BAB 3. METODOLOGI PENELITIAN

3.1 Alat dan Bahan Penelitian

3.1.1 Alat Penelitian

PC sebagai *node* yang dilindungi dalam skenario ini, dikonfigurasi untuk menjalani *service*/layanan *web* dengan spesifikasi sebagai berikut:

1. Prosesor Intel Core 2 Duo 1.7 GHz
2. RAM 2 GB
3. 40 GB HDD
4. *Output device* (monitor)
5. *Input device* (mouse dan keyboard)
6. 100BASE-T Interface Card *On-Board* (FastEthernet 100MBps)
7. 100BASE-T Interface Card *Add-on* (FastEthernet 100MBps)

Sistem Operasi yang digunakan untuk menjalankan *webserver* adalah GNU/Linux Ubuntu Server Edition versi 9.04 sedangkan Sistem Operasi yang digunakan untuk menjalankan Honeyweb adalah GNU/Linux Slackware64 versi 13.0.

Perangkat lunak yang dimanfaatkan dalam penelitian ini, baik dimanfaatkan secara langsung maupun secara tidak langsung adalah:

1. libpcap
2. libnids-1.21

3. Snort-2.8.4.1-3
4. BlockIt-1.4.3
5. IPTables-1.4.1.1
6. MySQL-5.0
7. Honeyweb-0.4
8. Perl
9. AWK
10. Python
11. Apache2
12. PHP-5.12
13. php5-dev (P.E.A.R. & P.E.C.L.)
14. libssh2
15. dan *web browser*

3.1.2 Bahan Penelitian

Sedangkan bahan penelitian yang penulis gunakan, meliputi infrastruktur dan *tools exploit* (terutama protokol HTTP) diantaranya sebagai berikut:

1. nmap
2. Telnet
3. Ping
4. Worm
5. HTTP *Normal Request* (dari *web browser*)

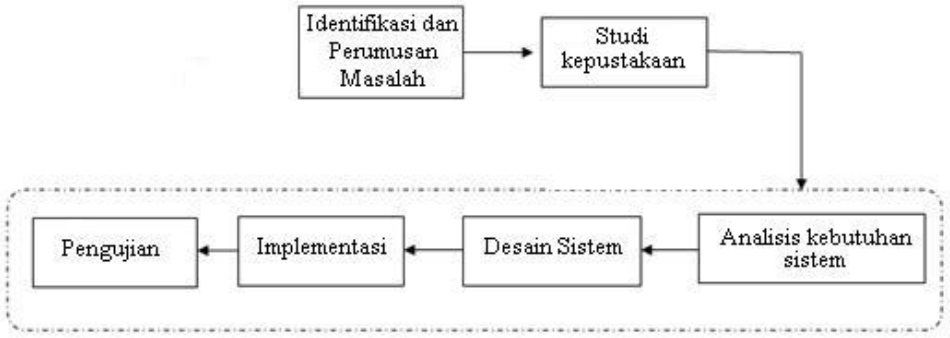
3.2 Desain Penelitian

Desain penelitian yang dilakukan dapat dilihat pada gambar 3.1 dibawah ini. Pada dasarnya penelitian ini terpisah antara pengembangan Honeyweb dan IPS. Keduanya merupakan sistem yang dapat berdiri sendiri tetapi akan dimanfaatkan untuk tujuan tertentu sehingga akan diintergrasikan.

Gambar 3.1 Desain Penelitian

3.2.1 Model Proses

Model proses yang digunakan oleh penulis dalam penelitian ini adalah sekuensial *linier* (Pressman, 2001), bisa dilihat seperti pada gambar 3.2 berikut.



Gambar 3.2 Tahap-Tahap Penelitian

3.2.1.1 *Identifikasi dan Perumusan Masalah*

Dalam sebuah penelitian berangkat dari suatu permasalahan, setelah masalah diidentifikasi dan dibatasi kemudian permasalahan tersebut dirumuskan. Perumusan masalah dalam penelitian ini dapat dilihat pada bagian pendahuluan dalam perumusan masalah.

3.2.1.2 *Studi Kepustakaan*

Metode pengumpulan data yang dilakukan penulis dengan mempelajari literatur yang berkaitan dengan teori keamanan jaringan, pembahasan mengenai masalah keamanan jaringan, dan pengumpulan informasi beberapa contoh metode penyerangan yang biasa dilakukan yang bisa mengancam keamanan jaringan.

