DESAIN DAN IMPLEMENTASI KEAMANAN JARINGAN *FIREWALL* PADA *EMBEDDED SYSTEM*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

YOGA PRADANA PAMUNGKAS

NIM. 0910683099

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

MALANG

2013

PERNYATAAN
ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 30 November 2013

Mahasiswa,

Yoga Pradana Pamungkas
NIM. 0910683099

LEMBAR PERSETUJUAN

DESAIN DAN IMPLEMENTASI KEAMANAN JARINGAN *FIREWALL* PADA *EMBEDDED SYSTEM*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

YOGA PRADANA PAMUNGKAS

NIM. 0910683099

Telah diperiksa dan disetujui oleh:

Pembimbing I	Pembimbing II
Barlian Henryranu P, S.T, M.T.	**Eko Setiawan,S.T.,M.Eng.**
NIP. 821024 06 1 1 0254	**NIK. 870610 06 1 1 0256**

LEMBAR PENGESAHAN

DESAIN DAN IMPLEMENTASI KEAMANAN JARINGAN *FIREWALL* PADA *EMBEDDED SYSTEM*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

YOGA PRADANA PAMUNGKAS

NIM. 0910683099

Skripsi ini diuji dan dinyatakan lulus pada
tanggal 20 Desember 2013

Penguji I	Penguji II
Adharul Muttaqin, S.T., M.T.	**Wijaya Kurniawan, S.T., M.T.**
NIP. 19760121 200501 1 001	**NIP. 820125 16 1 1 0481**

Penguji III

Gembong Edhi Setyawan, S.T., M.T.
NIK. 761201 16 1 1 0373

Mengetahui,
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji., M.T.
NIP. 19670801 199203 1 001

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena hanya dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan proposal skripsi dengan judul "*Desain dan Implementasi Keamanan Jaringan Firewall pada Embedded System*". Penulisan skripsi ini disusun untuk memenuhi syarat menjadi Sarjana Komputer.

Dalam pelaksanaan dan penulisan skripsi ini penulis mendapatkan banyak bantuan dari berbagai pihak baik secara moril dan materiil. Dalam kesempatan ini penulis ingin mengucapkan terimakasih sebesar - besarnya kepada:

1. Papa dan Mama yang selalu memberikan dorongan semangat dan doa kepada penulis dalam mengerjakan skripsi.
2. Barlian Henryranu P, S.T, M.T. selaku pembimbing utama yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis.
3. Eko Setiawan,S.T.,M.Eng. selaku pembimbing pendamping yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis
4. Bapak Ir. Sutrisno, M.T. selaku Ketua Program Teknologi Informasi dan Ilmu Komputer dan selaku dosen penasihat akademik penulis.
5. Bapak Drs. Mardji, M.T. selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer.
6. Segenap dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan.
7. Segenap staff dan pegawai Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.
8. Teman-teman kontrakan Griya Shanta E714, Bayu Priyo J. , Edwin Panigoro, Fathul Dani Liatama , Muhammad Dika Hidayat, dan Amin Sugeng Laksono atas kebersamaan rumah tangga selama pengerjaan skripsi.
9. Adien Faishol, Antarangga B. , Bintang Karunia A. , Delis Sukmawati, I Putu Arie Bayu Antara, Mohammad Halimi, M. Rizki Gus Sofwan, Rizky

Kurnia Aziz Musthofa, Om Alizar Erwinsyah Uber, dan Lik Yahya Ansori serta segenap Keluarga Besar Parkiran atas semua bantuan yang diberikan kepada penulis selama pengerjaan skripsi.

10. Seluruh teman-teman TIF C UB 09 tercinta atas dukungan dan kebersamaannya dari awal perkuliahan sampai akhir.
11. Teman-teman Program Studi Informatika/Ilmu Komputer angkatan 2009 tercinta yang selalu memberikan semangat, dorongan, dan bantuan pikiran.
12. Serta semua pihak yang telah membantu dan memberikan pengalaman berharga bagi penulis selama penulis menjalani masa perkuliahan.

Akhirnya atas segala bantuan semua pihak semoga mendapat balasan yang setimpal dari Allah SWT. Penulis sadar bahwa skripsi ini masih jauh dari kesempurnaan, untuk itu dengan segala kerendahan hati penulis mengharapkan saran dan kritik yang membangun demi kesempurnaan penulisan selanjutnya. Harapan penulis semoga skripsi ini dapat memberikan manfaat pada semua pihak.

Malang, 30 November 2013

Penulis

ABSTRAK

Yoga Pradana Pamungkas. 2013. : Desain dan Implementasi Keamanan Jaringan *Firewall* pada *Embedded System*. Penelitian Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing: Barlian Henryranu P, ST, MT. dan Eko Setiawan, ST., M.Eng.

Keamanan jaringan penting untuk diperhatikan karena merupakan aspek pertahanan suatu jaringan komputer. *Firewall* merupakan salah satu sistem keamanan yang berfungsi untuk mencegah penggunaan yang tidak sah serta pembatasan akses tertentu pada suatu perangkat. Penggunaan *embedded firewall system* dapat meningkatkan efisiensi dalam perancangan jaringan komputer. Perangkat keras yang digunakan untuk membuat *embedded firewall system*, adalah minikomputer *Raspberry Pi* model B. *Raspberry Pi* mempunyai ukuran dan harga yang ekonomis serta kinerja yang baik dengan *processor ARM* dan *512 MB RAM*. Mekanisme *firewall* yang digunakan adalah *packet filtering*, karena dapat melakukan *filter* terhadap paket-paket tertentu yang keluar masuk pada *embedded firewall system*. *Iptables firewall* diimplementasikan pada *Raspberry Pi*, sehingga didapatkan suatu *embedded system* yang bekerja dengan *firewall*. Tujuan dari penelitian ini adalah melakukan analisa *embedded firewall system* dengan simulasi serangan *scanning port, DoS (SYN flood, ICMP flood* dan *UDP flood),* terhadap *resource CPU usage, RAM usage, network,* dan akses *web server*. Berdasarkan hasil pengujian komputer target dalam keadaan normal, hasil ini ditunjukkan nilai rata-rata presentase resource *CPU usage* adalah 4 %, *RAM usage* 47 % dan *network* 0,08 %. Akses *web server* berhasil dilakukan ketika simulasi serangan ICMP flood dengan paket perdetik 1.000 dan 10.000 paket. Simulasi serangan *SYN flood* dan *UDP flood* dengan paket 1.000 per detik.

Kata Kunci : *Embedded System, Firewall, Raspberry Pi, Scanning Port, DoS*

ABSTRACT

Yoga Pradana Pamungkas. 2013. : *Design and Implementation of Network Firewall Security on Embedded System.* Penelitian Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

***Supervisors*: Barlian Henryranu P, ST, MT. dan Eko Setiawan, ST., M.Eng.**

Network security is important. It is an aspect of a computer network defense. Firewall is one of the security systems that prevents unauthorized usage and gives restricted access to a device. Embedded firewall system can increase efficiency in the design of computer networks. Raspberry PI model B is used to build embedded firewall system. Raspberry Pi model B is equipped with ARM processor and 512 MB of RAM, so that it make the good performance and economical design. Firewall packet filtering it choosen as the embedded firewall system. It can filters on certain packages. Iptables firewall is implemented on the Raspberry Pi, to obtain an embedded system with firewalls. The purpose of this research is analyzing embedded firewall system with scanning port, and DoS (SYN flood, ICMP flood, and UDP flood) simulation attack. Based on test result of attack simulation on computer target, CPU resource of usage is 4 %, RAM usage is 47 % and resource of network is 0,08 %. Web server access success when ICMP flood simulation attacks is 1,000 and 10,000 packets per second. It also SYN flood and UDP flood simulation attacks is 1,000 packets per second.

Keywords: *Embedded System, Firewall, Raspberry Pi, Scanning Port, DoS*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

2.1 Latar Belakang

Embedded System merupakan pengembangan teknis baru yang mengikuti perkembangan dunia teknologi informasi (TI) yang memiliki kinerja yang baik, berukuran lebih kecil, daya yang diperlukan lebih rendah dan memiliki fitur yang menonjol untuk aplikasi berorientasi industri. *Embedded system* mempunyai kelebihan yaitu desain ukuran lebih yang lebih kecil, dan tidak menuntut banyak sumber daya, sehingga lebih efisien dalam hal biaya, ukuran, dan dapat digunakan secara *portabel*. [MZ-10] *Embedded system* umumnya dirancang untuk bekerja dengan tujuan yang lebih spesifik sesuai dengan kebutuhan yang dibutuhkan.

Embedded system dapat digunakan untuk membantu tugas dari sebuah komputer atau *hardware* utama, karena *embedded system* mempunyai kelebihan desain ukuran yang lebih kecil, daya rendah dan tujuan yang lebih spesifik. E*mbedded system* dengan tujuan yang lebih spesifik dapat digunakan sebagai *router firewall* yang berfungsi sebagai keamanan jaringan dari sebuah *web server*. Keamanan jaringan yang dibutuhkan harus mencakup sebuah sistem keamanan jaringan *firewall*. Sistem keamanan jaringan *firewall* digunakan sebagai pengamanan jaringan komputer ketika tersambung dengan jaringan komputer lainnya.

Firewall didefinisikan sebagai sebuah komponen atau kumpulan komponen yang membatasi akses antara sebuah jaringan yang diproteksi dari *internet*, atau antara kumpulan-kumpulan jaringan lainnya. *Firewall* dapat berupa *hardware* dan *software*. Dengan menggunakan *firewall* maka kita dapat memproteksi paket-paket data yang dikirim ke jaringan internal [DCZ-00].

Dalam sebuah *survei* oleh *Computer Security Institut* tahun 2009 terhadap macam-macam serangan yang terjadi pada jaringan *internet* dan Komputer, serangan *Denial of Service (Dos/DDos)* menempati peringkat 3 (tiga) dibawah serangan *Malware Function*, dan *Message Phissing* sedangkan pada tahun

2010/2011, serangan *Denial of Service (Dos/DDos)* menempati peringkat 4 (empat) dibawah serangan *Malware Function, Bots on Network, dan Message Phissing*.[CSI-11] Dilihat dari survey tersebut *Dos/DDos* masih menjadi ancaman untuk merusak atau melumpuhkan sebuah *host* atau server. serangan *Dos/DDos* ini mencegah pengguna menggunakan suatu layanan, salah satu jenis serangan ini adalah *SYN flood*, dimana *host* eksternal mencoba untuk membanjiri mesin server dengan mengirimkan permintaan koneksi *TCP*. Sedangkan *scanning port* adalah teknik pengumpulan informasi teknis yang penting pada sebuah komputer *host*, dengan *scanning port* penyerang dapat menemukan informasi *port* yang terbuka pada sebuah komputer *host* berikut sistem yang digunakan dan layanan yang sedang berjalan. *Scanning port* merupakan tahapan awalan serangan yang dilakukan lebih berbahaya nantinya, setelah mengetahui *port* yang terbuka dan informasi yang diketahui dari sebuah sistem.[BBK-10]

Dalam penelitian ini penulis akan merancang dan menganalisis *firewall* pada *embedded syst*em. Implementasi *embedded firewall system* nantinya akan menggunakan minikomputer *Raspberry Pi*. Perangkat keras *Raspberry Pi* akan diinstalasi dengan sistem operasi *linux*, kemudian mengkonfigurasi perangkat tersebut sesuai dengan kebutuhan sistem keamanan jaringan *firewall*. Untuk mendapatkan *embedded firewall system* yang baik penelitian ini akan menganalisa *embedded firewall system* terhadap simulasi serangan *Scanning port* dan *DoS(SYN flood,ICMP flood,UDP flood).*

2.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan diatas maka dapat dirumuskan permasalahan pada skripsi ini yaitu sebagai berikut:

1. Bagaimana perancangan dari sistem keamanan jaringan *firewall* dengan menggunakan *embedded system*.
2. Bagaimana implementasi suatu keamanan jaringan *firewall* berbasiskan *embedded system* dalam mencegah serangan *Scanning port dan Denial of Service/DoS (SYN flood, ICMP flood, UDP flood).*
3. Bagaimana keberhasilan sistem keamanan jaringan *firewall* berbasiskan *embedded system* dalam mencegah serangan *Scanning port dan Denial of*

Service/DoS (SYN flood, ICMP flood, UDP flood) dengan parameter pengujian *CPU Usage, RAM Usage,* dan *Network.*

4. Bagaimana keberhasilan sistem keamanan jaringan *firewall* berbasiskan *embedded system* dalam melindungi dan memberikan akses *web server* terhadap serangan *Denial of Service/DoS (SYN flood, ICMP flood, UDP flood).*

2.3 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal:

1. Sistem keamanan Jaringan perancangan yang dilakukan adalah sistem Keamanan Jaringan *Firewall.*
2. Perancangan firewall berdasarkan arsitektur *firewall* yaitu *packet filtering*
3. Perangkat keras yang digunakan adalah minikomputer *Raspberry Pi.*
4. Sistem Operasi yang digunakan dalam *embedded firewall system* adalah *Rasbian Wheezy.*
5. Pengujian dilakukan dengan simulasi serangan *Scanning port dan Denial of Service(DoS)* pada komputer target *web server* lokal.
6. Serangan *DoS* yang akan diuji adalah *ICMP flood, SYN flood, dan UDP flood.*
7. Parameter pengujian yang digunakan adalah *CPU usage, RAM usage, dan network.*

2.4 Tujuan

Tujuan dilakukannya penelitian ini sesuai dengan rumusan masalah yang telah diberikan yaitu:

1. Membangun suatu sistem keamanan jaringan firewall berbasiskan *embedded system* menggunakan *Raspberry Pi,* agar didapatkan sistem keeamanan jaringan yang cukup handal.
2. Menganalisa tingkat keamanan *firewall* berbasiskan *embedded system* terhadap serangan *Scanning port dan DoS (SYN flood, ICMP flood, dan UDP flood),* serta menganalisa keberhasilan *embedded firewall system* dalam mencegah serangan tersebut.

2.5 Manfaat

Penulisan skripsi ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun manfaat yang diharapkan adalah sebagai berikut:

- Bagi Penulis
  1. Sebagai media untuk pengimplementasian ilmu pengetahuan teknologi pada bidang *Keamanan Jaringan* yang didesain pada *Embedded System*
  2. Mendapatkan pengetahuan dan wawasan terkait metode – metode yang digunakan untuk Perancangan Sistem Keamanan Jaringan *firewall* pada *Embedded System*
- Bagi pembaca/pengguna
  1. Mendapatkan wawasan akan pengimplementasian dari Sistem Keamanan Jaringan *firewall* pada *Embedded System.*
  2. Memberikan sistem keamanan jaringan *firewall* dengan menggunakan *Embedded System* pada pengguna.

2.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Bab I Pendahuluan memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, metodologi pembahasan, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar teori

Bab II Kajian Pustaka dan Dasar Teori menguraikan tentang dasar teori dan referensi yang mendasari pembuatan Sistem Keamanan Jaringan *firewall* pada *Embedded System*

BAB III Metode Penelitian dan Perancangan

Bab III Metode Peneltian dan Perancangan membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur,

perancangan sistem perangkat lunak, implementasi sistem perangkat lunak, pengujian dan analisis, serta penulisan laporan. Selain itu juga membahas tentang perancangan sistem keamanan Jaringan *Embedded System Firewall*

BAB IV Implementasi

Bab IV Implementasi membahas implementasi dari *Embedded System Firewall* terhadap Serangan *Scanning port* dan *Dos(SYN flood, ICMP flood, UDP flood)* yang sesuai dengan perancangan sistem yang telah dibuat.

BAB V Pengujian dan Analisis

Bab V Pengujian dan Analisis memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VI Penutup

Bab VI penutup memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi ini serta saran – saran untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahas penelitian sebelumnya yang menjadi dasar pembuatan penelitian ini. Dasar teori yang dibutuhkan dalam penelitian ini meliputi aspek-aspek Keamanan Jaringan, *Firewall, Iptables,* Serangan *Scanning port dan Denial of Service*(Dos), *Embedded System* dan *Raspberry Pi*.

2.1 Kajian Pustaka

Kajian pustaka yang digunakan sebagai acuan pembuatan penelitian ini adalah penelitian sebelumnnya (Guo Meng dan Li Zhen : 2010) yang berjudul *"Discussion on Internet Security Strategy of Embedded System"*. Penelitian ini membahas tentang tipe baru strategi keamanan jaringaan pada *embedded system*, menganalisa masalah keamanan sistem dan beberapa strategi keamanan jaringan. Pada penelitian ini memfokuskan terutama pada *embedded system* dengan sistem operasi *real-time migrasi*, *system remote*, monitoring perangkat lunak, termasuk desain aplikasi komunikasi embedded *protokol stack TCP/IP*, dan *monitoring host*.

Kajian pustaka lain yang digunakan adalah penelitian sebelumnya (P. Usha Rani dan D.Vara Prasada Rao : 2011) yang berjudul "A Novel Implementation of ARM based Design of Firewall to prevent SYN Flood Attack", dalam penelitian ini membahas Salah satu jenis serangan pada keamanan tentang jaringan adalah *Dos/DDos* salah satunya ini dikenal sebagai *SYN Flood*, di mana *host* eksternal mencoba untuk membanjiri mesin server dengan mengirimkan aliran koneksi *TCP* secara terus menerus. Dalam paper ini mencoba mencegah jenis serangan dengan menggunakan *iptables firewall*. Eksperimen dilakukan untuk mempelajari sejauh mana *firewall* dapat mencegah serangan dan mengamankan terutama serangan *SYN Flood*. Hasil dari analisis menyatakan jika serangan *SYN Flood* terjadi di *ARM board*, maka *web server* tidak dapat diakses, dan setelah aturan *Iptables* diimplementasikan ke dalam *ARM board*, jika

penyerang akan mengirimkan serangan untuk *ARM board*, maka *web server* bisa diakses karena *firewall Iptables* telah ditanamkan pada *ARM board.* [RP-11]

Berdasarkan kajian pustaka pertama menjelaskan tentang tipe baru strategi keamanan jaringan pada embedded system. Kajian pustaka kedua mengimplementasikan *firewall* pada *ARM board* untuk melindungi serangan *SYN flood*. Penulis ingin melakukan penelitian dengan mengimplementasikan sistem keamanan jaringan *firewall packet filtering* pada *embedded system*. Keamanan jaringan *firewall embedded system* berfungsi untuk melindungi sebuah *host* atau *web server*. *Embedded firewall system* dimplementasikan pada minikomputer *Raspberry Pi* dengan *processor ARM 11*. *Embedded firewall system* yang mampu melindungi sebuah *host* atau *web server* dilakukan pengujian dengan simulasi serangan *Scanning Port* dan *DoS(SYN flood, ICMP flood,* dan *UDP flood*)

2.2 Aspek Dasar Keamanan Jaringan

Sistem Keamanan Jaringan Komputer yang terhubung dengan beberapa *host* dengan jaringan luar maupun jaringan internet harus direncanakan dan dipahami dengan baik agar dapat melindungi investasi dan sumber daya di dalam jaringan secara *efektif* dan *efisien*. Untuk membuat sistem keamanan jaringan komputer, harus ditentukan terlebih dahulu tingkat ancaman (*threat)* yang harus diatasi, dan resiko yang harus diambil dan yang akan dihindari.

Dalam *handbook* Budi Rahardjo menjelaskan bahwa keamanan komputer melingkupi 4 aspek dasar yaitu : *privacy*, *integrity*, *authentication* dan *availability*. [QB-12]

A. *Privacy* atau *Confidentiality*

Aspek ini adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses ke suatu *sistem*. *Privacy* menekankan kepada data-data yang sifatnya pribadi, sedangkan *confidentiality* biasanya berhubungan dengan data yang diberikan kepada pihak lain untuk keperluan tertentu.

B. Integrity

Integity menekankan bahwa informasi tidak diperbolehkan diubah tanpa seizin dari pemilik informasi. Adanya *virus, Trojan Horse* atau pemakai lain yang mengubah informasi tanpa izin merupakan contoh yang harus

dihadapi. Contohnya sebuah komunikasi antar surat elektronik atau *e-mail* dapat dimodifikasi isi dari *e-mail* kemudian diteruskan kepada tujuan sbeleumnya. Contoh serangan lain adalah "*Man In The Midle Attack*" dimana seorang menempatkan ditengah pembicaraan dan menyamar sebagai orang lain.

C. Authentication

Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi benar-benar asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, atau *server* yang menjadi tujuan menyatakan benar *server* asli. Dalam hal ini pengguna harus menunjukkan bukti bahwa pengguna adalah pengguna yang sah, dengan menggunakan *password*.

D. Availability

Aspek *availability* atau ketersediaan berhungan dengan kertersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang dapat menghambat akses ke dalam informasi, contoh hambatan adalah serangan yang sering disebut "*Denial of Service Attack*" dimana *server* akan mendapat *request* permintaan palsu yang dilakukan secar terus-menerus sehingga tidak dapat melayani permintaa lain dan bahkan sampai d*own, hang*, dan *crash*.

