server jaringan namun masuk ke *server honeypot*
3	*Ddos Brute Force Attack*	-	√	Gagal menampilkan *username* dan *password router microtik*
4	*Ddos Attack SQL Injection*	-	√	Gagal menampilkan *password* dan *username admin*

Dalam implementasi keamanan server dengan menggunakan metode *port knocking* dan *honeypot*, *port knocking* dapat menghentikan memblok serangan yang masuk pada jaringan dengan menggunakan *microtik*, sedangkan *honeypot* sebagai pengalihan agar *intruder* atau penyusup masuk pada server tiruan, dengan *honeypot* kita bisa melihat *log*/aktivitas yang dikerjakan penyerang terhadap server jaringan. Dapat dilihat dalam penelitian ini bahwa *honeypot* dan *port knocking* mampu mengamankan serangan pada server jaringan dari *ddos attack request flooding*, *ddos attack traffick flooding*, *ddos brute force attack* dan *ddos attack SQL injection* khususnya pada *sistem operasi windows*.

4.2 Pembahasan

Berdasarkan hasil pengujian serangan *ddos attack request flooding, ddos atack traffick flooding, ddos brute force attack* dan *ddos attack SQL injection* dapat dibandingkan untuk dianalisis sebelum dan sesudah digunakannya sistem keamanan jaringan *honeypot* dan *port knocking* terhadap kemanan server jaringan. Keseluruhan pembahasan terhadap pengujian mendeteksi serangan *ddos atack* dengan *honeypot* dan *port knocking* dapat dilihat pada tabel 4.14.

Tabel 4.14 Perbadingan hasil deteksi serangan *ddos attack* dengan *honeypot* dan *port knocking*

Pengujian serangan	**Konfigurasi honeypot dan port knocking**		**Hasil pengujian**	
	Sebelum	**Sesudah**	**Honeypot**	**Port knocking**
Ddos Attack Request Flooding	Berhasil	Gagal	Terdeteksi	Terdeteksi
Ddos Attack Traffic Flooding	Berhasil	Gagal	Terdeteksi	Terdeteksi
Ddos Brute Force Attack	Berhasil	Gagal	Terdeteksi	-
Ddos Attack SQL Injection	Berhasil	Gagal	Terdeteksi	-

Tabel 4.14 menunjukan hasil dari pengujian serangan sebelum konfigurasi *honeypot* dan *port knocking* dilakukan 100% berhasil. Pada serangan *ddos attack request flooding request* berupa *timeout* yang masuk pada server jaringan saat pengujian 5 kali serangan terjadi sebanyak 2987428 *seconds,* 2735891 *seconds,* 2583947 *seconds*, 3310978 *seconds* dan 2872984 *seconds* sedangkan saat serangan *ddos attack traffick flooding* berhasil masuk berupa byte data sebanyak 7.117052e+8 kb, 5.363787e+8 kb, 7.023877e+7 kb, 2.116352e+9 kb dan 1.080268e+9 kb, pada hasil tersebut server jaringan pun menjadi lambat dan

lumpuh kinerja jaringan pada *client-server* terganggu, komputer serverpun *performance* kinerjanya meningkat yang sebelumnya rata-rata *performance* pada *CPU usage* 5%-12%, 5%-23%, 5%-16%, 5%-21%, 5%-18% dan *performance* rata-rata pada jaringan masih konsisten 1%-5% pada 5 kali pengujian serangan *ddos attack*, setelah adanya serangan ddos masuk performance pada *CPU usage* meningkat mencapai 71%-85%, 75%-90%, 70%-89%, 74%-96%, 70%-92% dan *performance* jaringan mencapai 20%-25%, 20%-30%, 20%-25%, 25%-45%, 25%-35%.

Setelah proses *konfigurasi honeypot* dan *port knocking* diterapkan pada server jaringan terdapat hasil dari serangan *ddos attack request flooding* mencapai 1572283 *seconds*, 1094759 *seconds*, 1948750 *seconds*, 1749371 *seconds*, 1430156 *seconds* dan pengujian serangan *ddos attack traffick flooding* 2.910889e+6 kb, 2.063770e+7 kb, 3.055146e+7 kb, 2.780476e+7 kb, 2.509194e+7 kb namun paket serangan ini telah dialihkan ke server *honeypot* dan telah dilakukan tindakan pemblokiran *user* pada *port knocking* sehingga *performance* kinerja pada *CPU usage* menjadi stabil dari 5 kali proses pengujian hanya mencapai 5%-19%, 5%-22%, 5%-25%, 5%-24% dan 5%-17% sedangkan *percormance* pada jaringan rata-rata 1%-5%, 1%-7%, 1%-5%, 1%-8%, dan 1%-5% yang artinya dalam keadaan stabil karena tidak banyak melayani *user* dan koneksi dari jaringan *client-server* tidak terganggu, sehingga serangan *Ddos Attack Request Flooding* dan *Ddos Attack Traffic Flooding* dinyatakan tidak berhasil mengganggu server jaringan.

