

## BAB II

### KAJIAN PUSTAKA DAN DASAR TEORI

#### 2.1 Kajian Pustaka

Sebuah server proxy dapat ditanamkan pada *embedded system*, dalam penelitian yang dilakukan oleh Ferrianto Gozali dan Muchammad Fachri yang berjudul “Perancangan dan Analisis Sistem Embedded Linux Firewall” tahun 2012, pada penelitian tersebut dilakukan pengembangan perangkat keamanan jaringan komputer berupa *packet filter* dan *server proxy* yang ditanam pada satu perangkat *embedded* jaringan komputer Aewin SCB-6971 menggunakan sistem operasi linux yang bersifat open source. Aplikasi yang digunakan adalah HAVP yang merupakan *transparent proxy server* tanpa cache dengan penyaringan anti virus. Setelah dilakukan pengujian fungsi *server proxy* pada embedded linux firewall didapatkan hasil sebagai berikut:

1. Fungsi *server proxy* pada sistem dapat melakukan *blocking* terhadap seluruh URL yang termasuk kedalam *blacklisted* URL.
2. *Proxy log* juga menunjukkan bahwa sistem dapat melakukan pemblokiran lebih dari 1 client yang berbeda dengan pengaksesan *blacklisted* URL yang berbeda pada waktu yang bersamaan dan dari *server proxy* log dapat diketahui 8 *internal client* yang berbeda tidak dapat mengakses URL yang di *block* oleh *server proxy*.

Penelitian mengenai fungsi *caching* server proxy dilakukan oleh Muhammad Hafizh dengan judul “Load Balancing dengan Metode Per Connection Classifier (PCC) Menggunakan Proxy Server Sebagai *Caching*”. Dalam penelitian tersebut peneliti membangun load balancing yang optimal dengan menggunakan proxy server server sebagai *cache* video. Aplikasi proxy server yang digunakan peneliti adalah squid. Hasil dari penelitian tersebut salah satunya adalah penggunaan proxy server yang diakses oleh 13 *client* sebagai *cache* memberikan optimasi pada jaringan internet dan dapat menghemat *bandwidth* sebesar 40.10%.

*Embedded system* menjadi piranti alternatif karena memiliki *suplay* listrik yang rendah sehingga dapat menghemat energi. Dalam penelitian yang berjudul “*Design and Implementation of An Embedded-Linux Downloading Server and Network Proxy*” tahun 2008 oleh Shengjie Liu, Xiangqian Ding, Haiyong Zheng dan Xiaodong Wang membuat sebuah server download dan network proxy berbasis embedded linux menggunakan perangkat embedded Gemini SL3512 yang terbukti *available* dan *effective* melindungi *enviroment* dan hemat energi. Prototipe yang dibuat oleh peneliti sudah diterapkan di laboratorium selama enam bulan hingga sekarang.

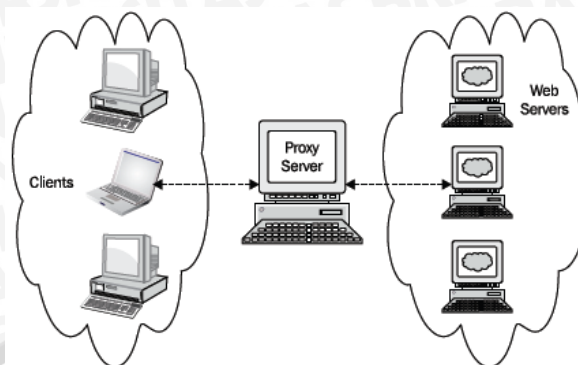
## 2.2 Dasar Teori

### 2.2.1 Server

Server merupakan komputer yang memiliki *speed* tinggi dengan RAM dan harddisk berkapasitas yang cukup besar, dan dilengkapi dengan *network card*. Kinerja server yang tinggi diperlukan dukungan prosesor dan peralatan pendukung lainnya untuk membangun sebuah server dengan kualitas baik karena server akan melayani permintaan secara terus-menerus. Selain itu, *operating system*, *application software* dan file data yang dibutuhkan juga dipilih untuk men-*support* kerja server. [Wagito, 2005]

### Server Proxy

*Server proxy* adalah server yang berguna sebagai perantara antara client dengan server gateway sebelum berhubungan ke Internet. Server proxy sering digunakan sebagai *cache*, filtering konten dan manajemen bandwidth. Tempat untuk menyimpan respon *request* ke dalam server proxy lokal adalah *cache*. Dengan adanya *cache* maka server proxy tidak perlu lagi mengirimkan *request* ke server tujuan, cukup mengirimkan *request* ke *cache* lokal untuk kemudian mengirimkan berupa respon kepada *client*.



