Forensik Serangan Brute Force pada Cloud Public Menggunakan Logika Fuzzy

Tugas Akhir

Oleh :

Ade Rahmad

09011281419059

Jurusan Sistem Komputer

Fakultas Ilmu Komputer

Universitas Sriwijaya

2019

BAB I

PENDAHULUAN

1.1 Latar Belakang

Serangan *brute force* adalah suatu serangan dimana penyerang menggunakan kumpulan *password* yang telah ditetapkan dalam *wordlist* untuk menyerang target hingga berhasil[1]. Berhasil atau tidaknya serangan bergantung pada kumpulan dari jumlah kemungkinan *password* yang telah ditetapkan. Jika jumlah kemungkinan *password* yang ditetapkan banyak maka serangan *brute force* mempunyai kemungkinan berhasil yang tinggi tetapi akan memakan lebih banyak waktu. Jika dalam *wordlist* yang digunakan terdapat kata yang cocok dengan *password* maka serangan *brute force* berhasil dilakukan.

Laporan ancaman internet yang dibuat oleh McAfee pada September 2017 menyatakan serangan *brute force* menjadi *top network attack* pada tahun 2017 dengan persentase serangan sebesar 20%. Serangan *brute force* dapat dilakukan oleh siapa pun karena sudah banyak *tools brute force* dan *wordlist* yang beredar di Internet[2].

NIST (*National Institute of Standards and Technology*) mendefinisikan *cloud computing* sebagai sebuah model yang memungkinkan untuk mengakses *resources*(Seperti jaringan,*server*,*storage*,aplikasi,dan servis) melalui jaringan baik jaringan lokal maupun jaringan internet [3].

Cloud mempunyai poin-poin utama dalam keamanan yaitu : Kerahasiaan, integritas, ketersediaan, akuntabilitas, dan privasi [4]. *Cloud* rentan akan serangan terhadap *cybercrime* yang semakin hari semakin canggih [5], sehingga tantangan terbesar dalam *cloud* adalah bagaimana mengidentifikasi serangan yang terjadi pada lingkungan *cloud* [6]

Pada penelitian sebelumnya, membahas tentang pembuktian serangan *brute force* pada dataset DARPA 2000. Hasil penelitian tersebut menggunakan berbagai metode dalam pembuktiannya dan didapatkan metode yang paling akurat digunakan adalah metode NFS-FLES (Fuzzy Logic Expert System) dengan akurasi 0.9834 dan mendeteksi sebesar 91.5 % terhadap semua serangan[7].

Pada penelitian sebelumnya, membahas penggunaan metode fuzzy dalam menyelesaikan masalah forensik. Hasil dari penelitian didapatkan metode fuzzy mendapatkan hasil akurat untuk analisa data forensik[8].

Pada penelitian ini, penulis akan membuktikan serangan *brute force* yang dilakukan pada *cloud* menggunakan metode fuzzy. Penggunaan metode ini sesuai dengan yang ada pada penelitian sebelumnya yang disebutkan metode *fuzzy* merupakan metode yang paling akurat dalam analisa data forensik.

1.2 Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Menerapkan metode fuzzy untuk analisa data yang didapat dari *cloud* dan penyerang.
2. Membuktikan serangan *brute force* pada *cloud*.
3. Menjelaskan kronologi serangan pada *cloud*.

1.3 Manfaat

Adapun manfaat yang dapat diambil dari penelitian ini adalah:

1. Pembuktian serangan *brute force* yang terjadi pada *cloud*.
2. Mampu menjelaskan secara rinci kronologis ketika penyerangan dan aksi yang dilakukan pelaku maupun sistem atau korban.

1.4 Perumusan dan Batasan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan dan batasan masalah yang ada pada penelitian ini adalah:

Perumusan Masalah

1. Bagaimana serangan *brute force* yang terjadi pada *cloud*?
2. Apakah karakter serangan *brute force* sama seperti *login* normal pada *cloud*?
3. Bagaimana hasil yang didapatkan dari penerapan metode fuzzy pada penelitian ini?

Batasan Masalah

1. Penelitian dilakukan pada *cloud* yang bersifat publik.
2. Serangan *brute force* dilakukan pada API OCS pada *cloud*
3. Metode yang digunakan untuk menganalisis data forensik menggunakan metode fuzzy.
4. Data yang digunakan didapat dari *server cloud* dan penyerang.
5. Hasil data yang diolah akan menghasilkan informasi tentang penyerang yaitu IP *address, operation system, username login, password login* dan aktivitas penyerang.
6. Penggunaan sistem snort dalam membuktikan adanya terjadi serangan pada *cloud*.

1.5 Metodologi Penulisan

Metodologi yang digunakan dalam penulisan tugas akhir, akan melewati beberapa tahapan sebagai berikut:

1. Tahap pertama (Perumusan Masalah)

Tahap ini ialah tahap yang menentukan permasalahan yang ada pada *cloud computing* yang telah dibahas pada penelitian sebelumnya yaitu keamanan pada *cloud computing* untuk mengidentifikasi serangan yang terjadi dan membuktikan serangan tersebut.

2. Tahap kedua (Study Pustaka / Literature Review)

Tahap ini ialah tahap mencari referensi atau literature ilmiah yang berhubungan dengan judul tugas akhir untuk menunjang penelitian yang dilakukan.

3. Tahap ketiga (Perancangan)

Tahap ini ialah tahap perancangan sistem yang akan dibuat sesuai dengan rumusan masalah penelitian.Dalam tahap ini melakukan instalasi *operation system* ,membangun jaringan *cloud* dan konfigurasi *cloud* tersebut.

4. Tahap keempat (Pengujian)

Tahap ini ialah tahap pengujian dari sistem yang telah dirancang. Ditahap ini akan diuji serangan *brute force* menggunakan kali linux kepada *cloud* yang telah dibangun.

5. Tahap kelima (Analisis)

Tahap ini ialah tahap analisa dari hasil pengujian. Disini akan dianalisa bagaimana serangan tersebut dilakukan dan oleh siapa serta dibuktikan dengan bukti yang jelas dan kronologis.

6. Kesimpulan dan Saran

Pada tahap ini ditarik kesimpulan dari hasil analisa penelitian dan dibuat saran sebagai referensi apabila penelitian ini dilanjutkan.

1.6 Sistematika Penulisan

Penyusunan laporan tugas akhir ini terdiri dari beberapa bab agar pembahasan lebih sistematik dan spesifik dengan rincian sebagai berikut:

BAB I. PENDAHULUAN

Pada bab I berisikan penjelasan secara sistematis mengenai topik penelitian yang diambil meliputi latar belakang, tujuan, manfaat, rumusan masalah, batasan masalah, metodologi penulisan dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Pada bab II berisikan mengenai dasar teori dari penelitian terkait mengenai *Brute Force Attack, Cloud Computing, Network Forensic, Fuzzy Logic, Snort* yang berkaitan dengan penelitian. Bab ini akan menjadi tinjauan atau landasan dalam menganalisis batasan masalah yang telah dikemukakan pada bab sebelumnya.

BAB III. METODOLOGI

Bab III berisikan tentang penjelasan secara bertahap mengenai proses penelitian yang dilakukan. Penjelasan tersebut meliputi tahapan perancangan sistem dan penerapan metode penelitian.

BAB IV. PENGUJIAN DAN ANALISIS

Bab ini menjelaskan mengenai hasil dari pengujian yang telah dilakukan selama penelitian tugas akhir. Hasil dari pengujian tersebut akan dianalisi dari serangan *Brute Force* yang dilakukan pada *Cloud*.

BAB V. KESIMPULAN DAN SARAN

Bab V berisi kesimpulan akhir dari pembahasan penelitian yang telah dilakukan. Pada bab ini juga terdapat saran yang diperlukan untuk pengembangan penelitian selanjutnya dari pengujian dan analisis tugas akhir.

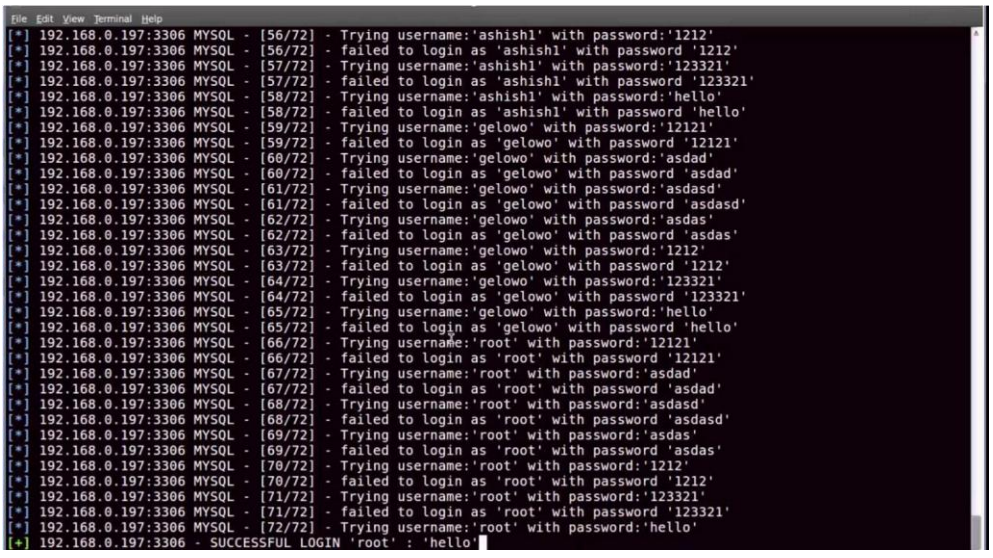
BAB II

TINJAUAN PUSTAKA

2.1 Serangan Brute Force

Serangan *brute force* adalah serangan yang digunakan untuk mendapatkan akses ke suatu *host* atau suatu data dengan mencoba semua kombinasi karakter yang ada. Istilah *brute force* dipopulerkan oleh Kenneth Thompson dengan motto "*When in doubt, use brute-force*". [1] Serangan *brute force* dibagi menjadi dua yaitu : *Brute force attack* dan *reverse brute force attack*. Yang membedakan dua serangan *brute force* ini ialah *brute force attack*, serangan ini berusaha menebak *password* yang ada dengan *username* yang telah diketahui. Sedangkan untuk *reverse brute force* berusaha menebak *username* dengan *password* yang telah diketahui.

Serangan *brute force* dapat mengganggu kinerja *server* karena serangan *brute force* membanjiri *server* dengan *traffic* yang tinggi sampai *password* yang dicari berhasil didapatkan. Sehingga dari sisi *server* diperlukan sebuah metode keamanan login yang bisa melindungi *server* dari serangan *brute force* seperti waktu jeda untuk login kembali apabila sudah mencapai batas gagal login pada suatu *server* atau dengan metode CAPTCHA (*Completely Automated Public Turing test to tell Computer and Human Apart)* untuk setiap kali login. Brute force bekerja diawali dengan melakukan *scanning port* yang terbuka terhadap suatu IP (*Internet Protocol address)* dimana pada IP tersebut terdapat suatu *service* berjalan yang akan di *brute force*. Setelah didapatkan *port* mana yang terbuka maka serangan *brute force* akan ditargetkan ke *port* tersebut.



Gambar 2.1 Serangan *brute force* pada MYSQL

Cara kerja serangan *brute force* untuk mendapatkan *username* dan *password* dengan membuat suatu kamus yang berisi kata-kata umum yang bisa dipilih menjadi suatu *username* atau *password*. [9] Banyaknya isi dalam kamus ini ditentukan oleh persamaan berikut :

$$N = L_{(Min)} + L_{(Min+1)} + L_{(Min+2)} + L_{(Min+3)} + \ldots\ldots + L_{(Max)} \qquad \ldots\ldots(1)$$

Keterangan : N = Jumlah kemungkinan
L = Jumlah karakter yang ada
Min = Panjang minimum *username* atau *password*
Max = Panjang maksimal *username* atau *password*

Sebagai contoh, jumlah karakter yang kemungkinan dijadikan *password* berjumlah enam karakter. *Password* tersebut terdiri dari huruf alfabet dari karakter a sampai karakter z yang berjumlah dua puluh enam huruf. Sehingga penyelesaian dari jumlah kemungkinan *password* tersebut ialah :

$$N = 26_{(1)} + 26_{(2)} + 26_{(3)} + 26_{(4)} + 26_{(5)} + 26_{(6)}$$

$$=26 + 676 + 17.576 + 456.976 + 11.881.376 + 308915776$$

$$N = 321.272.406$$

Didapatkan jumlah kemungkinan *password* yang terdiri dari 6 karakter yang terdiri dari huruf alfabet adalah 321.272.406 kemungkinan. Kemungkinan *password* tadi akan dicoba satu persatu untuk menemukan *password*. Proses percobaan ini tentu akan memakan waktu yang lama karena harus mencoba satu persatu kemungkinan password yang ada. Kondisi ini ketika *password* hanya memiliki panjang enam karakter dan terdiri dari alfabet. Untuk mempersingkat waktu serangan *brute force*,serangan dilakukan dengan membagi tugas ke perangkat-perangkat lain sehingga ketika serangan diluncurkan dengan banyak perangkat waktu yang akan dibutuhkan akan menjadi lebih cepat.

Serangan *brute force* memiliki karakteristik sebagai berikut :

1. Karena serangan *brute force* adalah serangan yang mencoba satu persatu kemungkinan *password* yang ada, sehingga pada *log server* pada informasi

kegagalan *login* akan terdapat alamat IP yang sama. Kecuali penyerang *brute force* membagi tugas kepada kelompoknya.

2. Urutan *password* atau *username* dalam percobaan *login* akan sesuai dengan urutan abjad.
3. Percobaan *login* terhadap satu akun dilakukan oleh banyak IP *address.* Serangan *brute force* pada prakteknya dapat membongkar semua jenis enskripsi akan tetapi akan memakan waktu yang banyak tergantung dari tingkat kerumitan *password* dan *username* dari sebuah *server* yang akan diserang.

2.2 Cloud Computing

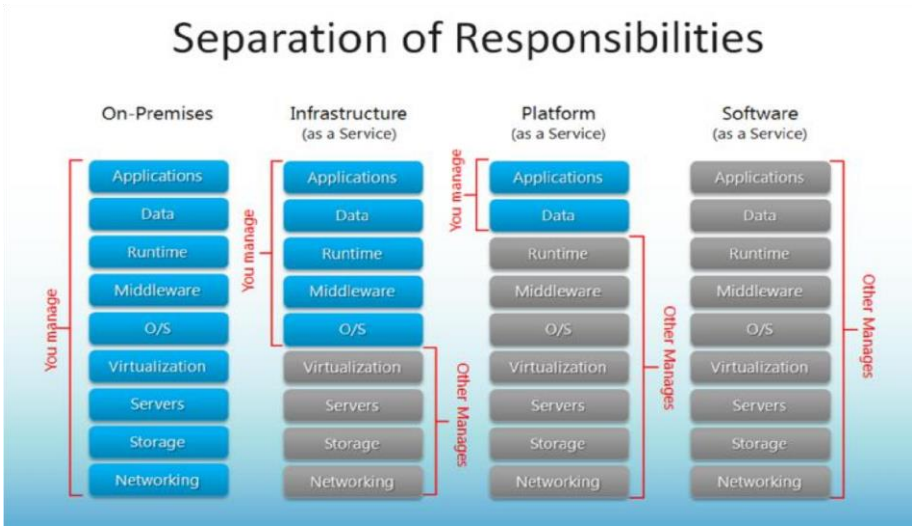
Komputasi awan merupakan sebuah perkembangan paradigma,dimana dengan komputasi awan yang sudah dikonfigurasi dapat diakses melalui media jaringan dengan manajemen yang efisien [3]. Pada komputasi awan semua data, aplikasi, *software* akan disimpan di *server* internet,apabila data dan aplikasi diperlukan dapat diakses melalui layanan *cloud* dengan internet tanpa harus menyimpan dan memasang data dan aplikasi tersebut ke komputer. [10] Komputasi awan dibedakan berdasarkan jenis layanannya menjadi 3 yaitu *Software as a Service (SaaS), Platform as a Service (PaaS), dan Infrastructure as a Service (IaaS).*

Software as a Service (SaaS) merupakan jenis layanan dalam bentuk *software* dari komputasi awan, dimana *user* tinggal memakai *software* yang tersedia di *cloud* tanpa harus menginstal aplikasi tersebut ke komputer *user*. Contoh dari SaaS ini ialah Gmail, Facebook, Whatsapp, dan Office 365. Kelebihan dari layanan SaaS ini adalah *user* tidak perlu menginstal aplikasi yang akan digunakan di komputer mereka dan *user* tidak perlu melakukan konfigurasi atau pemeliharaan terhadap aplikasi tersebut, *user* hanya tinggal pakai saja. Kekurangan dari layanan Saas ini *user* tidak memiliki kendali penuh atas aplikasi tersebut sehingga *user* memiliki akses yang terbatas.

Platform as a Service (PaaS) merupakan jenis layanan komputasi awan dalam bentuk *platform.* Dalam artian *user* hanya perlu memanajemen aplikasi yang

dibuatnya untuk diletakkan dalam layanan *cloud*. Infrastruktur dari *cloud* seperti jaringan, *server,* sistem operasi, dan *storage* tidak akan dimanajemen oleh *user* sehingga *user* hanya perlu fokus pada aplikasi yang ada pada cloud. Contoh dari layanan PaaS ini ialah Amazon *web service*, Microsoft Azure, aplikasi yang ada pada Facebook dan lain-lain.

Infrastructure as a Service (IaaS) merupakan jenis layanan komputasi awan dalam bentuk infrastruktur. *Provider* dari layanan *cloud* akan menyediakan fisik dari komputer, *server,* jaringan, *storage* dan lain-lain. Sehingga *user* hanya perlu menyediakan sistem operasi dan aplikasi pada *cloud* dan mengelola *storage* dan aplikasi yang ada pada *cloud.* Secara sederhana layanan IaaS merupakan layanan yang menyewa infrastruktur yang diperlukan seperti komputer, jaringan dan *storage.* Kelebihan dari layanan IaaS ini ialah *user* tidak perlu menyediakan perangkat fisik yang diperlukan.



Gambar 2.2 Jenis layanan *cloud* dan layanan yang diberikan.[10]

Komputasi awan dibedakan menjadi 4 menurut model penyebarannya yaitu *Public cloud, private cloud, hybrid cloud,* dan *community cloud*. *Public cloud* adalah layanan *cloud* yang disediakan untuk publik yang terbagi menjadi dua yaitu *free public cloud* dan *paid public cloud*. Contoh dari *free public cloud* di Indonesia seperti Gmail, Facebook, Youtube, dan lain-lain. Sedangkan untuk *paid public cloud* seperti Windows Azure, Amazon EC2, SalesForce dan lain-lain. Kelebihan dari *public cloud* adalah *user* hanya perlu menggunakan servis yang diperlukan

tanpa harus membeli dan merawat infrastruktur, *platform* maupun aplikasi. Kekurangan dari *public cloud* ialah keamanan dari data yang tersimpan di *cloud* tidak terjamin aman.

Private cloud merupakan jenis *cloud* yang bersifat *private.* Artinya tidak semua orang dapat menggunakan layanan *cloud* tersebut. *Private cloud* ini sering digunakan pada perusahaan-perusahaan yang akan dikelola oleh departemen IT dari perusahaan tersebut. Kelebihan dari *private cloud* ialah keamanan data yang terjamin karena dikelola sendiri oleh departemen IT, hemat *bandwith* internet ketika *private cloud* hanya menggunakan jaringan lokal saja sehingga apabila internet mati layanan *cloud* akan tetap berjalan. Sedangkan kelebihan dari *private cloud* ialah perusahaan harus menyediakan infrastruktur karena *cloud* akan dibangun sendiri. Setelah infrastruktur terbangun perusahaan harus menyediakan anggaran untuk pengelolaan dan *maintenance* dari *cloud* tersebut.

Hybird cloud merupakan jenis layanan *cloud* gabungan dari *private cloud* dan *public cloud*. Artinya layanan dalam *hybird cloud* akan dibagi mana yang bisa diletakkan pada *public cloud* dan mana layanan yang hanya boleh digunakan pada *internal* perusahaan. Dengan contoh, perusahaan X adalah sebuah perusahaan bank yang ada di Indonesia. Perusahaan X akan menyewa *public cloud* untuk memudahkan akses layanan kepada nasabah maupun calon nasabah, sedangkan data pribadi dari nasabah dan data *internal* perusahaan akan diletakkan pada *private cloud* agar data tidak diakses oleh publik.

Community cloud merupakan jenis layanan *cloud* yang digunakan dan dikelola oleh sekelompok orang atau komunitas yang mempunyai masalah dan tujuan yang sama [3].

Komputasi awan memiliki 5 karakteristik yaitu *On-demand selfservice, Broad network access, Resource polling, Rapid elasticity,* dan *Measured service. On-demand self-service*, user dapat menyediakan sendiri kemampuan komputasi seperti waktu *server* dan *network storage* yang diperlukan tanpa harus menghubungi pihak *provider. Broad network access* merupakan kemampuan yang tersedia pada jaringan dan dapat diakses pada berbagai perangkat yang berbeda seperti *handphone*, tablet, dan komputer. *Resource polling* merupakan kemampuan dimana sumber daya komputasi dari *provider* akan dikumpulkan untuk melayani

banyak *user* menggunakan *multi-tenant* model, dengan sumber daya fisik dan sumber virtual yang berbeda. Secara dinamis akan ditugaskan dan ditugaskan ulang sesuai dengan permintaan *user*. *Rapid elasticity* adalah kemampuan *cloud* bersifat elastis dimana layanan *cloud* dapat dibuka secara tidak terbatas dan bisa disesuaikan dengan kuantitas kapan saja. *Measured service* adalah kemampuan dimana sistem *cloud* akan secara otomatis mengontrol dan mengoptimalkan *resources* dengan memanfaatkan kemampuan pengukuran pada tingkat abstraksi yang sesuai dengan jenis layanan. Penggunaan *resources* akan dapat dipantau, dikontrol dan dilaporkan sehingga memiliki sifat transparansi bagi *provider* maupun *user* [3].

Cloud menghadapi tantangan besar dalam keamanan terhadap *cybercrime* yang semakin berkembang. Diperlukan metode keamanan yang kuat untuk menghadapi *cybercrime* sehingga pengguna *cloud* akan merasa tenang menggunakan teknologi *cloud* [4]. Masalah keamanan ada karena *cloud* bersifat *shared,virtualized,* dan *public* [5]. Pendekatan yang dilakukan untuk melindungi data *user cloud* dari *cybercrime* adalah mengimplementasikan enkripsi *end-point, firewalls* dan *antivirus*. [6] *Intrusion Detection System (*IDS) dapat digunakan untuk mendeteksi dan mencegah adanya serangan *cybercrime* pada lapisan *network cloud*. Keamanan adalah aspek yang sangat penting pada *cloud* untuk menjaga integritas, kerahasiaan, ketersediaan, dan privasi data pada *cloud*.

2.3 Network Forensic

Forensik jaringan adalah ilmu yang mempelajari tentang pencatatan, perekaman, dan analisa dari trafik jaringan untuk mendeteksi gangguan serta menginvestigasi gangguan tersebut [11]. Forensik jaringan melakukan pemantauan trafik jaringan, apabila ada anomali yang terjadi pada trafik maka dengan forensik jaringan dapat menentukan sifat atau karakteristik dari serangan tersebut. Forensik jaringan bertujuan untuk mengumpulkan bukti serangan untuk menangkap pelaku penyerangan dan diadili apabila dibawa ke ranah hukum.

Tools forensik jaringan dapat digunakan untuk merekonstruksi ulang kejadian ketika penyerangan terjadi secara berurutan. Dari rekonstruksi ulang tersebut akan didapatkan informasi penting seperti alamat IP, operasi sistem, dan pola penyerangan. Dari informasi yang didapat ini akan digunakan untuk mencegah

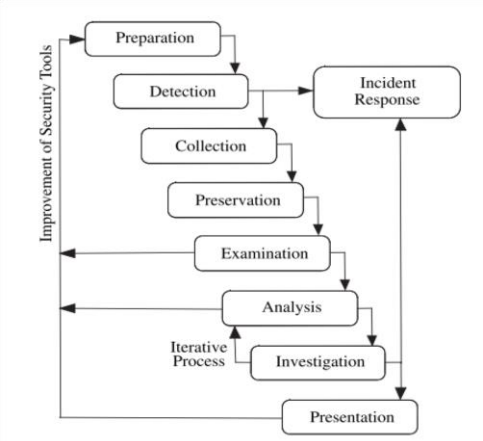
serangan yang sama di masa yang akan datang. Forensik jaringan dapat digunakan untuk menganalisa bagaimana penyerangan terjadi, siapa yang terlibat dalam penyerangan, durasi penyerangan, dan metode yang digunakan untuk menyerang. Selain itu forensik jaringan dapat digunakan sebagai *tool* untuk memonitor aktifitas *user*, menganalisa transaksi bisnis dan penentuan sumber masalah performa.

Keamanan jaringan melindungi sistem terhadap serangan dimana forensik jaringan fokus pada perekaman alat bukti penyerangan. Dalam forensik jaringan data *log* pada jaringan akan diperoleh dari produk keamanan jaringan yang ada untuk kemudian dianalisa untuk mendapatkan karakteristik atau pola dari serangan dan menginvestigasi pelaku penyerangan. Proses ini dapat membantu meningkatkan kualitas *tools* pada jaringan dengan menutup celah yang telah dimasuki oleh pelaku penyerangan.

Dalam forensik jaringan, *network investigator* dan penyerang berada dalam tingkat yang sama. Artinya ialah penyerang menggunakan sebuah *tools* untuk menyerang *server* dan *network investigator* akan menggunakan *tools* yang sama juga untuk menganalisa serangan tersebut. Konsep dari forensik jaringan adalah menganalisa data trafik yang masuk dan keluar dari satu *host* ke *host* yang lain dan menganalisa data trafik yang masuk ke *firewall* atau *intrusion detection system* pada suatu jaringan.