2.2.1 Aspek-aspek Ancaman Keamanan Jaringan

Serangan terhadap sistem komputer atau sistem informasi dapat dilihat dari jaringan komputer yang berfungsi sebagai penyedia informasi. Menurut W. Stalling (6:1999), ada beberapa kemungkinan serangan, contohnya :

A. Interruption

Perangkat *sistem* menjadi rusak atau tidak tersedia, Serangan ditujukan pada ketersediaan *(availability)* dari *sistem* sehingga informasi dan data yang ada dalam *sistem* komputer dirusak dan dihapus yang berdampak jika informasi dan data dibutuhkan maka data dan informasi tersebut tidak ada lagi. Contoh serangan adalah *"Denial of Sevice Attack"*

B. ***Interception***

Merupakan ancaman terhadap kerahasiaan *(secretry).* Pihak yang tidak berwenang berhasil mengakses aset dan informasi yang telah disimpan, Contoh serangan ini adalah penyadapan *(wiretapping)*

C. Modification

Merupakan ancaman terhadap integritas. Pihak yang tidak berwenang tidak saja berhasil mengakses, tetapi dapat juga aset atau isi. Contoh dari serangan ini antara lain adalah mengubah isi dari website dengan pesan-pesan yang merugikan bagi pemilik website.

D. Fabrication

Pihak yang tidak berhak dapat memalsukan suatu informasi yang ada sehingga pihak yang menerima informasi tersebut menyangka informasi tersebut berasal dari pihak yang dikehendaki oleh penerima informasi. Contoh dari serangan jenis ini adalah memasukkan pesan-pesan palsu seperti *e-mail* palsu kedalam jaringan komputer.[SW-99]

2.3 Firewall

Firewall adalah perangkat atau sebuah program yang berfungsi untuk mengontrol aliran lalu lintas jaringan antar jaringan atau *host* yang menerapkan keamanan yang berbeda. *Firewall* sering dibahas dalam sebuah konteks konektivitas *internet,* tetapi *firewall* juga memiliki penerapan pada konektivitas jaringan lainnya. Dapat diambil sebuah contoh, banyak jaringan dalam perusahaan menggunakan atau mengimplementasikan *firewall* untuk membatasi konektivitas menuju dan dari suatu jaringan internal yang digunakan untuk melayani fungsi yang lebih *spesifik* seperti personalia dan akuntansi. *Firewall* digunakan untuk mengontrol *konektivitas* masing-masing bagian dalam suatu jaringan untuk mecegah akses tidak sah ke dalam *sistem*. Penggunaan *firewall* yang tepat dapat memberikan keamanan jaringan yang baik dalam suatu jaringan maupun sebuah *host*.

Firewall sering dikombinasikan dengan teknologi lainnya terutama dengan *routing,* selain itu banyak teknologi yang dikaitkan dengan *firewall* bahwa *firewall* yang lebih akurat adalah bagian dari teknologi selain *firewall*. Sebagai

contoh, teknologi *Network Address Translation (NAT)* dianggap sebagai teknologi *firewall*, tetapi sebenarnnya adalah teknologi routing. *Firewall* juga mencakup fitur penyaringan konten dalam membuat kebijaksaan organisasi dalam menjaga keamanan jaringan. Beberapa *firewall* termasuk dalam *teknologi Intrusion Prevention System (IPS)*, teknologi ini dapat bereaksi terhadap serangan yang telah terdeteksi sebelumnya untuk mencegah kerusakan pada *sistem* yang dilindungi *firewall*. *Firewall* sering ditempatkan di *perimeter* jaringan, *firewall* tersebut dapat dikatakan memiliki antarmuka *eksternal* dan *internal*. [SH-09]

2.3.1 *Packet Filtering*

Fitur yang paling dasar pada *firewall* adalah paket *filtering*. Paket *filtering* pada perangkat *routing* disediakan untuk fungsi kontrol akses untuk alamat *host* dan sesi komunikasi. Perangkat seperti ini juga dikenal sebagai *stateless inspection firewall*, jadi tidak melacak keadaaan setiap arus lalu lintas yang lewat, ini berarti *firewall* tidak dapat menghubungkan beberapa permintaan dalam satu sesi yang sama. Paket *filtering* adalah inti dari *firewall* paling modern, fungsi kontrol akses *firewall* diatur oleh satu set intruksi sebagai sebuah *rule set*. Kemampuan paket *filtering* yang dibangun ke dalam sebagian besar sistem operasi dan perangkat *routing* yang paling umum digunakan adalah sebuah *router* jaringan yang menggunakan daftar kontrol akses. Dalam bentuk yang paling sederhana, *firewall* dengan paket *filter* beroperasi pada lapisan jaringan. Hal ini menyediakan kontrol akses jaringan berdasarkan beberapa informasi yang berada dalam sebuah paket, seperti :

- Alamat sumber-IP, alamat *host* dari mana paket berasal (ex: 192.168.1.1)
- Alamat Tujuan, alamat *host* dimana paket akan sampai (ex: 192.168.2.1)
- Network atau protocol trans*port* yang digunakan untuk komunikasi antara *host* asal dan tujuan seperti TCP,*UDP*, dan ICMP

Statefull Inspection meningkatkan fungsi *filter* paket dengan melakukan *monitoring traffic* dan memblokir paket yang menyimpang dari *traffic* jaringan normal. Hal ini dilakukan dengan memberikan beban yang lebih besar pada lapisan *transport.* Seperti paket *filtering*, *statefull inspection* melakukan

penyadapan paket pada lapisan jaringan dan melakukan pemeriksaan terhadap paket tersebut apakah diizinkan oleh aturan *firewall* yang telah dibuat, *statefull inspection* melacak setiap koneksi dalam table *state*. Dalam tabel *state* mencakup *IP address host, IP address destination*, nomor *port*, dan *protocol transport*. Cara kerja *statefull inspection* pertama kali adalah memverifikasi bahwa paket data merupakan bagian dari koneksi yang telah ada dalam tabel *state*. [SH-09]

2.3.2 Teknik yang digunakan *Firewall*

Firewall menggunakan teknik-teknik, teknik yang digunakan berpedoman pada *access control list* yang merupakan bagian dari *policy* yang ditentukan *top management* [RP-11]. Berikut merupakan teknik-teknik yang digunakan :

1. *Service Control*

   Teknik ini ditentukan pada tipe-tipe layanan untuk akses kedalam maupun keluar yang melewati *firewall*. *Firewall* akan memeriksa IP address dan nomor *port* yang digunakan pada *protokol TCP* maupun *UDP*.
2. *Direction Control*

   Teknik ini ditekankan pada arah traffic dari permintaan layanan yang akan dikenali dan diijinkan melewati *firewall* baik *traffic input* maupun *output*.
3. *Behaviour Control*

   Teknik ini ditekankan pada perilaku mengenai seberapa banyak suatu layanan digunakan.
4. *User Control*

   Teknik ini ditekankan pada user untuk dapat meminta dan menjalankan suatu layanan dimana *user* tidaak bisa menjalankan layanan tersbut. Teknik ini biasanya digunakan untuk membatasi *user* jaringan lokal untuk mengakses keluar.
5. *Host* Control

   Teknik ini ditekankan paa *host* untuk dapat meminta dan menjalankan suatu layaan dimana terdapat suatu *host* yang boleh dan tdak boleh meminta dan menjalankan lyanan tersebut.

6. *Time Control*

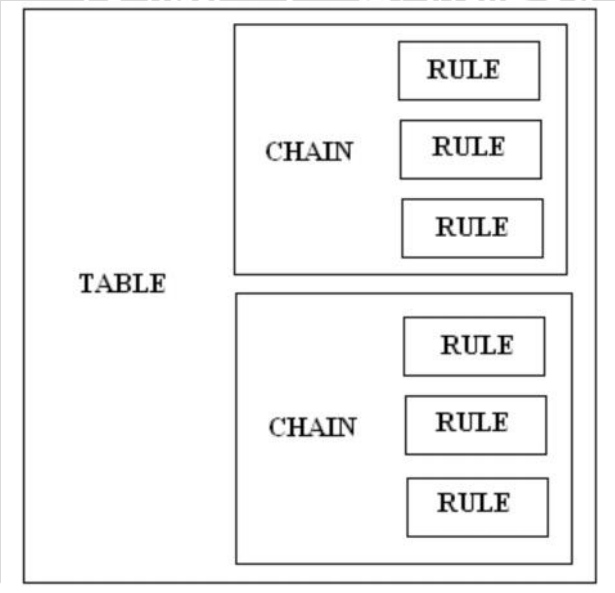
   Teknik ini ditekankan pada waktu dimana suatu layanan dapat diminta dan dijalankan. Teknik ini digunakan untuk mencegah terjadinya akses keluar disaat yang tidak diizinkan misalnya waktu jam kerja.

7. *Transmision Line Control*

   Teknik ini ditekankan pada jalur *transmisi* yang digunakan *sistem*. Jalur *transmisi* ini dapat berupa *private* maupun *public transmision line*.

2.4 Iptables

Iptables adalah *software* atau aplikasi pada sistem operasi *linux* yang memungkinkan administrator jaringan untuk mengkonfigurasi tabel yang disediakan oleh kernel *Linux firewall*. *Iptables* digunakan untuk mengatur, memelihara, dan memeriksa tabel aturan *packet filtering IP address* pada kernel *linux*. Setiap tabel berisi sejumlah *rule* dan *chains*. [JNK-12] Masing-masing *chain* adalah daftar aturan yang dapat sesuai dalam satu set paket *chain*.



Gambar 2.1 Bagan *Chain* dan *Rule Iptables*
Sumber: [RP-11]

Iptables *firewall* memiliki tiga komponen yaitu tabel, *chains*, dan *rules*.[PNG-04]

a. Tabel : Masing-masing *chains* yang telah dibuat akan dikelompokan ke dalam tabel, *iptables* memliki 3 default yaitu *FILTER, NAT,* dan *MANGLE* dengan masing-masing kebijakan yang berbeda-beda.

b. *Rule* : Terdiri dari kondisi untuk memilih paket dan menentukan apa yang harus dilakukan *iptables* dengan paket yang diterima.
c. *Chains* : Aturan yang telah dibuat dikelompokan ke dalam chains. Sebuah chains adalah seperangkat aturan yang digunakan untuk menentukan apa yang harus dilakukan sengan sebuah paket, tiga *chains default* adalah *INPUT, OUTPUT, FORWARD*

Filter Table merupakan tabel *default* untuk setiap aturan, tabel *filter* digunakan untuk mengijinkan dan memblokir lalu lintas dan tiga aturan yang biasa digunakan pada rantai

a. *INPUT* : paket diijinkan untuk melintasi rantai input *host*
b. *OUTPUT* : paket yang dibuat agar *host* dapat mengirimkan paket keluar melewati rantai *output*
c. *FORWARD* : paket diterima melalui *host* diteruskan untuk *host* yang lain melewati rantai *forward*

Tabel *NAT* merupakan rantai yang bertanggung jawab untuk melakukan *Network Address Translation (NAT). NAT* yaitu mengganti *field* asal atau alamat tujuan dari sebuah paket. Pada tabel ini terdapat 2 rantai, yaitu:

a. *Rantai Pre-Routing*: Merubah paket-paket *NAT* dimana alamat tujuan dari paket-paket tersebut terjadi perubahan. Biasanya dikenal dengan destination *NAT* atau *DNAT*.
b. *Rantai Post-Routing*: Merubah paket-paket *NAT* dimana alamat sumber dari paket-paket tersebut terjadi perubahan. Biasanya dikenal dengan source *NAT* atau *SNAT*.

Tabel *Mangle* : tabel ini digunkan untuk paket khusus yang mengalami perubahan. Tabel *mangle* digunakan untuk mengubah beberapa kolom tertentu dalam *header paket IP*. Sehingga dapat digunakan untuk mengubah *Time to Live* atau *TTL*, mengubah jenis layanan, dan menandai paket untuk penyaringan berikutnya. Rantai dari *tabel mangle* :

a. *Prerouting* : Rantai ini mengubah paket sebelum paket diarahkan.
b. *Postrouting* : Rantai ini mengubah paket setelah paket diarahkan.

2.5 Serangan Scanning port

scanning port adalah teknik pengumpulan informasi teknis yang penting pada sebuah komputer *host*, dengan *scanning port* penyerang dapat menemukan informasi *port* yang terbuka pada sebuah komputer *host* berikut sistem yang digunakan dan layanan yang sedang berjalan. *Scanning port* merupakan tahapan awalan serangan yang dilakukan lebih berbahaya nantinya, setelah mengetahui *port* yang terbuka dan informasi yang diketahui dari sebuah sistem. [BBK-10] Beberapa teknik scanning port sebagai berikut :

a. ***Stealth Scan*** : Scanning cara ini dirancang agar peralatan auditing tidak bisa mendeteksi. Scan ini mengirimkan paket kepada host tujuan dengan *stealth flags*. Beberapa *stealth flags* yang digunakan adalah *SYN, FIN* dan *NULL*
b. ***TCP Scanning*** : Sebuah koneksi *TCP* tidak pernah sepenuhnya terjadi selama proses *scanning*. Jika penyerang dapat mengetahui *port remote* menerima koneksi, penyerang dapat melakukan proses serangan selanjutnya. Beberapa scanning *TCP* dapat diidentifikasi dari *IP header dump SYN, FIN, ACK, XMAS,NULL* dan *TCP Fragment*.
c. ***UDP Scanning*** : Scanning ini menemukan port terbuka terkait dengan *protokol UDP*. *Protokol UDP* bersifat *connectionless* sehingga *scanning UDP* tidak sering digunakan oleh penyerang karena dapat dengan mudah untuk diblokir.

2.6 Serangan DoS atau DDoS

Distributed Denial of Service (DDoS) adalah bentuk serangan dimana banyak komputer atau yang disebut *zombie* (komputer *terinfeksi* yang berada dibawah kendali penyerang) digunakan baik secara langsung maupun tidak langsung digunakan untuk membanjiri *server* atau *host* targer korban,dengan mengirimkan paket dengan jumlah besar untuk mencegah pengguna yang sah untuk mengakses *server* tersebut. Ada dua jenis serangan DDoS yang pertama serangan yang menargetkan jaringan atau *bandwith*, serangan menargetkan kepada *choke bandwith internet* yang digunakan oleh *server* korban. Sehingga *user* tidak bisa mengakses *server* tersebut. Jenis serangan kedua adalah serangan yang menargetkan kerentanan dalam sebuah aplikasi terhadap sumber daya *server*

seperti *CPU*, *RAM*, dan *memory buffer*, sehingga membuat *server* tidak akan melayani *request* yang sah dari *user*. [CAK-11]

2.6.1 Tipe Serangan *DoS/DDoS*

Macam-macam tipe serangan *DoS/DDoS* dapat dilihat dari Protokol jaringan yang akan digunakan, seperti *TCP*, *UDP*, *ICMP*.

1. Serangan *SYN Flood*

   Serangan *SYN Flood* menggunakan perantara protokol *TCP*, korban akan dibanjiri dengan koneksi *SYN* secara terus menerus. Proses *sistem* dimulai dengan *client* mengirimkan pesan *SYN* ke *server*, kemudian *server* mengakui adanya pesan *SYN* dengan mengirimkan pesan *SYN - ACK* kepada *client*. Kemudian *client* selesai membuat koneksi dengan membalas dengan pesan *ACK*.

   Proses *SYN Flood* terjadi ketika muncul penyalahgunaan di titik dimana *sistem server* telah mengirimkan *SYN -ACK* kepada *client* tetapi belum menerima pesan *ACK* dari *client*. Menyerang *sistem* dengan mengirim pesan *SYN* kepada *server*. Hal ini tampak seperti koneksi sah tetapi pada kenyataannya *sistem client* tidak akan menerima pesan *SYN - ACK*, sehingga pesan *ACK* yang seharusnya dikirmkan ke *server* tidak akan pernah sampai kepada *server*. Serangan *SYN Flood* akan menyebabkan *host* dibanjiri permintaan secara terus menerus sehingga menjadi sibuk dan dapat terjadi *hang* pada sebuah *host*.

2. Serangan *ICMP Flood*

   ICMP Flood memanfaatkan *protokol Internet Control Message Protocol (ICMP)* sebagai perantara penyerangan. *User* akan mengirimkan paket echo ke remote *host* untuk mengecek apakah *server* masih aktif. Selama serangan *ICMP Flood DoS* agen akan mengirim *ICMP_ECHO_REPLY (PING)* dalam volume yang besar kepada korban atau *server*. Hal tersebut akan mengakibatkan *saturation bandwith koneksi* jaringan korban.

3. Serangan *UDP Flood*

Pada *UDP Flood* penyerang mengirimkan paket *UDP* dalam jumlah besar ke *sistem* korban, sehingga muncul *saturation* jaringan dan berkurangnya *bandwith* yang tersedia untuk melayani permintaan layanan yang sah untuk *sistem* korban. Serangan *UDP Flood* dimungkinkan ketika penyerang mengirimkan paket *UDP* pada *port* acak pada *sistem* korban. Ketika *sistem* korban menerima paket *UDP*, maka *sistem* akan menentukan *aplikasi* yang menunggu di destination *port*. Ketika *sistem* mengetahui bahwa tidak ada aplikasi yang sedang menunggu maka akan menghasilkan *ICMP* paket *"Destination Port Unreachable"*. Jika paket *UDP* yang dikirimkan dengan volume besar akan mengakibatkan *sistem down*.[QB-12]

2.7 Embedded System

Embedded system adalah sistem berbasiskan *mikroprosessor* yang dibangun untuk mengontrol fungsi atau berbagai fungsi, *embedded system* tidak dirancang untuk diprogram dengan cara yang sama dengan penggunaan komputer.[HS-03] *Embedded System* mempunyai fungsi yang sangat *spesifik*, jika dibandingkan dengan sebuah *komputer* atau *PC* yang dirancang untuk tujuan yang sama. *Embedded System* menghasilkan sebuah perangkat yang mempunyai ukuran lebih kecil dari *PC* dengan kemampuan yang sama. Menurut Jazi Eko Istiyanto, Ph.D Salah satu *dosen Elektronika* dan *Instrumentasi* di *FMIPA* UGM menyatakan bahwa *Embedded System* adalah *"A special purpose computer built into a larger device"*. Jika diartikan menjadi "Sebuah komputer khusus yang dibangun untuk sebuah perangkat yang lebih besar". *Embedded System* biasanya merupakan bagian dari perangkat yang lebih besar dengan tujuan untuk meningkatkan kapabilitas yang lebih besar agar didapatkan perangkat yang sesuai dengan kenginan *user* atau pengguna.[IJE-08]

Beberapa karakteristik dari *Embedded System* adalah sebagai berikut:

a. Menjadi bagian dari *sistem* yang lebih besar

Embedded *sistem* di sini menjadi pendukung dari *sistem* yang besar, bisa juga menjadi pengatur dari *sistem-sistem* kecil yang lain pada *sistem*

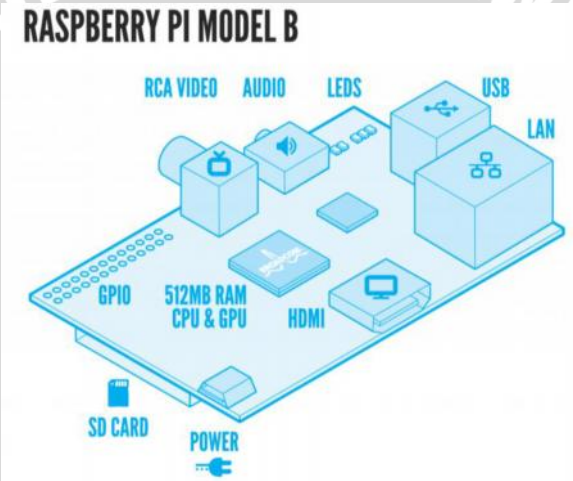
yang besar.

b. *Application-specific*

Perancangan dari *hardware* dan *software* dikhususkan untuk aplikasi yang spesifik sesuai kebutuhan.

2.8 Raspberry Pi

Raspberry Pi adalah sebuah *komputer* berukuran *kartu kredit* yang dapat dihubungkan ke *TV* dan *keyboard*. Ini adalah *PC* kecil yang dapat digunakan untuk banyak hal selayaknya *PC desktop*, seperti *spreadshee*t, pengolah kata dan permainan. *Raspberry Pi* juga memainkan *video* dengan definisi tinggi [RPF-13].



Gambar 2.2 *Raspberry Pi Model B*

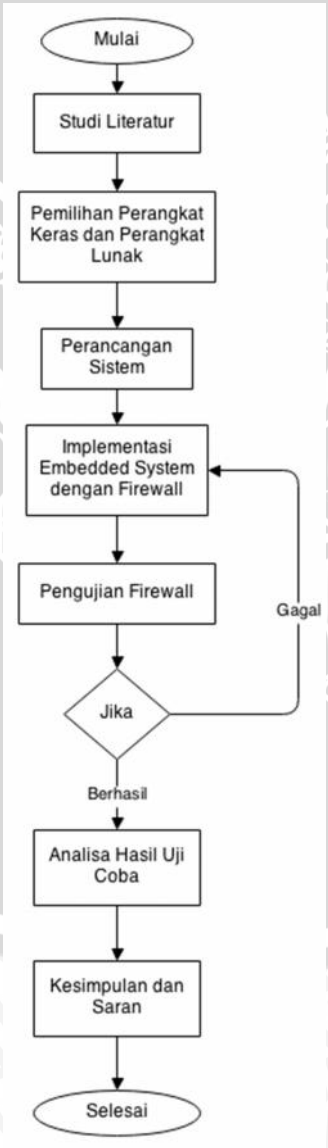
Sumber: [RPF-13]

BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN

Pada bab ini akan dijelaskan tentang metode penelitian dan perancangan sistem yang akan digunakan dalam tahapan-tahapan penelitian dan penyusunan penelitian.