Pada serangan *ddos brute force attack* sebelum *port FTP* dan *SSH* pada *microtik* dialihkan dengan metode *port knocking*, *exploit* pada *microtik* berhasil mengidentifikasi *username* dan *password* dengan benar. Namun setelah *port* pada *microtik* dikonfigurasi pada *port kncoking* dengan syarat *login* harus melewati *port-port* yang sudah ditentukan dengan cara melakukan 4 kali ketukan atau ping pada saat hendak *login*, serangan *brute force atack* pun gagal mengidentifikasi *username* dan *password* dengan benar.

Pengujian pada serangan *ddos attack SQL Injection* sebelum *honeypot* dan *port knocking* terpasang berhasil menginjeksi struktur tabel pada *database* sehingga penyerang berhasil mendapatkan *username* dan *password* untuk dapat *login* kedalam aplikasi web *CBT* dengan menggunakan *user admin*. Namun setelah *port* diamankan dengan menggunakan konfigurasi *port knocking* serangan ini pun tidak berhasil mengexploitasi *struktur database* pada aplikasi web tersebut.

Dari tabel 4.14 didapatkan presentasi hasil deteksi dari masing-masing serangan dan hasil deteksi dari *honeypot* dan *port knocking*, dengan menggunakan rumus:

Ph = (Jt/Jp)x100%

Keterangan:

Ph = Presentasi hasil

Jt = Jumlah terdeteksi

Jp = Jumlah pengujian

Dengan begitu didapatkan hasil presentasi deteksi dari *honeypot* dan *port knocking*, yaitu pada *honeypot* 100% mendeteksi adanya serangan pada 4 uji coba serangan *ddos attack* dan masing-masing serangan terdeteksi pada server *honeypot*, sedangkan pada *port knocking* 50% dari hasil pengujian 4 serangan *ddos attack*, mendeteksi 2 serangan pada log aktivitas *microtik* dan 2 serangan lagi tidak terdeteksi karena pada percobaan awal sudah gagal dilakukan, namun pada dasarnya sistem kemanan jaringan pada *honeypot* dan *port knocking* tersebutlah yang menggagalkan semua serangan *ddos attack* pada server jaringan.

BAB V
PENUTUP

Pada bagian ini akan dijelaskan kesimpulan dari hasil penelitian yang sudah dilakukan berdasarkan tujuan dan perumusan masalah penelitian yaitu bagaimana analisis dari implementasi *honeypot* dan *prort knocking* dalam mendeteksi serangan *distributed denial of servis attack (ddos attack)* pada server jaringan.

5.1 Kesimpulan

Dari hasil penelitian yang sudah dilakukan dapat disimpulkan bahwa peran *honeypot* dan *port knocking* dalam keamanan server jaringan telah teruji dan bekerja dengan baik sehingga sangat penting setiap sistem pada server jaringan untuk menghindari adanya serangan dari orang yang tidak bertanggung jawab yang hendak mengeksploitasi data, merusak jaringan dan gangguan lainnya yang dapat merugikan pemilik jaringan untuk menggunakan sistem keamanan ini selain konfigurasi dalam penggunaan *softwarenya* mudah juga metode ini sangat ampuh untuk mencegah bahkan menggagalkan serangan dari segala jenis *distributed denial of servis attack (ddos attack)* khususnya pada sistem operasi windows. Pada pengujian serangan *honeypot* mendeteksi 100% dari semua serangan yang dilakukan sedangkan *port knocking* mendeteksi 50% serangan pada log aktivitas *server microtik,* namun dalam menggagalkan serangan *honeypot* dan *port knocking* 100% menggagalkan semua serangan *ddos attack* yang dilakukan pada pengujian penelitian ini.

Berikut hasil kesimpulan yang telah didapatkan selama proses pengujian penelitian implementasi *honeypot* dan *port knocking* pada keamanan server jaringan dalam mendeteksi serangan dari beberapa *distributed denial of servis attack (ddos attack)* pada server jaringan;

1. Pengujian serangan pada server jaringan *localhost* ujian jamin *offline* dari serangan *ddos attack request flooding* dan *ddos attack traffick flooding* berhasil terdeteksi oleh *honeypot* berupa paket *request timeout and off connection* dan paket berupa *byte* data, serangan tersebut telah dialihkan dan masuk pada *server honeypot* sehingga tidak mengganggu server utama jaringan dan serangan berhasil diblokir menggunakan metode *port knocking*. Dapat dilihat dari 5 kali proses pengujian *performance CPU usage history* sebelum serangan rata-rata mencapai 5%-18% *performance* pada jaringan 1%-5%, setelah adanya serangan *ddos attack performance CPU usage history* kinerjanya meningkat rata-rata mencapai 72%-90% pada jaringan 22%-32% dan setelah terpasang sistem keamanan jaringan *honeypot* dan *port knocking performance CPU usage history* kembali stabil rata-rata mencapai 5%-21% sedangkan kinerja *performnace* pada jaringan 1%-6%.
2. Pengujian pada serangan *ddos brute force attack* yaitu mencoba untuk mengexploitasi *microtik server* untuk mendapatkan *username* dan *password* gagal dilakukan, serangan ini tidak bisa masuk pada *port* yang sudah dikonfigurasi dengan metode *port knocking* sehingga proses *eksploitasi* tidak berhasil dan *honeypo*t pun mendeteksi adanya log