Forensik jaringan penting dilakukan karena dengan forensik jaringan akan membuat penyerang lebih berhati-hati dalam menyerang untuk menutup identitasnya yang akan membuat serangan akan lebih rumit dilakukan.

Tools analisa forensik jaringan atau NFATs(*Network Forensic Analysis Tools)* [12] memungkinkan admin untuk memonitor jaringan, mendapatkan semua informasi tentang *anomaly* trafik, membantu investigasi kejahatan jaringan dan membantu dalam menghasilkan hasil respon insiden yang sesuai. NFAT juga membantu menganalisa pencurian dan penyalahgunaan *resources*, memprediksi serangan berikutnya terjadi, melakukan penilaian risiko dan mengevaluasi kinerja jaringan.



Gambar 2.3 Model proses untuk forensik jaringan [12]

Digambarkan model proses untuk forensik jaringan dimulai dari *preparation, detection, incident response, collection, preservation, examination, analysis, investigation,* dan *presentation* [12].

Preparation, forensik jaringan akan dapat bekerja ketika *network security tools* seperti *firewalls, packet analyzers, intrusion detection system* dipasang dan disebar pada titik-titik vital jaringan seperti *server*. Ketika *tools* ini tidak ada pada jaringan maka forensik jaringan tidak dapat dilakukan.

Detection, pada tahapan ini *alerts* akan terjadi bila sistem mendeteksi suatu *anomaly*. Bila *anomaly* terjadi maka data *anomaly* tersebut akan dianalisa dengan berbagai ketentuan atau *parameter.* Setelah dianalisa maka akan ditentukan apakah *anomaly* tersebut serangan atau hanya trafik data normal. Apabila data trafik tersebut memang serangan maka proses forensik jaringan akan dilanjutkan namun apabila trafik data tersebut merupakan trafik normal yang artinya terjadi *false alarm* maka proses forensik jaringan dihentikan. *Tools* yang digunakan pada tahapan ini adalah Wireshark, TCPDump, Snort, Bro, P0f, PADS, Ntop dan Sebek.

Incident response, pada tahapan ini apabila terjadi serangan maka sistem akan merespon serangan tersebut. Respon sistem bergantung terhadap tipe serangan yang terjadi. Apabila serangan tersebut merupakan tipe serangan yang baru maka sistem akan mengumpulkan informasi tentang serangan tersebut untuk membuat suatu pertahanan apabila serangan tersebut menyerang kembali di masa yang akan datang. Pada tahapan ini akan ditentukan apakah proses forensik jaringan diteruskan atau dihentikan. Proses forensik jaringan dapat dihentikan ketika

serangan yang menyerang merupakan serangan kecil dan bisa dilanjutkan apabila serangan tersebut mengakibatkan kerusakan sistem dan membutuhkan tindakan lebih lanjut untuk memperbaiki kerusakan tersebut.

Collection, pada tahapan ini data trafik akan dikumpulkan dari *network security tools.* Tahapan ini sangat penting untuk mendapatkan jejak dari serangan yang terjadi karena data trafik akan berubah secara cepat dan jejak yang ditimbulkan oleh serangan tadi mungkin tidak akan terjadi lagi dilain waktu, sehingga pada tahap ini diperlukan *hardware* dan *software* yang cepat dan handal untuk mengumpulkan jejak serangan yang digunakan sebagai bukti. *Tools* yang bekerja pada tahap *collection* adalah Wireshark, TCPDump, Snort, PADS, NfDump, Sebek, SiLK, TCPFlow dan Bro.

Preservation, pada tahap ini data trafik hasil dari *logs* korban akan disimpan dalam sebuah perangkat *backup.* Data trafik ini akan disalin ke sebuah perangkat forensik jaringan untuk kemudian diuji coba melakukan serangan yang diduga apakah hasilnya akan sama atau tidak. Data trafik asli akan diawetkan sehingga tidak akan disentuh dan diuji coba, hal ini agar menjaga keaslian dari data trafik tersebut. *Tools* yang bekerja pada tahap ini ialah Wireshark, TCPDump, Snort, PADS, NfDump, Sebek, SiLK, TCPFlow, dan Bro.

Examination, pada tahap ini data trafik yang disalin tadi akan dilakukan analisis. Masalah yang sering terjadi pada tahap ini adalah informasi yang berlebihan dan waktu yang saling tindih dalam artian terjadi secara bersamaan sehingga memerlukan suatu perkiraan. Bukti yang terkumpul akan diekstrak untuk mendapatkan indikator yang spesifik dari serangan yang terjadi. *Tools* yang bekerja pada tahap ini ialah Wireshark, TCPDump, TCPFlow, Flow-tools, PADS, Argus, NfDump, Nessus, Sebek, Ntop, TCPTrace, NetFlow, Ngrep, SiLK, TCPStat, TCPDstat, TCPXtract,P0f, TCPReplay, Snort, Bro, dan Nmap.

Analysis, pada tahap ini akan dilakukan analisa terhadap indikator yang didapatkan dari proses *examination* . Indikator-indikator ini akan dikumpulkan dan dicari hubungan antar indikator untuk menyimpulkan sebuah pengamatan dengan menggunakan pola serangan yang ada. Beberapa indikator penting akan berhubungan dengan pembentukan koneksi jaringan, *query* DNS, fragmentasi paket, protokol, dan sidik jari dari operasi sistem. Pola serangan akan disatukan,

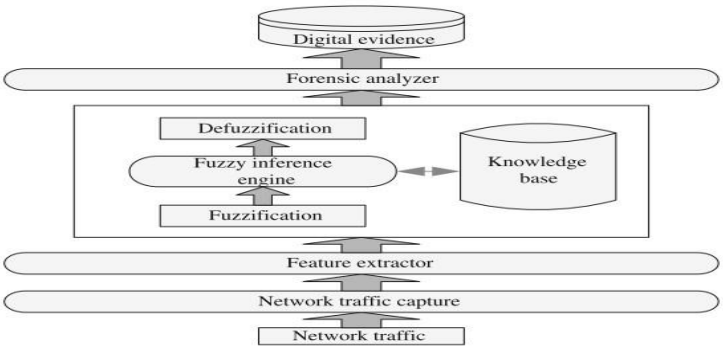
direkonstruksi dan dilakukan uji coba untuk mendapatkan informasi bagaimana serangan ini terjadi dan apa tujuan dari serangan. Tools yang bekerja pada tahap ini ialah Wireshark, TCPDump, TCPFlow, Flow-tools, PADS, Argus, NfDump, Nessus, Sebek, Ntop, TCPTrace, NetFlow, Ngrep, SiLK, TCPStat, TCPDstat, TCPXtract,P0f, TCPReplay, Snort, Bro, dan Nmap.

Investigation, Pada tahap ini *network forensic investigator* akan menganalisa dan menentukan jalur yang dilalui oleh penyerang sampai ke jaringan korban.. Data dari tahap *analysis* akan digunakan dan disatukan dengan data pada tahap ini untuk mendapatkan kesimpulan. Hasil data trafik digunakan untuk mendapatkan atribut penyerangan dan menentukan identitas dari penyerang.

Presentation, tahap ini merupakan tahap akhir dari proses forensik jaringan. Pada tahap ini akan dipresentasikan hasil dari pengamatan terhadap data yang didapat. Bukti-bukti yang didapat akan dipresentasikan untuk mengadili pelaku penyerangan. Dan juga mempresentasikan bagaimana serangan tersebut terjadi dan bagaimana pola serangan tersebut sehingga dapat mencegah serangang yang sama di masa yang akan datang.

2.4 Fuzzy Expert System for Network Forensics

Logika *fuzzy* adalah teknik yang digunakan untuk menghadapi dan menyesuaikan dengan penalaran manusia dan proses pengambilan keputusan [7]. Logika fuzzy adalah logika yang mempunyai nilai kabur atau samar sesuai dengan arti dari kata *fuzzy* yang artinya samar atau kabur.Sebelum logika *fuzzy* ditemukan, ada logika tegas yang dipakai manusia untuk kehidupan sehari-hari. Logika tegas memiliki nilai 0 dan 1 saja atau benar dan salah.



Gambar 2.4 Skema forensik jaringan menggunakan logika fuzzy [7]

NFS-FLES *(Network Forensics System – Fuzzy Logic Expert System)* mempunyai 7 bagian yaitu : *Traffic capture*, fitur ekstrasi, *fuzzification, fuzzy interfence engine*,*knowledge base*, *defuzzification,* dan analisa forensik [7].

Traffic capture, pada bagian ini melakukan penangkapan trafik. Data trafik yang ditangkap akan dianalisa untuk melakukan forensik jaringan. *Tools* yang digunakan pada bagian ini adalah Snort, Wireshark, TCPdump dan lain-lain.

Fitur ektraksi, pada bagian ini dilakukan ekstraksi dari data trafik yang ditangkap. Akan banyak fitur dari data trafik yang bisa diekstraksi. Pada [8], data forensik dapat didefinisikan menjadi 6 bagian yaitu fd_{pro}(informasi proses), fd_{call}(informasi panggilan sistem), fd_{status}(informasi status sistem), fd_{net}(informasi deteksi jaringan), fd_{file}(informasi operasi file), fd_{vul}(informasi pemindai kerentanan).

Fuzzification, pada bagian ini tiap input nilai dari variabel akan dilakukan *fuzzification* ke nilai bahasa penalaran manusia kemudian dilakukan proses pengambilan keputusan sesuai dengan yang ada pada *knowledge base*.

Fuzzy interference engine dan *knowledge base,* didalam *knowledge base* disimpan aturan-aturan dalam jaringan kemudian digunakan oleh *fuzzy interference engine* untuk mendapatkan suatu kesimpulan. Dalam sistem logika *fuzzy* algoritma yang digunakan dalam aturan adalah IF-THEN.

Defuzzification, pada bagian ini data yang telah dilakukan proses *fuzzification* akan dilakukan *defuzzification* ke bentuk nilai tegas.

Analisa forensik, pada bagian ini akan menganalisa dan menentukan apakah data trafik yang telah dianalisa termasuk serangan atau bukan. Apabila termasuk serangan maka bukti-bukti serangan akan dikumpulkan untuk kemudian mengadili penyerang.

Logika fuzzy mempunyai keunggulan menganalisis data forensik dibandingkan dengan metode lain. Logika fuzzy dapat mendeteksi 91,5% serangan dalam berbagai kondisi dan menghasilkan proses pengambilan keputusan yang efisien dan cepat diimplementasikan untuk menganalisis data forensik, sehingga logika fuzzy adalah metode yang cocok digunakan untuk menganalisis data forensic [7].

2.5 Snort

Snort adalah sebuah sistem deteksi dan proteksi yang digunakan untuk melindungi sistem dari serangan. Snort dapat dijalankan dalam 3 mode, yaitu *sniffer, packet logger, network intrusion detection system (NIDS)*.

1. *Sniffer Mode*

   Mode yang berfungsi untuk membaca dan menampilkan paket jaringan secara kontinyu.

2. *Packet Logger Mode*

   Mode yang berfungsi untuk menyimpan paket jaringan dalam bentuk *logs* pada *disk.*

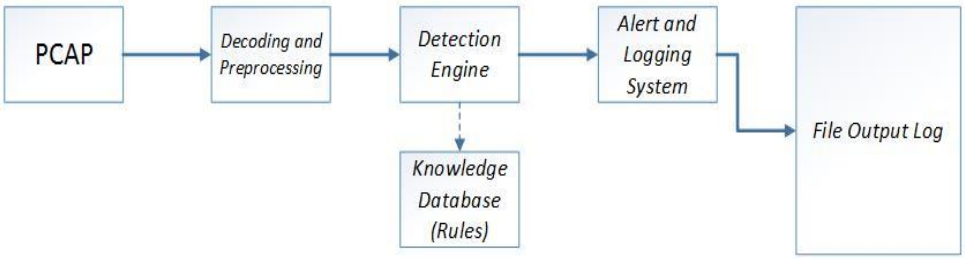
3. *Network Intrusion Detection System (NIDS)*

   Mode yang berfungsi untuk menganalisa paket jaringan dan mendeteksi apabila terdapat anomali pada paket jaringan. Pada mode NIDS, diperlukan *rules* untuk mendeteksi anomali yang terjadi.

Mode NIDS akan dipakai penulis dalam melakukan pembuktian serangan *brute force* pada *cloud* untuk menentukan paket serangan dan paket normal yang terjadi pada *cloud.*

2.5.1 Cara Kerja Snort

Snort akan melakukan beberapa proses pada paket data untuk mendeteksi *intrusion* dalam sebuah jaringan[13]. Alur proses snort dalam mendeteksi *intrusion* ditunjukkan pada Gambar 2.6.



Gambar 2.5 Cara kerja snort pada mode NIDS[13]

a. PCAP

Snort menggunakan PCAP *library* untuk menangkap paket pada jaringan.

b. *Decoding and Preprocessing*

Decoding berfungsi untuk mengambil data dan melakukan pemisahan data protokol yang dapat diproses lebih lanjut. Paket yang telah dipisahkan tersebut diteruskan ke *preprocessing* yang bertugas untuk memberikan *pre-alert* mengenai paket jaringan yang ditangkap sesuai dengan format protokol untuk diteruskan ke *Detection Engine.*

c. *Detection Engine*

Detection Engine merupakan bagian terpenting snort. Pada *detection engine* terdapat *rules* snort mengenai anomali atau serangan sehingga paket yang telah tertangkap akan dibandingkan dengan *rules* yang ada pada *detection engine.* Apabila paket jaringan yang tertangkap memiliki kesamaan pola dengan *rules* snort maka snort akan melakukan tindakan, yaitu *alert* dan *log.* Berikut contoh *rules* snort:

alert icmp any any -> $Home any (msg: "ICMP Packet Found";)

- *alert* menunjukkan perintah snort memberikan *alert* pada *console*
- *icmp* menunjukkan jenis protokol
- *any* menunjukkan *IP Address* yang masuk
- *any* menunjukkan *port* serangan yang masuk
- *$HOME* menunjukkan *IP Address* dari perangkat yang dilindungi
- *any* menunjukkan port dari perangkat yang dilindungi
- *msg:"ICMP Packet Found"* adalah pesan yang akan muncul.

d. *Alert and Logging System*

Ketika paket jaringan yang tertangkap mempunyai pola yang sama dengan *rules* snort maka snort akan mengirimkan alert kepada perangkat atau *server* yang dilindungi snort. Ketika selesai menjalankan snort pada satu sesi, kumpulan *alert* akan disimpan dalam satu *file log. File log* akan disimpan pada *hardisk* perangkat.

2.5.2 Performa Snort Sebagai *Intrusion Detection System*

Performa pengujian trafik menggunakan snort terbagi menjadi 4 jenis yang menentukan tingkat akurasi snort sebagai *intrusion decection system.* Berikut jenis-jenis performa snort dalam pengujian trafik berdasarkan *confusion matrix*[13].

a. *True Positive*

True Positive (TP) adalah kondisi data serangan yang terindikasi sebagai serangan oleh sistem.

b. *False Positive*

False Positive (FP) adalah kondisi data normal yang terindikasi sebagai serangan oleh sistem.

c. *False Negative*

False Negative (FN) adalah kondisi data serangan tidak terindikasi sebagai serangan oleh sistem.

d. *True Negative*

True Negative (TN) adalah kondisi data normal dan sistem tidak mendeteksi adanya serangan.

Tabel 1

Confusion Matrix[13]

Kategori Data	**Prediksi**	
	Normal	Serangan
Normal	*TN*	*FP*
Serangan	*FN*	*TP*

Berdasarkan tabel 5 terdapat persamaan perhitungan tingkat akurasi *confusion matrix*. Persamaan tersebut sebagai berikut.

a. *True Positive Rate* (TPR)

$$\text{TPR} = \frac{TP}{TP+FN} \qquad \ldots\ldots(2)$$

b. *True Negative Rate* (TNR)

$$\text{TNR} = \frac{TN}{TN+FP} \qquad \ldots\ldots(3)$$

c. *False Positive Rate* (FPR)

$$\text{FPR} = \frac{\boldsymbol{FP}}{\boldsymbol{TN+FP}} \quad \ldots\ldots(4)$$

d. *False Negative Rate* (FNR)

$$\text{FNR} = \frac{\boldsymbol{FN}}{\boldsymbol{TP+FN}} \quad \ldots\ldots(5)$$

e. *Precision*

$$\textit{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad \ldots\ldots(6)$$

f. *Accuracy*

$$\textit{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad \ldots\ldots(7)$$

BAB III

METODOLOGI

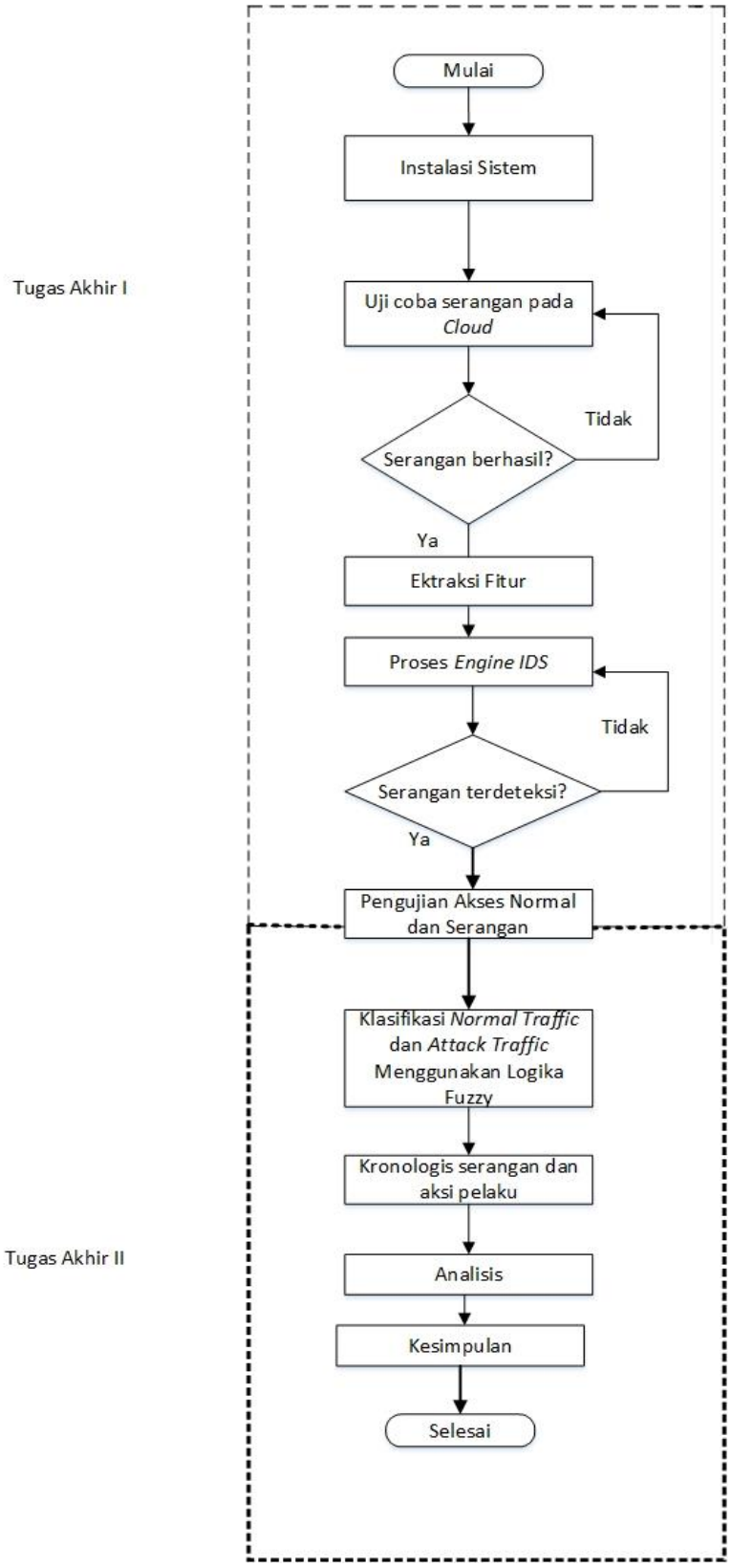
3.1 Pendahuluan

Pada bab ini dijelaskan mengenai metodologi yang digunakan dalam menyelesaikan penelitian. Penelitian ini dirancang untuk dapat membuktikan serangan *brute force* yang terjadi pada sebuah *cloud* yang bersifat publik dengan menggunakan metode logika fuzzy. Penelitian akan melalui beberapa tahapan yang dipresentasikan ke dalam kerangka kerja. Pada kerangka kerja menjelaskan tahapan penelitian yang meliputi perancangan, pengembangan, pengujian, dan proses menganalisis.

3.2 Kerangka Kerja Penelitian

Pembuktian serangan *brute force* pada *cloud* dengan mengimplementasikan logika fuzzy akan melalui beberapa tahapan kerja. Tahapan kerja pada penelitian ini mengikuti kerangka kerja yang sudah dirancang agar penelitian berjalan secara terstruktur.

Tahapan awal dari penelitian adalah melakukan instalasi dan konfigurasi *cloud* pada PC A sebagai server, instalasi kali linux pada PC B sebagai penyerang, dan membangun topologi untuk *cloud*. Tahap selanjutnya adalah melakukan uji coba serangan *brute force* pada *cloud* menggunakan PC B, menangkap trafik jaringan menggunakan *tool wireshark* pada PC A dan B ketika serangan berlangsung. Tahap selanjutnya adalah menganalisa hasil *capture tool wireshark* pada PC A dan B untuk menentukan pola serangan *brute force* yang dilakukan oleh PC B. Tahap selanjutnya ialah menggunakan *engine IDS Snort* pada PC A untuk memvalidasi serangan yang dilakukan oleh PC B. Tahapan selanjutnya adalah membedakan trafik serangan dengan trafik normal menggunakan logika fuzzy. Tahap terakhir adalah menyimpulkan kronologis serangan dan aksi yang dilakukan oleh PC B. Berikut Gambar 3.1, yang merupakan bentuk diagram alir dari kerangka kerja yang dilakukan:



Gambar 3.1 Kerangka Kerja Penelitian

3.3 Instalasi Sistem

Penelitian tugas akhir ini menggunakan perangkat keras dan perangkat lunak serta perancangan topologi *cloud* untuk mendukung sistem yang dipakai pada penelitian.

3.3.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian adalah 1 PC sebagai *server*, 1 PC sebagai penyerang, 1 *router* untuk membangun jaringan *cloud*. Spesifikasi kebutuhan perangkat keras dijelaskan pada tabel 2.

Tabel 2

Spesifikasi Kebutuhan Perangkat Keras

Sistem	***Tools***	**Keterangan**
Server	PC	Ubuntu 16.04
Penyerang	PC	Kali Linux v.4.13
Jaringan	Router	Huawei HG8245A

3.3.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian merupakan *tools* owncloud, snort, wireshark dan pyhton. Spesifikasi kebutuhan perangkat lunak dijelaskan pada tabel 3.

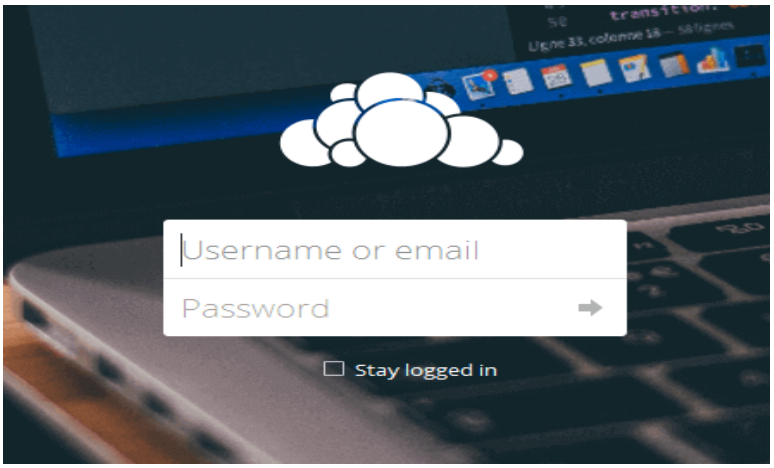
Tabel 3

Spesifikasi Kebutuhan Perangkat Lunak

Sistem	**Tools**	**Keterangan**
Cloud	Owncloud	Owncloud X
Capture Traffic	Wireshark	Wireshark 2.44
NIDS	Snort	Snort 2.9.11
Normal Traffic dan *Attack Traffic*	Python	Python 3

3.3.3 Owncloud sebagai Cloud

Owncloud adalah perangkat lunak *open source client-server* untuk membuat layanan *cloud* yang dapat dipasang dan dioperasikan pada *server*. Owncloud dibangun menggunakan bahasa PHP dan akan bekerja dengan sistem database seperti MySQL, MariaDB, SQLite dan lain-lain. Owncloud bersifat fleksible sehingga pemilik *cloud* dapat mengatur konfigurasi *cloud*.



Gambar 3.2 Tampilan *Login* Owncloud

3.3.4 Snort sebagai NIDS

Snort adalah sebuah sistem deteksi dan proteksi yang digunakan untuk melindungi sistem dari serangan. Snort dapat dijalankan dalam 3 mode, yaitu *sniffer, packet logger, network intrusion detection system (NIDS)*. Mode yang akan digunakan pada penelitian ini ialah mode NIDS.