3.1 Metode Penelitian

Metode Penelitian yang akan dilakukan pada penelitian ini secara umum ditunjukkan diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1.1 Studi Literatur

Studi literature digunakan untuk menambah studi pustaka dan pengetahuan yang dilakukan untuk mengerjakan penulisan laporan dan penelitian. Studi literature dilaksanakan dengan cara mengumpulkan dan mempelajari segala macam informasi yang berhubungan dengan keamanan komputer, *embedded system*, *firewall*, pola serangan *Scanning Port, DoS* dan segala hal yang berhubungan dengan model pemrogramannya.

3.1.2 Pemilihan Perangkat Keras dan Perangkat Lunak

Pemilihan perangkat keras dan perangkat lunak bertujuan untuk mendapatkan *embedded system* yang dapat diimplementasikan dengan keamanan jaringan *firewall*, Perangkat keras yang digunakan adalah *Raspberry Pi* model B. Pemilihan ini ditinjau dari ukuran, *CPU, Memory(RAM),* dan *network interface* yang ada pada *Raspberry Pi* model B. Pemilihan sistem operasi juga diperlukan agar lebih mudah dalam mengimplementasikan keamanan jaringan *firewall*. Sistem Operasi yang akan digunakan adalah *Raspbian Whezzy*. Sistem operasi *Raspbian wheezy* merupakan sistem operasi utama yang digunakan untuk minikomputer *Raspberry Pi*. Sistem operasi *raspbian whezzy* support dengan semua hardware yang ada di *Raspberry Pi* contohnya *port interface* eth0. Berikut adalah spesifikasi dari *Raspberry Pi* model B ditunjukkan pada tabel 3.1.

Tabel 3.1 Spesifikasi Raspberry Pi

Chip	Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
CPU	700 MHz ARM1176JZF-S core (ARM6 family)
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
Memory (SDRAM)	512 MB (shared with GPU)
USB 2.0 *ports*	2 (via integrated USB hub)
Video outputs	• Composite RCA (PAL & NTSC), HDMI (rev 1.3 & 1.4), raw LCD Panels via DSI • 14 HDMI resolutions from 640×350 to 1920×1200 plus various PAL and NTSC standards.

Audio outputs	3.5 mm jack, HDMI
Onboard storage	SD / MMC / SDIO card slot
Onboard network:	10/100 Ethernet (RJ45)
Low-level peripherals	8 × GPIO, UART, I²C bus, SPI bus with two chip selects, +3.3 V, +5 V,
Power ratings	700 mA (3.5 W)
Power source	5 volt via MicroUSB or GPIO header
Size	85.60 × 53.98 mm (3.370 × 2.125 in
Weight	45 gram (1.6 oz)

3.1.3 Perancangan Sistem

Pada tahap ini dilaksanakan perancangan sistem yang akan dibuat berdasarkan hasil study literature tentang keamanan jaringan *firewall* pada *embedded system*. Perancangan sistem memberikan kemudahan dalam proses pembuatan sistem yang akan dibuat dalam penelitian. Perancangan sistem mempunyai tujuan sebagai bahan acuan dalam tahapan implementasi sistem.

3.1.4 Implementasi dan Konfigurasi

Dalam tahap ini, dilakukan implementasi berdasarkan studi literatur dan perancangan sistem. Setelah spesifikasi sistem telah ditentukan maka implementasi yang pertama kali dilakukan adalah melakukan instalasi sistem operasi *Raspbian Whezzy* pada perangkat keras *Raspberry Pi*. Langkah selanjutnya adalah melakukan konfigurasi pada sistem operasi *Rasphian Wheezy* sehingga didapatkan keamanan jaringan dengan *embedded system*. *Firewall* menggunakan *iptables* dipilih sebagai kemanan jaringan yang akan ditanam pada *embedded system*. Kemudian melakukan konfigurasi *iptables* terhadap beberapa serangan yaitu *Scanning port, SYN flood, UDP flood,* dan *ICMP flood*. Untuk selanjutnya melakukan konfigurasi *interface wlan0* menggunakan *wifi nano adapter*. Setelah semuai tahapan implementasi selesai maka akan didapat sebuah *embedded system* yang memiliki keamanan jaringan berbasiskan *firewal.*

3.1.5 Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba *firewall* untuk mencoba sistem keamanan jaringan terhadap beberapa serangan, yaitu *Scanning port* , serangan *DoS* berupa *SYN flood, UDP flood,* dan *ICMP flood.* Pemilihan simulasi serangan untuk tahapan pengujian berdasarkan dari *survei* oleh *Computer Security Institut* terhadap macam-macam serangan yang terjadi pada jaringan *internet.* Hasil dari survei menjelaskan bahwa serangan *DDoS/DoS* pada tahun 2009 menempati peringkat ke-3 dan tahun 2011 pada peringkat ke-4.[CSI-11] Untuk pemilihan macam serangan *DoS* berupa *SYN flood, UDP flood* dan *ICMP flood* karena protokol komunikasi yang sering digunakan oleh komputer adalah protokol *TCP,UDP,* dan *ICMP*.

Pengujian menggunakan metode *penetrasi testing Grey-box hacking* yaitu metode yang dikenal dengan *internal testing* yang dilakukan dalam suatu jaringan perusahaan. Metode ini memiliki asumsi bahwa *peretas* mengetahui informasi sistem yang digunakan tetapi dalam tahap yang terbatas.

Simulasi uji coba :

Pengujian dengan 1 *web server*, 1 *attacker*, 1 *user* dan *embedded firewall system*

Dalam tahap skenario terdapat kriteria pengujian yaitu :

a. Estimasi jumlah paket perdetik dengan menggunakan *tool Hping*
b. Melihat perubahan pada komputer client terhadap *CPU usage, RAM usage*, dan *Network*.
c. Akses user terhadap *web server* lokal.

3.1.6 Analisa Hasil Uji Coba

Pada tahap ini dilakukan analisa uji coba terhadap sistem yang telah dibuat, jika terjadi kekurangan dan kesalahan terhadap *embedded system* dengan *firewall* yang telah selesai dibuat. Diharapkan *embedded system* dengan *firewall* tersebut mengalami uji coba sehingga menghasilkan *output* sesuai dengan rumusan dan batasan masalah.

3.1.7 Kesimpulan dan Saran

Kesimpulan didapatkan setelah melakukan perancangan, implementasi, pengujian dan analisis kepada sistem nantinya. Kesimpulan disusun berdasarkan hasil pengujian dan analisa pengujian yang telah dibuat. Isi dari kesimpulan diharapkan dapat menjadi acuan pada penelitian lain untuk mengembangkan *embedded system* dengan *firewall*. Selain itu, pada akhir penulisan terdapat saran yang bertujuan untuk penyempurnaan penelitian ini.

3.2 Perancangan

Perancangan berfungsi untuk menentukan kebutuhan perangkat lunak dan perangkat keras yang digunakan untuk melakukan perancangan sistem dan tahapan pengujian sistem. Kebutuhan perangkat keras dan perangkat lunak yang digunakan untuk memenuhi *minimum system requirement* dari masing-masing kebutuhan, seperti kebutuhan sebagai komputer penyerang, komputer target, dan komputer host.

3.2.1 Kebutuhan Perangkat Keras dan Lunak

Kebutuhan perangkat keras dan lunak digunakan untuk menyediakan kebutuhan sistem keamanan jaringan *firewall* pada *embedded system*, serta untuk memenuhi tahapan pengujian sistem. Dalam tahapan ini peneliti akan menentukan seluruh kebutuhan untuk perancangan dari perangkat lunak dan perangkat keras berdasarkan *minimum system requirement*. Berikut kebutuhan seluruh perangkat lunak dan perangkat keras yang dibutuhkan :

A. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan pada penelitian disesuaikan dengan kebutuhan dari komputer penyerang, komputer target, komputer host dan peralatan jaringan untuk memenuhi perancangan sistem, kebutuhan perangkat keras dapat dilihat pada tabel 3.2.

Tabel 3.2 Kebutuhan Perangkat Keras

Komputer Penyerang	Komputer Target	Komputer User	Embedded Firewall System	Peralatan Jaringan Komputer
Processor Intel Coleron CPU	Processor Intel Core i5 CPU	Ipad 4	Raspberry Pi Processor ARM	Kabel UTP Cat 5
1 GB RAM	2 GB RAM	1 GB RAM	512 MB RAM	Konektor RJ-45
120 GB hardisk Drive	465 GB hardisk Drive	16 GB hardisk	16 GB SD Card	Nano Wifi Adapter

B. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak digunakan untuk menunjang perangkat keras yang sudah dispesifikasikan sebelumnya. Kebutuhan perangkat lunak pada penelitian ditunjukkan pada Tabel 3.3.

Tabel 3.3 Kebutuhan Perangkat Lunak

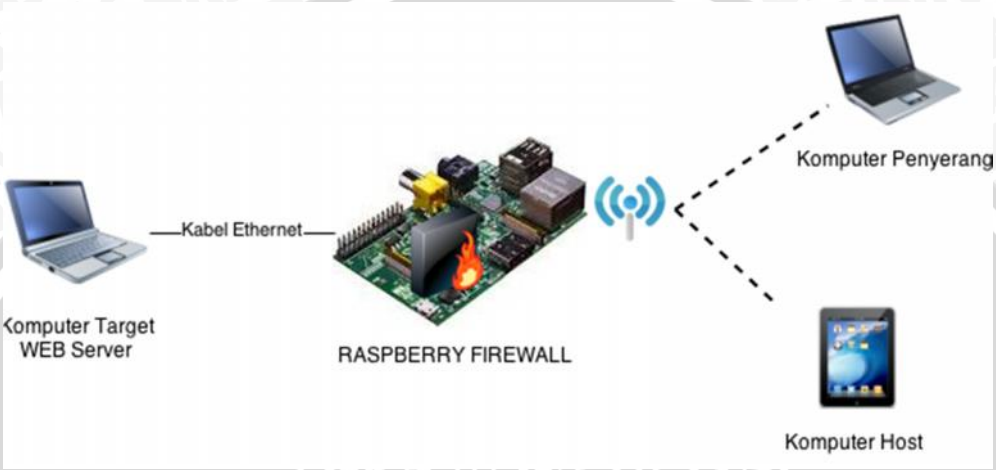
Komputer Target	Komputer Penyerang	Embedded System Firewall
Sistem Operasi Windows 7	Sistem Operasi Backtrack 5 R3	Sistem Operasi Raspbian Wheezy
Wireshark	Hping3	Iptables
Web Server Apache	-	*Host*apd, Dnsmasq

3.2.2 Perancangan Perangkat Keras dan Lunak

Perancangan sistem keamanan jaringan *embedded system* dengan *firewall* dilakukan setelah melakukan spesifikasi tentang kebutuhan perangkat keras dan perangkat lunak yang dibutuhkan. Perancangan keamanan jaringan *firewall* berdasarkan teori tentang *firewall* dan cara mekanisme kerja *firewall*. Sistem *Firewall* yang akan dibuat menggunakan mekanisme *firewall packet filtering* dengan mengimplementasikan *Iptables* pada sistem operasi *Raspbian Whezzy*.

3.2.2.1 Topologi Jaringan

Penelitian dilakukan pada jaringan *LAN* yang terdiri dari *2 buah* komputer sebagai komputer penyerang dan komputer target,ditambah 1 user untuk akses web server serta *embedded firewall system* yang berfungsi sebagai *router*. Topologi jaringan secara umum dapat digambarkan pada Gambar 3.2.



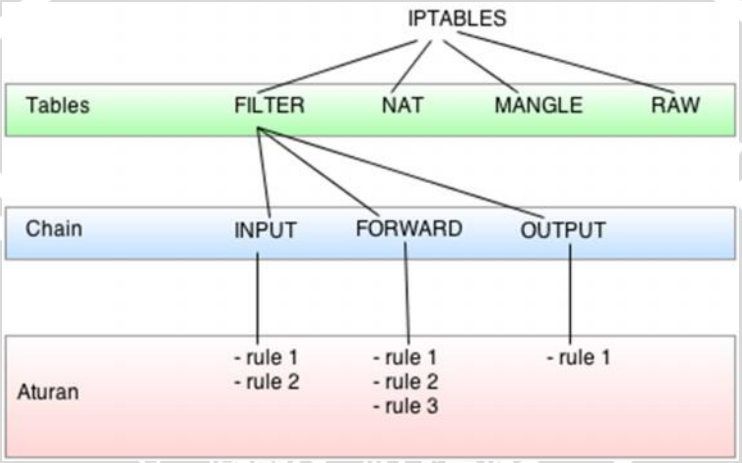
Gambar 3.2 Topologi Jaringan

Keterangan Gambar :

- **Raspberry Pi** : Sebagai *Embedded Firewall System*
- **Komputer Target dan Web Server** : Komputer yang melakukan koneksi melalui *interface eth0 embedded system* yang akan dijadikan target serangan
- **Komputer Penyerang** : Komputer yang melakukan koneksi melalui *interface wlan0 embedded system* yang akan melakukan serangan.
- **Komputer *Host*/user** : Komputer yang melakukan koneksi melalui *interface wlan0 embedded system* untuk akses pada *web server*

3.2.2.2 Arsitektur Perangkat Lunak

Perangkat lunak yang dijadikan sebagai *firewall* adalah *Iptables*. Pemilihan perangkat lunak *iptables* berdasarkan buku (Michael Rush:2007) dengan judul "*Linux Firewalls*". Michael Rush menjelaskan bahwa *iptables* pada sistem operasi *Linux* merupakan perangkat lunak *firewall* utama. *Iptables firewall* menyediakan perangkat lunak yang efektif dalam mengontrol sistem lalu lintas jaringa. Bagan arsitektur dari *Iptables* ditunjukkan pada Gambar 3.3.



Gambar 3.3 Arsitektur Iptables

Pada gambar 3.3 Arsitektur *Iptables* menunjukkan bahwa 3 bagian utama dari arsitektur iptables adalah tables, *chain*, dan aturan.

Kebijakan dan aturan yang akan diimplementasikan pada *firewall* dilakukan dengan menentukan *ruleset* dan *chain* pada konfigurasi *iptables* untuk *packet filtering*. Kebijakan yang dilakukan pada *packet filterin*g adalah terhadap serangan *Scanning port, SYN flood, ICMP flood,* dan *UDP flood*. *Firewall* yang akan dibangun dijelaskan sebagai berikut :

a. *Route the Packet* : *firewall* harus mempunyai kemampuan seperti *router* yaitu dengan *memforward* setiap paket data yang melewati *firewall*.
b. *Packet Filtering* : *firewall* mempunyai kemampuan dalam memfilter paket data yang lewat berdasarkan *IP source* atau *IP destination* dan juga berdasarkan *port number*

c. *Network Address Translation* : kemampuan *firewall* adalah untuk membungkus satu atau banyak *IP address* milik jaringan internal dan menggantikannya dengan *IP address* milik *firewall*.

Firewall yang dibangun berdasarkan *IPtables* sebagai berikut*:*

a. Ruleset *Iptables*

   Ruleset berisikan tentang kebutuhan-kebutuhan terhadap keamanan jaringan dengan *iptables* yang akan dikonfigurasi pada *embedded system*

b. Penyimpanan *Iptables*

   Iptables yang dibuat akan disimpan di sistem operasi pada direktori /etc/default/ dengan file bernama iptables

c. Log *Iptables*

   Log *iptables* digunakan untuk mencatat proses yang terjadi pada *iptables*

BAB IV
IMPLEMENTASI

Pada bab implementasi dibahas mengenai langkah-langkah dalam implementasi dan konfigurasi keamanan jaringan *embedded firewall system*. Langkah-langkah dalam implementasi mengacu pada tahapan perancangan yang terdiri dari sebuah komputer sebagai target serangan dan sebuah komputer sebagai penyerang yang berfungsi untuk membuat simulasi keamanan jaringan serta *embedded system* yang digunakan untuk perangkat keras *firewall*. Implementasi meliputi lingkungan perangkat keras dan implementasi perangkat lunak.

4.1 Implementasi Perangkat Keras dan Perangkat Lunak

Perangkat keras dan perangkat lunak yang digunakan untuk membuat sistem keamanan jaringan *embedded firewall system* dapat dilihat pada tabel 4.1.

Tabel 4.1 Kebutuhan Perangkat Keras dan Perangkat Lunak

Perangkat Keras	*Raspberry Pi*
Storage	*SDHC Card 16 Gb*
Interface network	*Ethernet LAN, Nano Wireless Adapter*
Sistem operasi	*Linux Raspbian wheezy*
Perangkat Lunak	*Iptables, Hostapd*

Perangkat keras yang digunakan adalah *Raspberry P*i model B dengan *memory RAM* sebesar 512 MB dan *storage* sebesar 16GB, tampilan perangkat keras *raspberry Pi* ditunjukkan pada gambar 4.1. Perangkat *interface network* yang digunakan pada *embedded firewall system* adalah *usb nano adapter*, tampilan perangkat *interface network* ditunjukkan pada gambar 4.2.

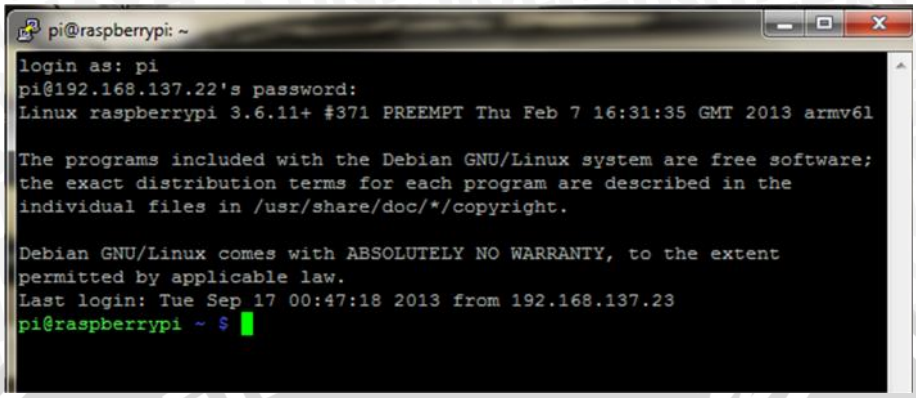
Gambar 4.1 Perangkat keras *raspberry Pi* model B

Gambar 4.2 Perangkat keras wifi adapter

4.2 Instalasi dan Konfigurasi *Embedded Firewall System*

Perangkat keras yang digunakan sebagai *embedded firewall system* adalah *Raspberry Pi*. Instalasi dan konfigurasi dilakukan dengan menginstalasi sistem operasi *raspbian wheezy* pada *Raspberry Pi*. Sistem operasi *Raspbian wheezy* merupakan sistem operasi utama yang digunakan untuk minikomputer Raspberry Pi. Sistem operasi raspbian whezzy support dengan semua hardware yang ada di Raspberry Pi contohnya port interface eth0. Raspbian wheezy berbasiskan linux, sehingga dapat dikonfigurasi untuk membuat sebuah sistem keamanan jaringan firewall. Langkah konfigurasi *Raspberry Pi* bertujuan agar perangkat keras dapat digunakan sebagai *embedded system firewall*. Sehingga dihasilkan sistem keamanan jaringan *firewall* yang diinginkan. Untuk memudahkan membuat *embedded system firewall* langkah-langkah konfigurasi dapat dilakukan

menggunakan remote *SSH putty*, tampilan remote *SSH Raspberry* menggunakan *putty* ditunjukkan pada gambar 4.3.



Gambar 4.3 Remote SSH menggunakan *putty*

4.3 Implementasi *Ruleset Firewall Iptables*

Ruleset firewall digunakan sebagai mekanisme untuk menerapkan kontrol dalam keamanan jaringan *firewall*. *Firewall* yang digunakan adalah *firewall Packet Filtering*. Dasar penyusunan *ruleset* yang digunakan untuk tahapan implementasi berdasarkan kebutuhan dari *embedded firewall system*. *Embedded firewall system* berfungsi sebagai *router firewall* yang memberikan akses *web server* dan melakukan *filtering* ketika serangan *Scanning Port, SYN flood, ICMP flood,* dan *UDP flood*. Berikut ruleset yang digunakan untuk tahapan impementasi:

a. Menuliskan *ruleset* dengan memberikan aturan memperbolehkan *(accept),* menolak *(deny),* dan membuang *(drop)* paket dengan tidak memberikan tanggapan ke pengirim paket.
b. Menuliskan *ruleset* NAT (Network Address Translation).
c. Memberikan *ruleset* untuk akses FORWARD dan OUTPUT pada port 80 yang digunakan untuk akses *web server*.
d. Memberikan *ruleset* untuk akses *SSH* hanya untuk IP 192.18.1.25.
e. Memberikan *ruleset* untuk komunikasi antar *host* dengan *protokol ICMP*(akses *PING*).
f. Memberikan *ruleset* terhadap serangan *Scanning port* dengan filter INPUT dan FORWARD untuk *destination port* 1-65535.

g. Memberikan *ruleset* terhadap serangan *SYN flood* dengan filter INPUT, OUTPUT dan OUTPUT untuk paket *SYN* dengan *hitcount* 20
h. Memberikan *ruleset* terhadap serangan *ICMP flood* dengan *filter* INPUT, OUTPUT dan OUTPUT untuk paket ICMP dengan *hitcount* 20
i. Memberikan *ruleset* terhadap serangan *UDP flood* dengan *filter* INPUT, OUTPUT dan OUTPUT untuk paket *UDP* dengan *hitcount* 20.