3.2.1.3 ***Tahap Analisis Kebutuhan Sistem***

Pada tahap ini, dilakukan analisis terhadap penggunaan IPS sebagai prevensi intrusi untuk keamanan dan keberlangsungan layanan *web*. Diantaranya:

1. Identifikasi resiko-resiko keamanan yang dihadapi (*Assets*, *Vulnerabilities*, *Threats*). Mendefinisikan peran komputer dan aset jaringan, kelemahan sistem, dan ancamannya.
2. Mengumpulkan kebutuhan-kebutuhan *Network Administrator* terhadap masalah resiko-resiko keamanan tersebut.

3.2.1.4 ***Tahap Desain Sistem***

Pada tahap desain ini, diterjemahkan kebutuhan-kebutuhan yang akan dicapai pada tahap analisis ke sebuah bentuk desain/perancangan sebelum melakukan implementasi yang nyata terhadap sistem di jaringan. Hal-hal itu meliputi:

1. Perancangan topologi jaringan
2. Rencana peletakan *node* dalam topologi jaringan dan tugas-tugasnya terhadap keamanan jaringan

3.2.1.5 ***Tahap Implementasi***

Tahap ini adalah tahap untuk mulai membuat dan melakukan aktifitas dalam penelitian yang sesuai dengan perancangan pada tahap sebelumnya. Implementasi yang dilakukan dapat dijabarkan sebagai berikut:

1. Penjadwalan kegiatan penelitian

Penjadwalan kegiatan penelitian dilakukan dalam rangka memberikan target progres penelitian dalam waktu tertentu. Penjadwalan penilitan sistem keamanan jaringan dipecah menjadi beberapa sub-sub sistem. Sub-sub sistem tersebut memiliki fungsi tersendiri untuk satu tujuan sistem akhir, yaitu sistem terintegrasi *Intrusion Prevention System* dan Honeypot. Adapun sub-sub sistem yang dibangun dan fungsinya adalah sebagai berikut:

1) IDS

   Yaitu sistem independen untuk mendeteksi aktifitas intrusi terhadap suatu node yang dimonitor oleh sistem tersebut.

2) Honeypot

   Yaitu sistem independen untuk melakukan emulasi *service* palsu dan perekaman aktifitas *host* penyerang.

3) Modul yang dapat menjembatani IDS dengan *Firewall*. Modul tersebut akan memanfaatkan *alert* IDS sebagai inputan kemudian inputan tersebut dijadikan sebagai parameter untuk mengkonfigurasi *firewall* sistem. Kombinasi modul ini dengan IDS, akan menciptakan suatu *Intrusion Prevention System* (IPS).

2. Penentuan lingkup dan batasan implementasi

Dalam implementasi pembangunan pembangunan perangkat lunak mencakup bidang yang cukup luas. Agar implementasi ini dapat terlaksana maka perlu diberikan batasan yang jelas. Adapun lingkup implementasi yang akan diterapkan, penulis batasi sebagai berikut:

1) Sistem yang akan dibangun adalah sistem pencegah terhadap aktifitas intrusi pada jaringan komputer untuk melindungi server terhadap *request* yang tidak selayaknya untuk dilayani, sehingga pelayanan server akan tetap optimal. Sistem akan otomatis mencegah terjadinya intrusi dari *host* yang melakukannya dengan membatasi koneksi yang boleh dilakukan oleh *host* tersebut dalam satu waktu.
2) Objek yang diteliti adalah *webserver* dan SSH server.

3. Pemilihan, penginstallan, dan konfigurasi perangkat lunak serta perangkat keras untuk menjalankan sistem keamanan jaringan dan keamanan *host* (*webserver*). Untuk perangkat lunak, diutamakan perangkat lunak yang *Open Source*.

A. Kebutuhan Perangkat Lunak

Agar sebuah sistem dapat terealisasikan dengan baik maka diperlukan aplikasi atau *software* yang mendukung. Agar sistem dapat berjalan sesuai dengan yang telah direncanakan, maka dalam penelitian ini

penulis menggunakan beberapa aplikasi yang sebelumnya telah disebutkan dalam sub bab alat dan bahan penelitian. Diantaranya sebagai berikut:

1) Snort, yaitu aplikasi sensor intrusi yang dimanfaatkan oleh penulis sebagai pemilah antara *Normal* HTTP *Request* dengan *Malicious* HTTP *Request*. Untuk dapat melakukan itu, penulis telah meng-*update database signature ruleset* Snort yang didapatkan dari SnortVRT di internet. Dengan *database signature ruleset* terbaru diharapkan dapat membuat Snort yang dikembangkan dapat mengenal semua jenis pola-pola serangan terkini, khususnya intrusi pada protokol HTTP.
2) BlockIt, aplikasi *parser* IP *address* pada *log* Snort, sebagai jembatan antara IDS dan *Firewall* membentuk IPS (BlockIt *homepage*, 2010). BlockIt yang telah dikembangkan oleh penulis membuat IPS memiliki toleransi sebelum benar-benar memblokir *traffic* yang dianggap berbahaya dari *host* tertentu, tanpa pengaruh bagi *host* lainnya. BlockIt dikembangkan dalam bahasa pemrograman PERL.
3) MySQL sebagai RDBMS (*Relational Data Base Management System*) yang dengan keterbukaan dan kesederhanaanya, telah di-*support* banyak bahasa pemrograman. Termasuk bagi program BlockIt penulis, yang dikembangkan dengan bahasa pemrograman

PERL. Untuk berkomunikasi dengan *socket* MySQL server, tinggal memanfaatkan DBI, sebuah database *driver* bahasa pemrograman PERL. MySQL selain dimanfaatkan sebagai tempat referensi statistik pada program BlockIt, juga sebagai tempat memproses *query* untuk disajikan ke *web* sebagai *interface* berisi informasi yang dapat dengan mudah dipahami oleh *Network Administrator*. *Interface* tersebut penulis kembangkan dalam bahasa pemrograman PHP.

4) Honeyweb, sebagai aplikasi yang akan mengemulasi *service* HTTP dan mencatat aktifitas dari tiap *host* yang berinteraksi dengannya.

B. Kebutuhan Perangkat Keras

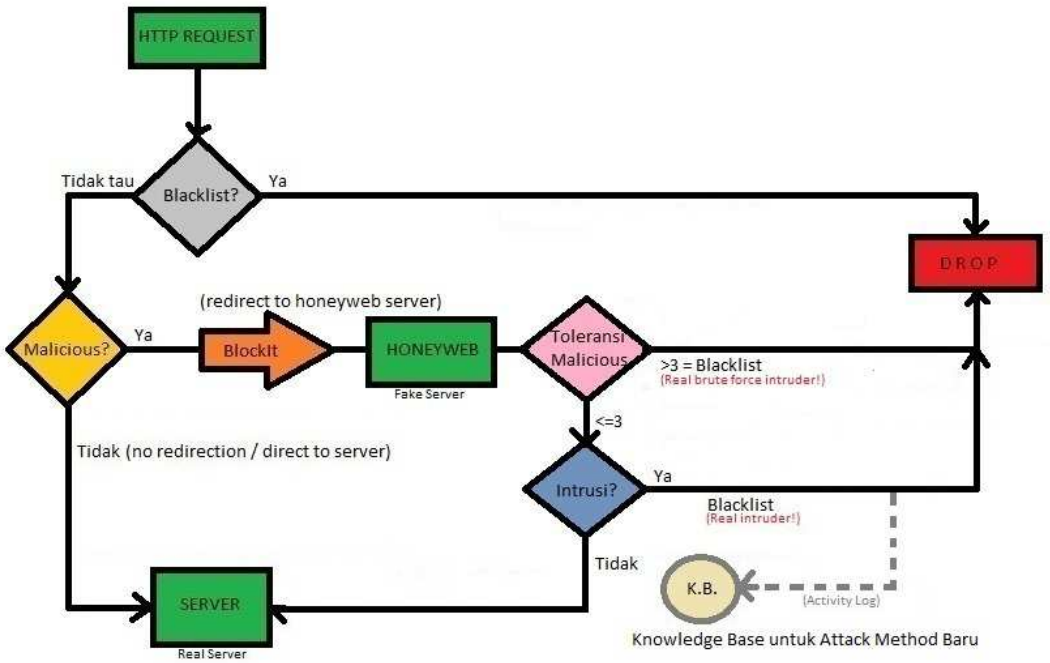
Dalam penelitian ini diperlukan seperangkat komputer dengan spesifikasi yang cukup untuk menjalankan aplikasi yang dibutuhkan dalam pembuatan sistem ini, sebelumnya telah disebutkan dalam sub bab alat penelitian.

4. Pengembangan perangkat lunak pendukung

Untuk tujuan yang ingin dicapai, akan dilakukan pengembangan lebih lanjut pada BlockIt sebagai aplikasi yang menjadi bagian penting dalam intergrasi IDS dan *Firewall*.