Pada mode NIDS, snort bekerja dengan komponen *packet decoder, pre-processor, detection engine* dan *output.* Komponen utama yang digunakan pada penelitian ini ialah *detection engine* untuk membaca dan mendeteksi data dalam *traffic* apakah termasuk serangan atau bukan. Pada *detection engine* terdapat *rules, rules* adalah aturan-aturan mengenai pola yang terdapat dalam data pada *traffic* yang masuk atau keluar. *Rules* inilah yang akan mencocokkan pola data dari *traffic* yang masuk maupun yang keluar apabila pola datanya sama dengan yang ada di *rules* maka snort akan mendeteksi *traffic* data tersebut. Snort *rules* terdapat 2 bagian, yaitu *rule header* dan *rule options. Rule header* berisi *rules action, protocol, source* dan *destination* IP, dan *source* dan *destination port*. Pada *rule option* berisi

alert message yang berfungsi untuk menentukan *rules action* yang akan dilakukan. *Rules action* terdiri dari 8 *action* yang digambarkan pada tabel 4.

Tabel 4. *Action Rule* Snort

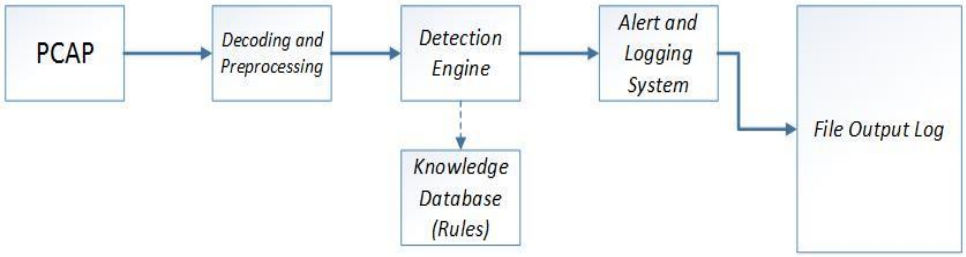
Rules Action	**Deskripsi**
Alert	Memberikan *alert* apabila terdapat paket yang mempunyai pola yang sama pada *rules*
Log	Membuat *log* paket
Pass	Mengabaikan paket
Active	Mengaktifkan *dynamic rule* lain
Dynamic	*Rule* siaga sampai diaktifkan oleh *Activate rule* kemudian bertindak sebagai *log rule*
Drop	Memblokir paket dan membuat log paket
Reject	Memblokir paket, *log it*
Sdrop	Memblokir paket

Alert tcp any any -> $HOME_NET 80
(msg:”Brute Force Detected”; sid:1000001;rev:1;)

Gambar 3.3. Contoh *Rules* Snort

Pada *rules* diatas, menunjukkan *rules action* adalah *alert* yang berasal dari protokol TCP dengan *source* IP *any* dengan *destination* IP $HOME_NET, *source*

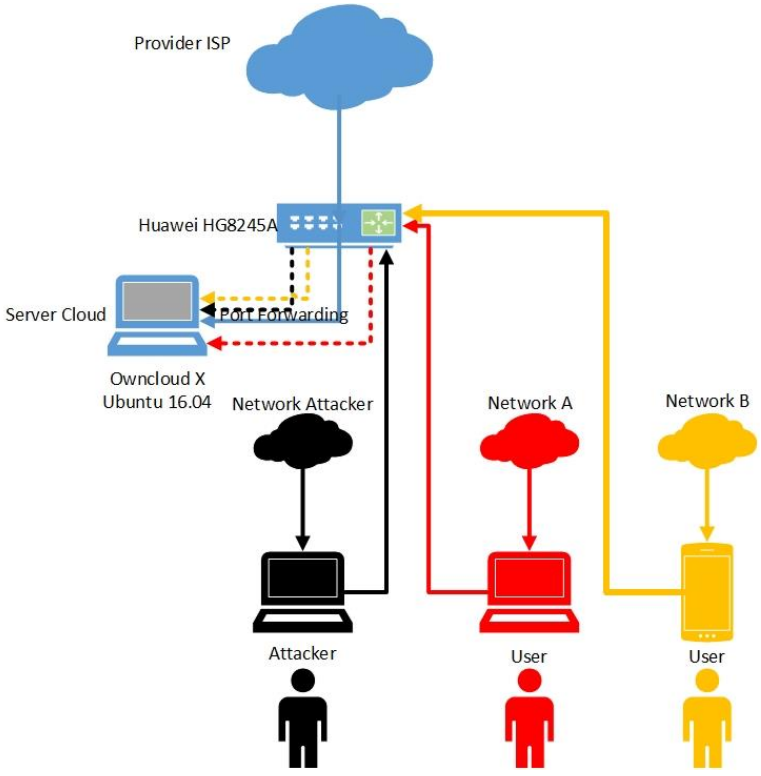
port any dengan *destination port* 80. (msg:"Brute Force Detected"; sid:1000001;rev:1;) memberikan pesan "Brute Force Detected" apabila pola dari data yang masuk sama dengan pola yang ada di *rules*.



Gambar 3.4 Cara Kerja Snort

3.4 Perancangan Topologi

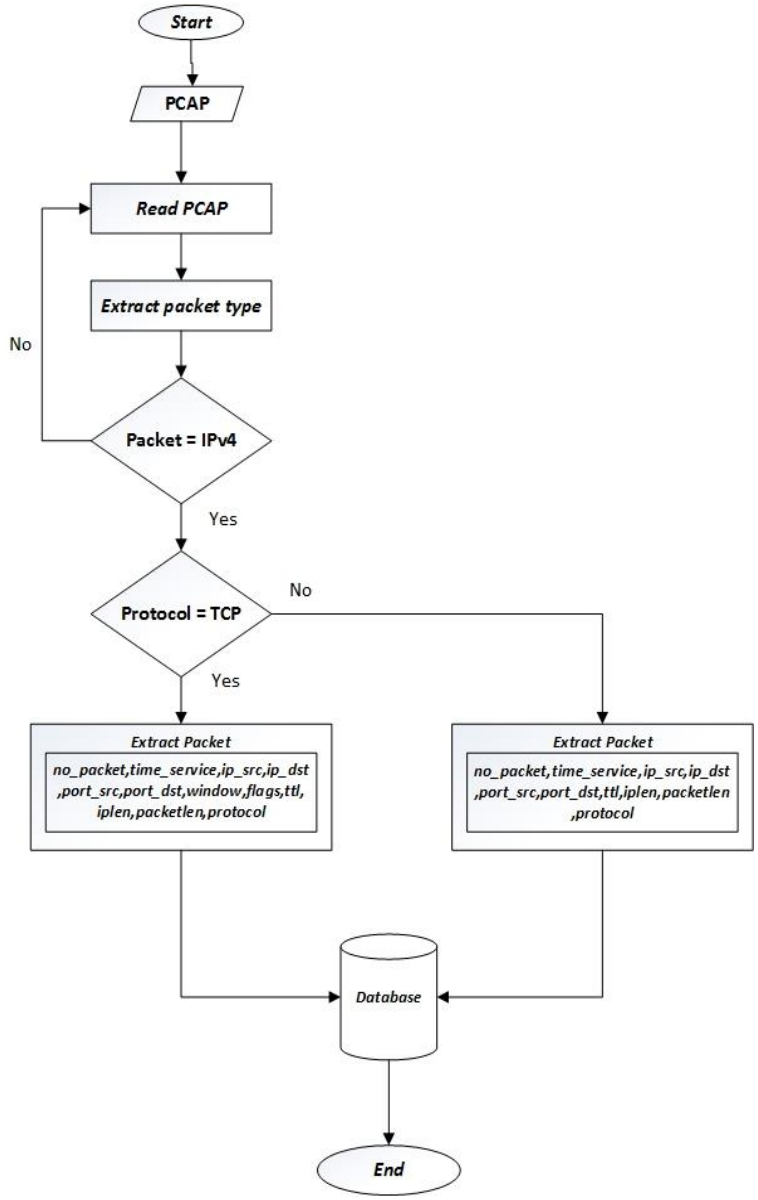
Pada penelitian tugas akhir, akan dirancang topologi untuk membangun *cloud*. Topologi menggunakan 1 *router* dengan jaringan dari PT Telkom Indonesia kemudian di-*port forwarding* ke IP *server cloud* sehingga *cloud* dapat diakses dari luar jaringan (bersifat publik). Berikut Gambar 3.6 topologi penelitian tugas akhir.



Gambar 3.5 Topologi Penelitian

3.5 Feature Extraction

Setelah pengujian serangan *brute force* yang direkam menggunakan wireshark dalam jenis file *packet capture* (pcap), penelitian dilanjutkan dengan melakukan *feature extraction*. *Feature extraction* berfungsi untuk membuat file *Comma Separated Value (CSV)* dari *raw* data hasil *capture* wireshark (pcap) yang bertujuan untuk mengenali pola serangan dari dataset yang telah dilakukan. Pada [14], *feature extraction* dilakukan menggunakan bahasa pemrograman python dengan *flowchart* sebagai berikut.



Gambar 3.6 *Flowchart Feature Extraction [14]*

Atribut yang diekstrak pada *feature extraction* dibagi menjadi dua yaitu atribut pada protokol Tcp dan protocol non Tcp. Berikut adalah atribut yang diekstrak pada serangan *brute force*.

Tabel 5

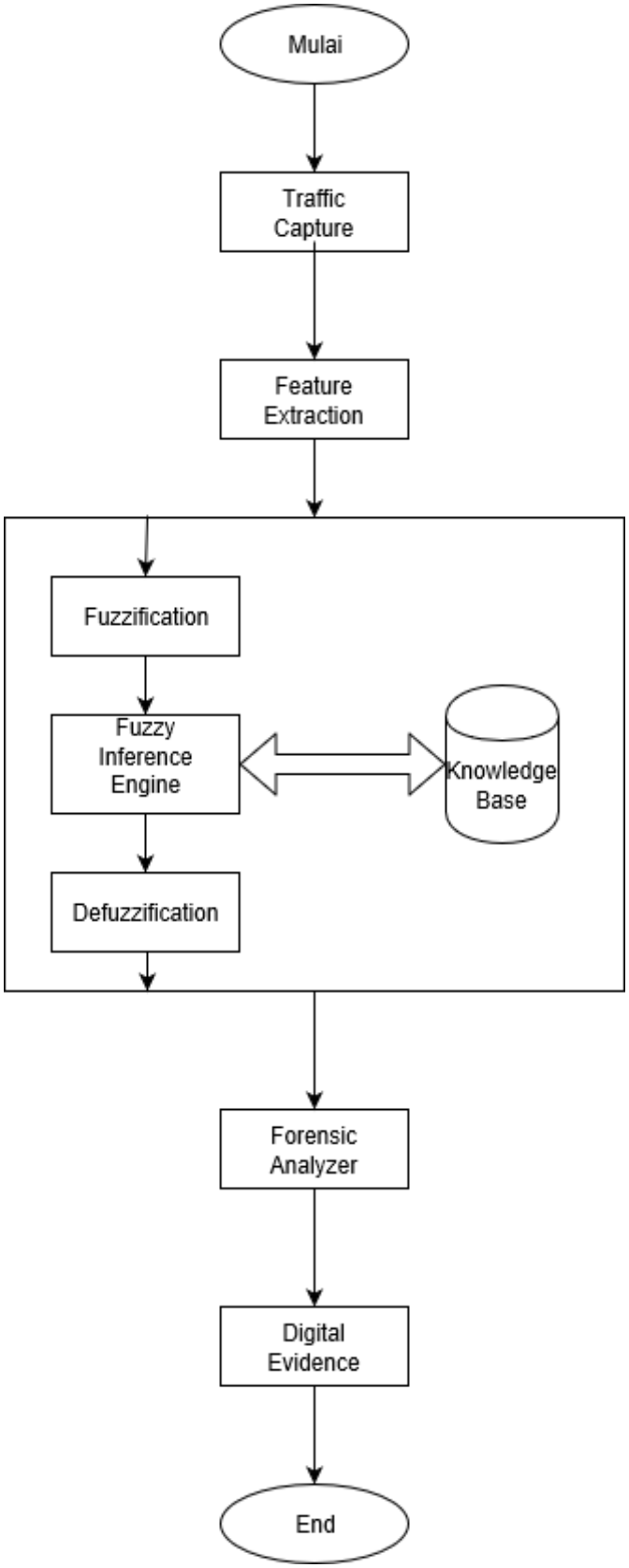
Atribut *Feature Extraction* [14]

No	**Atribut *Protocol* TCP**	**No**	**Atribut *Protocol TCP***
1	*Packet Number*	8	*Window*
2	*Timestamp*	9	*Flags*
3	*Service*	10	*Ttl*
4	*Ip source*	11	*Ip length*
5	*Ip destination*	12	*Tcp length*
6	*Port source*	13	*Protocol*
7	*Port destination*	14	*Payload*

No	**Atribut *Protocol Non-TCP***	**No**	**Atribut *Protocol Non-Tcp***
1	*Packet Number*	7	*Port destination*
2	*Timestamp*	8	*Ttl*
3	*Service*	9	*Ip length*
4	*Ip source*	10	*Tcp length*
5	*Ip destination*	11	*Protocol*
6	*Port source*	12	*Payload*

3.6 Perancangan Logika Fuzzy untuk Forensik Jaringan

Logika fuzzy untuk forensik jaringan dibagi menjadi tujuh tahapan yaitu : *the traffic capture, the feature extraction, the fuzzification, the fuzzy inference and knowledge base, the defuzzification, forensic analyzer*, dan *digital evidence* [7].



Gambar 3.7 *Flowchart* Perancangan Logika Fuzzy untuk Forensik Jaringan

3.6.1 *Traffic Capture*

Pada tahap ini digunakan wireshark untuk menangkap data trafik jaringan dari skenario pengujian.

3.6.2 *Feature Extraction*

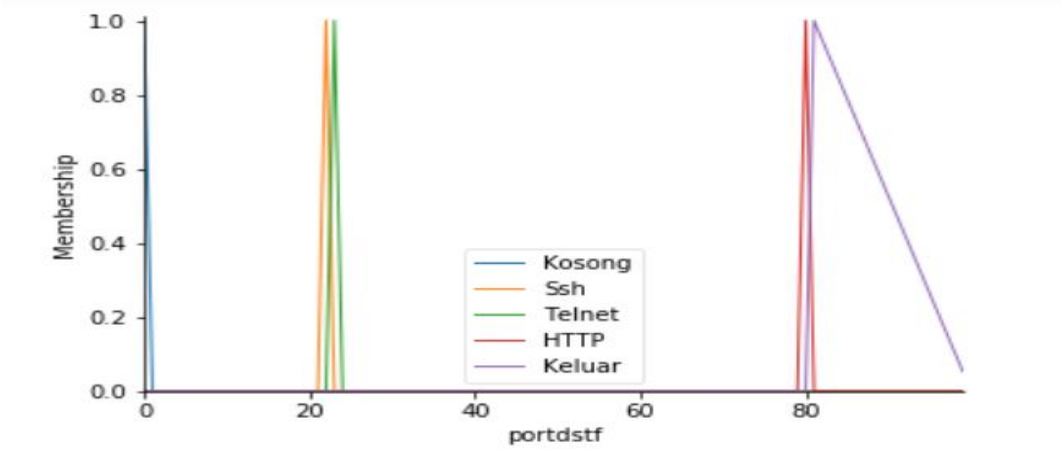
Pada tahap ini dari data trafik jaringan yang telah ditangkap menggunakan wireshark dalam bentuk file pcap akan disalin ke bentuk file csv dengan tujuan mengekstrak fitur-fitur yang ada pada *raw* data (pcap) menggunakan Bahasa pemrograman Python.

3.6.3 *Fuzzification*

Setiap nilai dari input atau fitur-fitur atau variabel dari data trafik jaringan yang dalam bentuk nilai bilangan real atau nilai *crisp* (tegas) perlu diterjemahkan kedalam nilai *linguistic* atau nilai fuzzy.Variabel-variabel ini akan disatukan dengan *rules* fuzzy dengan konjungsi AND menggunakan metode min dengan konsekuen menggunakan metode max.

- Variabel Port Destination

  Variabel input Port Destination didefinisikan dengan variabel linguistik sebanyak lima variabel yaitu : Kosong, SSH, Telnet, HTTP, dan Keluar. Fungsi keanggotaan variabel port destination dijelaskan pada gambar 3.8 berikut :



Gambar 3.8 Grafik Fungsi Keanggotaan Variabel Port Destination

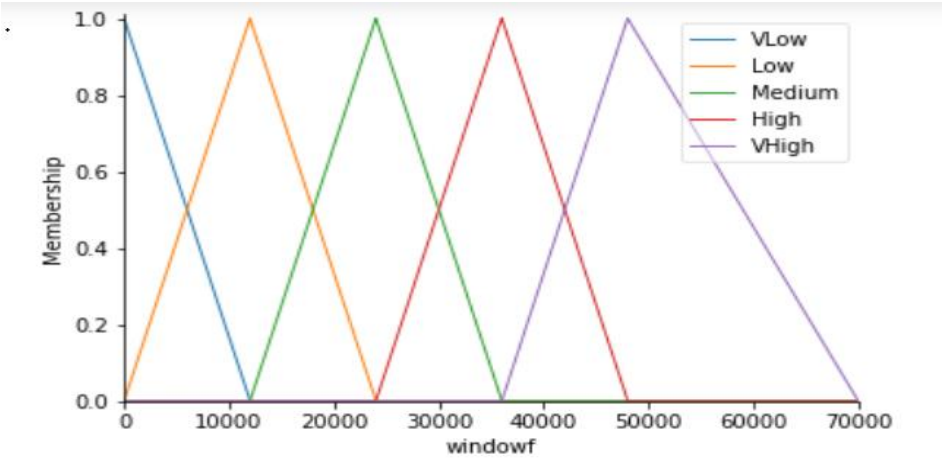
Dari gambar 3.8, nilai derajat keanggotaan variabel port destination dapat ditentukan pada tabel persamaan berikut:

Tabel 6 Persamaan Derajat Keanggotaan Variabel Port Destination

Variabel Linguistik	Rentang Nilai	Derajat Keanggotaan (μ)
Kosong	x = 0	μ Kosong= 1
SSH	x = 22	μ SSH = 1
Telnet	x = 23	μ Telnet= 1
Http	x = 80	μ Http= 1
Keluar	80 < x <= 100.000	μ Keluar= 1

❖ Variabel Window

Variabel input Window didefinisikan dengan variabel linguistik sebanyak lima variabel yaitu : *very low*, *low*, *medium*, *high*, dan *very high*. Fungsi keanggotaan variabel window dijelaskan pada gambar 3.9 berikut :



Gambar 3.9 Grafik Fungsi Keanggotaan Variabel Window

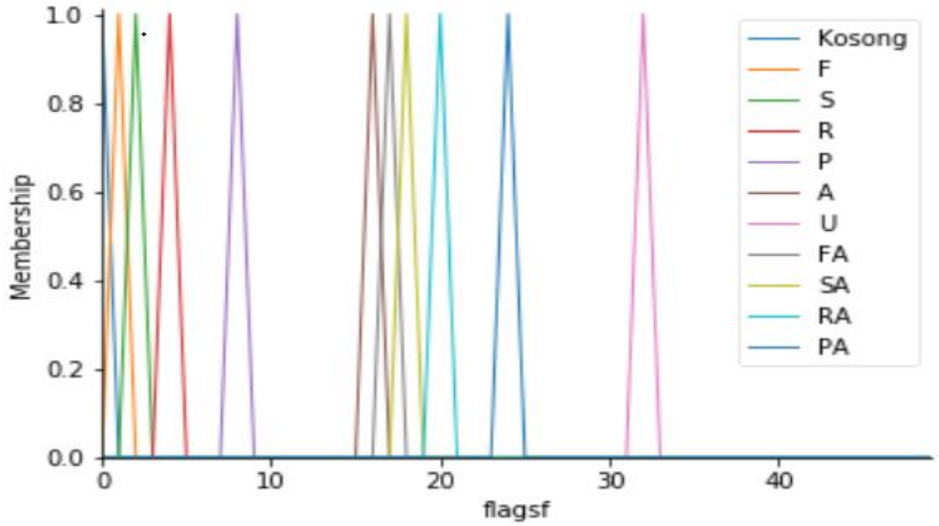
Dari gambar 3.9, nilai derajat keanggotaan variabel window dapat ditentukan pada tabel persamaan berikut:

Tabel 7 Persamaan Derajat Keanggotaan Variabel Window

Variabel Linguistik	**Rentang Nilai**	**Derajat Keanggotaan (μ)**
Very low	x <= 8000	μ Very low= 1
Very low ∩ low	0<=x<8000	μ Very low= (8000 - x) / (8000 - 0) μ Low= (x - 0) / (8000 - 0)
Low	0<=x<= 16000	μ Low= 1
Low ∩ medium	8000<=x<16000	μ Low= (16000 – x) / (16000 – 8000) μ Medium= (x – 8000) / (16000 – 8000)
Medium	8000<=x<=24000	μ Medium= 1
Medium ∩ High	16000<=x<24000	μ Medium= (24000 - x) / (24000 - 16000) μ High= (x - 16000) / (24000 - 16000)
High	16000<=x<32000	μ High= 1
High ∩ Very high	24000<=x<32000	μ High= (32000 – x) / (32000 – 24000) μ Very High= (x – 24000) / (32000 – 24000)
Very High	24000<=x<=68000	μ Very High= 1

❖ Variabel Flags

Variabel input Flags didefinisikan dengan variabel linguistik sebanyak sebelas variabel yaitu : Kosong, F, S, R, P, A, U, FA, SA, RA, dan PA. Fungsi keanggotaan variabel port destination dijelaskan pada gambar 3.10 berikut :



Gambar 3.10 Grafik Fungsi Keanggotaan Variabel Flags

Dari gambar 3.10, nilai derajat keanggotaan variabel port destination dapat ditentukan pada tabel persamaan berikut:

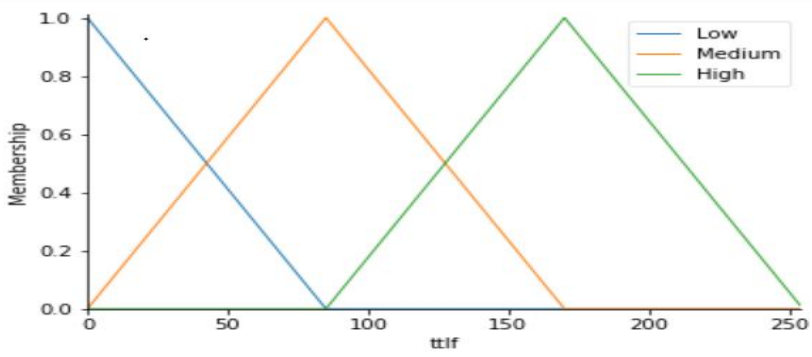
Tabel 8 Persamaan Derajat Keanggotaan Variabel Flags

Variabel Linguistik	Rentang Nilai	Derajat Keanggotaan (μ)
Kosong	x = 0	μ Kosong= 1
F	x = 1	μ F = 1
S	x = 2	μ S= 1
R	x = 4	μ R= 1
P	x < 8	μ P= 1

A	x = 16	μ A= 1
U	x = 32	μ U= 1
FA	x = 17	μ FA= 1
SA	x = 18	μ SA= 1
RA	x = 20	μ RA= 1
PA	x = 24	μ PA= 1

❖ Variabel TTL (*Time To Live*)

Variabel input ttl (*time to live*) didefinisikan dengan variabel linguistik sebanyak tiga variabel yaitu : *low*, *medium*, dan *high*. Fungsi keanggotaan variabel ttl sebagai berikut :



Gambar 3.11 Grafik Fungsi Keanggotaan Variabel TTL

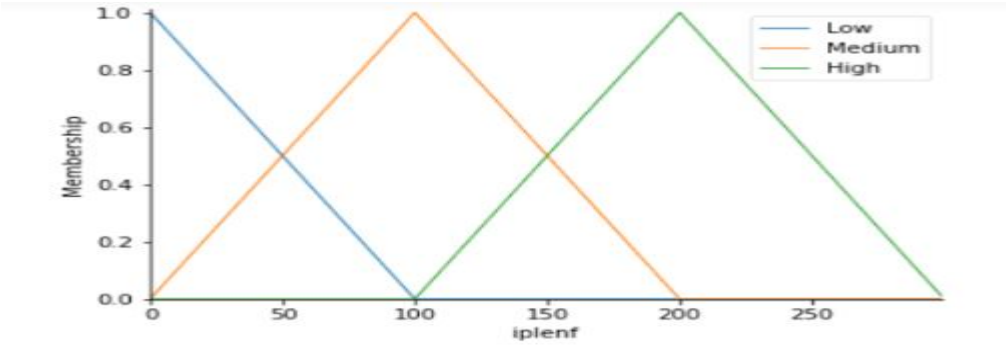
Tabel 9 Persamaan Derajat Keanggotaan Variabel *Time to Live*

Variabel Linguistik	**Rentang Nilai**	**Derajat Keanggotaan (μ)**
Low	x < 85	μ Low= 1
Low ∩ medium	0<=x<85	μ Low= (85 - x) / (85-0) μ Medium= (x - 0) / (85-0)

Medium	0<=x<170	μ Medium= 1
Medium ∩ high	85<=x<170	μ Medium= (170 – x) / (170 – 85) μ High= (x – 85) / (170 – 85)
High	85<=x<255	μ High= 1

❖ Variabel IP Length

Variabel input Ip *length* didefinisikan dengan variabel linguistik sebanyak tiga variabel yaitu : *low*, *medium*, dan *high*. Fungsi keanggotaan variabel Ip *length* sebagai berikut :



Gambar 3.12 Grafik Fungsi Keanggotaan Variabel Ip *Length*

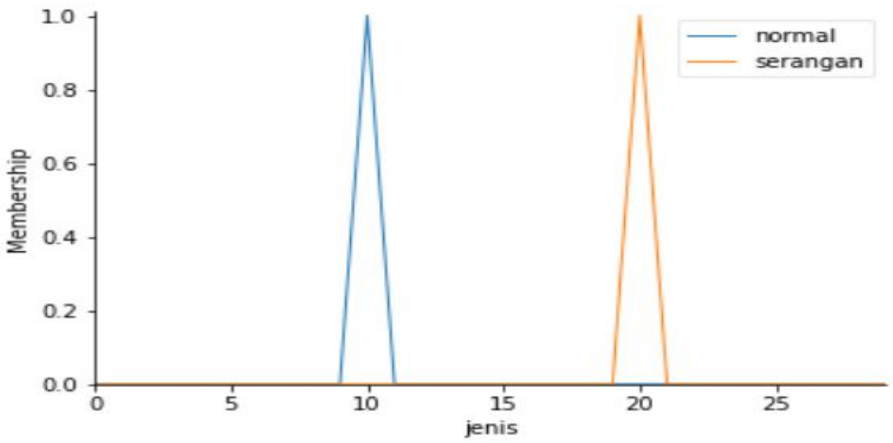
Tabel 10 Persamaan Derajat Keanggotaan Variabel IP *Length*

Variabel Linguistik	**Rentang Nilai**	**Derajat Keanggotaan (μ)**
Low	0<=x<100	μ Low= 1
Low ∩ medium	0<=x<100	μ Low= (100 - x) / (100 - 0) μ Medium= (x - 0) / (100-0)

Medium	0<=x<200	μ Medium= 1
Medium ∩ high	100<=x<200	μ Medium= (200 – x) / (200 – 100) μ High= (x – 100) / (200 – 100)
High	100<=x<=300	μ High= 1

❖ Variabel Jenis Trafik (*Output*)

Variabel jenis trafik memiliki dua variabel lingustik yaitu : trafik normal dan trafik serangan. Dengan bentuk fungsi keanggotaan berbentuk singleton yang artinya garis lurus yang mewakili konstanta tertentu. Nilai tunggal dari variabel trafik normal adalah 10 sedangkan untuk nilai tunggal dari variabel trafik serangan yaitu 20.