4.3.1 Iptables serangan *Scanning port*

Konfigurasi *firewall* menggunakan *iptables* pada *embedded system* terhadap serangan *Scanning port* dapat dilihat pada gambar 4.4

```
1  $iptables -A INPUT -p tcp -m tcp --dport 1:65535 -m recent
   --name portscan --set -j LOG --log-prefix "portscan : "
2  $iptables -A INPUT -p tcp -m tcp --dport 1:65535 -m recent
   --name portscan --set -j DROP
3  $iptables -A FORWARD -p tcp -m tcp --syn --dport 1:65535 -
   m recent --name portscan --set -j LOG --log-prefix
   "portscan : "
4  $iptables -A FORWARD -p tcp -m tcp --syn --dport 1:65535 -
   m recent --name portscan --set -j DROP
```

Gambar 4.4 Konfigurasi *Iptables Scanning Port*

Penjelasan dari konfigurasi *iptables scanning port*:

1. Membuat *chain rule* INPUT dengan *protokol tcp* dengan *destination port* antara *port* 1 sampai *port* 65535, menginisialisasi dengan set name "*port*scan", dimasukan kedalam LOG dengan inisialisasi *log-prefix portscan.*
2. Membuat *chain rule* INPUT dengan *protokol tcp* dengan *destination port* 1 sampai *port* 65535 dengan set name "*port*scan" kemudian set dengan DROP.
3. Membuat *chain rule* FORWARD dengan *protokol tcp* dengan *destination port* antara *port* 1 sampai *port* 65535 menginisialisasi dengan set name "*port*scan", dimasukan kedalam LOG dengan inisialisasi *log-prefix portscan*.

4. Membuat *chain rule* FORWARD dengan *protokol tcp* dengan *destination port* 1 sampai *port* 65535 dengan set name "*port*scan" kemudian set dengan DROP.

4.3.2. *Iptables* serangan *SYN Flood*

Konfigurasi *firewall* menggunakan *iptables* pada *embedded system* terhadap serangan *SYN Flood* dapat dilihat pada gambar 4.5.

```
1  $iptables -N SYN_FLOOD
2  $iptables -A INPUT  -i wlan0 -p tcp --syn -j SYN_FLOOD
3  $iptables –A FORWARD –i wlan0 –o eth0 –p tcp –-syn –j
   SYN_FLOOD
4  $iptables –A OUTPUT –o eth0 –p tcp –-syn –j SYN_FLOOD
5  $iptables -A SYN_FLOOD -m limit --limit 30/s --limit-burst
   60 -m comment --comment "rata-rata TCP SYN" -j RETURN
6  $iptables -A SYN_FLOOD -m recent –name SYNFLOOD –rcheck –
   second 1 –hitcount 20 -j LOG --log prefix " syn flood"
7  iptables -A INPUT -m recent --name SYNFLOOD --rcheck --
   second 1 --hitcount 20 -j DROP
8  iptables -A FORWARD -m recent --name SYNFLOOD --rcheck --
   second 1 --hitcount 20 -j DROP
```

Gambar 4.5 Konfigurasi *Iptables SYN Flood*

Penjelasan dari konfigurasi *iptables Syn flood* :

1. Membuat aturan *chain* baru dengan nama SYN_FLOOD.
2. Membuat aturan *rule* INPUT dari *interface wlan0* dengan *protokol tcp syn* kedalam *chain* SYN_FLOOD.
3. Membuat aturan *rule* FORWARD dari *interface wlan0* menuju *interface eth0* dengan *protokol tcp syn* kedalam chain SYN_FLOOD.
4. Membuat aturan *rule* OUTPUT dari *interface eth0* dengan *protokol tcp syn* kedalam chain SYN_FLOOD.
5. Pada *chain* SYN_FLOOD membatasi paket *tcp syn* yang masuk sebesar 30 *second* dengan inisialisasi *comment* "rata-rata TCP SYN" untuk di RETURN.

6. Pada *chain* SYN_FLOOD inisialisasi name dengan “SYNFLOOD” agar dilakukan *rcheck*, dengan waktu 1 *second* untuk *hitcount* 20 kemudian masukan dalam LOG “syn flood”.
7. Membuat aturan pada *rule* INPUT dengan name “SYNFLOOD” untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.
8. Membuat aturan pada *rule* FORWARD dengan name “SYNFLOOD” untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.

4.3.3. *Iptables* serangan *UDP Flood*

Konfigurasi *firewall* menggunakan *iptables* pada *embedded system* terhadap serangan *SYN Flood* dapat dilihat pada gambar 4.6

```
1  iptables -N UDP_FLOOD
2  iptables -A INPUT -i wlan0 -p udp -j UDP_FLOOD
3  iptables -A FORWARD -i wlan0 -o eth0 -p udp -j UDP_FLOOD
4  iptables -A OUTPUT -o eth0 -p udp -j UDP_FLOOD
5  iptables -A UDP_FLOOD -p udp -m limit --limit 3/s --
   limit-burst 8 -m comment --comment "Limit UDP rate" -j
   RETURN
6  iptables -A UDP_FLOOD -m recent --name UDPFLOOD --set -j
   LOG --log-prefix "UDP FLOOD"
7  iptables -A INPUT -m recent --name UDPFLOOD --rcheck --
   second 1 --hitcount 20 -j DROP
8  iptables -A FORWARD -m recent --name UDPFLOOD --rcheck --
   seconds 1 --hitcount 20 -j DROP
9  iptables -A OUTPUT -m recent --name UDPFLOOD --rcheck --
   seconds 1 --hitcount 20 -j DROP
```

Gambar 4.6 konfigurasi *Iptables UDP Flood*

Penjelasan dari konfigurasi *iptables UDP flood*:

1. Membuat aturan *chain* baru dengan nama UDP_FLOOD
2. Membuat aturan *rule* INPUT dari *interface wlan0* dengan protokol *tcp syn* kedalam *chain* UDP_FLOOD

3. Membuat aturan *rule* FORWARD dari *interface wlan0* menuju *interface eth0* dengan protokol *tcp syn* kedalam *chain* UDP_FLOOD
4. Membuat aturan *rule* OUTPUT dari *interface eth0* dengan protokol *tcp syn* kedalam *chain* UDP_FLOOD
5. Pada *chain* UDP_FLOOD dengan inisialisasi *comment* " UDP rata-rata" untuk di RETURN
6. Pada *chain* SYN_FLOOD inisialisasi name dengan "UDPFLOOD" agar dilakukan *rcheck*, dengan waktu 1 *second* untuk *hitcount* 20 kemudian masukan dalam LOG "udp flood".
7. Membuat aturan pada *rule* INPUT dengan name "UDPFLOOD" untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.
8. Membuat aturan pada *rule* FORWARD dengan name "UDPFLOOD" untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.
9. Membuat aturan pada *rule* OUTPUT dengan name "UDPFLOOD" untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP

4.3.2 *Iptables* Serangan *ICMP Flood*

Pada serangan ICMP Flood memanfaatkan protokol ICMP untuk melakukan serangan. Konfigurasi *firewall* terhadap serangan *ICMP Flood* dapat dilihat pada gambar 4.7

```
1  $iptables -N ICMP_FLOOD
2  $iptables -A INPUT –i wlan0 -p icmp -j ICMP_FLOOD
3  $iptables –A FORWARD –i wlan0 –o eth0 –p icmp –j
   ICMP_FLOOD
4  $iptables –A OUTPUT –o eth0 –p icmp –j ICMP_FLOOD
5  $iptables -A ICMP_FLOOD -m limit --limit 1/s --limit-
   burst 2 -m comment --comment "Limit ICMP rate" -j
   RETURN
6  iptables -A ICMP_FLOOD -m recent --name ICMPFLOOD --
```

```
    set -j LOG --log-prefix "ICMPFLOOD "
7   iptables -A INPUT -m recent --name ICMPFLOOD --rcheck
    --second 1 --hitcount 20 -j DROP
8   iptables -A FORWARD -m recent --name ICMPFLOOD --
    rcheck --seconds 1 --hitcount 20 -j DROP
```

Gambar 4.7 Konfigurasi *iptables icmp flood*

Penjelasan dari listing code iptables ICMP flood :

1. Membuat aturan *chain* baru dengan nama ICMP_FLOOD.
2. Membuat aturan *rule* INPUT dari *interface wlan0* dengan *protokol tcp syn* kedalam *chain* ICMP_FLOOD.
3. Membuat aturan *rule* FORWARD dari *interface wlan0* menuju *interface eth0* dengan *protokol tcp syn* kedalam *chain* ICMP_FLOOD
4. Membuat aturan *rule* OUTPUT dari *interface eth0* dengan *protokol tcp syn* kedalam *chain* ICMP_FLOOD.
5. Pada *chain* ICMP_FLOOD membatasi paket icmp yang masuk sebesar 3 *second* dengan inisialisasi comment “rata-rata ICMP” untuk di RETURN.
6. Pada chain ICMP_FLOOD inisialisasi name dengan “ICMPFLOOD” agar dilakukan *rcheck*, dengan waktu 1 *second* untuk *hitcount* 20 kemudian masukan dalam LOG “icmp flood”.
7. Membuat aturan pada *rule* INPUT dengan name “ICMPFLOOD” untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.
8. Membuat aturan pada *rule* FORWARD dengan name “ICMPFLOOD” untuk dilakukan *rcheck* dengan waktu 1 *second* untuk *hitcount* 20 untuk eksekusi DROP.

4.3.3 Implementasi wlan0 pada Raspberry Pi

Interface wlan0 pada *embedded firewall system* berfungsi membuat *raspberry pi* menjadi *router* untuk memberi koneksi antara *host/user* terhadap *web server* sehingga terjadi jaringan *WLAN* antara komputer *host*, komputer target dan komputer penyerang.

Konfigurasi *wlan0* pada *embedded firewall system* sebagai berikut :

1. Download File

```
wget https://github.com/jenssegers/RTL8188-hostapd/archive/v1.1.tar.gz
```

2. Konfigurasi file *host*apd.conf

```
# Basic configuration

interface=wlan0
ssid=raspberry
channel=10
#bridge=br0

# WPA and WPA2 configuration

macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=1234554321
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

# Hardware configuration

driver=rtl871xdrv
ieee80211n=1
hw_mode=g
device_name=RTL8188CUS
manufacturer=Realtek

```

```
27 beacon_int=100
28 auth_algs=3
29 wmm_enabled=1
```

Gambar 4.8 Konfigurasi file *host*apd.conf

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini akan dilakukan pengujian dan analisis terhadap hasil implementasi yang telah dilakukan pada tahapan penelitian. Pengujian dan analisa mempunyai tujuan untuk melihat kemampuan sistem keamanan jaringan *embedded firewall system* yang telah diimplementasikan.

5.1 Pengujian

Pengujian dilakukan menggunakan metode *penetrasi testing Grey-box hacking* yaitu metode yang dikenal dengan *internal testing* yang dilakukan dalam suatu jaringan. Metode ini memiliki asumsi bahwa peretas mengetahui informasi sistem yang digunakan tetapi dalam tahap yang terbatas. Sehingga dalam tahap pengujian diasumsikan bahwa peretas mengetahui *IP address* dan nomer *port* terbuka dari masing-masing *host*.

Keterangan komputer yang digunakan dalam pengujian :

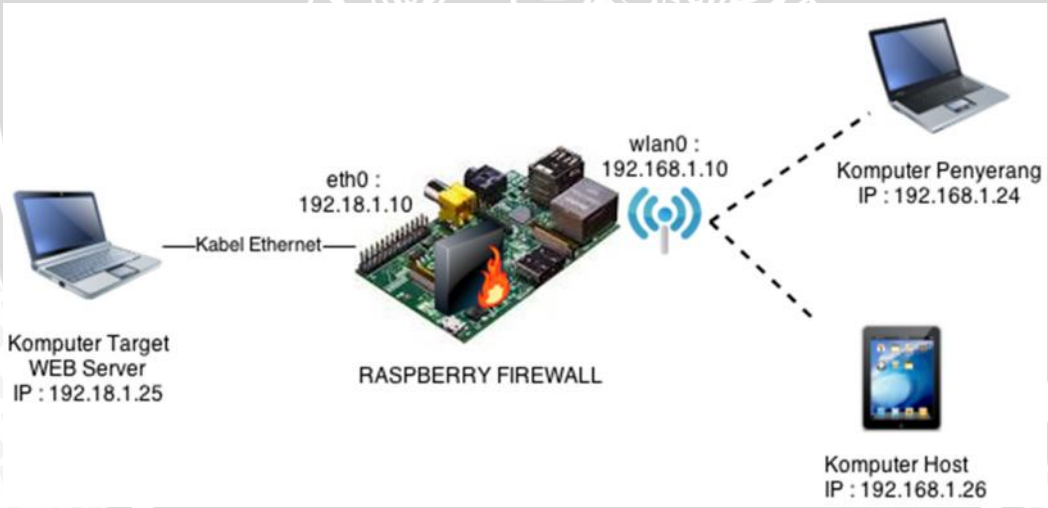
1. Komputer Attacker
   a. Intel pentium celeron
   b. Sistem operasi Windows 7
   c. 1 Gb RAM
   d. 120 Gb Hardisk
   e. IP address : 192.168.1.24
2. Komputer Target
   f. Intel core i5
   g. Sistem operasi Backtrack 5 R3
   h. 3 Gb RAM
   i. 60 Gb Hardisk
   j. IP address : 192.18.1.25

5.1.1 Skenario pengujian sistem

Pada skenario pengujian sistem dibuatkan beberapa keadaan dan kondisi dengan menggunakan *embedded firewall system.* Skenario pengujian dibuat berbeda untuk melakukan pengujian sistem keamanan *firewall iptables* yang telah diimplementasikan. Berikut kondisi skenario pengujian sistem yang digunakan :

1. 2 buah komputer sebagai penyerang dan target, 1 host sebagai akses *web server* dengan *embedded firewall system* dalam keadaan *firewall iptables* tidak aktif
2. 2 buah komputer sebagai penyerang dan target, 1 host sebagai akses *web server* dengan *embedded firewall system* dalam keadaan *firewall iptables* aktif.

Topologi jaringan yang digunakan dalam pengujian simulasi serangan dapat dilihat pada gambar 5.1.



Gambar 5.1 Topologi Jaringan

Untuk melakukan simulasi pengujian serangan digunakan *tool* yang ada pada *linux Backtrack 5* yaitu *nmap* dan *hping3*, *nmap* digunakan untuk melakukan *scanning port* yang bertujuan untuk mengetahui informasi dari komputer target berupa alamat *port* yang terbuka dan informasi sistem yang digunakan. Sedangkan tool *hping3* berfungsi melakukan serangan *DoS* berupa *SYN flood, UDP flood, dan ICMP flood* yang bertujuan untuk membanjiri komputer target

sehingga dapat menghabiskan *resource* yang dimiliki oleh komputer target. Dalam melakukan pengujian sistem digunakan parameter-paramater yang ditetapkan berupa jumlah paket perdetik yang dikirimkan oleh komputer penyerang, berikut estimasi jumlah paket perdetik yang digunakan dalam pengujian :

Tabel 5.1 Tabel estimasi paket perdetik saat pengujian

Estimasi	Penjelasan
u10	10.000 paket perdetik
u100	1.000 paket perdetik

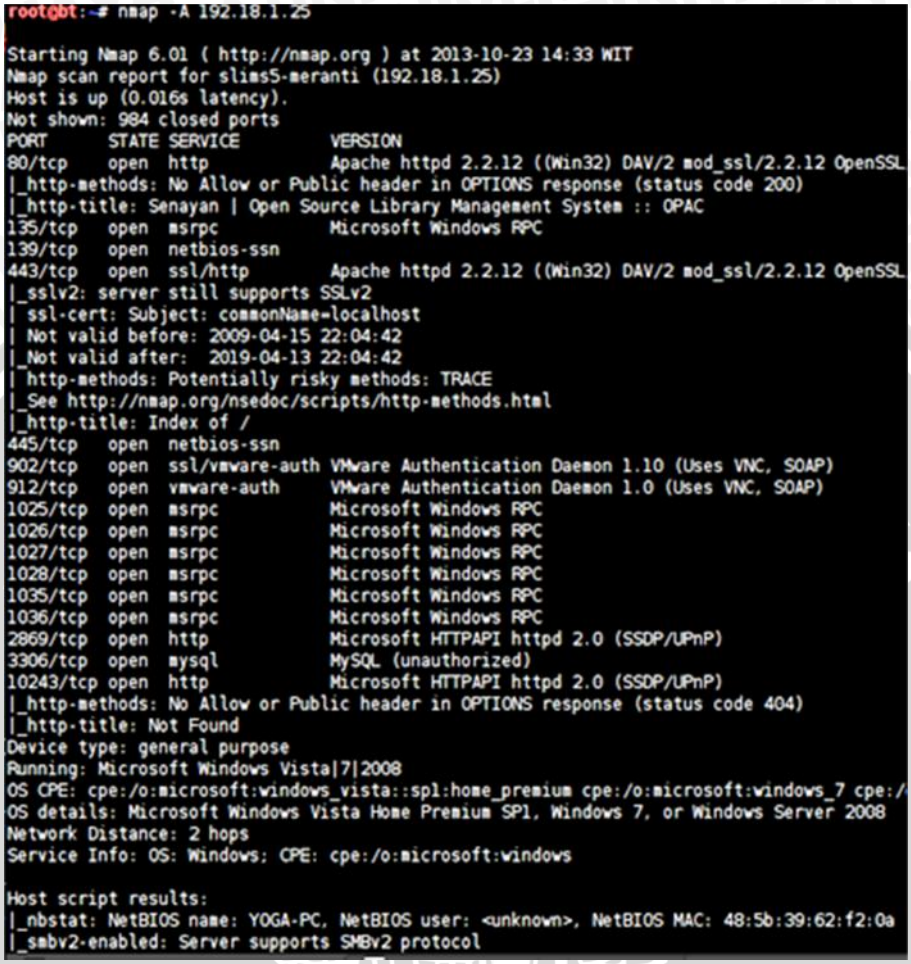
Untuk mengetahui perubahan *resource* dari komputer target, perlu diketahu keadaan awal komputer target ketika dalam kondisi normal. Kondisi normal dari komputer target diinisialisasikan bahwa nilai normal dari CPU usage dalam keadaan tidak ada proses yang sedang dijalankan dengan nilai 1-10 %. RAM usage diinisialisasikan dalam keadaan normal dengan proses yang berjalan dengan nilai 40 - 49% . Sedangkan Network komputer target diinisialisasikan dalam keadaan normal dengan proses yang berjalan dengan nilai 0,24%. *Resource* normal pada komputer target dapat dilihat pada lampiran 1, Lampiran 2 dan Lampiran 3.

5.1.2 Pengujian Simulasi Serangan *Scanning Port* dengan *Firewall* Tidak Aktif

Simulasi serangan *scanning port* menggunakan tool *Nmap* pada komputer target, *scanning port* bertujuan untuk mengetahui nomer *port* dari komputer target yang terbuka. Simulasi serangan ini akan dibagi menjadi dua skenario yaitu *scanning port* dengan *firewall* tidak aktif dan dengan *embedded firewall system* dalam keadaan *firewall* aktif.

Pengujian pertama dilakukan dengan menggunakan syntax : ***Nmap –A 192.18.25*** ,syntax ini digunakan untuk mengetahui semua nomer *port* yang terbuka dan informasi dari komputer target berupa sistem operasi dan *MAC*

address. Simulasi serangan pada komputer penyerang ditunjukkan pada gambar 5.2

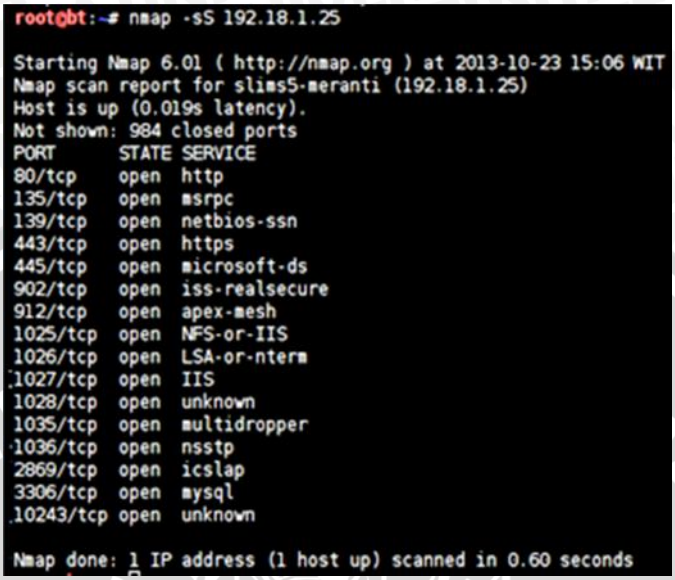


Gambar 5.2 Hasil scanning Nmap -A

Gambar 5.2 adalah keadaaan saat simulasi serangan *scanning port* menggunakan *nmap* pada komputer penyerang. *Scanning port* yang telah dijalankan dapat memperoleh informasi tentang komputer target berupa nomer *port* yang terbuka, sistem operasi yang digunakan, *hostname* komputer dan *MAC address* dari komputer target. Nomer *port* yang terbuka pada komputer target yaitu nomer *port* 80, 135, 139, 443, 445, 902, 912, 1025, 1026, 1027, 1028, 1036, 2869, 3306, dan 10243.

Pengujian kedua scanning *port* menggunakan tool nmap dengan syntax ***nmap –Ss 192.18.1.25***, *scanning port* ini mengirimkan paket *raw SYN* pada komputer target. Ketika *port* terbuka maka akan merespon dengan paket

SYN/ACK dan akan memberikan informasi nomer *port TCP* yang terbuka. Hasil scanning port ditunjukkan pada gambar 5.3.