aktivitas pada jaringan, sehingga *honeypot* dan *port knocking* berhasil mengamankan *server microtik* pada jaringan.

3. Pengujian serangan *ddos attack SQL injection* dimana target serangan yaitu aplikasi web *computer based tes (CBT)* yang dapat diakses dengan jaringan *localhost* dan dalam teknik serangannya mencoba untuk mendapatkan *usename* dan *password* agar bisa masuk sebagai *admin* dengan cara menginjeksi data-data yang terdapat pada struktur *database, table, column* dan *entry* data. Namun gagal dilakukan karena *virtualhost* pada *port* yang menjadi target serangan telah dialihkan dengan *port knocking* dan serangan ini pun dapat terdeteksi pada *server honeypot*.

5.2 Saran

Berdasarkan hasil dari kesimpulan diatas saran yang dapat diberikan kepada peneliti seanjutnya adalah;

1. Parameter dalam pengujian penelitian gunakan lebih dari satu metode, lakukan pengujian serangan jangan hanya pada server jaringan namun pada pengujian serangan target sasaranya yaitu pada *user client* tertentu misalnya model serangan *client to client* .
2. Dalam melakukan pengujian serangan dengan menggunakan metode *ddos attack* ataukah menggunakan teknik penyusupan atau serangan lainya sebaiknya target pengujian dalam serangan lebih banyak lagi baik secara *offline* maupun *online* agar dapat menjadi referensi dalam penelitian

lanjutan atau lakukan pengujian serangan pada lebih dari satu server jaringan.

3. Gunakan metode dalam keamanan jaringan lebih banyak lagi seperti *honeypot*, *port knocking*, *Snort IDS* dan yang lain sebagainya namun tetap bisa terkonfigurasi pada satu server jaringan agar lebih baik lagi dalam mengamankan sever pada jaringan.
4. Dalam pengujian penelitian selanjutnya baiknya menggunakan lebih dari satu sistem operasi seperti *windows*, *linux* atau *mecintosh* untuk melakukan serangan ataupun untuk keamanan server jaringan, agar dapat dianalisis model proses serangannya mana yang lebih baik dan model dari sistem keamanan jaringanya mana yang lebih baik, jika dikolaborasikan menggunakan lebih dari satu sistem operasi.

DAFTAR PUSTAKA

[1] Fakariah Hani Mohd Ali, Rozita Yunos, Mohd Azuan Mohamad Alias, "Simple Port Knocking Method", IEEE Software, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA 2013.

[2] Luigi Catuogno, Aniello Castiglione, Francesco Palmieri, "A Honeypot System with Honeyword-driven Fake Interactive Sessions", IEEE Software, Department of Computer Science, University of Salerno 2015.

[3] Mahajan, S., Adagale, A. M., & Sahare, C. (2016). Intrusion Detection System Using Raspberry PI Honeypot in Network Security. International Journal of Scientific and Engineering Research- IJESC, 6(3), 2792–2795.

[4] M. S. Zemene and P. S. Avadhani, "Implementing High Interaction Honeypot to Study SSH Attacks," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1898-1903, 2015

[5] Firrar, Utdirartatmo. 2005. Trik Menjebak Hacker Dengan Honeypot. Yogyakarta: ANDI OFFSET.

[6] ArborNetworks. (2014). Worldwide Infrastructure Security Report. Burlington: Arbor Networks Security Division.

[7] Zhao, T., Lo, D. C.-T., & Qian, K. (2015). A Neural Network Based DDoS Detection System Using Hadoop and HBase. IEEE 17th International Conference on High Performance Computing and Communication, 1326-1331.

[8] Eray, B., Jander, A., & A.Nur. (2014). Supervised Learning to Detect DDoS Attacks. IEEE Journal, 14.

[9] Unswagati, (2010). Keamanan Jaringan Komputer. Diakses dari http://unswagaticrb.ac.id/component/option,com_phocadownload/Itemid,73/download,55/id,11/view,category/ (diakses pada tanggal 13 Februari 2015).

[10] Sugiyono. 2011. Metode Penelitian Kuantitatif, Kualitatif dan R&D. Bandung: Afabeta

[11] R.Michael, Linux Firewalla, San Fransisco: No Strach Press Inc., 2007