Gambar 3.13 Grafik Fungsi Keanggotaan Variabel *Output* Jenis Trafik.

3.6.4 *Fuzzy Inference Engine and Knowledge Base*

Pada tahap ini dibuat aturan-aturan yang sesuai dengan fungsi keanggotaan dari data variabel *input* yang telah dibentuk. Logika fuzzy untuk forensik jaringan menggunakan aturan IF-THEN dengan *output* pada penelitian ini ialah: Trafik normal dan trafik serangan. Berikut tabel aturan fuzzy pada forensik serangan *brute force* pada *cloud* (terlampir). Setelah aturan-aturan fuzzy terbentuk, proses

inference akan berlanjut ke fungsi implikasi yang menggunakan metode min pada *input* dan max pada *output*.

$$\mu_{sf}[x] = \min(\mu kf1[x], \mu kf2[x], \mu kf3[x], \mu kf4[x], \mu kf5[x], ...) \quad \text{……(8)}$$

Keterangan:

$\mu_{sf}[x]$: nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}\, i[x]$: nilai keanggotaan konsekuen fuzzy setiap aturan ke-i, i = 1,2,3,.....

$$\mu_{sf}[x] = \max(\mu kf1[x], \mu kf2[x], \mu kf3[x], \mu kf4[x], \mu kf5[x], ...) \quad \text{……...(9)}$$

Keterangan:

$\mu_{sf}[x]$: nilai keanggotaan solusi fuzzy sampai aturan ke-i;

$\mu_{kf}\, i[x]$: nilai keanggotaan konsekuen fuzzy setiap aturan ke-i, i = 1,2,3,.....

3.6.5 *Defuzzification*

Pada tahap ini *output* yang didapat dalam bentuk fuzzy akan diubah menjadi kembali menjadi bentuk bilangan real atau kedalam bentuk *crisp* (tegas) dengan menggunakan rumus berikut :

$$Def = \sum \frac{[\mu(Normal).\widehat{normal} + \mu(Serangan).\widehat{Serangan}]}{[\mu(normal) + \mu(serangan)]} \quad \text{…..……(10)}$$

Keterangan:

Def : nilai akhir defuzifikasi

μ (x) : nilai derajat keanggotaan *output*

$\hat{x}$: nilai singleton variabel *output*.

3.6.6 *Forensic Analyzer* dan *Digital Evidence*

Pada tahapan terakhir, paket yang terdeteksi sebagai serangan akan dianalisa dan diambil data-data penting sebagai berikut : ip *address, timestamp,* dan *payload.* Ip *address* dibutuhkan untuk melacak jaringan yang digunakan pelaku dan melihat apakah penyerang sudah masuk ke system *cloud* setelah serangan *brute*

force dilakukan. *Timestamp* diperlukan untuk melihat waktu dan durasi dari serangan *brute force* yang dilakukan. *Payload* diperlukan untuk melihat informasi *login* dari pelaku ketika serangan *brute force* dilakukan.

3.7 Skenario Pengujian

Pada pengujian penelitian ini akan menggunakan beberapa skenario serangan maupun akses normal menggunakan beberapa sistem operasi. Berikut skenario pengujian pada Tabel 10.

Tabel 10
Skenario Pengujian

No	Skenario	Keterangan
1	Akses normal pada *cloud* menggunakan Windows 8	*Log in, action, log out*
2	Akses normal pada *cloud* menggunakan Kali Linux v.4.13	*Log in, action, log out*
3	Akses normal pada *cloud* menggunakan Andorid Nougat	*Log in, action, log out*
4	Serangan *brute force* pada *cloud* menggunakan Kali Linux v.4.13	Target *brute force* dengan id admin dilakukan menggunakan *wordlist* yang didalamnya terdapat *password* yang benar. Dilakukan selama 5 menit
5	Serangan *brute force* pada *cloud* menggunakan Kali Linux v.4.13 + akses normal menggunakan windows 8	Target *brute force* dengan id admin dilakukan menggunakan *wordlist* yang didalamnya terdapat *password* yang benar. Dilakukan selama 5 menit + *log in* secara berulang.

BAB IV

PENGUJIAN DAN ANALISIS (SEMENTARA)

4.1 Pendahuluan

Pada bab ini dilakukan pengujian dan pengambilan data yang akan digunakan untuk analisis. Metode pengambilan data yang dilakukan mengacu pada bab III metodologi penelitian. Pada hasil pengujian akan dilakukan *feature extraction* untuk melihat *attack pattern* dan *normal pattern* serta melakukan validasi data perbandingan hasil ektraksi fitur dengan data hasil *capture* pada *wireshark*.

4.2 Hasil Pengujian Akses Normal Menggunakan Windows 8

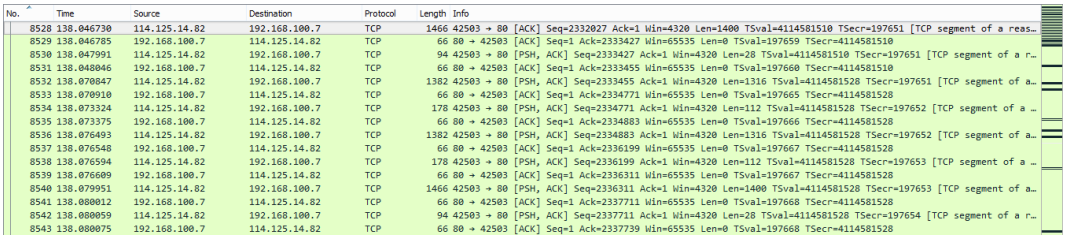
Hasil pengujian akses normal menggunakan windows 8 memiliki ukuran *raw data* 7,75 MB dengan jumlah paket sebanyak 16.438. Pengujian dilakukan selama 3 menit dengan aktivitas pada tabel 11.

Tabel 11

Aktivitas Akses Normal Menggunakan Windows 8

No	Aktivitas	Keterangan
1	Log-in gagal 5x	IP User 114.125.14.82 ID login= laptop Password = 2,3,4,5 dan 6
2	Log-in berhasil	ID login= laptop Password= 1
3	Upload file	Heart.csv dan proposal.rar
4	Ubah password	Password lama = 1 Password baru = 11
5	Log-out	

Hasil *capture traffic data* akan dihitung berdasarkan tiap tipe protokol yaitu protokol http, https, dns, mdns, ssh, tcp, dan udp.



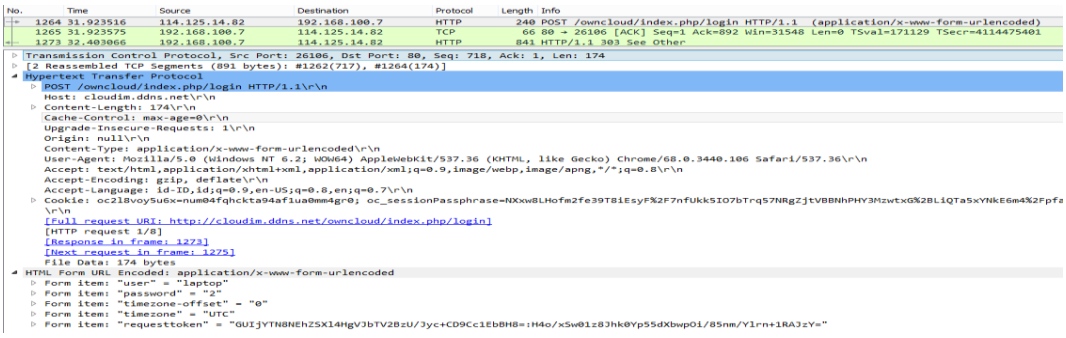
Gambar 4.1 *Traffic Data* Akses Normal Menggunakan Windows 8

Tabel 12

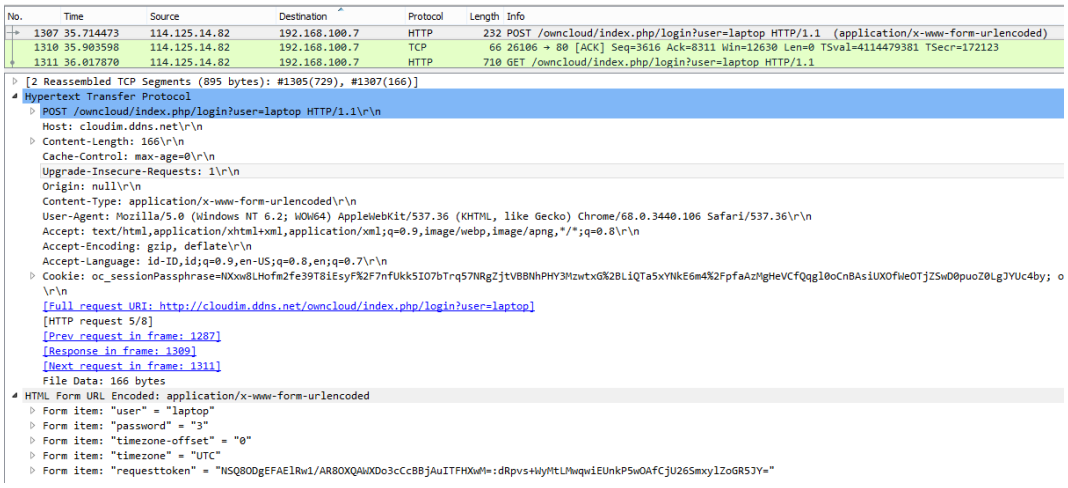
Jumlah Paket Berdasarkan Tipe Protokol

Protokol	Jumlah Paket
Http	16.344
Https	20
Dns	8
Mdns	27
Ssh	14
Tcp	22
Udp	2
Jumlah	**16.437**

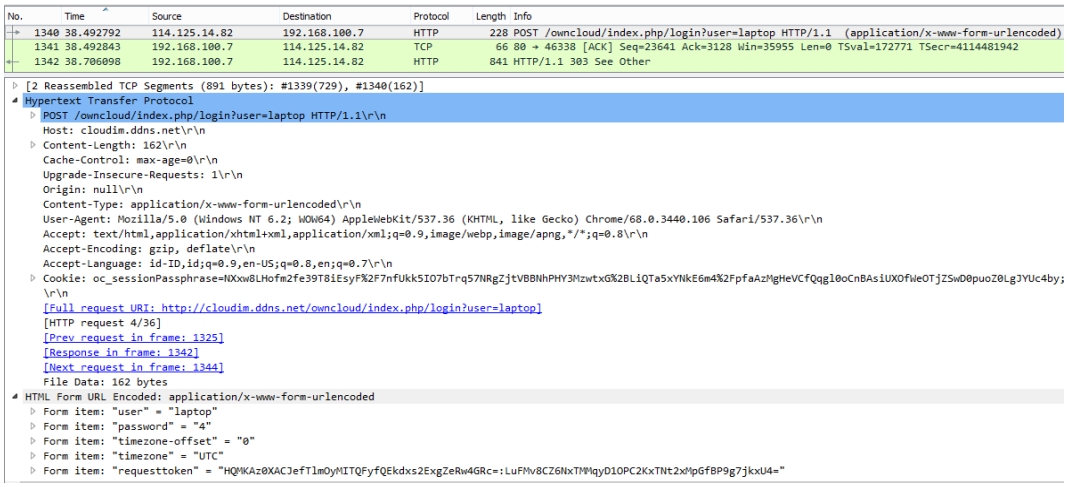
Hasil *capture data traffic* menggunakan wireshark dalam skenario pertama dijelaskan pada gambar dibawah.



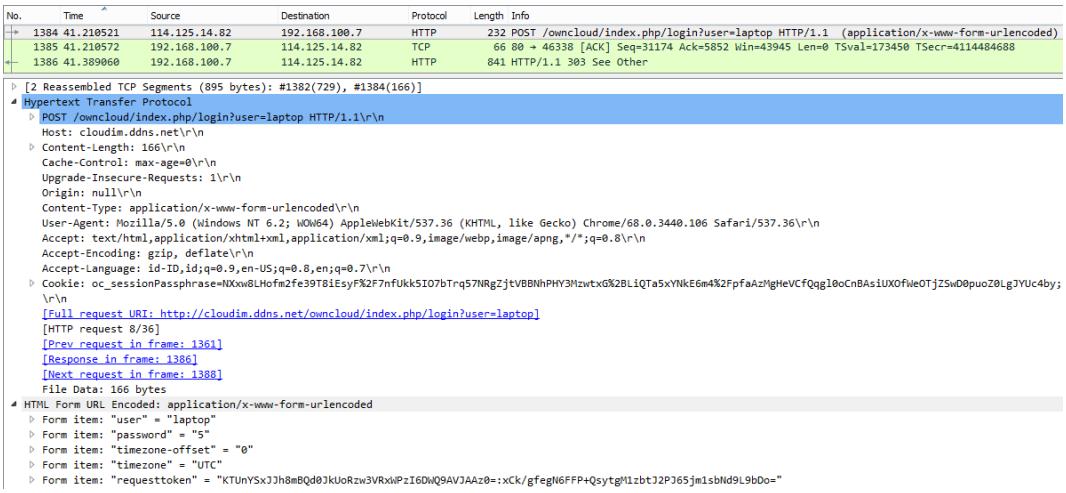
Gambar 4.2 *Traffic Data Windows8 Log-in* Password = 2.



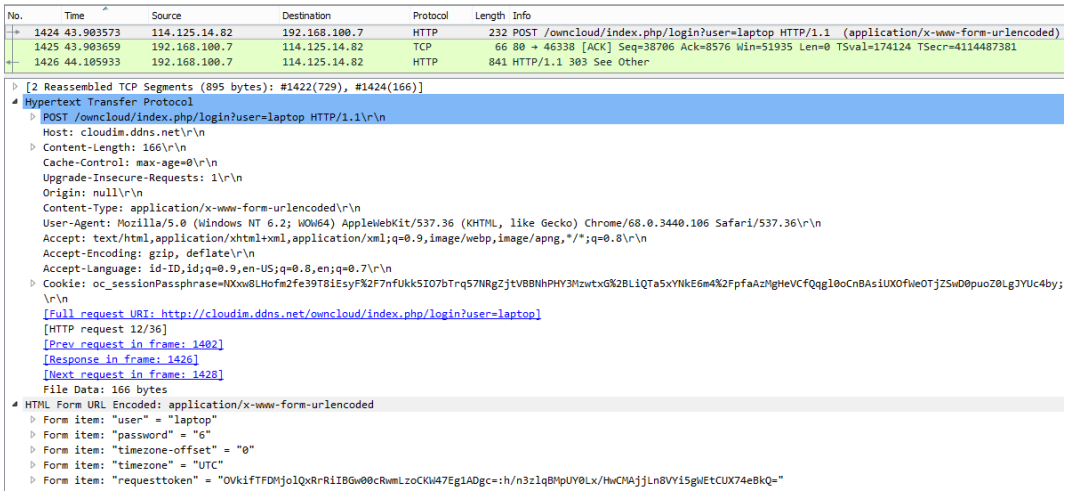
Gambar 4.3 *Traffic Data Windows8 Log-in* Password = 3.



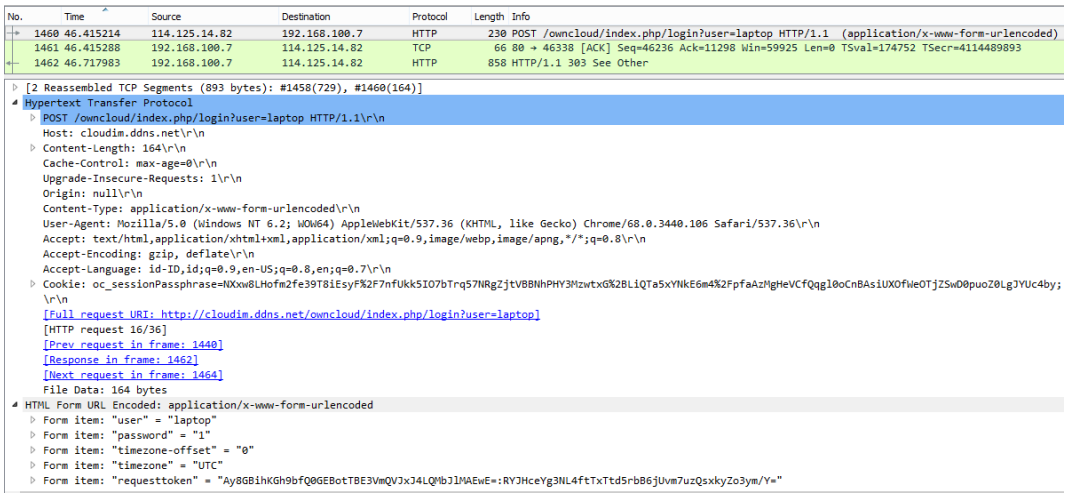
Gambar 4.4 *Traffic Data Windows8 Log-in* Password = 4.



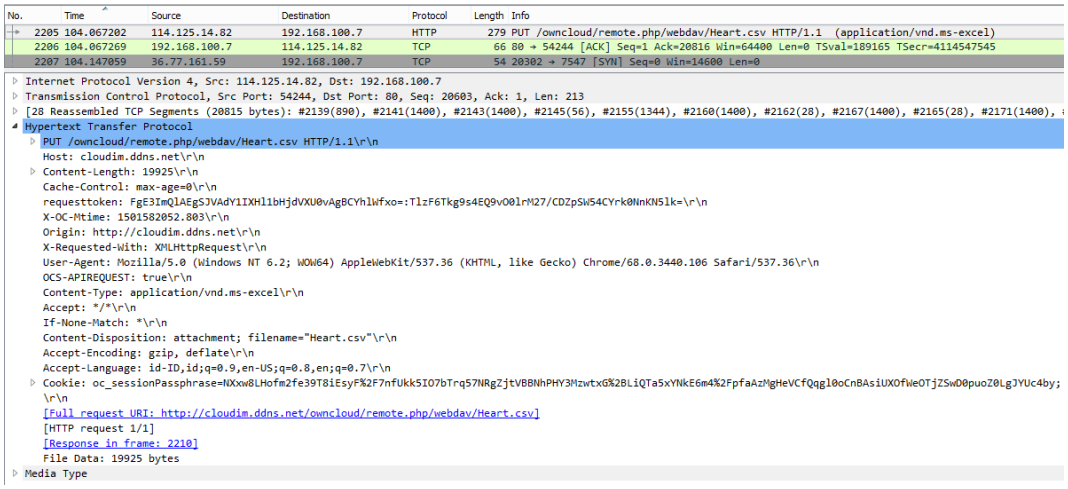
Gambar 4.5 *Traffic Data Windows8 Log-in* Password = 5.



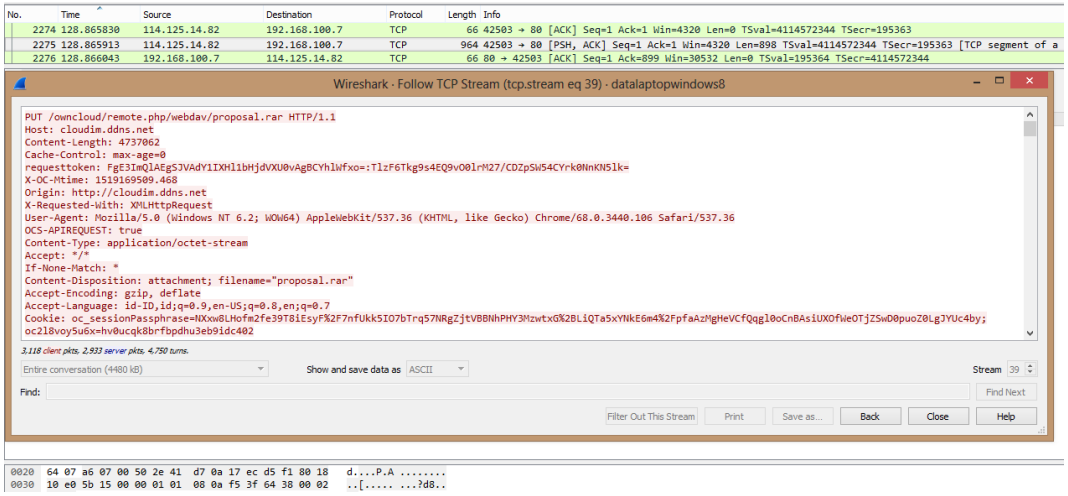
Gambar 4.6 *Traffic Data Windows8 Log-in* Password = 6.



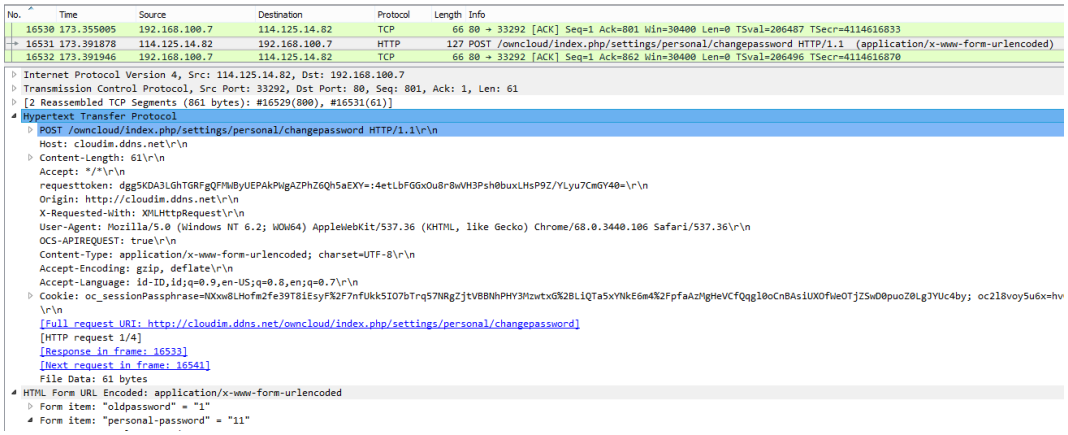
Gambar 4.7 *Traffic Data Windows8 Log-in* Password = 1.



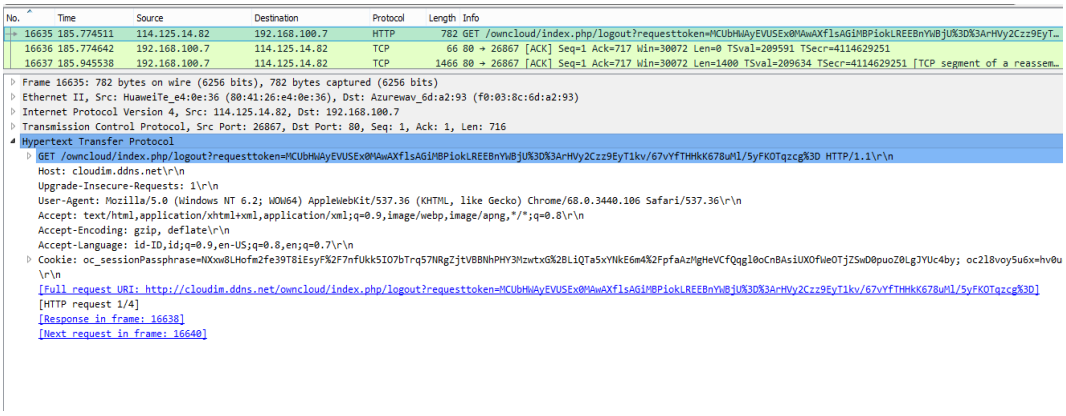
Gambar 4.8 *Traffic Data Windows8 Upload File* Heart.csv



Gambar 4.9 *Traffic Data Windows8 Upload File* Proposal.rar.



Gambar 4.10 *Traffic Data Windows8 Change Password*.