Gambar 5.3 Hasil scanning Nmap -sS

Gambar 5.3 adalah keadaaan saat simulasi serangan *scanning port* menggunakan *nmap* pada komputer penyerang. *Scanning port* yang telah dijalankan dapat memperoleh informasi tentang komputer target berupa nomer *port* yang terbuka. Nomer *port* yang terbuka pada komputer target adalah nomer *port* 80, 135, 139, 443, 445, 902, 912, 1025, 1026, 1027, 1028, 1035, 1036, 2869, 3306, dan 10243.

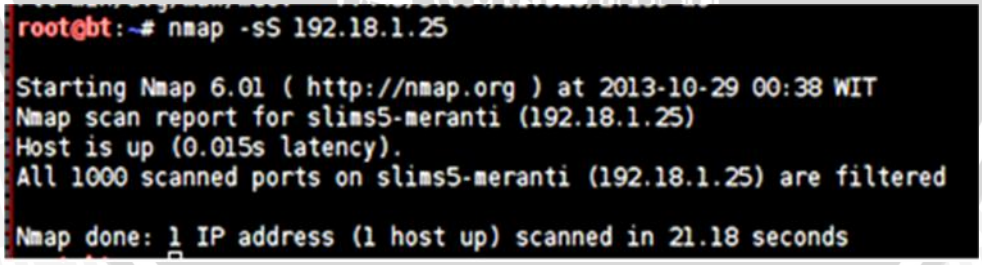
5.1.3 Pengujian Simulasi Serangan *Scanning Port* dengan *Firewall* Aktif

Simulasi serangan *scanning port* dengan *firewall* aktif menggunakan *embedded firewall system*. Simulasi ini bertujuan untuk menganalisa *embedded firewall system* dalam melakukan *filtering* ketika serangan *scanning port*.*Syntax nmap* yang digunakan adalah ***nmap –A 192.18.1.25***

Gambar 5.4 Hasil scanning Nmap dengan firewall aktif

Gambar 5.4 adalah simulasi serangan *scanning port* dengan kondisi *firewall* aktif menggunakan *nmap,* saat simulasi serangan berlangsung komputer penyerang tidak mendapatkan nomer *port* yang terbuka dari komputer target, hal ini ditunjukan pesan yang diterima oleh komputer penyerang yaitu *"all 1000 scanned ports in slims5-meranti (192.18.1.25) are filtered*". Selain itu *Mac Address* dari komputer target juga tidak ditampilkan dalam proses *scanning port*.

Pada pengujian kedua dengan menggunakan *syntax* ***nmap –sS 192.18.1.25***. Pada pengujian ini komputer penyerang juga tidak dapat melihat nomer *port* yang terbuka, karena *firewall* telah melakukan filtering terhadap *scanning port*. Keadaan ini dapat dilihat dari pesan yang diterima oleh komputer penyerang yaitu *"all 1000 scanned ports in slims5-meranti (192.18.1.25) are filtered"* ditunjukkan pada gambar 5.5.

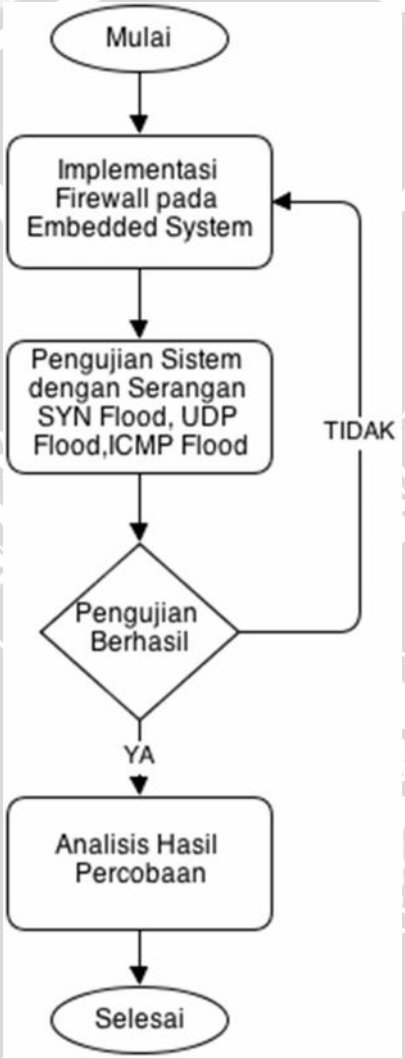


Gambar 5.5 Hasil scanning Nmap –sS dengan firewall aktif

5.1.4 Pengujian Simulasi Serangan *SYN flood* dengan *Firewall* Tidak Aktif

Pengujian dilakukan dengan melakukan simulasi serangan terhadap komputer target terhadap serangan *SYN flood*, *ICMP flood* dan *UDP flood*. *Flowchart* Gambar 5.6 menjelaskan alur mulai dari implementasi sistem kemudian melakukan pengujian *firewall* dengan melakukan simulasi serangan

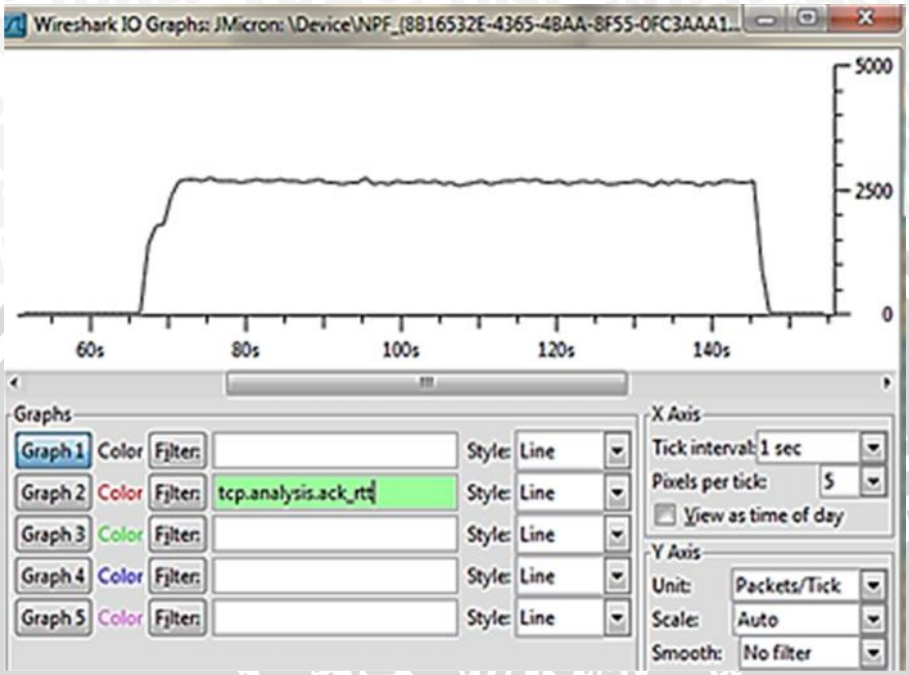
SYN flood, ICMP flood dan *UDP flood*. Simulasi serangan dinyatakan berhasil ketika sudah sesuai dengan rumusan masalah dan batasan masalah, apabila belum sesuai dengan rumusan masalah dan batasan masalah dalam melakukan pengujian akan kembali ke proses sebelumnya yaitu implementasi sistem. Tahapan selanjutnya adalah melakukan analisis terhadap hasil pengujian tersebut.



Gambar 5.6 Alur Simulasi serangan *SYN flood. ICMP flood,* dan *UDP flood*

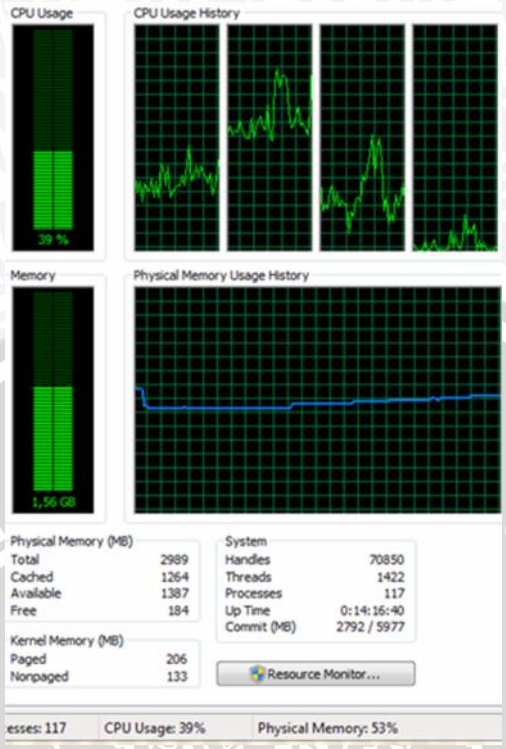
Serangan *SYN flood* adalah membanjiri request berupa paket *SYN* kepada komputer target sehingga komputer target sibuk menerima paket *SYN* yang dikirimkan. Untuk mengetahui akibat serangan *SYN flood* terhadap komputer target, simulasi serangan ini akan membandingkan terhadap keadaan komputer

target sebelum *firewall* aktif. Pengujian serangan dilakukan menggunakan *hping3* pada komputer penyerang dengan syntax : ***hping3 -i u100 --syn -p 80 192.168.1.24*** pada komputer penyerang.



Gambar 5.7 *Capture wireshark* simulasi serangan *SYN flood*

Gambar 5.7 adalah *capture wireshark* simulasi serangan *SYN flood* yang berlangsung selama kurang lebih 80 detik, dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* yaitu antara 2500 sampai 3000 paket. Dengan jumlah paket yang cukup besar dalam rentang waktu yang sedikit akan mengakibatkan konsumsi *resource* yang berlebihan pada komputer target. Saat simulasi serangan *SYN flood* mengakibatkan proses *three way handshake* tidak pernah terjadi. Banjir paket *SYN* terjadi karena paket yang yang dikirimkan komputer penyerang secara terus menerus sehingga komputer target akan sangat sibuk untuk membalas paket yang telah dikirimkan. Keadaan *resource* komputer target saat terjadi simulasi serangan *SYN flood* berlangsung ditunjukkan pada Gambar 5.8 dan Gambar 5.9.



Gambar 5.8 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *SYN flood* tanpa *firewall*

Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnected
Local Area Connection	1,35 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnected
Wireless Network Connection 4	0 %	-	Disconnected

Gambar 5.9 *Resource network* saat simulasi serangan *SYN flood* tanpa *firewall*

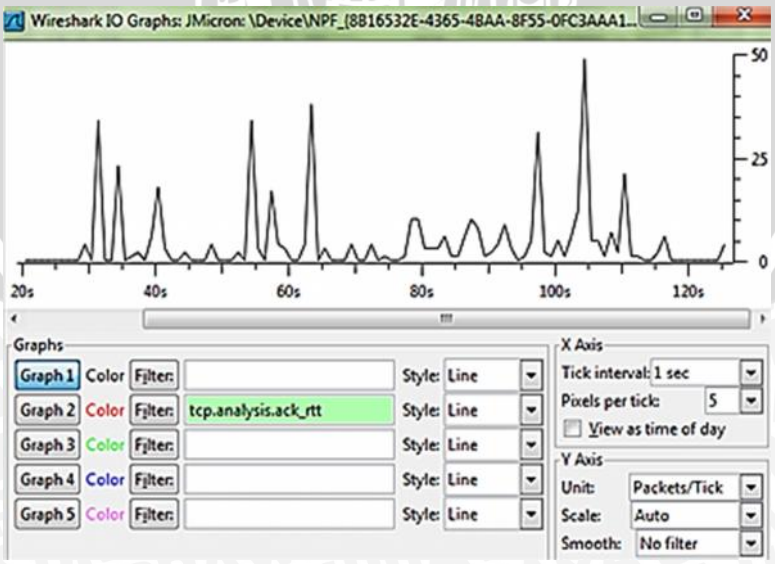
Gambar 5.8 adalah tampilan *windows task manager* komputer target saat simulasi serangan *SYN flood*, persentase nilai *resource* CPU usage mengalami kenaikan sebesar 39%. Nilai *RAM usage* mengalami kenaikan menjadi 53%. Sedangkan Gambar 5.9 pada *Local Area Connection* persentase nilai dari *network utilization* mengalami kenaikan menjadi 1,35%. *Resource* komputer target mengalami kenaikan saat simulasi serangan *SYN flood* berlangsung, hal ini dibuktikan dari jumlah kenaikan nilai *resource* dibandingkan dari keadaan normal *resource* komputer target. Untuk akses *web server* oleh komputer client, ketika akan mengakses *web server* terjadi kegagalan untuk mengakses halaman *web server*, hal ini ditunjukan pada Gambar 5.10.

Gambar 5.10 Akses *web server* tanpa *firewall*

5.1.5 Pengujian Simulasi Serangan *SYN flood* dengan *firewall* aktif

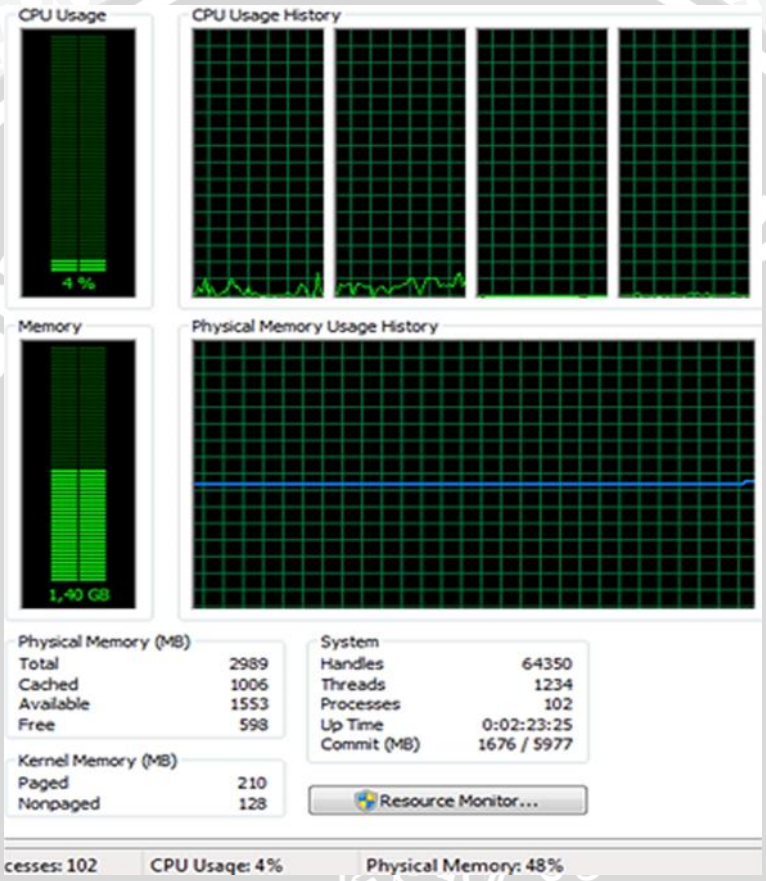
Pengujian simulasi serangan SYN flood dengan firewall aktif dilakukan dengan menggunakan *embedded firewall system*. Pengujian dilakukan dengan jumlah paket perdetik yang berbeda untuk melihat kemampuan *firewall* dalam melakukan *block* serangan *SYN flood*. Estimasi jumlah paket perdetik yang digunakan dalam simulasi serangan adalah 1000 dan 10000 paket perdetik

Pengujan pertama dilakukan dengan simulasi serangan *SYN flood* pada komputer penyerang dengan menggunakan tool *Hping3* dengan syntax ***hping3 -i u100 --syn -p 80 192.18.1.25***. Besar paket *SYN flood* yang di *capture wireshark* dapat dilihat pada komputer target saat serangan *SYN flood* berlangsung pada Gambar 5.11.



Gambar 5.11 *Capture Wireshark* saat simulasi serangan *SYN flood* pertama

Gambar 5.11 menjelaskan saat simulasi serangan *SYN flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* adalah 0 sampai 50 paket. Hal ini dikarenakan *firewall* dapat melakukan *filtering* ketika terjadi serangan *SYN flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* pada komputer target ditunjukkan pada Gambar 5.12 dan Gambar 5.13.

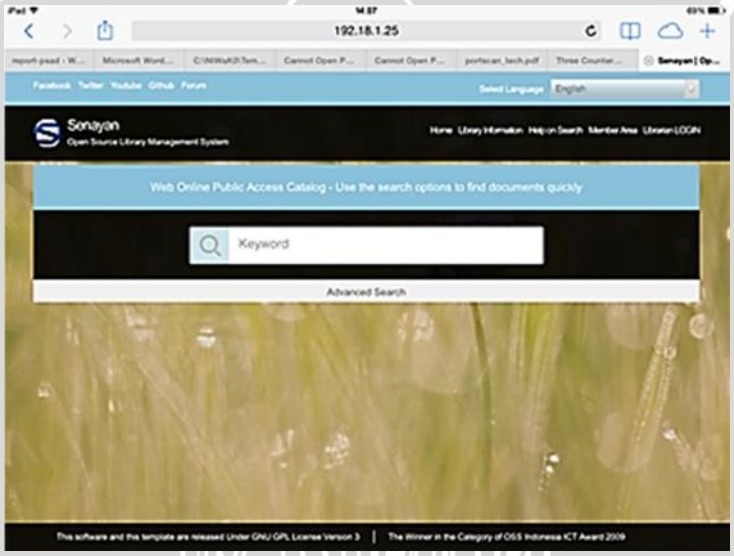


Gambar 5.12 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *SYN flood* pertama

Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnected
Local Area Connection	0,11 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnected
Wireless Network Connection 4	0 %	-	Disconnected

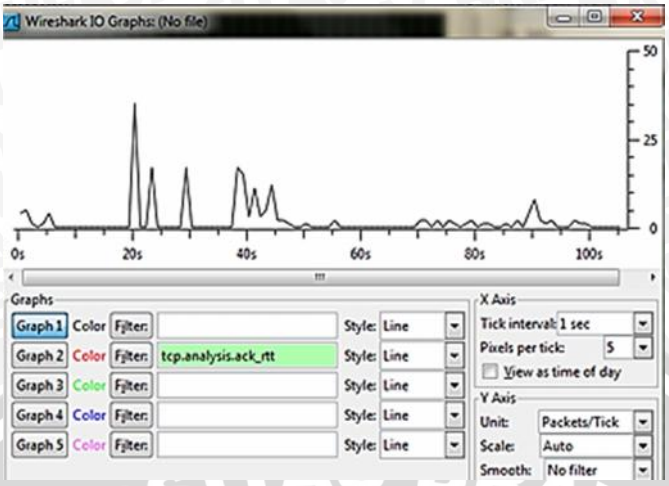
Gambar 5.13 *Resource network* saat simulasi serangan *SYN flood* pertama

Gambar 5.12 menunjukkan *windows task manager* pada komputer target, persentase nilai *resource* CPU usage sebesar 4% dan pada RAM usage sebesar 48%. Gambar 5.13 menunjukkan *resource network* dari komputer target, pada koneksi *Local Area Connection* persentase nilai *network utilization* sebesar 0,11%. Keadaan *resource* komputer target dalam keadaaan normal karena persentase nilai dari *resource* tidak melebihi persentase nilai normal komputer target. Saat akses *web server* dari komputer *host*, halaman *web server* dapat diakses dan ditampilkan dengan normal, ditunjukkan pada Gambar 5.14. Sehingga dapat diambil kesimpulan ketika simulasi serangan fi*rewall* dapat melakukan *block* serangan yang menuju komputer target, karena *web server* dapat diakses secara normal dan *CPU usage*, *RAM usage* serta *network* dalam keadaan normal.



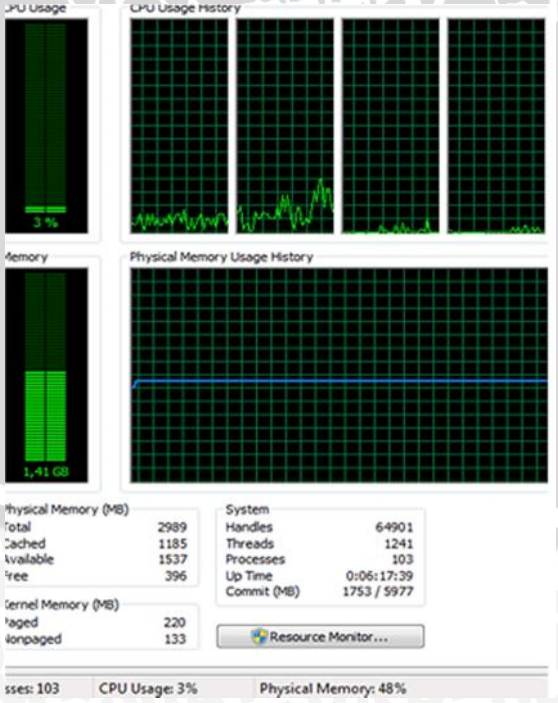
Gambar 5.14 Akses *web server* saat simulasi serangan *SYN flood* pertama

Pengujian kedua melakukan simulasi serangan *SYN flood* dengan estimasi 10.000 paket perdetik , menggunakan *tool hping* dengan syntax ***hping3 –i u10 --syn –p 80 192.18.1.25***. Dengan penambahan jumlah paket perdetik akan mengetahui kemampuan *firewall* dalam melakukan *block* serangan *SYN flood*. Hasil pengujian kedua *capture wireshark* pada komputer target ditunjukkan pada Gambar 5.15.