Gambar 2.1 Server proxy  
Sumber : Saini, 2011

Selain digunakan sebagai *caching*, server proxy dapat digunakan sebagai filtering konten. Dengan filtering konten, server proxy menjadi sebuah filter permintaan data dari suatu situs yang diizinkan dikunjungi atau tidak diizinkan dikunjungi. Server proxy juga memiliki kemampuan untuk manajemen bandwidth. Manajemen bandwidth digunakan untuk membatasi bandwidth yang dipakai berdasarkan ACL (*Access Control List*) yang telah ditetapkan. [Cartealy, 2013]

### 2.2.2 Squid

Pada tahun 1996 oleh Duane Wessel dari National Laboratory for Applied Network Research, mengembangkan sebuah produk yang diberi nama Squid [Wessel, 2004]. Squid adalah *open source proxy caching* untuk Web mendukung HTTP, HTTPS, FTP. Squid dapat menghemat konsumsi bandwidth dan meningkatkan respon dengan adanya *caching* dan dapat menggunakan kembali *web page* yang sering dikunjungi. Squid memiliki fitur tambahan seperti akses kontrol, otorisasi, logging, distribusi konten, manajemen trafik dan masih banyak lagi. Squid dapat berjalan di lingkungan sistem operasi Windows dan sistem operasi yang berada dibawah lisensi GNU GPL. Saat ini squid banyak digunakan oleh beberapa perusahaan yang memiliki *embedded squid* dalam perangkat *firewall* mereka, disisi lain Squid digunakan dalam web proxy skala besar untuk mempercepat akses internet *broadband* dan *dial up*. [Squid, 2013]

### Access Control List (ACL)

Menurut Imam Cartealy dalam bukunya yang berjudul “Linux Networking” tahun 2013, *Access Control List* (ACL) digunakan untuk mengelompokkan informasi yang diterima dari *client* berdasarkan kriteria tertentu. Apabila ditemukan kecocokan maka Squid akan memberikan akses sesuai dengan yang telah ditentukan.

Format penulisan ACL Squid sebagai berikut.

Tabel 2.1 Format ACL

```
acl nama-acl tipe rangkaian
```

Sumber : Cartealy, 2013

Deklarasi ACL didahului oleh direktif `acl` dilanjutkan dengan nama ACL tersebut. Setiap ACL mengenali suatu tipe aktivitas tertentu. Penulisan ACL secara default bersifat case-sensitive, untuk merubahnya menjadi case-insensitive dipergunakan opsi `-i`. Tipe ACL yang sering dipergunakan dapat dilihat pada tabel berikut.

Tabel 2.2 Tipe ACL

Type ACL	Keterangan dan Deklarasi
Src	Digunakan untuk mendefinisikan alamat IP asal. alamatIP/netmask ... alamatIP1-alamatIP2
Dst	Digunakan untuk mendefinisikan alamat IP tujuan. alamatIP/netmask ...
Port	Digunakan untuk mendefinisikan port tujuan. port ... port1-port2
Arp	Tipe ini digunakan untuk mendefinisikan alamat MAC dari klien. Tipe ini akan tersedia apabila Squid dibangun dengan opsi <code>-enable-arp-acl</code> .

dstdomain	Tipe ini digunakan untuk mendefinisikan domain tujuan.  namaDomain
Time	Tipe ini digunakan untuk mendefinisikan waktu dan mengakibatkan ACL yang bersangkutan hanya aktif pada waktu tertentu saja. Tipe ACL ini dapat mengenali hari dalam seminggu yang didefinisikan sesuai dengan singkatannya yaitu S ( <i>Sunday</i> ), M ( <i>Monday</i> ), T ( <i>Tuesday</i> ), W ( <i>Wednesday</i> ), H ( <i>Thursday</i> ), F ( <i>Friday</i> ), A ( <i>Saturday</i> ), dan juga waktu yang didefinisikan dengan rentang waktu h1:m1-h2:m2 dalam satuan 24 jam.  [hari] [h1:m1-h2:m2]
url_regex	Tipe ini digunakan untuk menyediakan kontrol akses terhadap bagian dari URL berdasarkan regex-nya.  [-i] ^http://...
proxy_auth	Tipe ini digunakan untuk otentikasi proxy.  Username ...

Sumber : Cartealy, 2013 dan Wagito, 2005

Dalam penggunaannya, tipe ACL dikombinasikan dengan *access list* agar dapat mengendalikan akses tertentu terhadap internet. Adapun cara menuliskan pernyataan *control access* menurut Wagito dalam bukunya “Jaringan Komputer:Teori dan Implementasi Berbasis Linux” adalah sebagai berikut.

Tabel 2.3 Penulisan *access list*

<pre>controlaccess allow deny ACLtype ... controlaccess allow deny [!]ACLtype ...</pre>
---

Sumber : Wagito, 2005

Aturan yang diberlakukan pada penulisan pernyataan dalam konfigurasi squid sebagai berikut.

- Aturan *control access* dapat memberikan kata kunci allow atau deny diikuti nama ACL.
- *Control access* dapat mengandung lebih dari satu aturan.

- *Control access* diperiksa berdasarkan urutan penulisan (dibaca dari atas ke bawah) sehingga penempatan urutan sangat berpengaruh apakah aturan tersebut dapat dijalankan atau tidak.