5. Pengintegrasian seluruh sistem keamanan dan mekanisme kerja global

Ketika semua sudah berjalan dengan baik dan dapat berdiri masing-masing secara independen, saatnya bagi penulis untuk mengkonfigurasi tiap entitas tersebut menjadi satu kesatuan mekanisme kerja IPS yang diharapkan. Sistem baru ini kemudian diberi nama SmartIPS. *Flowchart* mekanisme kerja SmartIPS diilustrasikan pada gambar 3.3 sebagai berikut.



Gambar 3.3 Flowchart Mekanisme Umum Kerja SmartIPS

Dari gambar dapat dijelaskan per bagian.

a. HTTP *Request* adalah semua jenis *request* dari *web browser client* terhadap *Server*.
b. Blacklist, pada dasarnya adalah semua *list* IP *address* yang targetnya adalah di-DROP pada *chain* INPUT *firewall*. Jadi ketika *header packet*

(IP *address*) melewati IPTables tidak cocok dengan *general matches* pada *chain* INPUT berarti bisa lewat seperti biasa. Tetapi yang cocok dengan *general matches chain* INPUT, berarti sudah termasuk *blacklist* IP *address* dan tidak akan sampai ke sensor Snort karena akan langsung di-DROP.

c. Jika tidak termasuk *blacklist*, maka sekarang adalah tahap bagi sensor Snort yang melakukan pemindaian *payload* data HTTP *request* tersebut. Pemindaian mereferensi ke *ruleset signature* Snort. Jika suatu *payload* tidak cocok dengan *ruleset signature* Snort maka langsung diteruskan ke *webserver* asli pada *host* A. Sebaliknya jika *payload* cocok dengan *ruleset signature* Snort, paket tersebut kemudian dianggap *malicious*. Snort menulis ke dalam *log*-nya.

d. BlockIt mem-*parsing* IP *address* dan *alertline* pada *log* Snort. Kemudian *data* yang di*parsing* dimasukkan ke *database*. Seluruh koneksi yang berasal dari IP *address* tersebut selanjutnya akan di-DNAT ke Honeyweb dan aktifitas *ping* akan diblokir untuk sementara. Dan dalam beberapa menit, BlockIt akan me-*release redirect* dan kembali mengizinkan protokol ICMP dari IP *address* tersebut dalam beberapa kali masa toleransi. Tetapi, *Network Administrator* bisa saja langsung mem-*blacklist* IP *address* tersebut jika diinginkan.

e. Honeyweb akan merekam aktifitas *host* yang pernah berinteraksi dengannya. Ini untuk dijadikan sebagai bahan pertimbangan seorang *Network Administrator* apakah suatu *client* melakukan kejahatan,

mengirim *bad request*, atau justru tidak termasuk tindakan jahat sama sekali. Jika semua *request-request client* tidak termasuk kejahatan yang mengeksploitasi HTTP, berarti ada kemungkinan terjadi *false positive* dengan terdeteksinya IP *address* tersebut yang dianggap oleh Snort melakukan kejahatan.

Selain untuk mereduksi *false positive* dengan analisis Honeyweb, bisa juga untuk merekam *request* aneh baru kiriman dari *client* dan mencari tahu apakah hal itu berbahaya atau tidak terhadap kelangsungan layanan *webserver*. Sehingga bisa menjadi bahan untuk *Network Administrator* melakukan penyelamatan dini pada server asli.

6. Pembuatan aplikasi berbasis *web* sebagai antarmuka antara perangkat sistem keamanan dengan *Network Administrator*.

Selanjutnya dikembangkan aplikasi berbasis *web* sebagai antarmuka pengolah *data* yang dihasilkan pada perangkat sistem keamanan. *Data* diolah menjadi informasi sebagai bahan analisis *Network Administrator*. Disamping sebagai panel untuk perangkat sistem keamanan, antarmuka berbasis *web* juga dapat memberikan kemudahan bagi *Network Administrator* dalam memantau server yang dilindungi.

3.2.1.6 ***Tahap Pengujian***

Proses untuk memastikan bahwa semua pernyataan sudah diuji, meliputi:

1. *Networking & Performance Testing* untuk menguji dampak dari instalasi perangkat keamanan, terhadap kemampuan pengaksesan informasi pada *webserver* (*latency* dan *throughput*).
2. *Security Performance Testing* untuk menguji kemampuan perangkat keamanan dalam mengenali serangan-serangan yang dilakukan dengan menggunakan bahan penelitian (*attack recognition* dan *repeatability*/*bruteforce*).

Selanjutnya akan mengarahkan penguji untuk menemukan kesalahan-kesalahan yang mungkin terjadi dan juga memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan dan diharapkan.