Gambar 5.15 *Capture Wireshark* saat simulasi serangan *SYN flood* kedua

Gambar 5.15 menjelaskan saat simulasi serangan *SYN flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* 0 sampai 35 paket. Hal ini dikarenakan *firewall* dapat melakukan filter ketika terjadi serangan *SYN flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* pada komputer target ditunjukkan pada gambar 5.16 dan gambar 5.17.



Gambar 5.16 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *SYN flood* kedua

Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnected
Local Area Connection	0 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnected
Wireless Network Connection 4	0 %	-	Disconnected

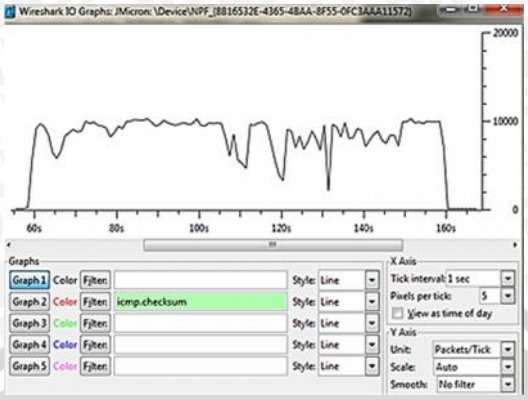
Gambar 5.17 *Resource network* saat simulasi serangan *SYN flood* kedua

Gambar 5.16 menunjukkan windows task manager pada komputer target, persentase nilai *resource* CPU usage sebesar 3% dan pada RAM usage sebesar 48 %. Gambar 5.17 menunjukkan *resource network* dari komputer target, pada koneksi *Local Area Connection* persentase nilai *network utilization* sebesar 0 %. Hal ini dikarenakan terjadi kegagalan saat akses *web server*, sehingga tidak ada proses yang terjadi pada *network utilization.* Keadaan akses web server ditunjukkan pada Gambar 5.14.

Gambar 5.18 Akses *web server* saat simulasi serangan *SYN flood* kedua

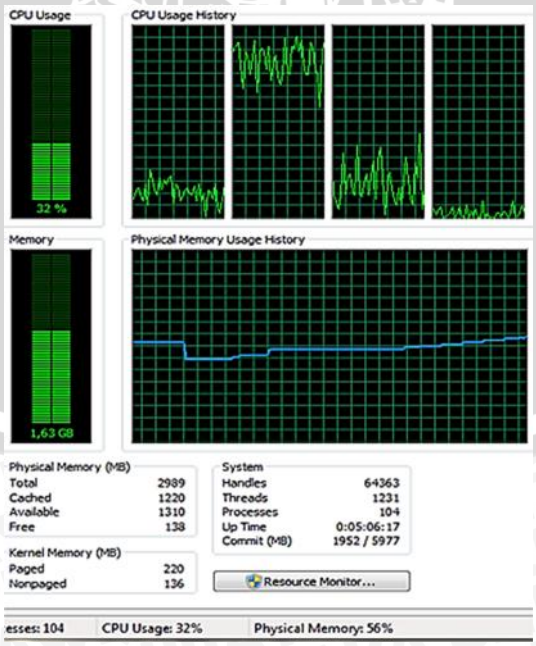
5.1.6 Pengujian Simulasi Serangan *ICMP flood* dengan *Firewall* Tidak Aktif

Serangan *ICMP flood* bekerja dengan membanjiri komputer target dengan menggunakan protokol *ICMP*, komputer penyerang melakukan *request ICMP* secara terus menerus sehingga mengharuskan komputer target untuk melakukan *reply* terhadap *request* tersebut. Untuk mengetahui akibat dari *ICMP flood* terhadap komputer target maka pengujian serangan dilakukan sebelum *firewall* aktif dan setelah *firewall* aktif. Pengujian serangan menggunakan *hping3* pada komputer penyerang dengan syntax : ***hping3 -i u100 --icmp -p 80 192.168.1.25***.

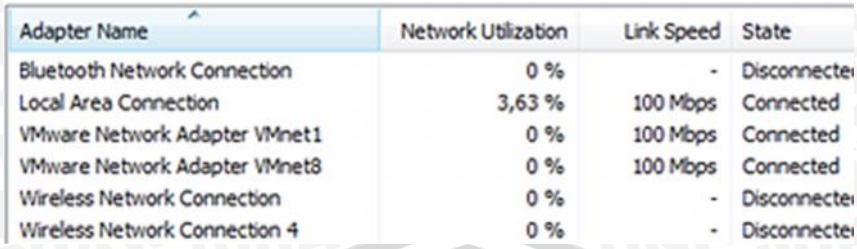


Gambar 5.19 *Capture Wireshark* saat simulasi serangan *ICMP flood* tanpa *firewall*

Gambar 5.19 menjelaskan saat simulasi serangan *ICMP flood* berlangsung selama kurang lebih 100 detik, besaran paket yang dapat ditangkap oleh *wireshark* yaitu antara 7000 sampai 10000 paket. Banjir paket ICMP terjadi karena *echo(ping)request* yang dikirimkan komputer penyerang secara terus menerus. Komputer target akan sangat sibuk untuk membalas paket yang dikirimkan, sehingga semua request tidak bisa untuk direspon kembali. Dengan jumlah paket yang cukup besar dalam rentang waktu yang sedikit akan mengakibatkan konsumsi *resource* yang berlebihan pada komputer target. Keadaan *resource* komputer target ketika terjadi simulasi serangan *ICMP flood* ditunjukkan pada Gambar 5.20 dan Gambar 5.21.



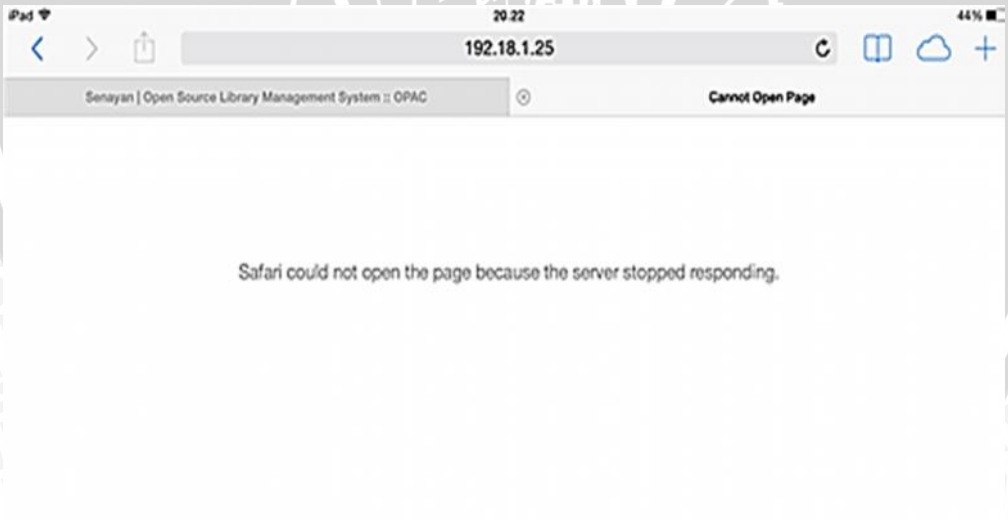
Gambar 5.20 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *ICMP flood* tanpa *firewall*



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnecte
Local Area Connection	3,63 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnecte
Wireless Network Connection 4	0 %	-	Disconnecte

Gambar 5.21 *Resource network* saat simulasi serangan *ICMP flood* tanpa *firewall*

Gambar 5.20 menunjukkan *windows task manager* dari komputer target, dapat dilihat dalam grafik pada *CPU usage* terlihat kenaikan persentase sebesar 49% dari keadaan normal *CPU usage* sebesar 1-5 %. Persentase nilai *RAM usage* mengalami kenaikan menjadi 56%. Sedangkan pada Gambar 5.21 pada *Local Area Connection* nilai dari *network utilization menjadi 3,63 %*. *Resource* komputer target mengalami kenaikan saat simulasi serangan *ICMP flood* berlangsung, hal ini dibuktikan dari jumlah kenaikan nilai *resource* dibandingkan dengan keadaan normal *resource* komputer target. Untuk akses web server oleh komputer client, ketika akan mengakses *web server* terjadi kegagalan untuk mengakses halaman *web server*, hal ini dapat ditunjukan pada Gambar 5.22.

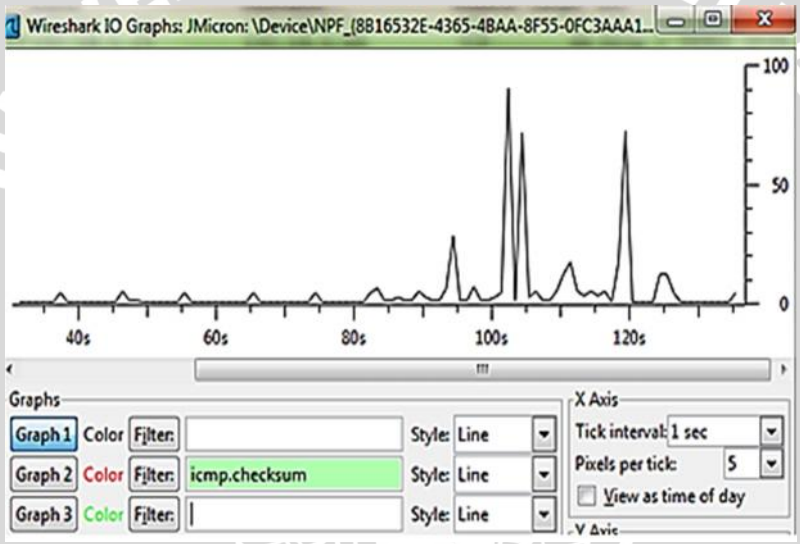


Gambar 5.22 Akses *web server* saat serangan *ICMP flood* tanpa *firewall*

5.1.7 Pengujian Simulasi Serangan *ICMP flood* dengan *Firewall* Aktif

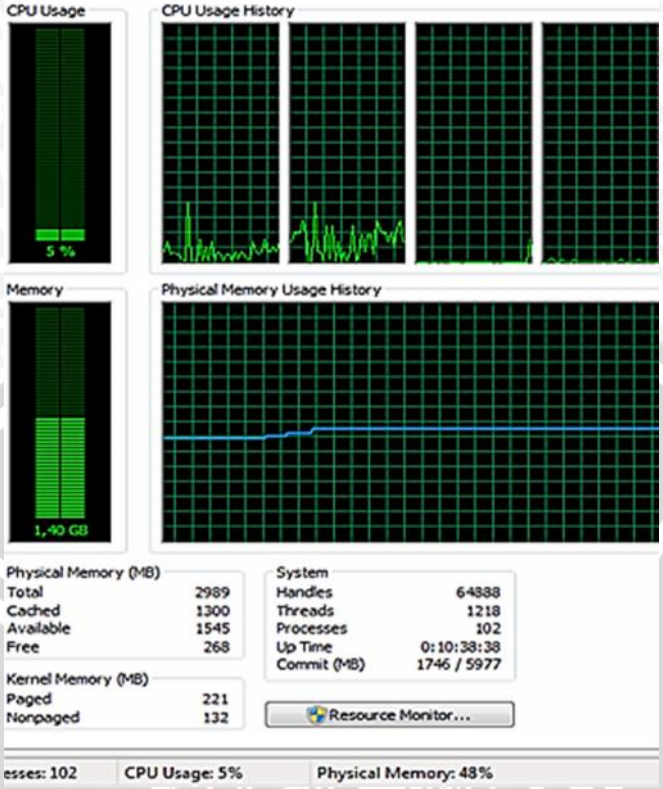
Pengujian simulasi serangan *ICMP flood* dengan *firewall* aktif dilakukan dengan jumlah paket perdetik yang berbeda-beda untuk melihat kemampuan *firewall* dalam melakukan *block* serangan *ICMP flood*. Estimasi jumlah paket perdetik yang digunakan dalam simulasi serangan adalah 1.000 dan 10.000 paket

Pengujan pertama dilakukan dengan simulasi serangan *ICMP flood* dengan menggunakan tool *Hping3* dengan syntax ***hping3 –i u100 --icmp –p 80 192.18.1.25***. Besar paket *ICMP flood* yang di *capture wireshark* pada komputer target ditunjukkan pada gambar 5.23.

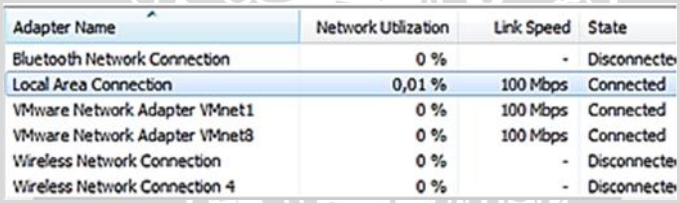


Gambar 5.23 *Capture Wireshark* saat simulasi serangan *ICMP flood* pertama

Gambar 5.23 menjelaskan saat simulasi serangan *ICMP flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* adalah 0 sampai 90 paket. Hal ini dikarenakan *embedded firewall system* dapat melakukan filter ketika terjadi serangan *ICMP flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* pada komputer target ditunjukkan pada Gambar 5.24 dan Gambar 5.25.



Gambar 5.24 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *ICMP flood* pertama

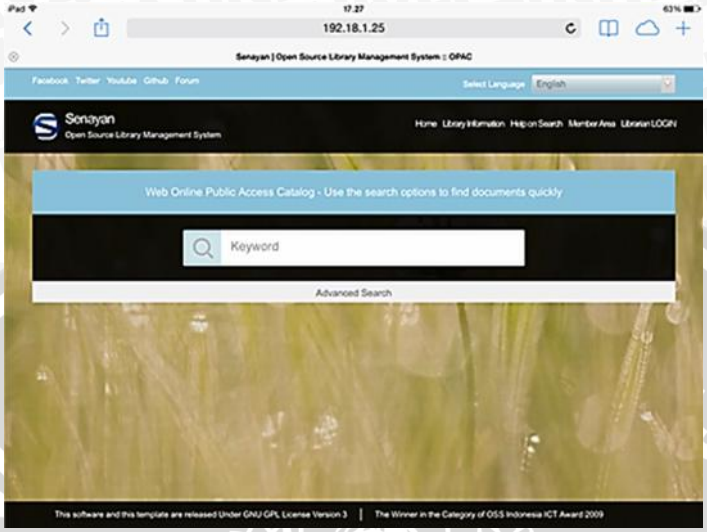


Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnecte
Local Area Connection	0,01 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnecte
Wireless Network Connection 4	0 %	-	Disconnecte

Gambar 5.25 *Resource network* saat simulasi serangan *ICMP flood* pertama

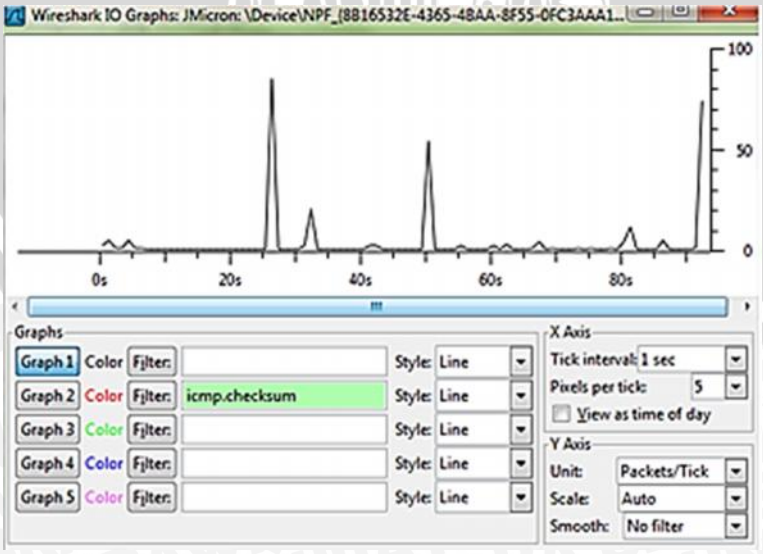
Gambar 5.24 menunjukkan persentase nilai dari *CPU usage* dan *RAM usage* saat terjadi simulasi serangan *ICMP flood*. Persentase nilai *CPU usage* sebesar 5 % dan *RAM usage* sebesar 48 %. Gambar 5.25 menunjukkan *resource network*, persentase nilai *network utilization* pada *Local Area Connection* sebesar 0,01 %. Keadaan *resource* komputer target dalam keadaaan normal karena nilai persentase dari *resource* tidak melebihi nilai persentase normal komputer target. Saat akses *web server* dari komputer *host*, halaman *web server* dapat diakses dan ditampilkan dengan normal, ditunjukkan pada gambar 5.26. Sehingga dapat diambil kesimpulan ketika simulasi serangan fi*rewall* dapat melakukan *block* serangan

yang menuju komputer target, karena *web server* dapat diakses secara normal dan *CPU usage*, *RAM usage* dan *network* dalam keadaan normal.



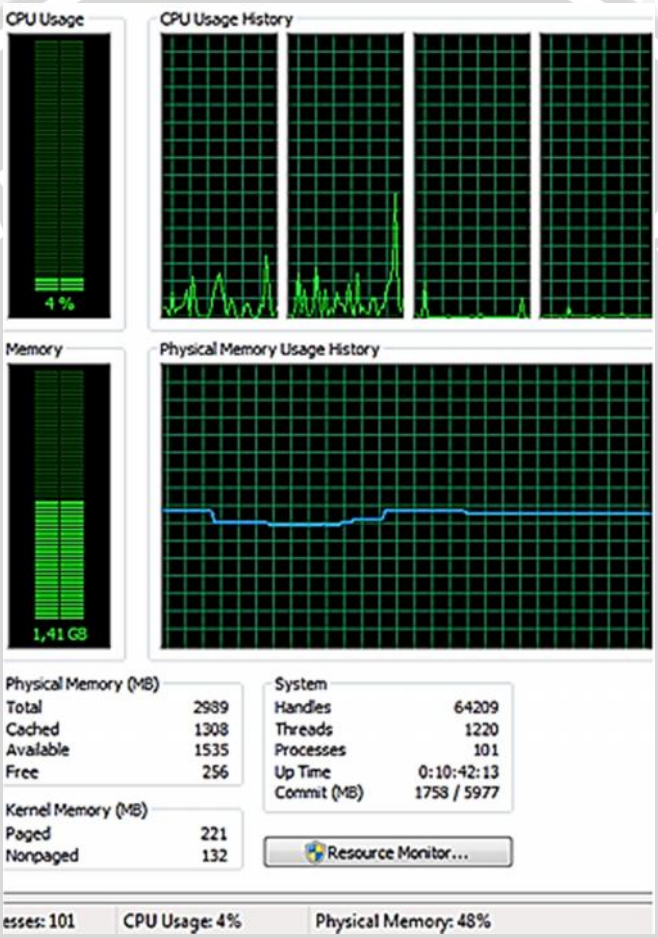
Gambar 5.26 Akses *web server* saat simulasi serangan *ICMP flood* pertama

Pengujian kedua dengan melakukan simulasi serangan *ICMP flood* dengan estimasi 10.000 paket per detik. Komputer penyerang melakukan Hping dengan syntax ***hping3 –i u10 – icmp –p 80 192.18.1.25***. Dengan penambahan jumlah paket perdetik akan mengetahui kemampuan *firewall* dalam melakukan *block* serangan *ICMP flood*. Hasil pengujian kedua *capture wireshark* pada komputer target ditunjukkan pada Gambar 5.27.

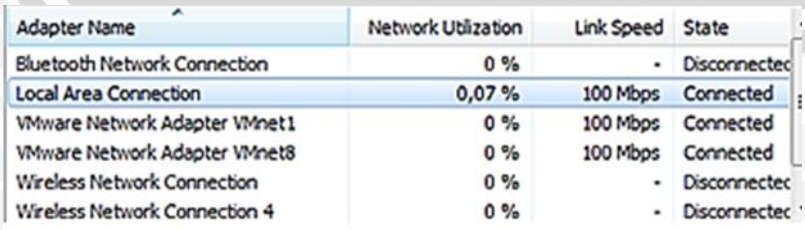


Gambar 5.27 *Capture Wireshark* saat simulasi serangan *ICMP flood* kedua

Pada gambar 5.27 menjelaskan saat simulasi serangan *ICMP flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* 0 sampai 90 paket. Hal ini dikarenakan *firewall* dapat melakukan filter ketika terjadi serangan *ICMP flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* pada komputer target ditunjukkan pada gambar 5.28 dan gambar 5.29.