Gambar 4.11 *Traffic Data Windows8 Logout.*

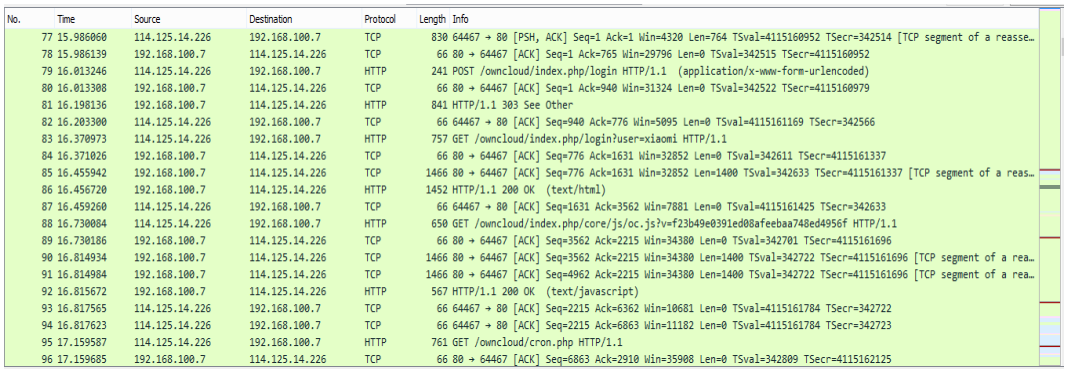
4.3 Hasil Pengujian Akses Normal Menggunakan Android

Hasil pengujian akses normal menggunakan android mempunyai ukuran *raw data* sebesar 1,06MB, dengan jumlah paket sebanyak 1.749. Pengujian dilakukan selama 3 menit dengan aktivitas pada tabel 13.

Tabel 13

Aktivitas Akses Normal Menggunakan Android

No	Aktivitas	Keterangan
1	Log-in gagal 5x	IP User 114.125.14.226 ID login= xiaomi Password = 2,3,4,5 dan 6
2	Log-in berhasil	ID login= xiaomi Password= 1
3	Upload file	Resume.doc,boarding.pdf,unnamed.gif
4	Ubah password	Password lama = 1 Password baru = 11
5	Log-out	



No.	Time	Source	Destination	Protocol	Length	Info
77	15.986060	114.125.14.226	192.168.100.7	TCP	830	64467 → 80 [PSH, ACK] Seq=1 Ack=1 Win=4320 Len=764 TSval=4115160952 TSecr=342514 [TCP segment of a reasse…
78	15.986139	192.168.100.7	114.125.14.226	TCP	66	80 → 64467 [ACK] Seq=1 Ack=765 Win=29796 Len=0 TSval=342515 TSecr=4115160952
79	16.013246	114.125.14.226	192.168.100.7	HTTP	241	POST /owncloud/index.php/login HTTP/1.1 (application/x-www-form-urlencoded)
80	16.013308	192.168.100.7	114.125.14.226	TCP	66	80 → 64467 [ACK] Seq=1 Ack=940 Win=31324 Len=0 TSval=342522 TSecr=4115160979
81	16.198136	192.168.100.7	114.125.14.226	HTTP	841	HTTP/1.1 303 See Other
82	16.203300	114.125.14.226	192.168.100.7	TCP	66	64467 → 80 [ACK] Seq=940 Ack=776 Win=5095 Len=0 TSval=4115161169 TSecr=342566
83	16.370973	114.125.14.226	192.168.100.7	HTTP	757	GET /owncloud/index.php/login?user=xiaomi HTTP/1.1
84	16.371026	192.168.100.7	114.125.14.226	TCP	66	80 → 64467 [ACK] Seq=776 Ack=1631 Win=32852 Len=0 TSval=342611 TSecr=4115161337
85	16.455942	192.168.100.7	114.125.14.226	TCP	1466	80 → 64467 [ACK] Seq=776 Ack=1631 Win=32852 Len=1400 TSval=342633 TSecr=4115161337 [TCP segment of a reas…
86	16.456720	192.168.100.7	114.125.14.226	HTTP	1452	HTTP/1.1 200 OK (text/html)
87	16.459260	114.125.14.226	192.168.100.7	TCP	66	64467 → 80 [ACK] Seq=1631 Ack=3562 Win=7881 Len=0 TSval=4115161425 TSecr=342633
88	16.730084	114.125.14.226	192.168.100.7	HTTP	650	GET /owncloud/index.php/core/js/oc.js?v=f23b49e0391ed08afeebaa748ed4956f HTTP/1.1
89	16.730186	192.168.100.7	114.125.14.226	TCP	66	80 → 64467 [ACK] Seq=3562 Ack=2215 Win=34380 Len=0 TSval=342701 TSecr=4115161696
90	16.814934	192.168.100.7	114.125.14.226	TCP	1466	80 → 64467 [ACK] Seq=3562 Ack=2215 Win=34380 Len=1400 TSval=342722 TSecr=4115161696 [TCP segment of a rea…
91	16.814984	192.168.100.7	114.125.14.226	TCP	1466	80 → 64467 [ACK] Seq=4962 Ack=2215 Win=34380 Len=1400 TSval=342722 TSecr=4115161696 [TCP segment of a rea…
92	16.815672	192.168.100.7	114.125.14.226	HTTP	567	HTTP/1.1 200 OK (text/javascript)
93	16.817565	114.125.14.226	192.168.100.7	TCP	66	64467 → 80 [ACK] Seq=2215 Ack=6362 Win=10681 Len=0 TSval=4115161784 TSecr=342722
94	16.817623	114.125.14.226	192.168.100.7	TCP	66	64467 → 80 [ACK] Seq=2215 Ack=6863 Win=11182 Len=0 TSval=4115161784 TSecr=342723
95	17.159587	114.125.14.226	192.168.100.7	HTTP	761	GET /owncloud/cron.php HTTP/1.1
96	17.159685	192.168.100.7	114.125.14.226	TCP	66	80 → 64467 [ACK] Seq=6863 Ack=2910 Win=35908 Len=0 TSval=342809 TSecr=4115162125

Gambar 4.12 *Traffic Data* Akses Normal Menggunakan Android

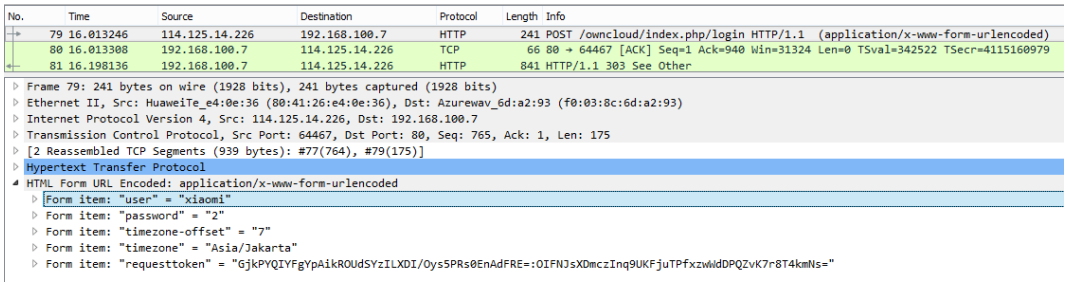
Hasil *capture traffic data* akan dihitung berdasarkan tiap tipe protokol yaitu protokol http, https, dns, mdns, ssh, tcp, dan udp. Hasilnya adalah protokol http mempunyai jumlah terbanyak dibanding protokol lain dengan jumlah 1.684 paket dari total keseluruhan 1.749 paket.

Tabel 14

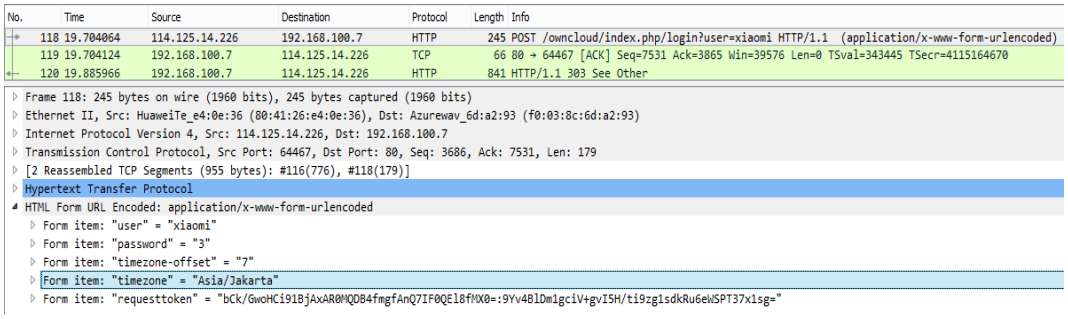
Jumlah Paket Berdasarkan Tipe Protokol

Protokol	Jumlah Paket
Http	1.684
Https	0
Dns	0
Mdns	29
Ssh	0
Tcp	26
Udp	8
Jumlah	**1.747**

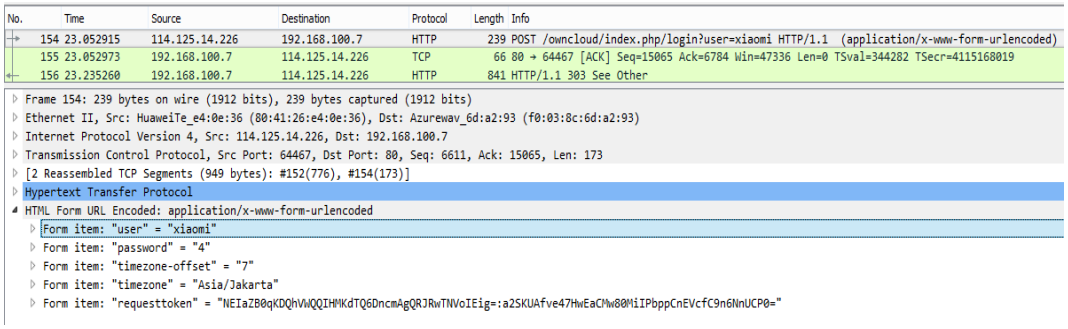
Hasil *capture data traffic* menggunakan wireshark dalam skenario kedua dijelaskan pada gambar dibawah.



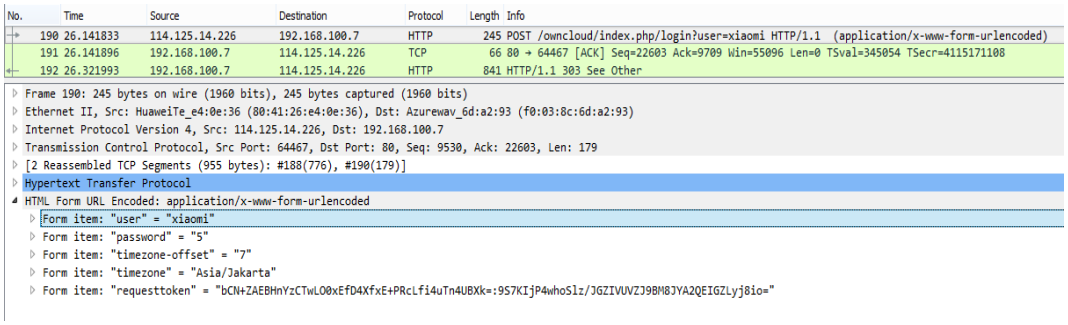
Gambar 4.13 *Traffic Data Android Log-in Password* =2.



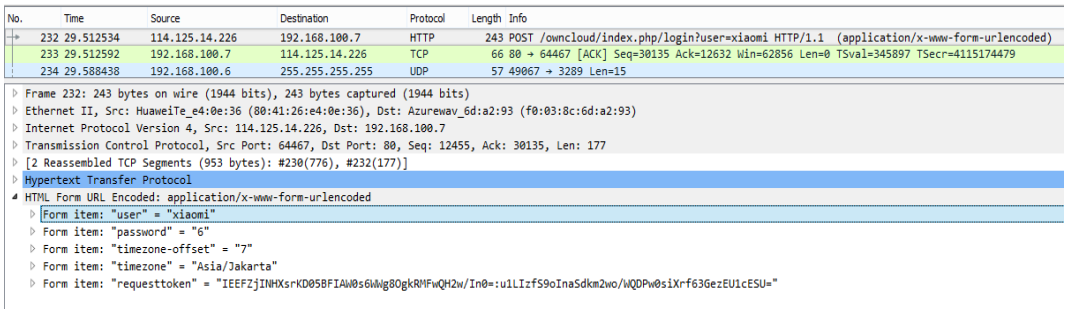
Gambar 4.14 *Traffic Data Android Log-in Password* =3.



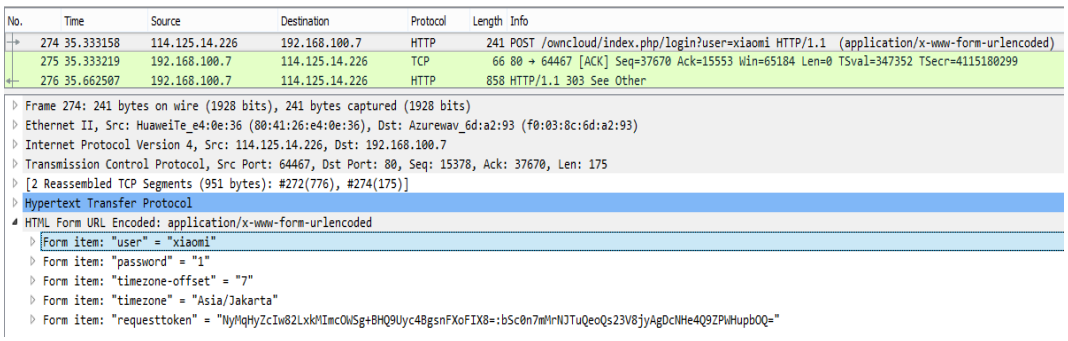
Gambar 4.15 *Traffic Data Android Log-in Password* =4.



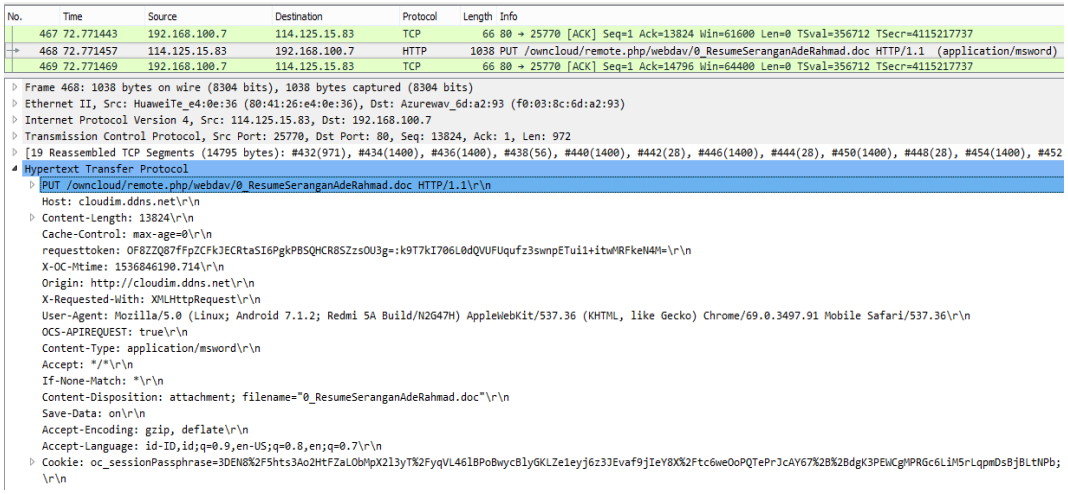
Gambar 4.16 *Traffic Data Android Log-in Password* =5.



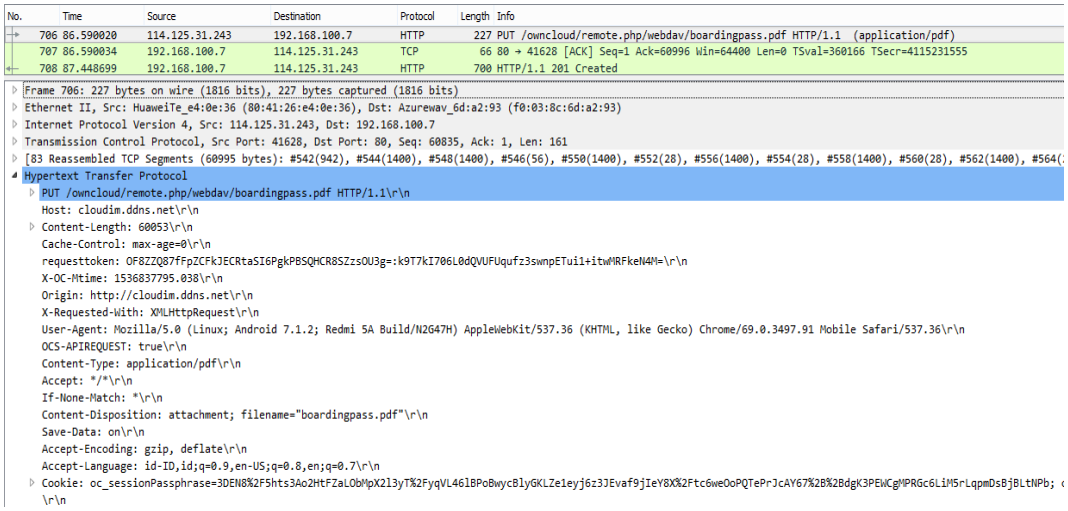
Gambar 4.17 *Traffic Data Android Log-in Password* =6.



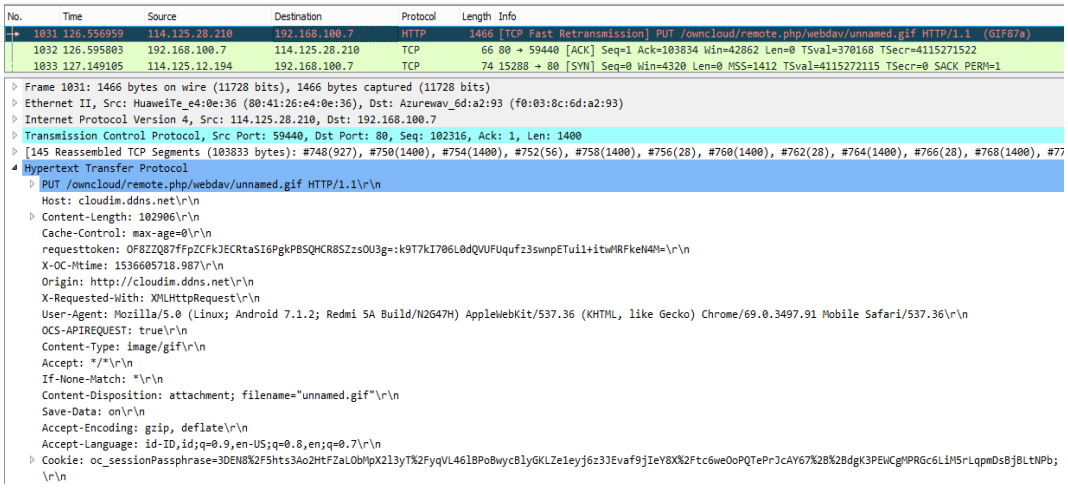
Gambar 4.18 *Traffic Data Android Log-in Password* =1.



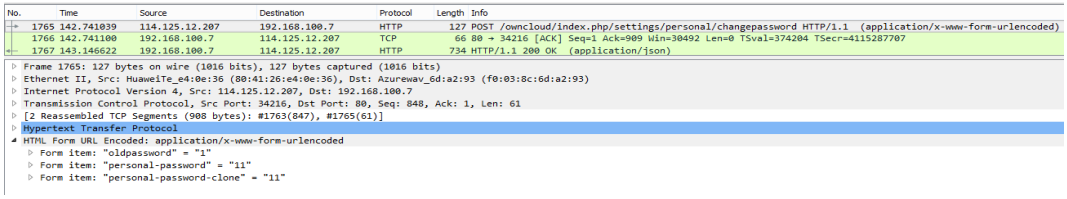
Gambar 4.19 *Traffic Data Android Upload File* Resume.doc



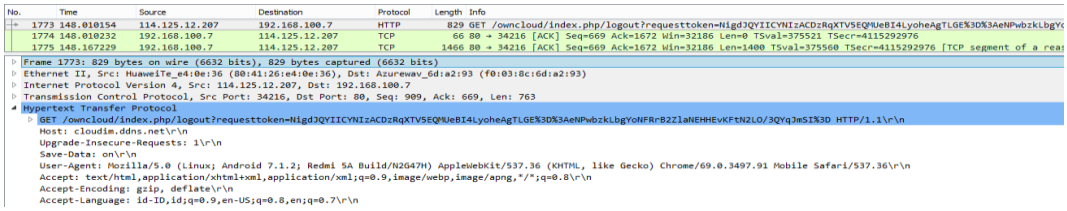
Gambar 4.20 *Traffic Data Android Upload File* boardingpass.pdf



Gambar 4.21 *Traffic Data Android Upload File* unnamed.gif



Gambar 4.22 *Traffic Data Android Change Password.*



Gambar 4.23 *Traffic Data Android Log-out.*

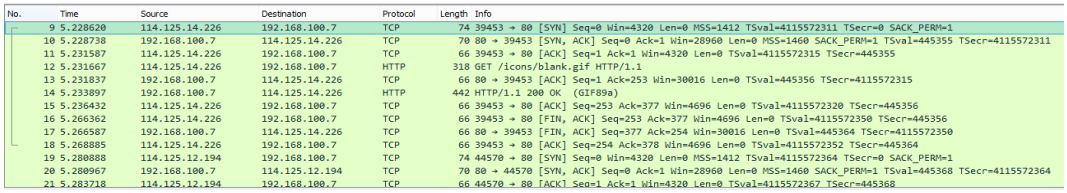
4.4 Hasil Pengujian Akses Normal Menggunakan Kali Linux

Hasil pengujian akses normal menggunakan kali linux mempunyai ukuran *raw data* sebesar 2,85MB, dengan jumlah paket sebanyak 4.942. Pengujian dilakukan selama 3 menit dengan aktivitas pada tabel 15.

Tabel 15

Aktivitas Akses Normal Menggunakan Kali Linux

No	Aktivitas	Keterangan
1	Log-in gagal 5x	IP User 114.125.15.83 ID login= sk Password = 2,3,4,5,6, dan 1
2	Log-in berhasil	ID login= sk1 Password= 1
3	Upload file	16.csv, android.csv
4	Ubah password	Password lama = 1 Password baru = 11
5	Log-out	



Gambar 4.24 *Traffic Data* Akses Normal Kali Linux.

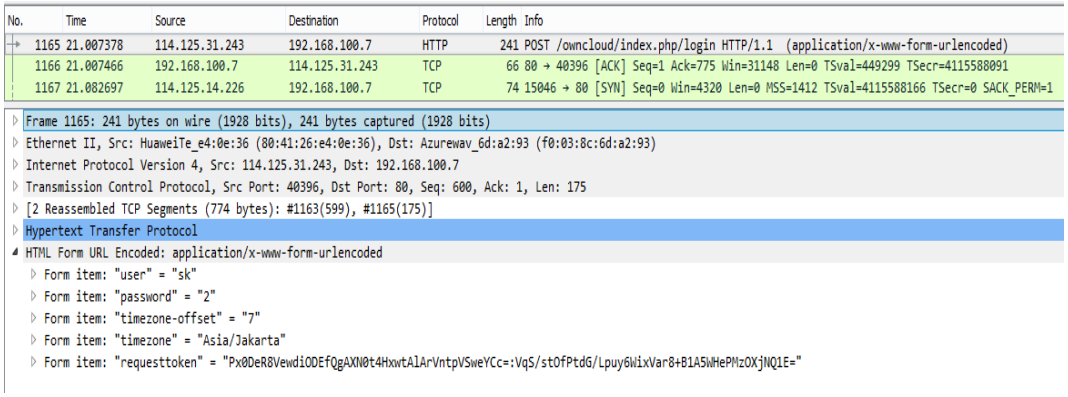
Hasil *capture traffic data* akan dihitung berdasarkan tiap tipe protokol yaitu protokol http, https, dns, mdns, ssh, tcp, dan udp. Hasilnya adalah protokol http mempunyai jumlah terbanyak dibanding protokol lain dengan jumlah 4.832 paket dari total keseluruhan 4.862 paket

Tabel 16

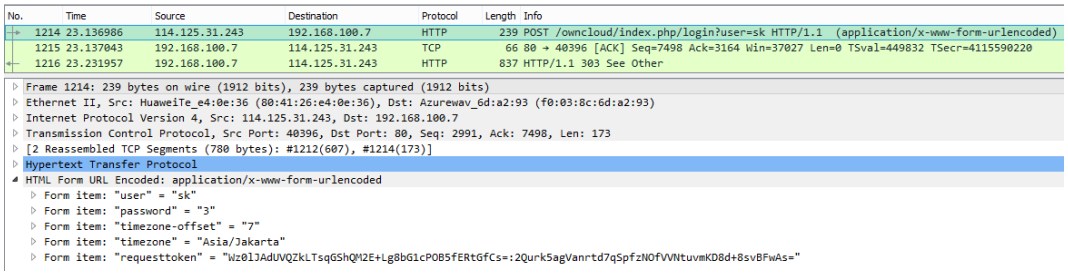
Jumlah Paket Berdasarkan Tipe Protokol

Protokol	Jumlah Paket
Http	4.832
Https	0
Dns	0
Mdns	0
Ssh	14
Tcp	14
Udp	2
Jumlah	**4.862**

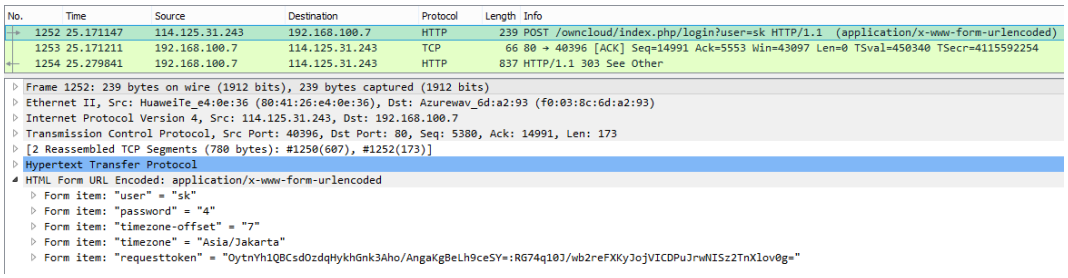
Hasil *capture data traffic* menggunakan wireshark dalam skenario ketiga dijelaskan pada gambar dibawah.