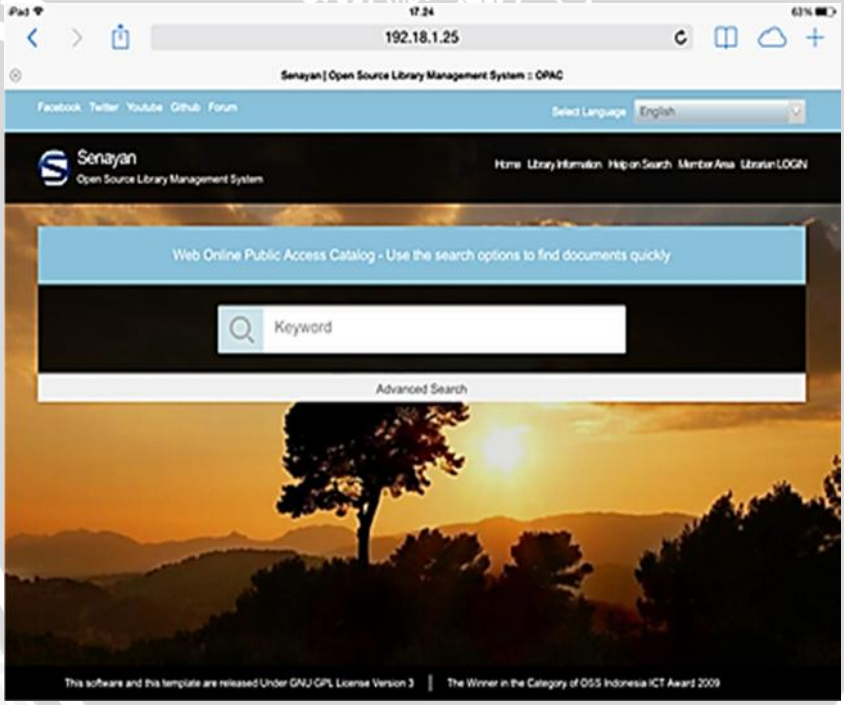


Gambar 5.28 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *ICMP flood* kedua



Gambar 5.29 *Resource network* saat simulasi serangan *ICMP flood* kedua

Gambar 5.28 menunjukkan nilai dari *CPU usage dan RAM usage* saat terjadi simulasi serangan *ICMP flood*. Persentase nilai CPU usage sebesar 4 % dan *RAM usage* sebesar 48 %. Gambar 5.29 menunjukkan *resource network*, persentase nilai *network utilization* pada *Local Area Connection* sebesar 0,07 %. Keadaan *resource* komputer target dalam keadaaan normal karena nilai persentase dari *resource* tidak melebihi nilai persentase normal komputer target. Saat akses *web server* dari komputer *host*, halaman *web server* dapat diakses dan ditampilkan dengan normal, ditunjukkan pada gambar 5.30. Sehingga dapat diambil kesimpulan ketika simulasi serangan *ICMP flood embedded* f*irewall system* dapat melakukan *block* serangan yang menuju komputer target, karena *web server* dapat diakses secara normal. Nilai persentase *CPU usage*, *RAM usage* dan *network* juga dalam keadaan normal.

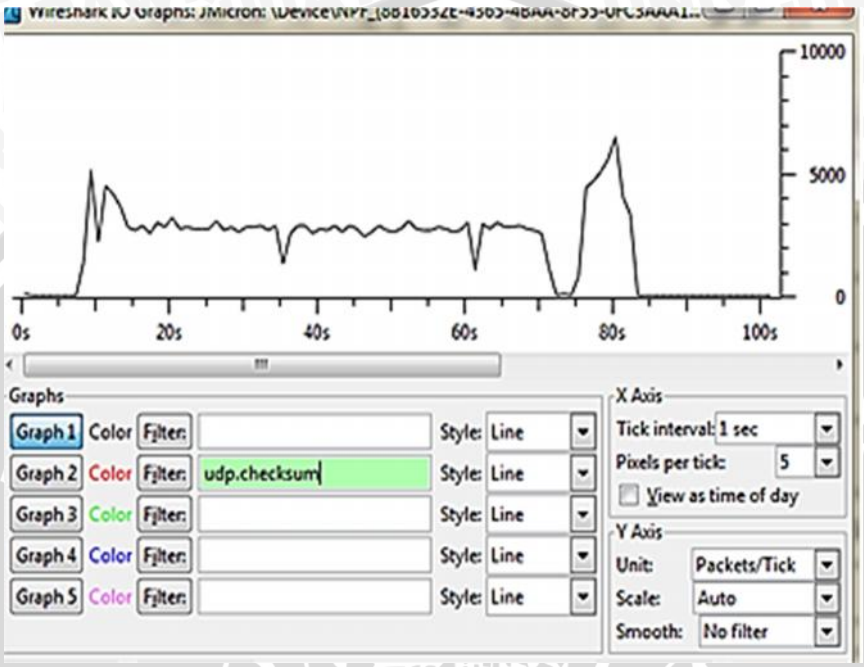


Gambar 5.30 Akses *web server* saat simulasi serangan *ICMP flood* kedua

5.1.8 Pengujian Simulasi Serangan *UDP flood* dengan *firewall* Tidak Aktif

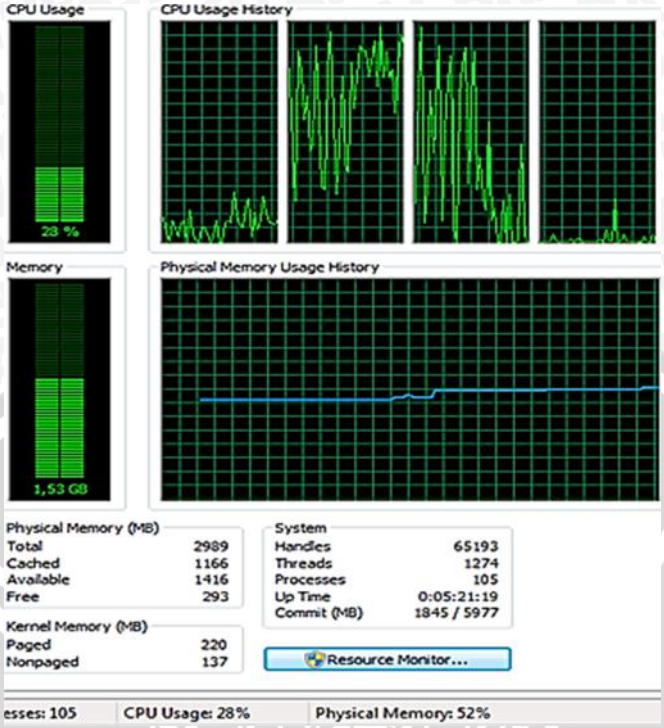
Serangan *UDP flood* bekerja dengan membanjiri komputer target dengan menggunakan protokol *UDP*, *UDP flood* merupakan serangan yang bersifat *connectionless*, yaitu tidak memperhatikan apakah paket yang dikirim ke komputer diterima atau tidak oleh komputer target. Untuk mengetahui akibat dari

UDP flood terhadap komputer target maka pengujian serangan dilakukan dengan mengnonaktifkan *firewall*. Pengujian simulasi serangan menggunakan *hping3* pada komputer penyerang dengan *syntax* : ***hping3 -i u100 --udp -p 80 192.168.1.24***.

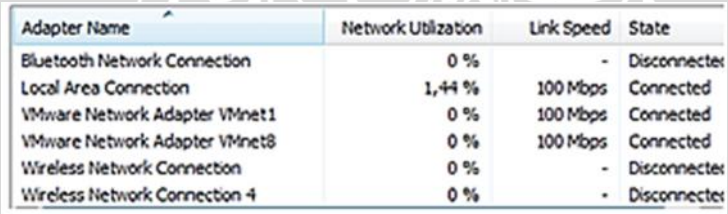


Gambar 5.31 *Capture Wireshark* saat simulasi serangan *UDP flood* tanpa *firewall*

Pada gambar 5.31 menjelaskan saat simulasi serangan *UDP flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* yaitu antara 2000 sampai 3000 paket. Banjir paket *UDP* terjadi ketika komputer target merespon dengan menghasilkan balasan *ICMP "Destination Port Unreachable"*, keadaan ini terjadi karena tidak ada aplikasi yang berjalan untuk menerima kembali paket *UDP* pada komputer penyerang. Sehingga komputer target akan sibuk melayani permintaan tidak valid yang dilakukan oleh komputer penyerang. Keadaan *resource* pada komputer target ketika simulasi serangan *UDP flood* ditunjukkan pada Gambar 5.32 dan Gambar 5.33.



Gambar 5.32 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *UDP flood* tanpa *firewall*



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnecte
Local Area Connection	1,44 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnecte
Wireless Network Connection 4	0 %	-	Disconnecte

Gambar 5.33 *Resource network* saat simulasi serangan *UDP flood* tanpa *firewall*

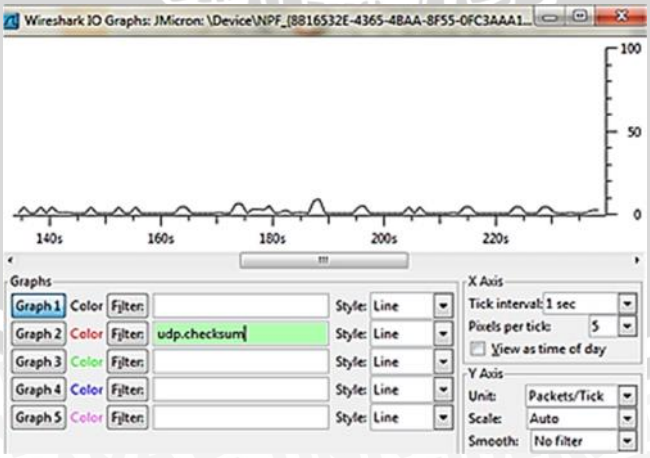
Gambar 5.32 menunjukkan *windows task manager* dari komputer target, pada *CPU usage* terlihat kenaikan persentase sebesar 28% dari keadaan normal *CPU usage* sebesar 1-5 % dan persentase nilai *RAM usage* mengalami kenaikan menjadi 52%. Gambar 5.33 menunjukkan *resource network*, pada *Local Area Connection* nilai persentase dari *network utilization menjadi 1,44 %*. *Resource* komputer target mengalami kenaikan saat simulasi serangan *UDP flood* berlangsung, hal ini dibuktikan dari jumlah kenaikan persentase nilai *resource* dibandingkan dari keadaan normal *resource* komputer target. Untuk akses *web server* oleh komputer *client*, ketika akan mengakses *web server* terjadi kegagalan untuk mengakses halaman *web server*, hal ini ditunjukan pada Gambar 5.34.

Gambar 5.34 Akses *web server* saat serangan *ICMP flood* tanpa *firewall*

5.1.9 Pengujian Simulasi Serangan *UDP flood* dengan *Firewall* Aktif

Pengujian simulasi serangan *UDP flood* dengan *firewall* aktif dilakukan dengan *embedded firewall system*. Pengujian dilakukan dengan jumlah paket perdetik yang berbeda-beda untuk melihat kemampuan *firewall* dalam melakukan *block* serangan *UDP flood*. Estimasi jumlah paket perdetik yang digunakan dalam simulasi serangan adalah 1.000 dan 10.000 paket

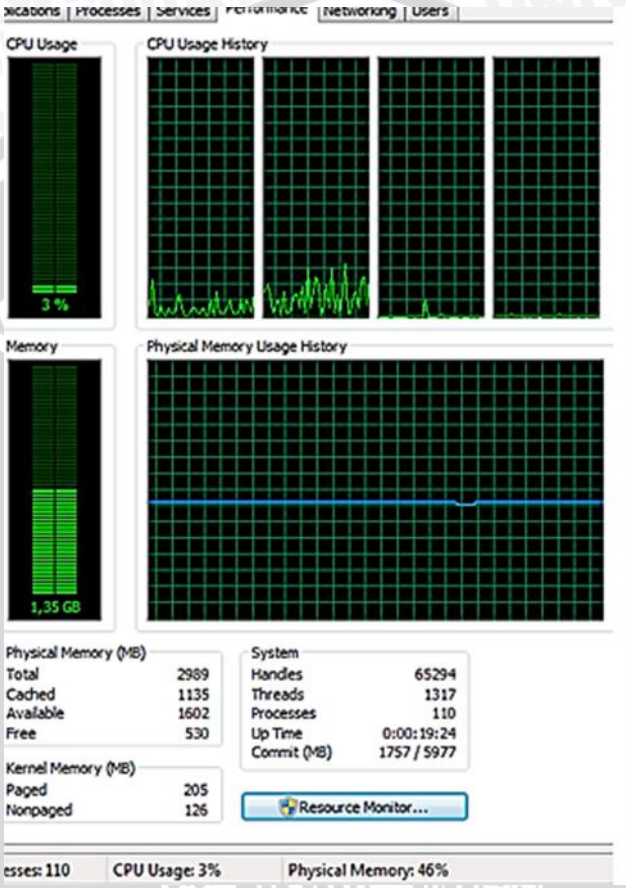
Pengujian pertama dilakukan dengan simulasi serangan *UDP flood* pada komputer penyerang dengan menggunakan tool *Hping3* dengan syntax ***hping3 –i u100 --icmp –p 80 192.18.1.25***. Besar paket *UDP flood* yang di *capture wireshark* dapat dilihat pada komputer target saat serangan *UDP flood* berlangsung pada gambar 5.35.



Gambar 5.35 *Capture Wireshark* saat simulasi serangan *UDP flood* pertama

Gambar 5.35 menjelaskan saat simulasi serangan *UDP flood* berlangsung selama kurang lebih 80 detik, dapat dilihat bahwa besaran paket yang dapat

ditangkap oleh *wireshark* adalah 0 sampai 10 paket. Hal ini dikarenakan *firewall* dapat melakukan filter ketika terjadi serangan *UDP flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* pada komputer target ditunjukkan pada Gambar 5.36 dan Gambar 5.37.



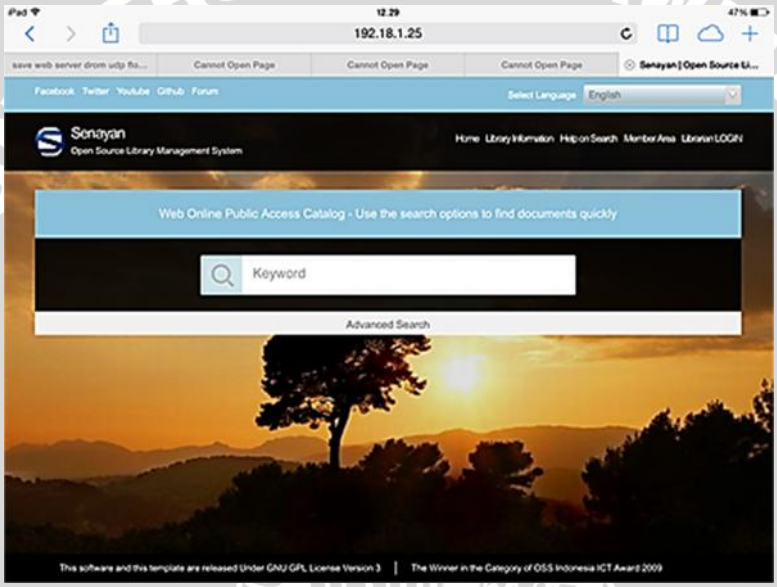
Gambar 5.36 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *UDP flood* pertama

Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnected
Local Area Connection	0,18 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnected
Wireless Network Connection 4	0 %	-	Disconnected

Gambar 5.37 *Resource network* saat simulasi serangan *UDP flood* pertama

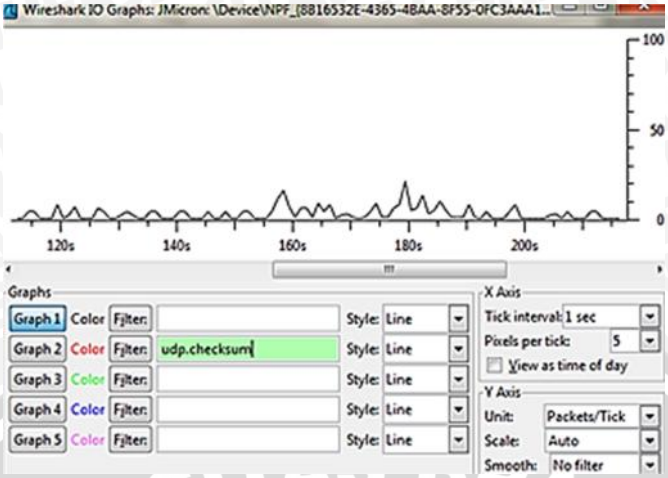
Gambar 5.36 dan Gambar 5.37 menunjukkan persentase nilai dari *CPU usage* dan *RAM usage* saat terjadi simulasi serangan *UDP flood*. Persentase nilai *CPU usage* sebesar 3 % dan *RAM usage* sebesar 46 %. Pada Gambar 5.37

menunjukkan *resource network*, dapat dilihat bahwa persentase nilai *network utilization* pada *Local Area Connection* sebesar 0,18 %. Keadaan *resource* komputer target dalam keadaaan normal karena persentase nilai dari *resource* tidak melebihi persentase nilai normal komputer target. Saat akses *web server* dari komputer *user*, halaman *web server* dapat diakses dan ditampilkan dengan normal, ditunjukkan pada gambar 5.38. Sehingga dapat diambil kesimpulan ketika simulasi serangan f*irewall* dapat melakukan *block* serangan UDP flood yang menuju komputer target, karena *web server* dapat diakses secara normal dan *CPU usage*, *RAM usage* dan *network* dalam keadaan normal.



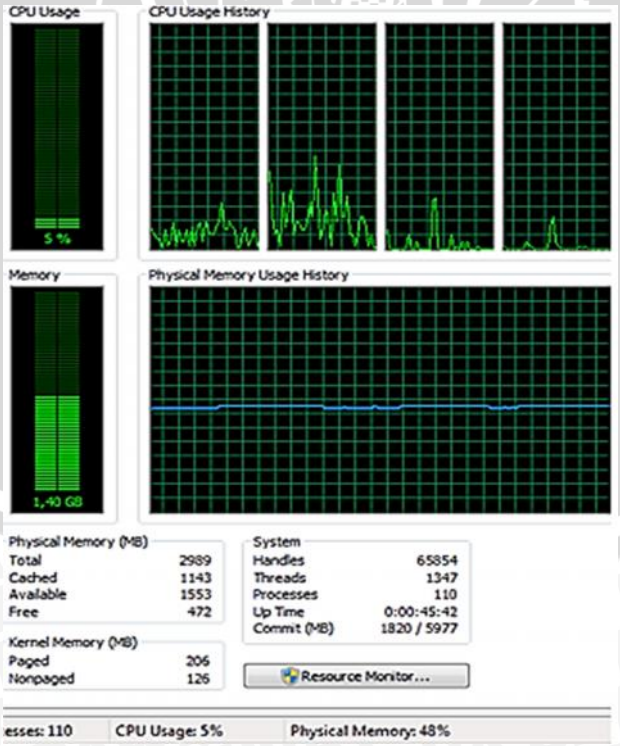
Gambar 5.38 Akses *web server* saat simulasi serangan *UDP flood* pertama

Pengujian kedua dengan melakukan simulasi serangan *UDP flood* dengan estimasi 10.000 paket per detik. Komputer penyerang melakukan *Hping* dengan syntax ***hping3 –i u10 --udp –p 80 192.18.1.25***. Dengan penambahan jumlah paket perdetik akan mengetahui kemampuan *firewall* dalam melakukan *block* serangan *UDP flood*. Hasil pengujian kedua *capture wireshark* pada komputer target ditunjukkan pada Gambar 5.39.

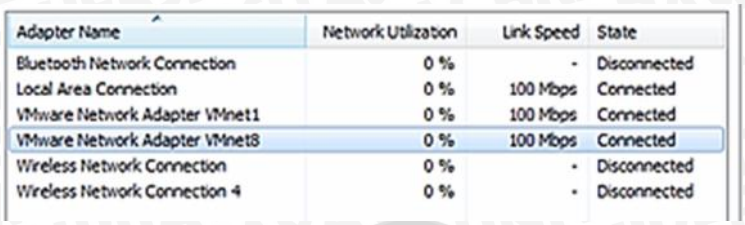


Gambar 5.39 *Capture Wireshark* saat simulasi serangan *UDP flood* kedua

Gambar 5.39 menjelaskan saat simulasi serangan *UDP flood* berlangsung selama kurang lebih 80 detik dapat dilihat bahwa besaran paket yang dapat ditangkap oleh *wireshark* 0 sampai 20 paket. Hal ini dikarenakan *firewall* dapat melakukan *filtering* ketika terjadi serangan *UDP flood*, sehingga komputer target tidak akan menerima paket berjumlah besar yang dikirimkan. Keadaan *resource* komputer target ditunjukkan pada Gambar 5.40 dan Gambar 5.41



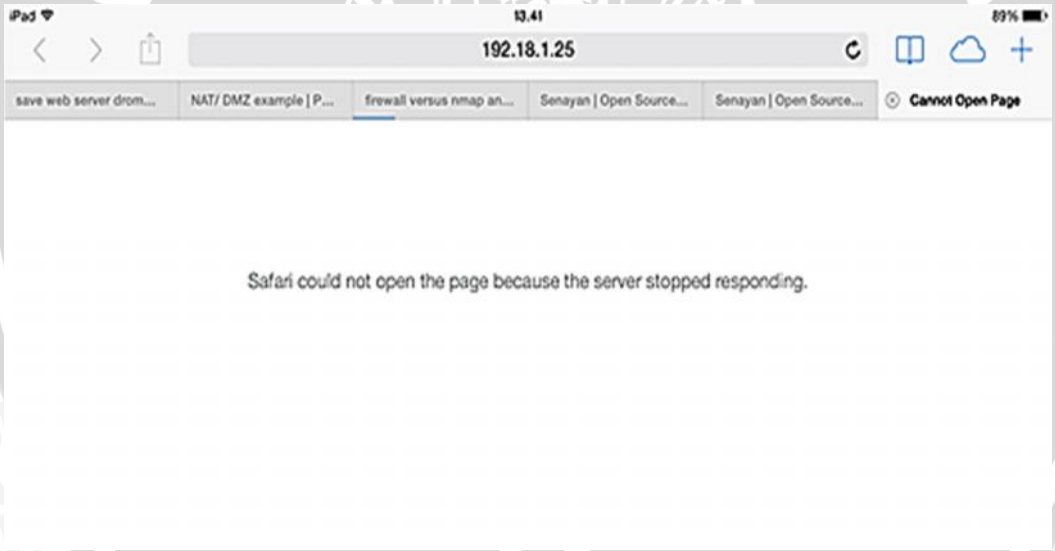
Gambar 5.40 *Resource CPU usage* dan *RAM usage* saat simulasi serangan *UDP flood* kedua



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Network Connection	0 %	-	Disconnected
Local Area Connection	0 %	100 Mbps	Connected
VMware Network Adapter VMnet1	0 %	100 Mbps	Connected
VMware Network Adapter VMnet8	0 %	100 Mbps	Connected
Wireless Network Connection	0 %	-	Disconnected
Wireless Network Connection 4	0 %	-	Disconnected

Gambar 5.41 *Resource network* saat simulasi serangan *UDP flood* kedua

Gambar 5.40 menunjukkan nilai dari CPU usage dan RAM usage saat terjadi simulasi serangan *UDP flood*. Persentase nilai CPU usage sebesar 5 % dan *RAM usage* sebesar 48 %. Pada Gambar 5.41 *resource network* dapat dilihat bahwa persentase nilai *network utilization* pada *Local Area Connection* sebesar 0%. Hal ini dikarenakan terjadi kegagalan saat akses *web server*, sehingga tidak ada proses yang terjadi pada *network utilization.* Keadaan akses web server ditunjukkan pada Gambar 5.42.