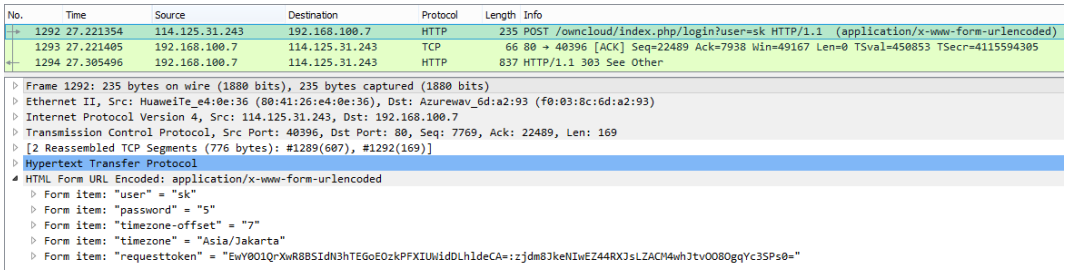
Gambar 4.25 *Traffic Data Kali Linux Log-in Password* = 2.



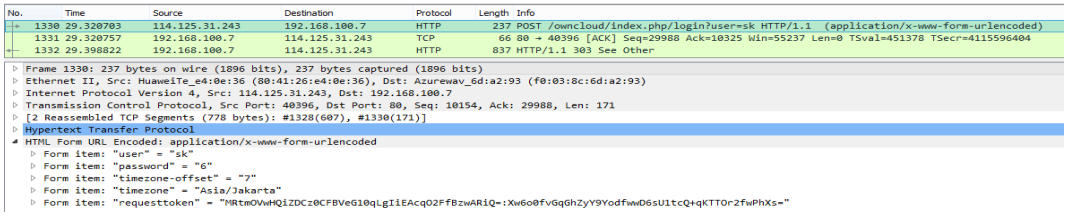
Gambar 4.26 *Traffic Data Kali Linux Log-in Password* = 3.



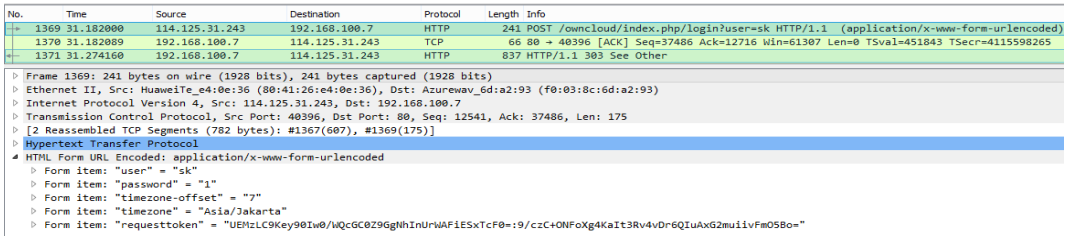
Gambar 4.27 *Traffic Data Kali Linux Log-in Password* = 4.



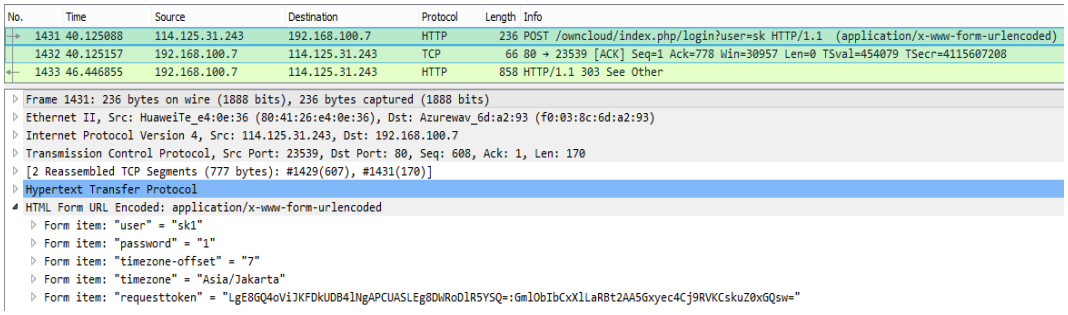
Gambar 4.28 *Traffic Data Kali Linux Log-in Password* = 5.



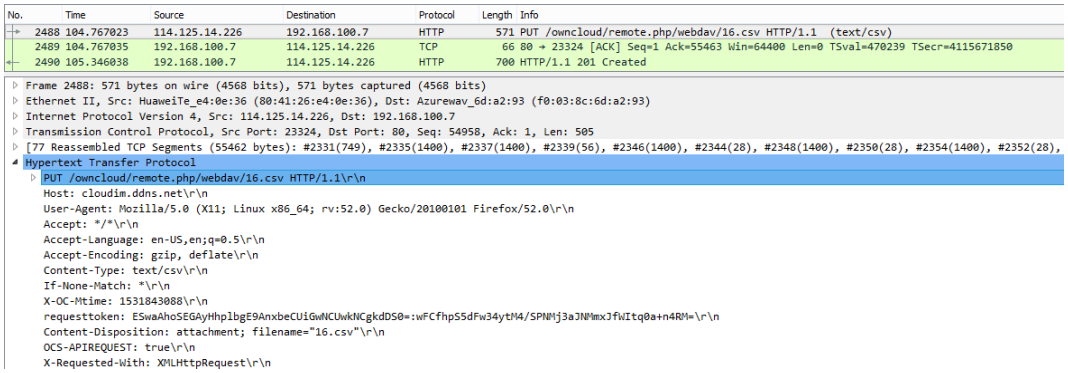
Gambar 4.29 *Traffic Data Kali Linux Log-in Password* = 6.



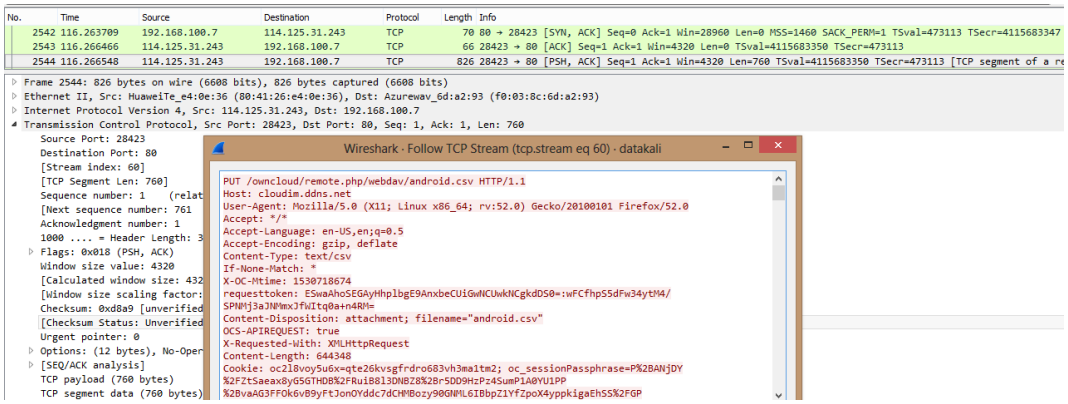
Gambar 4.30 *Traffic Data Kali Linux Log-in Id = sk Password* = 1.



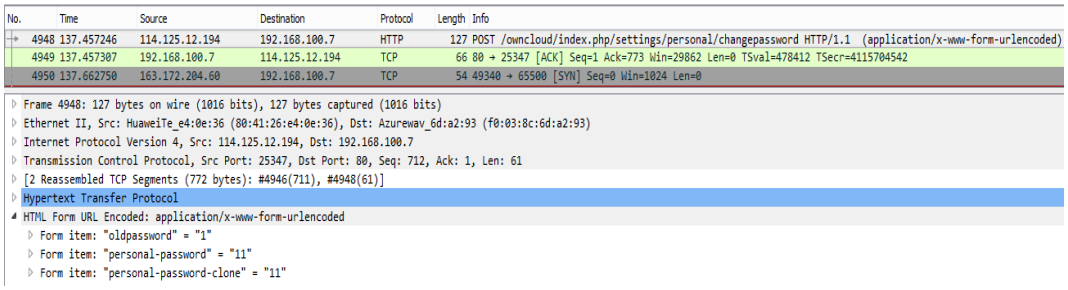
Gambar 4.31 *Traffic Data Kali Linux Log-in Id = sk1 Password* = 1.



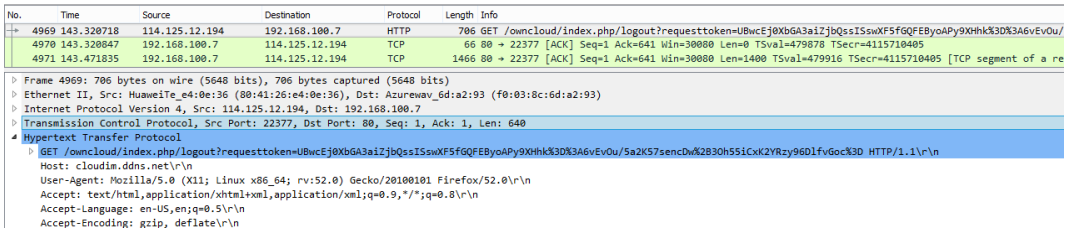
Gambar 4.32 *Traffic Data Kali Linux Upload File 16.csv*.



Gambar 4.33 *Traffic Data Kali Linux Upload File android.csv.*



Gambar 4.34 *Traffic Data Kali Linux Change Password.*



Gambar 4.35 *Traffic Data Kali Linux Log-out.*

4.5 Hasil Pengujian Serangan Menggunakan Kali Linux

4.5.1 Pengujian serangan *brute force* pada halaman *login cloud*

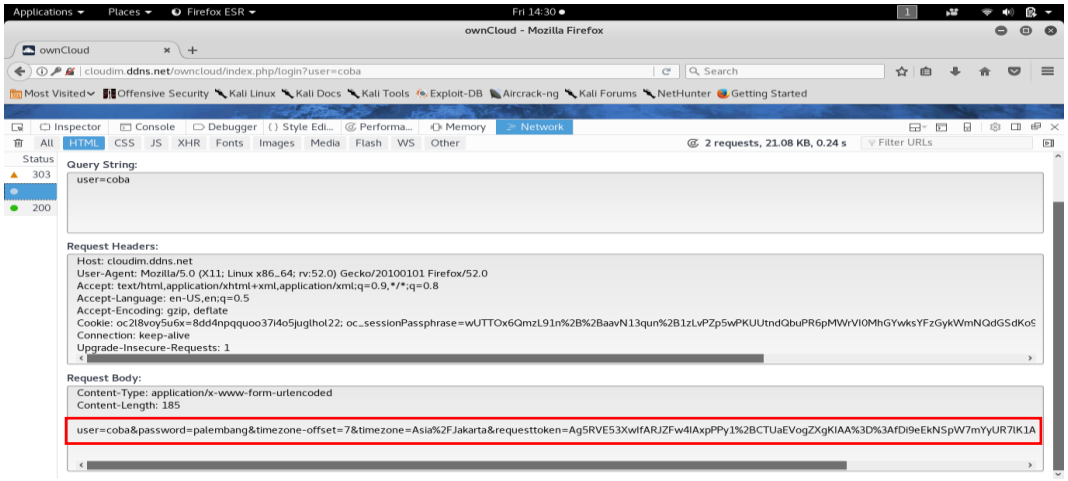
Pengujian serangan *brute force* pada halaman *login cloud* menggunakan *software* Hydra. Format untuk melakukan serangan *brute force* pada halaman *login* menggunakan Hydra sebagai berikut:

hydra -L <username list> -p <password list> <Target> <form parameters><failed login message>

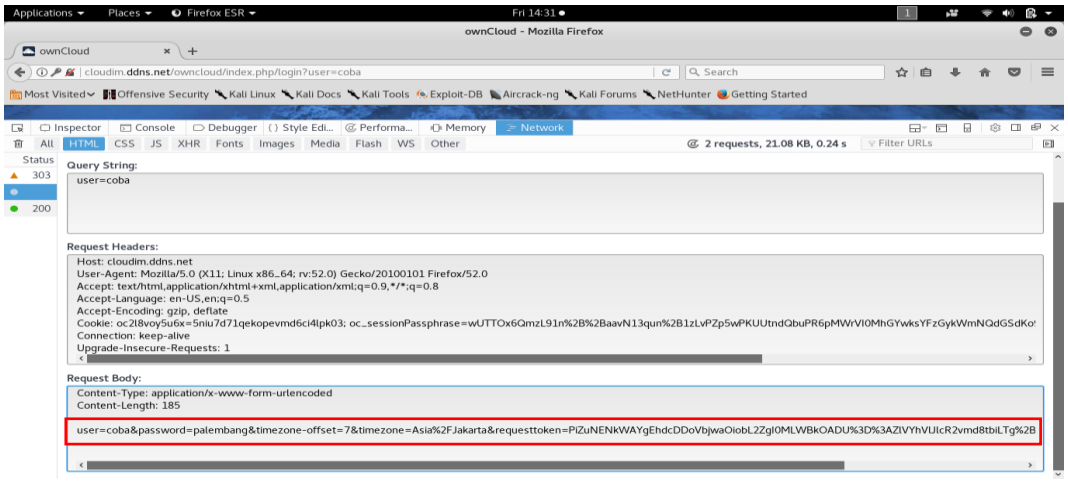
Gambar 4.36 Format Perintah Serangan *Brute Force* Menggunakan Hydra

Pada gambar 4.36, untuk melakukan serangan *brute force* menggunakan hydra pada halaman *login* diperlukan *form parameters. Form parameters* adalah parameter-parameter yang ada pada halaman *login* sebuah *website. Form parameters* pada halaman *login* owncloud didapatkan dari hasil percobaan gagal login sebanyak tiga kali menggunakan *user* dan *password* yang sama adalah sebagai berikut.

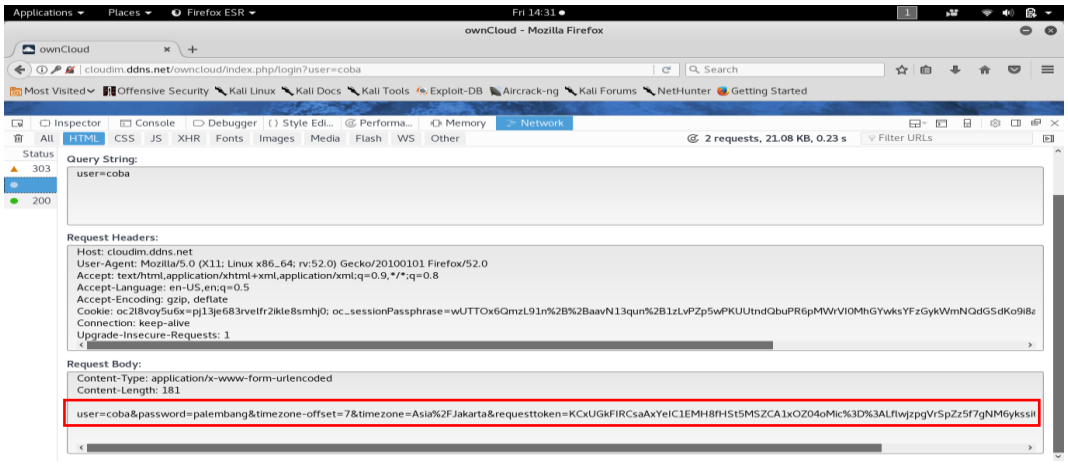
Gambar 4.37 Gagal *Login* pada Halaman *Login* Owncloud



Gambar 4.38 *Form Parameters* pada Owncloud (Gagal *Login* Pertama)



Gambar 4.39 *Form Parameters* pada Owncloud (Gagal *Login* Kedua)

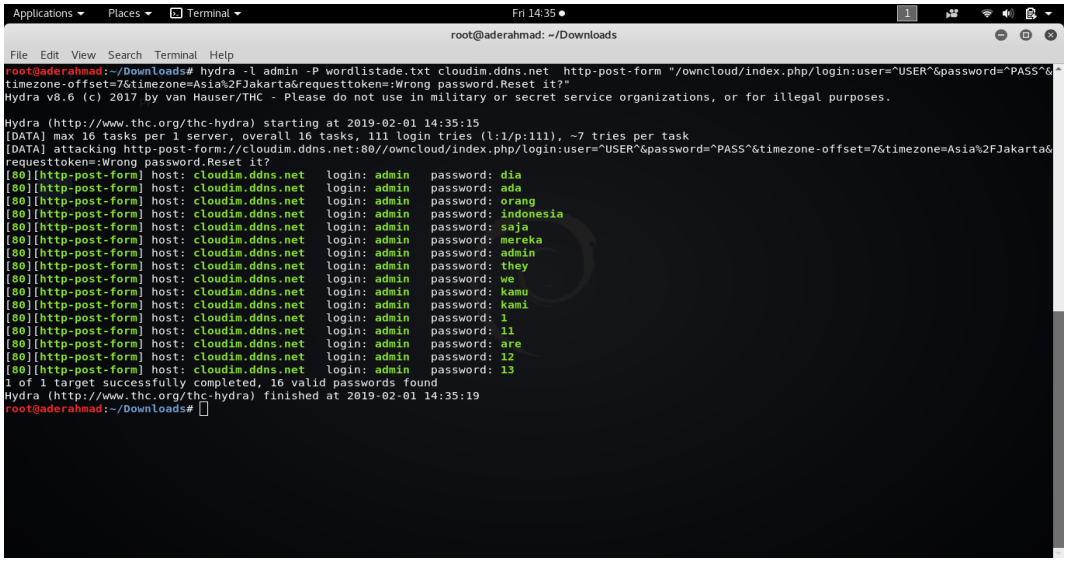


Gambar 4.40 *Form Parameters* pada Owncloud (Gagal *Login* Ketiga)

Pada gambar 4.38, 4.39, dan 4.40 terdapat lima *form parameters* pada halaman *login* owncloud yaitu : *User, password, timezone-offset, timezone,* dan *request token*. Dengan menggunakan *user* dan *password* yang sama ada empat nilai dari *form parameters* yang sama pada percobaan gagal *login* sebanyak tiga kali yaitu nilai *parameters* : *User, password, timezone-offset,* dan *timezone*. Sedangkan untuk nilai dari *parameters request token* pada gagal *login* sebanyak tiga kali berbeda-beda setiap *login* walaupun menggunakan perangkat, jaringan, *user* dan *password* yang sama. Sehingga disimpulkan untuk nilai dari *parameters request token* yang terdapat pada Owncloud bernilai *random*. Dari percobaan gagal *login* sebanyak tiga kali didapatkan *form parameters* yang dibutuhkan untuk melakukan serangan *brute force* pada halaman *login* owncloud sehingga *syntax* yang dijalankan pada hydra adalah sebagai berikut:

hydra -l admin -P wordlistade.txt Cloudim.ddns.net http-post-form "/owncloud/index.php/login:user=^USER^&password=^PASS^&timezone-offset=7&timezone=Asia%2FJakarta&requesttoken=:Wrong password.Reset it?"

Gambar 4.41 Format Perintah Serangan *Brute Force* Menggunakan Hydra

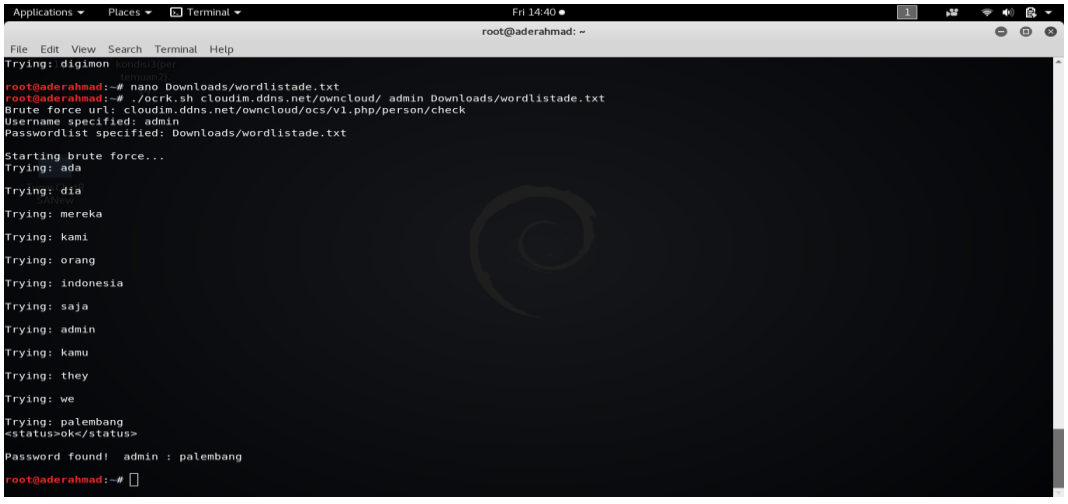


Gambar 4.42 Hasil Serangan *Brute Force* Menggunakan Hydra pada Halaman *Login*

Hasil serangan *brute force* pada halaman *login cloud* menggunakan Hydra pada gambar 4.42 menunjukkan ada enam belas *password* yang valid untuk *user* "admin" sedangkan *password* dari *user* "admin" adalah "palembang" sehingga serangan *brute force* pada halaman *login cloud* menggunakan hydra dinyatakan gagal karena hasil percobaan memberikan enam belas *password* yang salah tetapi dianggap benar oleh hydra. Hal ini dikarenakan adanya nilai *form parameters* yang tidak dapat dimasukkan pada *syntax* serangan *brute force* karena bersifat *random value* yaitu *parameters request token* sehingga serangan *brute force* pada halaman *login* owncloud tidak dapat dilakukan.

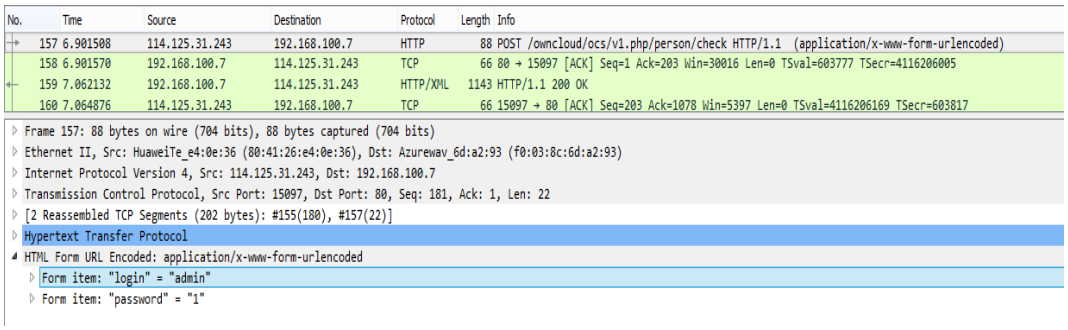
4.5.2 Pengujian serangan *brute force* pada API ocs *cloud*

Hasil pengujian serangan menggunakan program Ocrk.sh (terlampir) pada kali linux di API OCS dari sisi penyerang mempunyai ukuran *raw data* sebesar 950KB, dengan jumlah paket sebanyak 3.260. Sedangkan dari sisi *server* mempunyai ukuran *raw data* 254KB dengan jumlah paket sebanyak 1.387 paket.Pengujian dilakukan menggunakan *wordlist* yang mempunyai 111 baris kata dengan IP *address* penyerang 114.125.28.210, 114.125.15.83, 114.125.12.194, 114.125.31.243 dan target ID login= admin. Pengujian dilakukan untuk melihat pola serangan *brute force* pada *cloud* dengan melakukan *feature extraction* pada hasil *capture* wireshark dalam pengujian serangan *brute force*. Pola serangan *brute force* akan diimplementasikan ke *engine IDS* (Snort) sehingga snort dapat memberikan *alert* jika terjadi serangan *brute force* pada *cloud*.

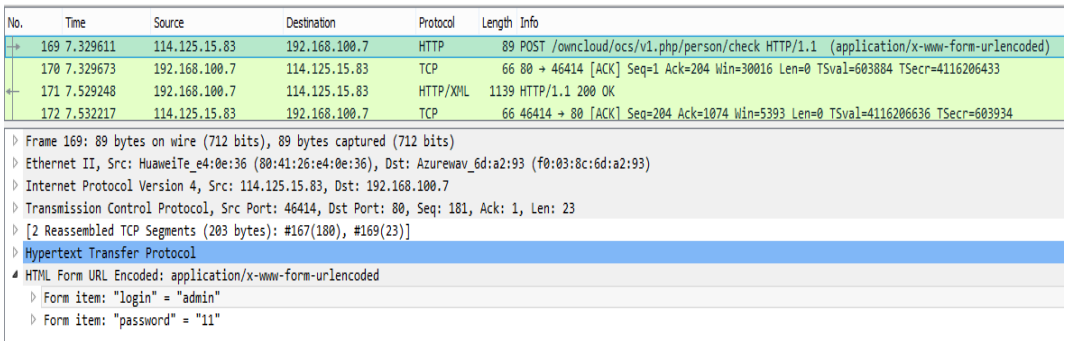


Gambar 4.43 Serangan *Brute Force* Menggunakan Kali Linux.

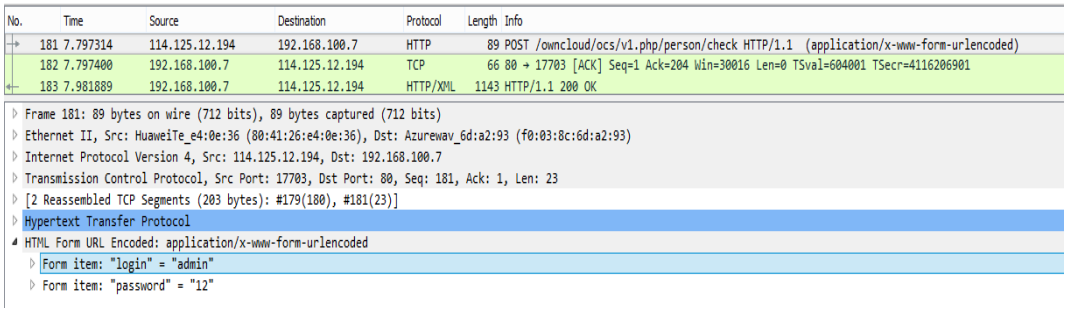
Hasil *capture data traffic* menggunakan wireshark dalam skenario keempat dijelaskan pada gambar dibawah.



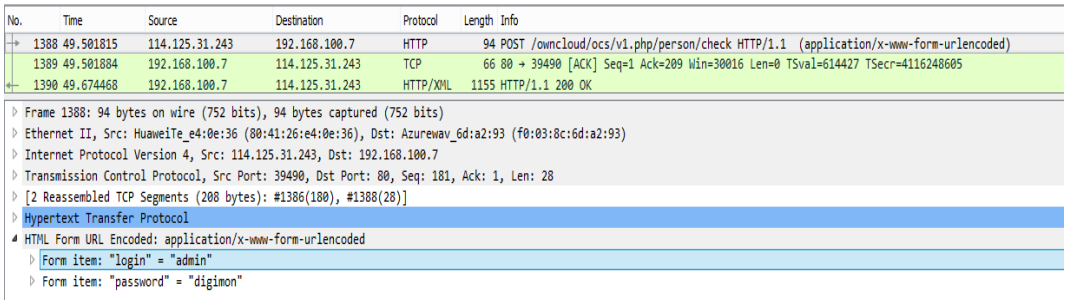
Gambar 4.44 *Traffic Data* Serangan Menggunakan Kali Linux Password = 1.



Gambar 4.45 *Traffic Data* Serangan Menggunakan Kali Linux Password = 11.

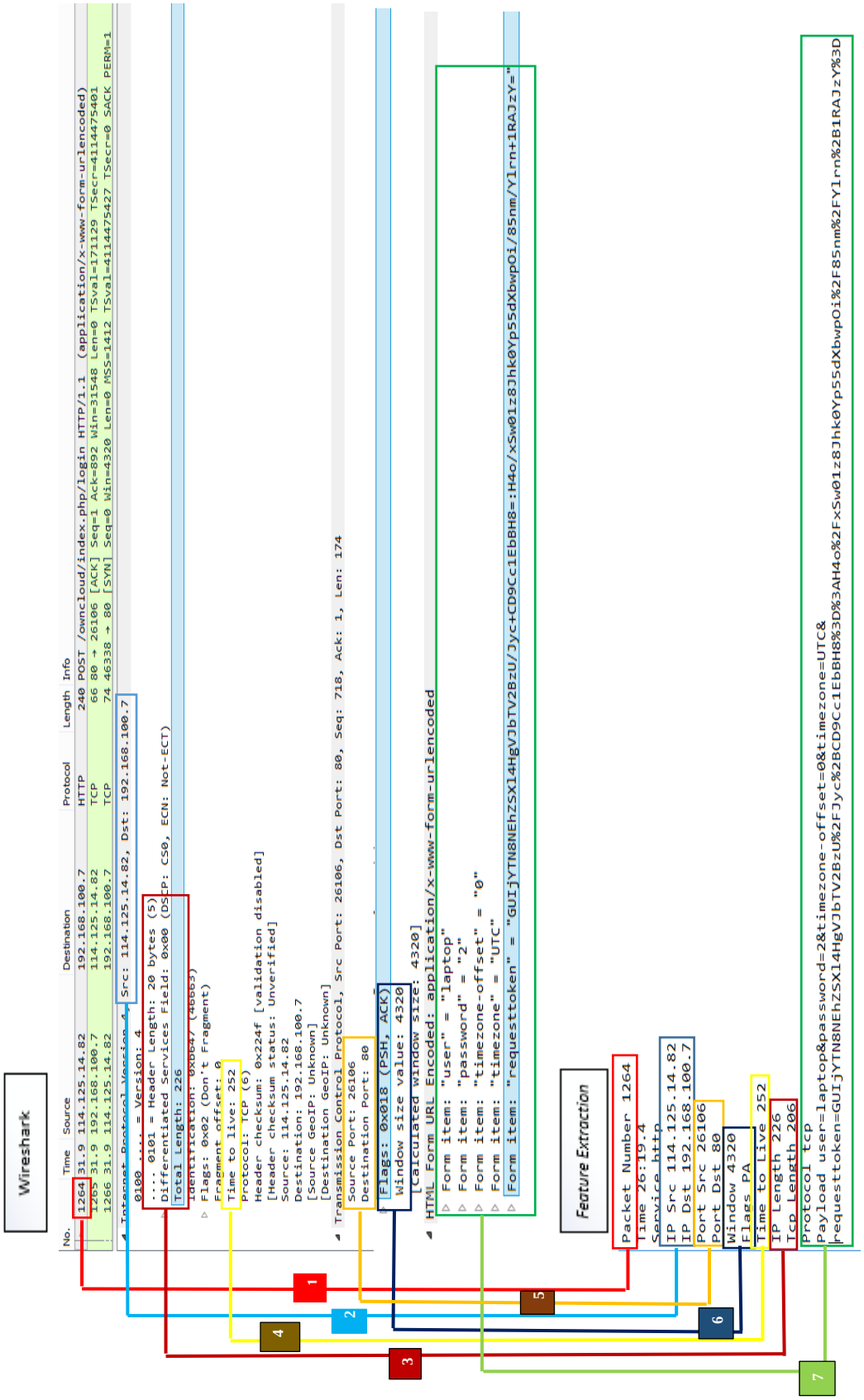


Gambar 4.46 *Traffic Data* Serangan Menggunakan Kali Linux Password = 12.



Gambar 4.47 *Traffic Data* Serangan Kali Linux Password = digimon.

4.6 *Feature Extraction Traffic Data* Hasil Pengujian



Gambar 4.48 Pencocokkan Hasil *Feature Extraction* dengan *Raw Data* (Pcap).

Pada gambar 4.48 terdapat tujuh (7) poin persamaan data antara hasil *feature extraction* dengan *raw data* (pcap). Berikut penjelasan mengenai tujuh poin data yang sama pada hasil *feature extraction* dan *raw data* (pcap);

1. Nomor satu (1) berisi nomor paket atau frame dari paket data. Pada gambar 4.48 nomor paket bernilai 1264.
2. Nomor dua (2) berisi ip *address source* dan *destination* dari paket data. Pada gambar 4.48 ip *address source* adalah 114.125.14.82 dan ip *address destination* adalah 192.168.100.7.
3. Nomor tiga (3) berisi ip *length* dan tcp *length* dari paket data. Pada gambar 4.48 nilai ip *length* adalah 226. Untuk mendapatkan nilai tcp *length*, nilai ip *length* dikurangi nilai *header length* dalam hal ini 20 sehingga nilai tcp *length* 206.
4. Nomor empat (4) berisi nilai *time to live* (ttl) dari paket data dengan nilai 252.
5. Nomor lima (5) berisi nilai *source port* dan *destination port* dari paket data. Pada gambar 4.48 *source port* bernilai 26106 dan *destination port* bernilai 80.
6. Nomor enam (6) berisi nilai *flags* dan *window* dari paket data. Pada gambar 4.48 nilai *flags* adalah PSH, ACK (PA) dan nilai *window* adalah 4320.
7. Nomor tujuh (7) berisi *payload* dari paket data. Pada gambar 4.48 payload berisi *user* = laptop, *password* = 2, *timezone-offset* = 0, *timezone* = UTC, dan *requesttoken*=GUIjYTN8NEhZSXl4HgVJbTV2BzU%2FJyc%2BCD9Cc1EbBH8%3D%3AH4o%2FxSw01z8Jhk0Yp55dXbwpOi%2F85nm%2FYlrn%2B1RAJzY%3D.

Pada hasil *feature extraction* terdapat 14 fitur yang diekstrak dari *raw data* (pcap) hasil skenario pengujian yang akan dianalisis untuk menentukan pola serangan dan pola akses normal pada *cloud*. Fitur tersebut adalah ; *packet number, timestamp, service, ip source, ip destination, port source, port destination, windows, flags, ttl, ip length, payload, tcp length,* dan *protocol*. Berikut adalah hasil *feature extraction* dari *raw data* (pcap) hasil pengujian skenario pertama (1) sampai skenario keempat (4) pada tabel 17, 18, 19, dan 20 (terlampir).

4.7 Pola Serangan *Brute Force*

Pada hasil *feature extraction* skenario pengujian terdapat 14 fitur yang digunakan untuk analisis penentuan pola serangan *brute force*. Pada tabel 17, 18, 19, dan 20 dijelaskan hasil *feature extraction raw data* (pcap) dari skenario pengujian yang dilakukan. Dari data skenario pengujian satu sampai tiga pada tabel 17, 18 , dan 19 nilai fitur ada yang sama dan ada yang berbeda. Berikut adalah fitur yang nilainya sama dan yang berbeda dari pengujian pertama sampai pengujian ketiga pada tabel berikut.

Tabel 21

Nilai Fitur yang Sama Pada Skenario Pengujian Pertama – Ketiga

	Ip dst	Port dst	Flags	Ttl	Protocol
Pengujian pertama (windows 8)	192.168.100.7	80	PA	252	TCP
Pengujian kedua (android)	192.168.100.7	80	PA	252	TCP
Pengujian ketiga (Kali Linux)	192.168.100.7	80	PA	252	TCP

Tabel 22

Nilai Fitur yang Berbeda Pada Skenario Pengujian Pertama – Ketiga

	Window	Ip Length	Tcp Length
Pengujian pertama (windows 8)	4320, 11855, 27960, 35493, 43025, 50555	226, 218, 214, 216, 942, 950, 113, 768	206, 198, 194, 196, 922, 930, 93, 748
Pengujian kedua (android)	4320, 11850, 19384, 26922,34454, 41989, 4988	227, 231, 225, 231, 229, 227, 1023, 994, 113, 815	207, 211, 205, 211, 209, 207, 1003, 974, 93, 795
Pengujian ketiga (Kali Linux)	4320, 11817, 19310, 26808, 34307	227, 225, 221, 223, 222, 801, 812, 113, 692	207, 205, 201, 203, 202, 781, 792, 93, 672

Pada tabel 21 dan 22 disimpulkan pola akses normal dari skenario pengujian pertama (1) sampai pengujian ketiga (3) memiliki pola nilai ip *address destination, port destination, flags, time to live, protocol* yang sama dan dengan rentang nilai ip *length* 113 sampai 1023. Dari tabel 20 hasil *feature extraction raw data* (pcap) skenario pengujian empat (4) *brute force* menggunakan kali linux terdapat fitur dengan nilai yang sama dan nilai yang berbeda. Berikut adalah fitur dari *raw data* (pcap) skenario pengujian empat (4) yang memiliki nilai yang sama pada tabel berikut.

Tabel 23

Nilai Fitur yang Sama Pada Skenario Pengujian Keempat

Ip dst	Port dst	Window	Flags	Ttl	Protocol
192.168.100.7	80	4320	PA	252	TCP

Tabel 24

Nilai Fitur yang Berbeda Pada Skenario Pengujian Keempat

Ip Src	Port Src	Ip Length	Tcp Length
114.125.12.194	38376	82	62
114.125.15.83	49144	78	58
114.125.31.243	15097	74	54
114.125.15.83	46414	75	55
114.125.12.194	17703	75	55
114.125.31.243	18041	75	55
114.125.12.194	65166	75	55
114.125.28.210	13523	75	55
114.125.28.210	56470	75	55
114.125.15.83	48671	75	55
114.125.31.243	39490	80	60

Pada data tabel 23 dan 24 disimpulkan pola serangan *brute force* menggunakan kali linux mempunyai pola nilai ip *address destination, port destination, window, flags, time to live, protocol* yang sama dan mempunyai rentang nilai ip *length* sebesar 74 sampai 82. Disimpulkan pola akses normal dengan pola

serangan *brute force* menggunakan kali linux hampir sama yang membedakan adalah nilai fitur window dan rentang nilai ip *length*.

Ip address source “any” port source “any” ip address destination “192.168.100.7” port destination “80” protocol “tcp” flags “PA” ttl “252” ip length “113-1023”

Gambar 4.49 Pola Akses Normal Menggunakan Windows 8, Android, Kali Linux

Ip address source “any” port source “any” ip address destination “192.168.100.7” port destination “80” protocol “tcp” window “4320” flags “PA” ttl “252” ip length “74-82”

Gambar 4.50 Pola Serangan *Brute Force* Menggunakan Kali Linux

4.8 Kinerja Snort Sebagai NIDS

Pola serangan *brute force* pada gambar 4.50 akan diimplementasikan ke dalam *engine* snort sebagai *rules* untuk mendeteksi serangan *brute* force.Untuk menguji pola serangan *brute force* pada gambar 4.50 dilakukan skenario pengujian kelima (5) yaitu akses normal menggunakan windows ketika serangan *brute force* dilakukan. Pengujian skenario kelima (5) menghasilkan *raw data* (pcap) sebesar 9.00 MB dengan jumlah paket data sebanyak 13.650 paket.

Tabel 25

Jumlah Paket Berdasarkan Protokol Skenario Pengujian Lima

Protokol	Jumlah Paket
Http	12.959
Mdns	102
Tcp	133
Jumlah	**13.194**

Tabel 26

Aktivitas Hasil Skenario Pengujian Lima

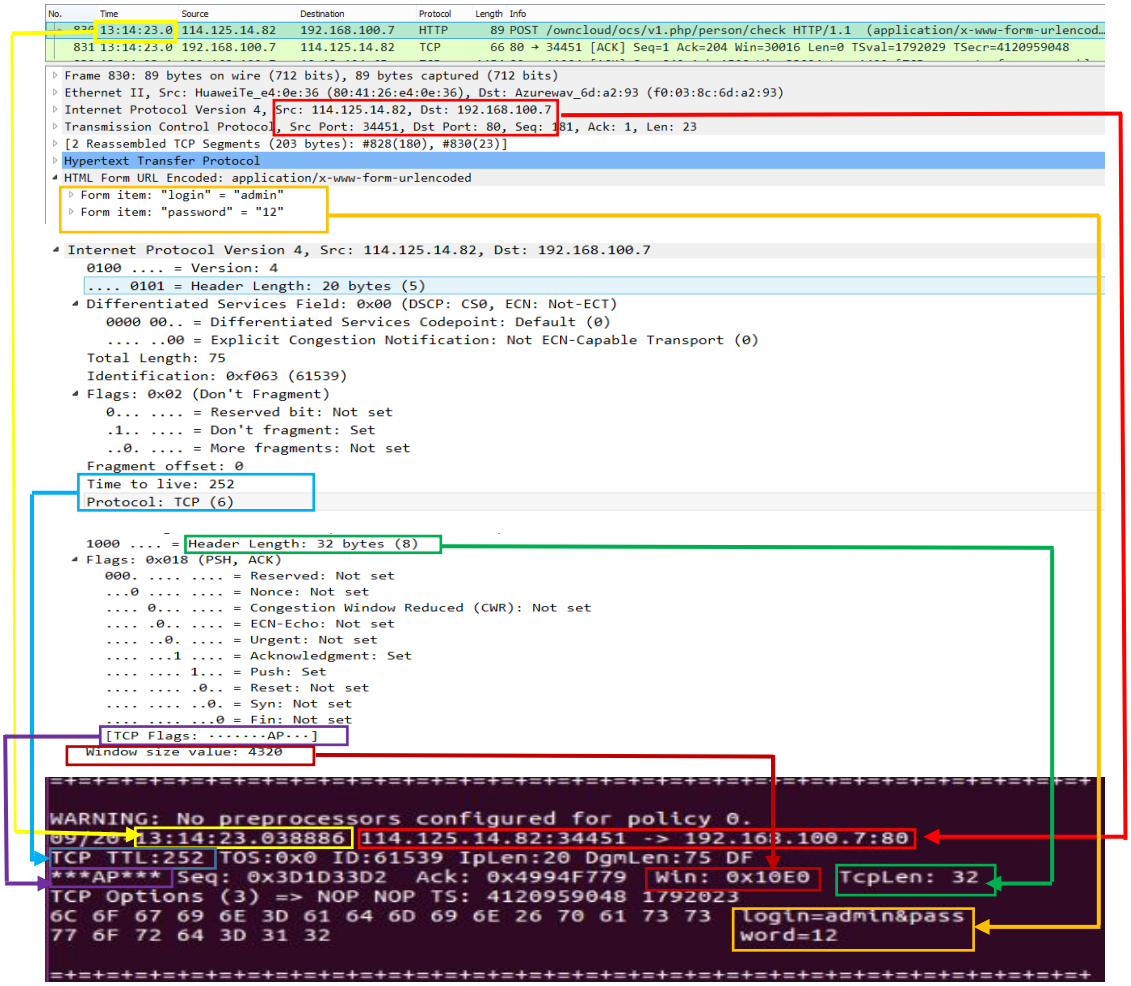
No Paket	Ip *Address*	Keterangan
476	10.13.124.65	*Login* normal menggunakan *broswer* chrome pada windows 7 dengan *user* "sayang" *password* "kamu"
511	10.13.124.65	*Login* normal menggunakan *broswer* chrome pada windows 7 dengan *user* "sayang" *password* "dia"
557	10.13.124.65	*Login* normal menggunakan *broswer* chrome pada windows 7 dengan *user* "sayang" *password* "saya"
601	10.13.124.65	*Login* normal menggunakan *broswer* chrome pada windows 7 dengan *user* "sk" password "2"
640 – 4192	114.125.12.207 114.125.14.226 114.125.28.210 114.125.31.243 114.125.15.83	Serangan *brute force* dilakukan menggunakan kali linux dengan *wordlist* berisi 111 kata.
808	10.13.124.65	*Login* normal menggunakan *broswer* chrome pada windows 7 dengan *user* "sk2" *password* "2"
1422	10.13.124.65	*Upload file* dengan judul Dimas Wahyudi_09011281320004_revisi.pdf menggunakan akun id "sk2"
10834	114.125.15.83	*Login* normal menggunakan *browser* firefox pada kali linux dengan *user* "admin" *password* "password"
11126	114.125.31.243	*Login* normal menggunakan *browser* firefox pada kali linux dengan *user* "admin" *password* "digimon"

12195	114.125.15.83	*Logout* dari akun id “admin”
12377	10.13.124.65	*Upload file* dengan judul Table 2.8 Waist loss.xls menggunakan akun id “sk2”
12409	10.13.124.65	*Upload file* dengan judul Titanic.csv menggunakan akun id “sk2”
12534	114.125.31.243	*Login* normal menggunakan *browser* firefox pada kali linux dengan *user* “admin” *password* “digimon”
12655	114.125.15.83	*Upload file* dengan judul Image (4).jpeg menggunakan akun id “admin”
12770	114.125.15.83	*Upload file* dengan judul IMG-20180304-WA0006.jpg menggunakan akun id “admin”
13192	10.13.124.65	*Logout* dari akun id “sk2”
13438	114.125.12.207	Mengganti *password* akun id “admin” dari “digimon” menjadi “palembang”
13538	114.125.12.207	Menghapus file Image (4).jpeg dari akun id “admin”
13599	114.125.31.243	*Logout* dari akun id “admin”

Tabel 27

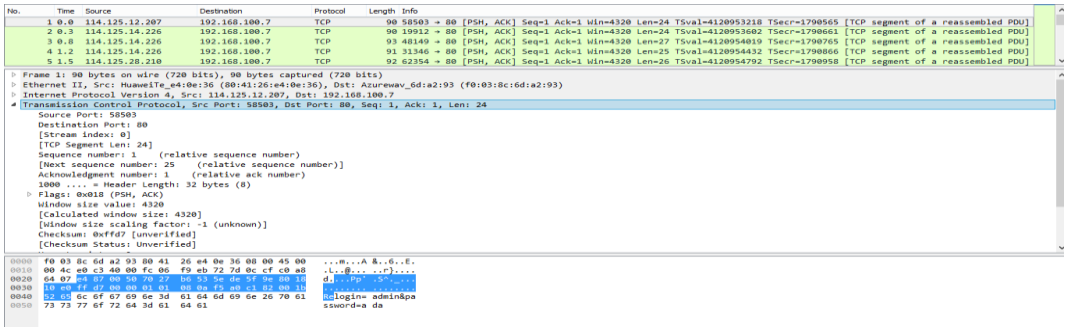
Aktivitas Hasil Skenario Pengujian Lima Berdasarkan Ip *Address*

Ip *address*	Keterangan
10.13.124.65	Gagal *login* sebanyak 4 kali, *login* menggunakan akun id “sk2”, *upload* 3 file, dan *logout*.
114.125.12.207 114.125.14.226 114.125.28.210 114.125.31.243 114.125.15.83	Gagal *login* sebanyak 112 kali dalam jangka waktu yang berdekatan, *login* menggunakan akun id “admin”, *upload* 2 file, mengganti *password* akun id “admin”, menghapus 1 file, dan *logout*.



Gambar 4.51 Validitas Hasil *Alert* Snort pada Skenario Pengujian Lima

Snort akan menghasilkan file *log* untuk menyimpan hasil *capture* snort berdasarkan *rules* yang aktif pada snort. Pada skenario pengujian kelima, snort menghasilkan snort.log.1537423604 dengan ukuran 14,1 KB. Wireshark dapat digunakan untuk membaca hasil *log* yang dihasilkan snort.



Gambar 4.52 *Log File* Snort pada Wireshark

Snort menghasilkan 136 *alert* paket data dari total jumlah paket 13.650 paket. *Wordlist* yang digunakan pada skenario pengujian kelima memiliki 111 kata sehingga serangan *brute force* yang dilakukan sebanyak 111 kali. Hasil perhitungan *confusion matrix* pada skenario pengujian lima ditunjukkan pada tabel 28.

Tabel 28

***Confusion Matrix* Skenario Pengujian Lima**

No	Hasil Kategori	Jumlah	Persentase (%)
1.	TP	111/111	-
2.	TN	13.514/13.539	-
3.	FP	25/13.539	-
4.	FN	0	-
5.	TPR	0,81	81%
6.	TNR	0,99	99%
7.	FPR	0,0018	0,18%
8.	FNR	0	0%
9.	*Precision*	0,81	81%
10.	*Accuracy*	0,99	99%

Dari data skenario pengujian lima pada tabel 28 dapat disimpulkan pola serangan *brute force* pada gambar 4.50 yang diimplementasikan dalam *engine* snort memiliki tingkat akurasi yang tinggi sebesar 99% dan menangkap seluruh paket serangan *brute force* yang dilakukan, sehingga dapat disimpulkan pola serangan *brute force* pada gambar 4.50 adalah benar.