Gambar 5.42 Akses *web server* saat simulasi serangan *UDP flood* kedua

5.2 Analisis

Pada sub bab ini dijelaskan tentang analisis dari hasil pengujian yang telah dilakukan sebelumnya. Analisis pengujian dibagi menjadi dua yaitu simulasi serangan dengan *Scanning port* dan *DoS* meliputi *SYN flood*, *ICMP flood*, dan *UDP flood*.

5.2.1 Analisis Pengujian Simulasi Serangan *Scanning port*

Analisis hasil simulasi serangan pada komputer target dibagi menjadi dua yaitu simulasi serangan dengan keadaan *firewall* tidak aktif dan dengan keadaan *firewall* aktif, ditunjukkan pada Tabel 5.2 dan Tabel 5.3.

Tabel 5.2 Hasil pengujian simulasi serangan scanning *port*

Serangan *Scanning port*	**Hasil Scanning *Port***	
	***Firewall* tidak aktif**	***Firewall* Aktif**
Nmap –A	**Nomer *port* :** 80, 135, 139, 443, 445, 902, 912, 1025, 1026, 1027, 1028, 1036, 2869, 3306, dan 10243 **Mac Address :** 48:5b:39:62:f2:0a ***Host*name** : Yoga-PC	Tidak ada
Nmap –sS	80, 135, 139, 443, 445, 902, 912, 1025, 1026, 1027, 1028, 1035, 1036, 2869, 3306, dan 10243	Tidak ada

Hasil simulasi serangan scanning *port* ditunjukkan pada tabel 5.2 dari hasil *scanning port* menggunakan *nmap*, komputer penyerang dapat mengetahui nomer *port* terbuka dari komputer target. Ketika simulasi serangan *scanning port* menggunakan "*nmap –A*" hasil dari scanning *port* adalah nomer *port* terbuka, alamat *MAC Address* dan *Hostname* dari komputer target. Saat simulasi serangan menggunakan *firewall* aktif, hasil yang diperoleh adalah komputer penyerang tidak bisa mengetahui nomor *port* yang terbuka dari komputer target, hasil ini diketahui saat simulasi serangan *scanning port* pesan yang diterima oleh komputer penyerang adalah "*all 1000 scanned ports in slims5-meranti (192.18.1.25) are filtered*".

5.2.2 Analisis Pengujian Simulasi Serangan *SYN flood*, *ICMP flood*, dan *UDP flood*

Analisis hasil simulasi serangan dari komputer penyerang kepada komputer target dapat dilihat pada Tabel 5.3 dengan keadaan *firewall* tidak aktif dan Tabel 5.4 dengan *firewall* dalam keadaan aktif

Tabel 5.3 Hasil simulasi serangan *SYN flood, ICMP flood,* dan *UDP flood firewall* tidak aktif

Serangan	*Resource*			*Capture* Packet *wireshark*	Akses Web Server
	CPU usage	*RAM usage*	*Network*		
SYN flood	39 %	53 %	1,35 %	2500-3000 paket	Gagal
ICMP flood	32 %	56 %	3,63 %	7000-10000 paket	Gagal
UDP flood	28 %	52 %	1,44 %	2000-3000 paket	Gagal

Tabel 5.3 menunjukkan hasil pengujian simulasi serangan menggunakan *firewall* tidak aktif, persentase nilai *CPU usage* pada komputer target yang paling besar adalah ketika simulasi serangan *SYN flood* yaitu sebesar 39 %. Untuk *resource RAM usage* dan *network* persentase nilai paling besar adalah saat simulasi serangan *ICMP flood* yaitu sebesar 56 % dan 3,63 %. Paket yang dapat di *capture wireshark* paling besar saat terjadi simulasi serangan *ICMP flood* yaitu sebesar 7000-10.000 paket. Sedangkan akses *web server* terjadi kegagalan ketika simulasi serangan *SYN flood*, *ICMP flood* dan *UDP flood*

Analisis selanjutnya adalah pengujian ketika simulasi serangan dengan *firewall* aktif menggunakan *embedded firewall system,* hasil dari pengujian ditunjukkan pad Tabel 5.4.

Tabel 5.4 Hasil simulasi serangan *SYN flood, ICMP flood,* dan *UDP flood firewall* aktif

Jenis Serangan	**Estimasi Paket perdetik (paket)**	***Capture* paket *wireshark* (paket)**	***CPU usage***	**RAM**	***Network***	**Akses Web Server**
SYN flood	1.000	50	4 %	48%	0,11%	Berhasil
	10.000	35	3 %	48%	0%	Gagal
ICMP flood	1 000	90	5 %	48%	0,01%	Berhasil
	10.000	90	4 %	48%	0,07%	Berhasil
UDP flood	1.000	10	3 %	46%	0,18%	Berhasil
	10.000	20	5 %	48%	0%	Gagal

Tabel 5.4 adalah hasil pengujian simulasi serangan dengan *firewall* aktif menggunakan 3 buah serangan yaitu *SYN flood, ICMP flood,* dan *UDP flood*. Jumlah paket perdetik yang digunakan saat simulasi serangan adalah 1.000 dan 10.000 paket. *Embedded firewall system* bekerja dengan melakukan filtering serangan *SYN flood*. *ICMP flood,* dan *UDP flood,* ketika simulasi serangan berlangsung *embedded firewall* langsung akan melakukan *drop* terhadap kiriman paket melebihi 20 paket per 1 detik yang terjadi secara terus menerus. Sehingga komputer target tidak akan menerima paket dalam jumlah besar. Hasil *capture wireshark* ditunjukkan pada Tabel 5.4 bahwa paket yang dapat ditangkap *wireshark* tidak melebihi 100 paket.

Pada *resource* komputer target yaitu *CPU usage, RAM usage* dan *network* saat simulasi serangan menggunakan *embedded firewall system* ditunjukkan pada tabel 5.4 semua *resource* dalam keadaan normal saat simulasi serangan. Karena persentase nilai dari *resource* komputer target tidak melebihi kondisi normal awal nilai *resource*. Sehingga *embedded firewall* system dapat melakukan *block* serangan sehingga melindungi *resourece* komputer target untuk bekerja tetap secara normal. Saat simulasi serangan *SYN flood* dan *UDP flood* dengan menggunakan *estimasi* paket perdetik 10.000 paket, persentase nilai *resource network* adalah 0%. Keadaan ini terjadi karena akses *web server* oleh komputer

user mengalami kegagalan, sehingga tidak ada proses yang terjadi pada *network utilization*

Analisa akses *web server* ketika simulasi serangan berlangsung terjadi gagal akses *web server* pada simulasi serangan *SYN flood* dan *UDP flood. Embedded firewall system* dapat bekerja dengan melakukan *drop* paket untuk melindungi *resource* komputer target. *Embedded firewall system* tidak bisa untuk melayani akses *web server* ketika simulasi serangan *SYN flood* dan *UDP flood* ketika estimasi paket perdetik sejumlah 10.000 paket.

Dari hasil analisis pertama dapat diambil kesimpulan bahwa simulasi serangan *scanning port* dapat dilakukan filtering oleh *embedded firewall system* sehingga komputer penyerang tidak bisa mengetahui informasi nomer *port* dari komputer target. Analisis kedua dapat diambil kesimpulan ketika *firewall* tidak aktif, simulasi serangan *SYN flood*, *ICMP flood*, dan *UDP flood* mempengaruhi *resource* komputer target, sehingga jika serangan terjadi secara terus-menerus akan membuat sistem *crash* atau *hang*. Simulasi serangan menggunakan *firewall* aktif, *embedded firewall system* mampu melakukan *filtering* serangan *SYN flood*, *ICMP flood*, dan *UDP flood* dengan melindungi *resource* komputer target. Sehingga *resource* komputer target tetap dalam keadaan normal. *Embedded firewall system* tidak bisa untuk melayani akses *web server* ketika simulasi serangan *SYN flood* dan *UDP flood* ketika estimasi paket perdetik sejumlah 10.000 paket.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian analisa yang telah dilakukan, maka diambil kesimpulan sebagai berikut :

1. Sistem keamanan jaringan *firewall* berdasarkan arsitektur *firewall packet filtering* dapat diimplementasikan pada *embedded system* menggunakan perangkat keras *raspberry pi*
2. Linux *iptables* dapat diimplementasikan sebagai keamanan jaringan untuk mencegah serangan *Scanning port dan Denial of Service/DoS (SYN flood, ICMP flood, UDP flood)* pada *embedded system* menggunakan perangkat keras *raspberry pi.*
3. Keamanan jaringan *firewall* berbasiskan *embedded system* dapat melakukan *filtering* ketika serangan *scanning port, SYN flood, ICMP flood*, dan *UDP flood* saat dilakukan simulasi dengan komputer penyerang. Hal ini berdasarkan hasil pengujian *scanning port*, komputer penyerang menerima pesan "*all 1000 scanned ports in slims5-meranti (192.18.1.25) are filtered"*. Pada simulasi serangan *SYN flood*, *ICMP flood*, dan *UDP flood* r*esource* komputer target dalam keadaan normal. Hal ini berdasarkan dari nilai persentase rata-rata *CPU usage* sebesar 4 %, *RAM usage* 47%, dan *network* 0,08%.
4. Sistem keamanan jaringan *firewall* berbasiskan *embedded system* berhasil memberikan akses *web server*, berdasarkan hasil pengujian simulasi serangan *ICMP flood* dengan estimasi paket perdetik 1.000 dan 10.000 paket. Pada simulasi serangan *SYN flood* dan *UDP flood* akses *web server* dapat dilakukan ketika estimasi paket 1.000 paket perdetik dan terjadi kegagalan ketika akses *web server* dengan estimasi 10.000 paket perdetik.

6.2 Saran

Saran yang diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut :

1. *Embedded firewall system* perlu untuk penambahan perangkat lunak *IDS (Intrusion Detection System*) sehingga didapatkan *firewall* yang dapat melakukan *block* terhadap serangan-serangan yang lebih bermacam dan dapat memberikan *warning* atau peringatan kepada administrator jaringan.
2. Implementasi algoritma sistem kecerdasan buatan kedalam *embedded firewall system* dapat membuat *datasheet* jenis-jenis serangan yang terjadi, sehingga dapat mengoptimalkan fungsi dari *firewall*.

DAFTAR PUSTAKA

.

[BBK-10] Bhuyan Monowar H, Bhattacharyya D K, Kalita J K., 2010. Surveying Port Scans and Their Detection Methodologies. *The Computer Journal*

[CSI-11] Computer Security Institute,2009-2011, New York, Amerika Serikat

[CAK-11] C.S Abhilash & Kumar Sunil ,2011, Mitigation of Distributed Denial of Service (DDoS) Threats. International Journal of Computer Application

[DCZ-00] D.B. Chapman & E.D.Zwicky, 2000, *Building Internet Firewalls*, O'Reilly & Associates, United States.

[IJE-08] Istiyanti Jazi Eko, Embedded System, Program Studi Elektronika dan Instrumentasi Jurusan Fisika FMIPA UGM

[JNK-12] J. Network, 2012, Application Note SECURING AND HARDENING NSM USING IPTABLES, Juniper Network Inc.

[HS-03] Heath, S. 2003. *Embedded-Systems Design, Second Edition*. Newnes.

[MZ-10] Meng, G. dan Zhen, L. 2010, Discussion on Internet Security Strategy of Embedded System, *2nd International Conference on Computer Engineering and Technology,* Vol. 3

[PNG-04] Purdy N. Gregor. 2004, Linux Iptables Pocket Reference, O'Reilly Media, Amerika Serikat.

[QB-12] Qasim B. ,2012, Mitigating Dos/DDoS Att*ACK* Using Iptables. International Journal of Engineering & Technology IJET-IJENS Vol: 12 No:03

[RB-02] Rahardjo B. ,2002, Keamanan Sistem Informasi Berbasis Internet. Insan Komunikasi, Bandung Indonesia

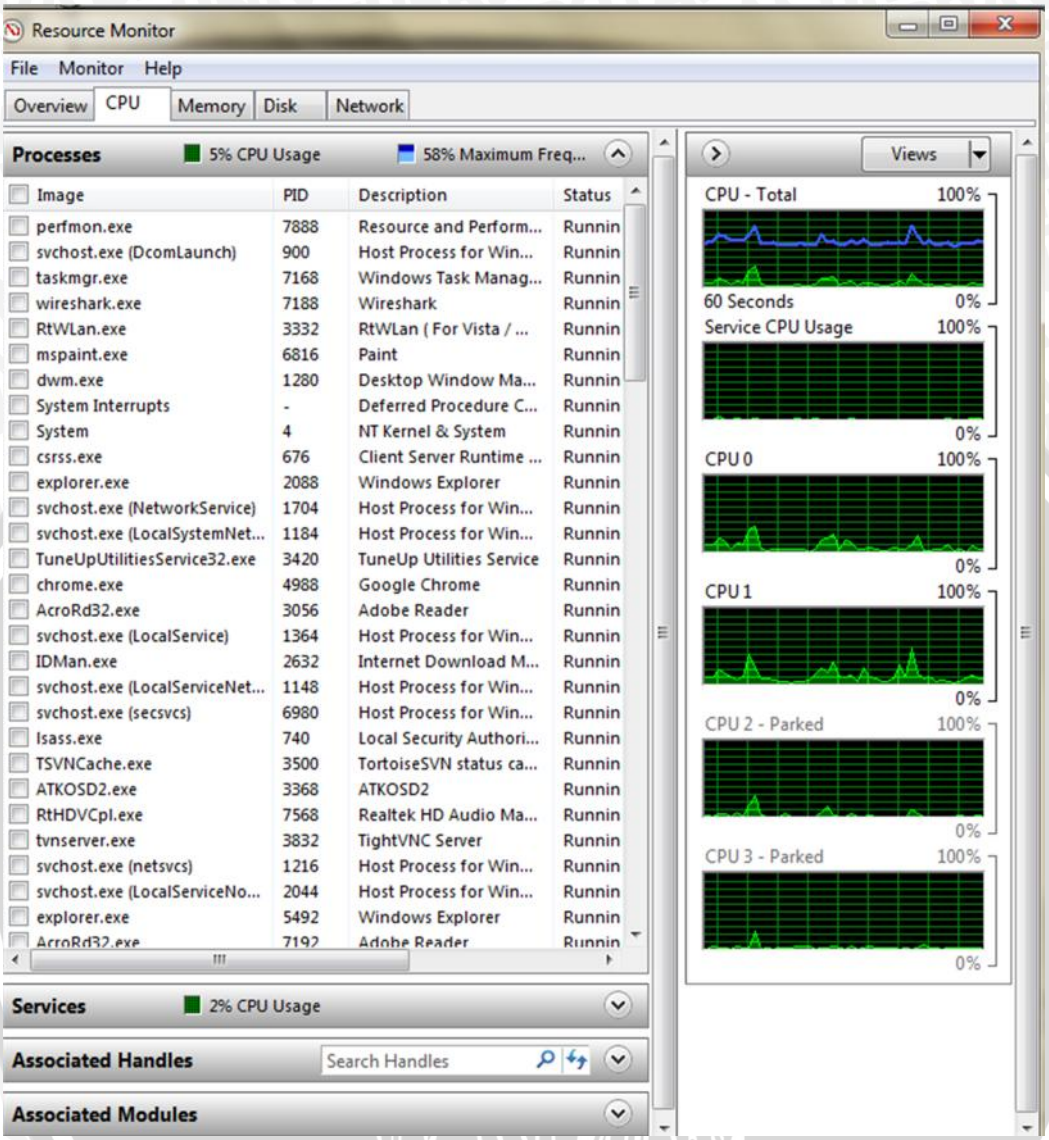
[RPF-13] Raspberry PI Foundation, 2013. Diakses melalui http://www.raspberrypi.org/faqs. [23 September 2013]

[RP-11] Rani P. Usha, Prasada D. Vara,. 2011, A Novel Implementation of ARM based Design Firewall to prevent SYN Flood Attack, International Journal of Computer Trends and Technology, Vol 2, Issue 2

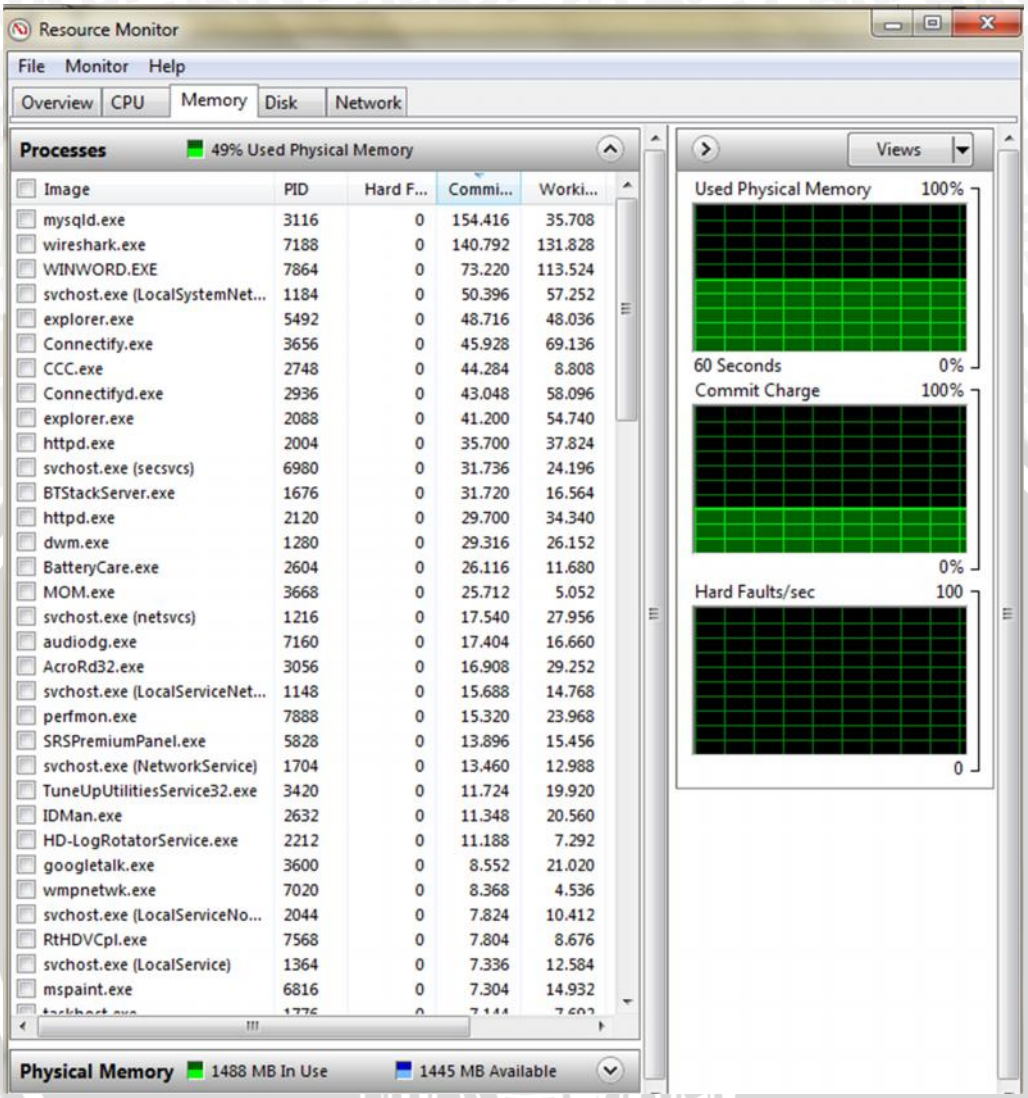
[SH-09] Scarfone K. & Hoffman P. ,2009, Guidelines on *firewall*s and firewall policy, Department of Commerce, Amerika Serikat

[SW-99] S. William. 1999, Cryptography and network security : principles and practice. Prentice-Hall, New Jersey Amerika Serikat

Lampiran 1. Keadaan Normal CPU usage Komputer Target



Lampiran 2. Keadaan Normal RAM usage Komputer Target



Lampiran 3. Keadaan Normal network Komputer Target