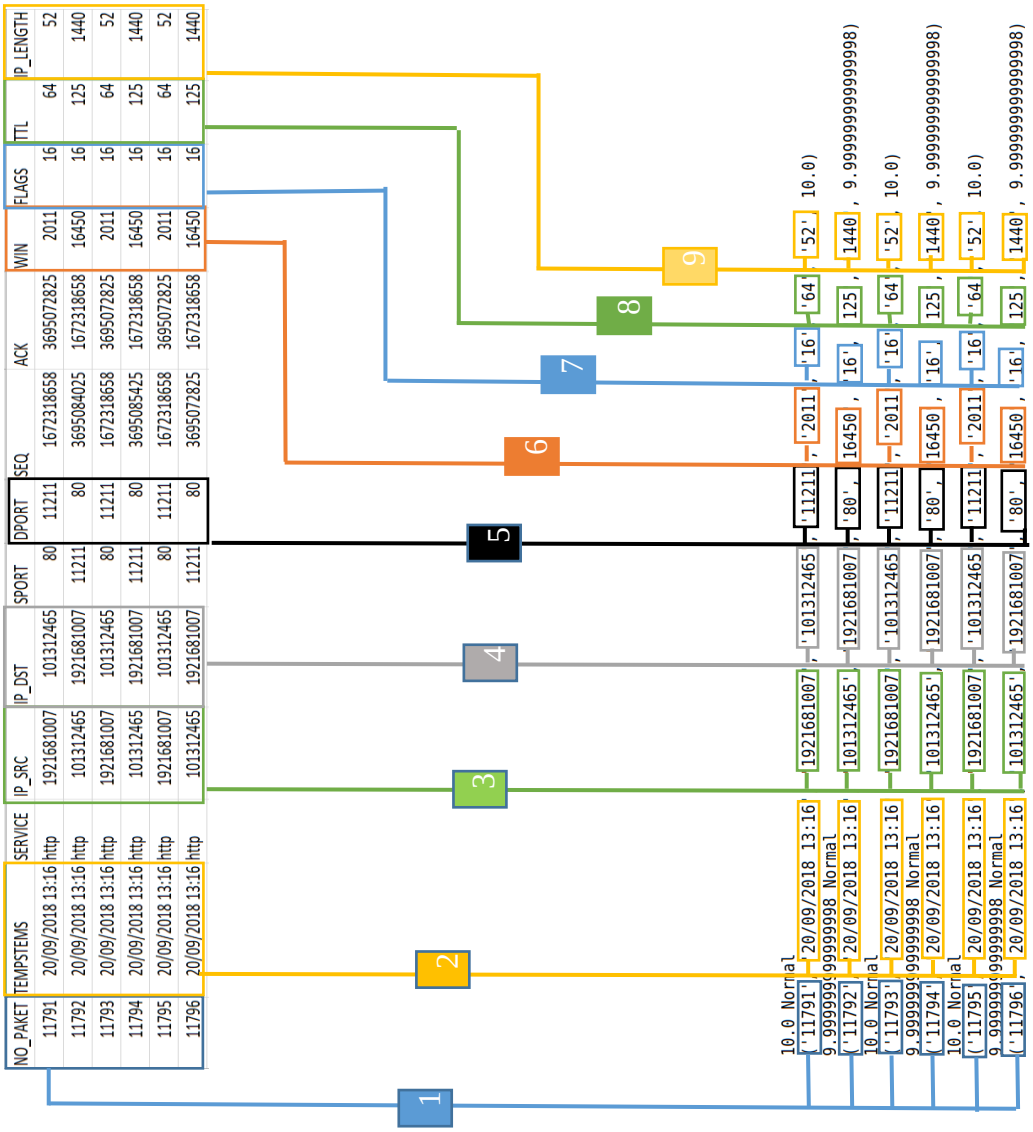
4.9 Implementasi Logika Fuzzy

Pada hasil kinerja snort yang telah dimasukkan pola serangan brute force sebagai NIDS didapatkan akurasi sebesar 99% dalam mendeteksi serangan brute force. Pola serangan brute force pada gambar 4.50 terdapat 5 variabel yang bisa digunakan sebagai input dari system logika fuzzy dalam menentukan paket normal dan paket serangan brute force yaitu : Port destination, window, flags, ttl, dan ip

length. Hasil dari system logika fuzzy pada penentuan jenis paket terhadap serangan brute force dijelaskan pada gambar 4.53.

```
10.0 Normal
('11791', '20/09/2018 13:16', '1921681007', '101312465', '11211', '2011', '16', '64', '52', 10.0)
9.999999999999998 Normal
('11792', '20/09/2018 13:16', '101312465', '1921681007', '80', '16450', '16', '125', '1440', 9.999999999999998)
10.0 Normal
('11793', '20/09/2018 13:16', '1921681007', '101312465', '11211', '2011', '16', '64', '52', 10.0)
9.999999999999998 Normal
('11794', '20/09/2018 13:16', '101312465', '1921681007', '80', '16450', '16', '125', '1440', 9.999999999999998)
10.0 Normal
('11795', '20/09/2018 13:16', '1921681007', '101312465', '11211', '2011', '16', '64', '52', 10.0)
9.999999999999998 Normal
('11796', '20/09/2018 13:16', '101312465', '1921681007', '80', '16450', '16', '125', '1440', 9.999999999999998)
```

Gambar 4.53 Output Sistem Logika Fuzzy Pada Dataset Skenario Pengujian Lima



Gambar 4.54 Validasi Output Sistem Logika Fuzzy Terhadap Raw Data

Pada gambar 4.54 terdapat sembilan (9) poin persamaan data antara hasil system logika fuzzy dengan *raw data* (csv). Berikut penjelasan mengenai sembilan poin data yang sama pada hasil system logika fuzzy dan *raw data* (csv):

1. Nomor satu berisi data tentang nomor paket. Pada hasil system logika fuzzy dan raw data (csv) nomor paket bernilai 11791, 11792, 11793, 11794, 11795, dan 11796.
2. Nomor dua berisi data tentang waktu dan tanggal paket. Pada hasil system logika fuzzy dan raw data (csv) waktu dan tanggal paket bernilai 20/09/2019 13:16.
3. Nomor tiga berisi data tentang ip source. Pada hasil system logika fuzzy dan raw data (csv) ip source bernilai 192.168.100.7 dan 10.13.124.65.
4. Nomor empat berisi data tentang ip destination. Pada hasil system logika fuzzy dan raw data (csv) ip destination bernilai 192.168.100.7 dan 10.13.124.65.
5. Nomor lima berisi data tentang port destination. Pada hasil system logika fuzzy dan raw data (csv) port destination bernilai 11211 dan 80.
6. Nomor enam berisi data tentang nilai window. Pada hasil system logika fuzzy dan raw data (csv) nilai window bernilai 2011 dan 16450.
7. Nomor tujuh berisi data tentang flags. Pada hasil system logika fuzzy dan raw data (csv) nilai flags bernilai 16.
8. Nomor delapan berisi data tentang ttl. Pada hasil system logika fuzzy dan raw data (csv) nilai ttl bernilai 64 dan 125.
9. Nomor sembilan berisi data tentang ip length. Pada hasil system logika fuzzy dan raw data (csv) nilai ip length bernilai 52 dan 1440.

Data pada raw data (csv) dan data pada output system logika fuzzy bernilai sama sehingga tidak ada perubahan data ketika raw data diproses oleh system logika fuzzy. Output nilai dari system logika fuzzy akan dibandingkan dengan hitungan secara manual untuk melihat ketepatan hitungan system logika fuzzy. Akan diambil masing-masing dua paket serangan dan paket normal untuk digunakan sebagai sample dalam perbandingan output nilai system logika fuzzy

dengan output nilai secara manual. Berikut output system logika fuzzy dan perhitungan secara manual :

❖ Paket Normal

NOPAKET	TIME	IP_SOURCE	IP_DESTINATION	DPORT	WINDOW	FLAGS	TTL	IP_LENGTH	OUTPUT_FUZZY
1	20/09/2018 13:06	176119758	1921681007	40187	1024	2	243	40	10
2	20/09/2018 13:06	1921681007	176119758	42493	0	20	64	40	10

Gambar 4.55 Output Paket Normal Sistem Logika Fuzzy

Pada gambar 4.55 terdapat dua paket normal dengan nilai 10 hasil output system logika fuzzy. Perhitungan secara manual pada paket gambar 4.55 sebagai berikut :

- Nomor Paket 1
  a) Variabel Input Destination Port

     Nilai input destination port pada paket satu adalah 40187 dengan merujuk pada gambar 3.8 maka nilai 40187 terletak pada nilai lingusitik ‘keluar’. Nilai linguistic ‘keluar’ mempunyai nilai 81 sampai 100.000.

     μKosong = 0 ; μtelnet =0 ; μssh = 0 ; μhttp = 0; μkeluar = 1;

  b) Variabel Input Window

     Nilai input variabel window pada paket satu adalah 1024 dengan merujuk pada gambar 3.9 maka nilai 1024 terletak pada nilai linguistic ‘Vlow’ dan ‘Low’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 7 sebagai berikut :

     μVlow = (8000-1024/8000-0) = 0,872;
     μLow = (1024-0/8000-0) = 0,128;
     μMedium = 0; μHigh = 0; μVhigh = 0;

c) Variabel Input Flags

Nilai input variabel flags pada paket satu adalah 2 dengan merujuk pada gambar 3.10 maka nilai 2 terletak pada nilai linguistic ‘S’. Nilai linguistic ‘S’ mempunyai nilai 2.

μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 0; μRA = 0;
μS = 1;

d) Variabel Input TTL

Nilai input variabel ttl pada paket satu bernilai 243 dengan merujuk pada gambar 3.11 maka nilai 243 terletak pada nilai linguistic ‘High’. Nilai linguistic ‘High’ mempunyai rentang nilai dari 85 sampai 255.

μLow = 0; μMedium = 0; μHigh = 1;

e) Variabel Input Ip Length

Nilai input variabel ip length pada paket satu bernilai 40 dengan merujuk pada gambar 3.12 maka nilai 40 terletak pada nilai linguistic ‘Low’ dan ‘Medium’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 10 sebagai berikut:

μLow = (85-40/85-0) = 0,53;
μMedium = (40-0/85-0) = 0,47;
μHigh = 0;

f) Interfensi Logika Fuzzy

(Dport) μKosong = 0 ; μtelnet =0 ; μssh = 0 ;
μhttp = 0; μkeluar = 1;

(Window) μVlow = 0,872 ; μLow = 0,128; μMedium = 0; μHigh = 0; μVhigh = 0;
(Flags) μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0; μFA = 0; μSA =0; μPA = 0; μRA = 0; μS = 1;
(TTL) μLow = 0; μMedium = 0; μHigh = 1;
(IP Length) μLow = 0,53; μMedium = 0,47; μHigh = 0;

(**Rule 52**) jika destination port 'keluar' & window 'vlow' & flags 's' & ttl 'High' & Ip Length 'Low' maka jenis paket 'normal'

(**Rule 62**) jika destination port 'keluar' & window 'Low' & flags 's' & ttl 'High' & Ip Length 'Medium' maka jenis paket 'normal'

μNormal = 1;
μSerangan = 0;

g) Defuzifikasi

Defuzifikasi dari sampel pertama dapat dihitung melalui persamaan sepuluh (10) sebagai berikut:

$$\text{Def} = \sum \frac{[\mu(\text{Normal}).\widehat{\text{normal}}+\mu(\text{Serangan}).\widehat{\text{Serangan}}]}{[\mu(\text{normal})+\mu(\text{serangan})]}$$

$$\text{Def} = \sum \frac{[\mu(1).10+\mu(0).20]}{[\mu(1)+\mu(0)]}$$

Defuzifikasi = 10

Pada gambar 4.55 nilai defuzifikasi dari paket normal dengan nomor paket satu adalah 10.

- Nomor Paket 2

a) Variabel Input Destination Port

Nilai input destination port pada paket dua adalah 42493 dengan merujuk pada gambar 3.8 maka nilai 42493 terletak pada

nilai lingusitik ‘keluar’. Nilai linguistic ‘keluar’ mempunyai nilai 81 sampai 100.000.

μKosong = 0 ; μtelnet =0 ; μssh = 0 ; μhttp = 0; μkeluar = 1;

b) Variabel Input Window

Nilai input variabel window pada paket dua adalah 0 dengan merujuk pada gambar 3.9 maka nilai 0 terletak pada nilai linguistic ‘Vlow’ dan ‘Low’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 7 sebagai berikut :

μVlow = (8000-0/8000-0) = 1;
μLow = (0-0/8000-0) = -;
μMedium = 0; μHigh = 0; μVhigh = 0;

c) Variabel Input Flags

Nilai input variabel flags pada paket dua adalah 20 dengan merujuk pada gambar 3.10 maka nilai 20 terletak pada nilai linguistic ‘S’. Nilai linguistic ‘S’ mempunyai nilai 20.

μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 0; μRA = 1;
μS = 0;

d) Variabel Input TTL

Nilai input variabel ttl pada paket dua bernilai 64 dengan merujuk pada gambar 3.11 maka nilai 64 terletak pada nilai linguistic ‘Low’ dan ‘Medium’.

μLow = (85-64/85-0) = 0,247;
μMedium = (64-0/85-0) = 0,753;
μHigh = 0;

e) Variabel Input Ip Length

Nilai input variabel ip length pada paket dua bernilai 40 dengan merujuk pada gambar 3.12 maka nilai 40 terletak pada nilai linguistic ‘Low’ dan ‘Medium’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 10 sebagai berikut:

μLow = (85-40/85-0) = 0,53;
μMedium = (40-0/85-0) = 0,47;
μHigh = 0;

f) Interfensi Logika Fuzzy

(Dport) μKosong = 0 ; μtelnet =0 ; μssh = 0 ;
μhttp = 0; μkeluar = 1;
(Window) μVlow = 0,872 ; μLow = 0,128; μMedium = 0; μHigh = 0; μVhigh = 0;
(Flags) μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 0; μRA = 0; μS = 1;
(TTL) μLow = 0; μMedium = 0; μHigh = 1;
(IP Length) μLow = 0,53; μMedium = 0,47; μHigh = 0;

(**Rule 46**) jika destination port ‘keluar’ & window ‘vlow’ & flags ‘RA’ & ttl ‘Low’ & Ip Length ‘Low’ maka jenis paket ‘normal’

(**Rule 59**) jika destination port ‘keluar’ & window ‘Low’ & flags ‘s’ & ttl ‘Medium’ & Ip Length ‘Medium’ maka jenis paket ‘normal’
μNormal = 1;
μSerangan = 0;

g) Defuzifikasi

Defuzifikasi dari sampel kedua dapat dihitung melalui persamaan sepuluh (10) sebagai berikut:

$$Def = \sum \frac{[\mu(Normal).normal+\mu(Serangan).Serangan]}{[\mu(normal)+\mu(serangan)]}$$

$$Def = \sum \frac{[\mu(1).10+\mu(0).20]}{[\mu(1)+\mu(0)]}$$

Defuzifikasi = 10

Pada gambar 4.55 nilai defuzifikasi dari paket normal dengan nomor paket dua adalah 10.

❖ Paket Serangan

NO PAKET	TIME	IP SOURCE	IP DESTINATION	PORT DESTINATION	WINDOW	FLAGS	TTL	IPLEN	OUTPUT_FUZZY
640	43363,55139	11412512207	1921681007	80	4320	24	252	76	20
653	43363,55139	11412514226	1921681007	80	4320	24	252	76	20

Gambar 4.56 Output Paket Serangan Sistem Logika Fuzzy

- Nomor Paket 640

a) Variabel Input Destination Port

Nilai input destination port pada paket 640 adalah 80 dengan merujuk pada gambar 3.8 maka nilai 80 terletak pada nilai lingusitik ‘HTTP’. Nilai linguistic ‘HTTP’ mempunyai nilai 80.

μKosong = 0 ; μtelnet =0 ; μssh = 0 ; μhttp = 1; μkeluar = 0;

b) Variabel Input Window

Nilai input variabel window pada paket 640 adalah 4320 dengan merujuk pada gambar 3.9 maka nilai 4320 terletak pada nilai linguistic ‘Vlow’ dan ‘Low’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 7 sebagai berikut :

μVlow = (8000-4320/8000-0) = 0,46;

μLow = (4320-0/8000-0) = 0,54;
μMedium = 0; μHigh = 0; μVhigh = 0;

c) Variabel Input Flags

Nilai input variabel flags pada paket 640 adalah 24 dengan merujuk pada gambar 3.10 maka nilai 24 terletak pada nilai linguistic ‘PA’. Nilai linguistic ‘PA’ mempunyai nilai 24.

μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 1; μRA = 0;
μS = 0;

d) Variabel Input TTL

Nilai input variabel ttl pada paket 640 bernilai 252 dengan merujuk pada gambar 3.11 maka nilai 252 terletak pada nilai linguistic ‘High’.

μLow = 0;
μMedium = 0;
μHigh = 1;

e) Variabel Input Ip Length

Nilai input variabel ip length pada paket 640 bernilai 76 dengan merujuk pada gambar 3.12 maka nilai 76 terletak pada nilai linguistic ‘Low’ dan ‘Medium’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 10 sebagai berikut:

μLow = (85-76/85-0) = 0,105;
μMedium = (76-0/85-0) = 0,895;
μHigh = 0;

f) Interfensi Logika Fuzzy

(Dport) μKosong = 0 ; μtelnet =0 ; μssh = 0 ;
μhttp = 0; μkeluar = 1;
(Window) μVlow = 0,46 ; μLow = 0,54; μMedium = 0; μHigh = 0; μVhigh = 0;
(Flags) μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0; μFA = 0; μSA =0; μPA = 1; μRA = 0; μS = 0;
(TTL) μLow = 0; μMedium = 0; μHigh = 1;
(IP Length) μLow = 0,105; μMedium = 0,895; μHigh = 0;

(**Rule 7**) jika destination port ‘http’ & window ‘vlow’ & flags ‘PA’ & ttl ‘High’ & Ip Length ‘Low’ maka jenis paket ‘Serangan’

(**Rule 17**) jika destination port ‘http’ & window ‘Low’ & flags ‘PA’ & ttl ‘High’ & Ip Length ‘Medium’ maka jenis paket ‘Serangan’
μNormal = 0;
μSerangan = 1;

g) Defuzifikasi

Defuzifikasi dari sampel ketiga dapat dihitung melalui persamaan sepuluh (10) sebagai berikut:

$$\text{Def} = \sum \frac{[\mu(\text{Normal}).\widehat{\text{normal}}+\mu(\text{Serangan}).\widehat{\text{Serangan}}]}{[\mu(\text{normal})+\mu(\text{serangan})]}$$

$$\text{Def} = \sum \frac{[\mu(0).10+\mu(1).20]}{[\mu(0)+\mu(1)]}$$

Defuzifikasi = 20

Pada gambar 4.55 nilai defuzifikasi dari paket serangan dengan nomor paket 640 adalah 20.

- Nomor Paket 653

a) Variabel Input Destination Port

Nilai input destination port pada paket 653 adalah 80 dengan merujuk pada gambar 3.8 maka nilai 80 terletak pada nilai lingusitik ‘HTTP’. Nilai linguistic ‘HTTP’ mempunyai nilai 80.

μKosong = 0 ; μtelnet =0 ; μssh = 0 ; μhttp = 1; μkeluar = 0;

b) Variabel Input Window

Nilai input variabel window pada paket 653 adalah 4320 dengan merujuk pada gambar 3.9 maka nilai 4320 terletak pada nilai linguistic ‘Vlow’ dan ‘Low’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 7 sebagai berikut :

μVlow = (8000-4320/8000-0) = 0,46;
μLow = (4320-0/8000-0) = 0,54;
μMedium = 0; μHigh = 0; μVhigh = 0;

c) Variabel Input Flags

Nilai input variabel flags pada paket 653 adalah 24 dengan merujuk pada gambar 3.10 maka nilai 24 terletak pada nilai linguistic ‘PA’. Nilai linguistic ‘PA’ mempunyai nilai 24.

μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 1; μRA = 0;
μS = 0;

d) Variabel Input TTL

Nilai input variabel ttl pada paket 653 bernilai 252 dengan merujuk pada gambar 3.11 maka nilai 252 terletak pada nilai linguistic ‘High’.

μLow = 0;
μMedium = 0;
μHigh = 1;

e) Variabel Input Ip Length

Nilai input variabel ip length pada paket 653 bernilai 76 dengan merujuk pada gambar 3.12 maka nilai 76 terletak pada nilai linguistic ‘Low’ dan ‘Medium’. Untuk menentukan derajat keanggotaan yang berpotongan dapat menggunakan persamaan pada tabel 10 sebagai berikut:

μLow = (85-76/85-0) = 0,105;
μMedium = (76-0/85-0) = 0,895;
μHigh = 0;

f) Interfensi Logika Fuzzy

(Dport) μKosong = 0 ; μtelnet =0 ; μssh = 0 ;
μhttp = 0; μkeluar = 1;
(Window) μVlow = 0,46 ; μLow = 0,54; μMedium = 0; μHigh = 0; μVhigh = 0;
(Flags) μKosong = 0; μF = 0; μR =0; μP =0; μA = 0; μU = 0;
μFA = 0; μSA =0; μPA = 1; μRA = 0; μS = 0;
(TTL) μLow = 0; μMedium = 0; μHigh = 1;
(IP Length) μLow = 0,105; μMedium = 0,895; μHigh = 0;

(**Rule 7**) jika destination port ‘http’ & window ‘vlow’ & flags ‘PA’ & ttl ‘High’ & Ip Length ‘Low’ maka jenis paket ‘Serangan’

(**Rule 17**) jika destination port ‘http’ & window ‘Low’ & flags ‘PA’ & ttl ‘High’ & Ip Length ‘Medium’ maka jenis paket ‘Serangan’

μNormal = 0;

μSerangan = 1;

g) Defuzifikasi

Defuzifikasi dari sampel keempat dapat dihitung melalui persamaan sepuluh (10) sebagai berikut:

$$Def = \sum \frac{[\mu(Normal).normal+\mu(Serangan).Serangan]}{[\mu(normal)+\mu(serangan)]}$$

$$Def = \sum \frac{[\mu(0).10+\mu(1).20]}{[\mu(0)+\mu(1)]}$$

Defuzifikasi = 20

Pada gambar 4.55 nilai defuzifikasi dari paket serangan dengan nomor paket 653 adalah 20.

Hasil perbandingan perhitungan secara manual dan output system logika fuzzy bernilai sama, sehingga perhitungan system logika fuzzy adalah benar. System logika fuzzy menghasilkan output paket normal sebanyak 13.327 paket sedangkan untuk paket serangan berjumlah 192 paket. Dari 111 serangan brute force yang dilakukan system logika fuzzy mampu menangkap semua serangan tersebut akan tetapi terdapat false positive pada system logika fuzzy sebanyak 82 paket. Hal ini disebabkan karena 82 paket tersebut mempunyai perbedaan yang sangat kecil yang ditunjukkan pada gambar 4.56 dan 4.57.

4134	43363,55208	11412528210	1921681007	80	4320	24	252	76	20	login=admin&password=110
4192	43363,55208	1141251482	1921681007	80	4320	24	252	80	20	login=admin&password=digimon
12665	43363,55347	1141251583	1921681007	80	4320	24	252	80	20	....S.....YNG.v.\p..h...A
12669	43363,55347	1141251583	1921681007	80	4320	24	252	80	20	.....[..C-.. ....g.....S...

Gambar 4.56 Perbedaan Paket Serangan Terhadap Paket False Positive

12772	43363,55347	1141251583	1921681007	80	4954	24	252	80	20	.Xq....A.....Q....z..y.8.
12776	43363,55347	1141251583	1921681007	80	4954	24	252	80	20	m9...*t...jx....0.OJ.w.3....
12780	43363,55347	1141251583	1921681007	80	4954	24	252	80	20	ql..@.1....^t.3..:.Z.J....c.
12786	43363,55347	1141251583	1921681007	80	4954	24	252	80	20	q.q.+rA..G..c=h..k..u._..l<.
12790	43363,55347	1141251583	1921681007	80	4954	24	252	80	20	f.&....qk..qMF..... `..1>.

Gambar 4.57 Paket False Positive Output Sistem Logika Fuzzy

Pada gambar 4.56 dam 4.57 terdapat perbedaan antara paket serangan dengan paket false positive. Pada gambar 4.56 perbedaan paket serangan dan paket false positive terletak pada isi dari payload, sedangkan untuk gambar 4.57 perbedaan paket serangan dan paket false positive terletak pada nilai window dan isi payload.

Tabel 29

***Confusion Matrix* Skenario Pengujian Lima Menggunakan Logika Fuzzy**

No	Hasil Kategori	Jumlah	Persentase (%)
1.	TP	111/111	-
2.	TN	13.347/13.539	-
3.	FP	81/13.539	-
4.	FN	0	-
5.	TPR	1	100%
6.	TNR	0,99	99%
7.	FPR	0,006	0,6%
8.	FNR	0	0%
9.	*Precision*	0,58	58%
10.	*Accuracy*	0,99	99%

Dari hasil confusion matrix pada tabel 29, metode logika fuzzy mampu menangkap 111 dari 111 serangan dan mempunyai false positive sebanyak 81 paket. Dengan tingkat akurasi sebesar 99% metode logika fuzzy dapat digunakan untuk mendeteksi serangan brute force pada cloud dengan hasil yang sangat baik.

BAB V

KESIMPULAN (SEMENTARA)

5.1 Kesimpulan

Setelah dilakukan lima kali pengujian dan analisa data hasil pengujian, maka dapat disimpulkan bahwa:

1. Owncloud memiliki kelemahan pada *Application Programming Interface* (API) sehingga dapat dilakukan serangan *brute force* pada API tersebut.
2. Owncloud pada halaman *login* tidak membatasi jumlah maksimum gagal login pada *user*. Pada halaman *login* owncloud, *user* diminta untuk memasukkan *username* dan *password,* selain itu sistem *user* akan otomatis mengirim data *user* tentang *timezone, timezone-offset* , dan *request token*. Nilai *request token* bersifat *random value* sehingga tidak dapat dilakukan serangan *brute force* pada halaman *login* owncloud
3. Pola akses normal pada owncloud memiliki beberapa nilai fitur yang sama yaitu : *protocol* "tcp", *flags*"PA", *ttl* "252", *ip length* "113-1023".

Ip address source "any" port source "any" ip address destination "192.168.100.7" port destination "80" protocol "tcp" flags "PA" ttl "252" ip length "113-1023"

4. Pola serangan *brute force* pada owncloud memiliki beberapa nilai fitur yang sama yaitu : *protocol* "tcp", *flags* "PA", *window* "4320", *ttl* "252", *ip length* "113-1023".

Ip address source "any" port source "any" ip address destination "192.168.100.7" port destination "80" protocol "tcp" window "4320" flags "PA" ttl "252" ip length "74-82"

5. Perbedaan pola serangan *brute force* dan akses normal terletak pada nilai fitur *window* dan nilai ip *length*. Nilai ip *length* dipengaruhi oleh besarnya data. Pada paket normal ketika *login, user* akan mengirim 5 data yaitu *username, password, timezone, timezone-offset, request token* sehingga data yang dikirimkan *user* menjadi besar. Pada paket serangan *brute force*, data yang dikirim oleh *attacker* hanya 2 yaitu *username* dan *password* sehingga data yang dikirimkan berukuran kecil.
6. Variabel input dalam mendeteksi serangan brute force pada cloud menggunakan logika fuzzy adalah : Destination port, window, flags, ttl, dan ip length.
7. Penggunaan metode fuzzy dalam mendeteksi serangan brute force pada cloud memiliki tingkat akurasi sebesar 99%.
8. Perbedaan paket serangan dengan paket false positive terletak pada nilai window yang berdekatan dan isi payload paket
9. Penggunaan metode fuzzy dengan 118 rules akan memakan waktu empat sampai enam detik untuk memproses satu data dari raw data (csv).
10. Diperlukan sebuah metode yang lebih efisien dalam memproses raw data dengan menggunakan banyak rules.
11. Diperlukan metode yang dapat mendeteksi serangan brute force pada cloud menggunakan payload dari raw data (csv)