*Access Control* yang sering dipergunakan dapat dilihat pada tabel berikut.

Tabel 2.4 *Access Control*

Access Control	Keterangan
http_access	Ijin akses HTTP.
http_reply_access	Ijin bagi <i>client</i> untuk menerima jawaban permintaan.
icp_access	Ijin <i>query</i> menuju <i>cache</i> dengan ICP
miss_access	Ijin bagi <i>client</i> untuk meneruskan <i>cache</i> yang hilang ke <i>cache server proxy</i> .
no_cache	Mendefinisikan tanggapan yang tidak disimpan dalam <i>cache</i> .
redirector_access	Mengendalikan permintaan yang dikirim melalui pool pengarah
ident_lookup_access	Mengendalikan permintaan yang memerlukan tipe ACL Ident.
always_direct	Mengendalikan permintaan yang selalu dilewatkan secara langsung pada asal server.
never_direct	Mengendalikan permintaan yang tidak boleh dilewatkan secara langsung pada asal server.
snmp_access	Mengendalikan akses SNMP pada <i>cache</i> .
broken_posts	Menentukan squid untuk mengirim pasangan CRLF setelah <i>body</i> permintaan PUT/POST.
cache_peer_access	Mengendalikan permintaan yang dapat diteruskan ke <i>cache</i> lain.

Sumber : Wagito, 2005

### Object Cache

Optimasi object cache dalam squid hal yang perlu diperhatikan adalah kapasitas memori dan kapasitas cache. `cache_mem` merupakan direktif yang digunakan untuk menentukan jumlah maksimum pemakaian memori dalam menyimpan objek yang populer. Untuk menentukan objek yang akan disimpan atau dihapus, ada beberapa parameter yang dapat digunakan adalah sebagai berikut.

- **LRU (*Least Recently Used*)**  
Parameter LRU menyimpan objek berdasarkan umur objek. Objek yang berumur lama akan dihapus dari cache.
- **LFUDA (*Least Frequently Used with Dynamic Aging*)**  
Parameter LFUDA menyimpan objek yang populer tanpa memperhatikan ukuran objeknya. Objek-objek yang tidak populer akan dihapus dari cache.
- **GDSF (*Greedy-Dual Size Frequency*)**  
Parameter GDSF menyimpan objek berukuran kecil. Objek yang berukuran besar lebih diutamakan untuk dihapus dari cache.

### 2.2.3 Wireless LAN (WLAN)

Wireless LAN adalah jaringan yang berbasis wireless. Jaringan wireless ini memanfaatkan gelombang radio elektromagnetik dalam mengirimkan data kepada komputer komputer yang berada pada sebuah *Local Area Network* tanpa melibatkan fisik kabel jaringan. [Rafiudin, 2006]

#### Wireless Access Points (WAP)

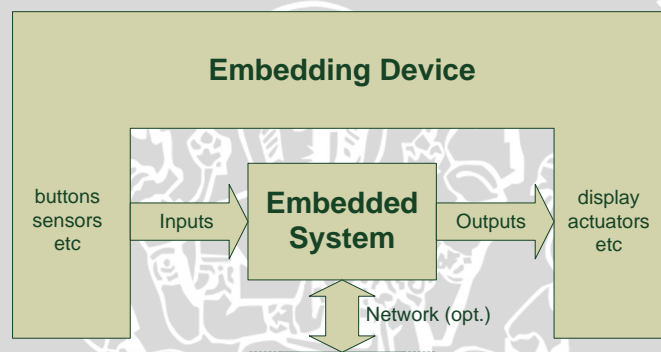
Konfigurasi dasar WLAN, transmitter atau receiver radio dinamakan *Wireless Access Point* (WAP) dikoneksikan ke jaringan kabel Ethernet (IEEE 802.3) yang bersifat tetap. *Access point* bekerja receiving, buffer dan transmitter data antara WLAN dengan infrastruktur jaringan kabel.

Sebuah *access point wireless* dapat melayani satu grup user pada waktu yang bersamaan (IEEE 802.11). *Range* yang dapat dijangkau oleh *access*

*point* adalah beberapa ratus meter tanpa halangan yang berarti. [Rafiudin, 2006]

#### 2.2.4 Embedded System

*Embedded system* adalah sistem komputer yang dirancang khusus untuk tujuan tertentu demi meningkatkan fungsi suatu mesin. Sesuai artinya, “*embedded*” yang berarti “mencocokkan”, maka bagian yang dicocokkan meliputi peranti keras dan bagian mekanis lain. Hal ini berlawanan dengan sistem umum seperti yang kita kenal pada *Personal Computer* (PC) yang bisa menjalankan banyak perintah sekaligus tergantung pada pemrogramannya.



Gambar 2.2 Skema *Embedded System*

Sumber : [Heryana, 2012]

*Embedded system* terdiri atas pemroses (processor) dan perangkat lunak. Perangkat lunak tersebut harus tersimpan secara permanen dan harus terdapat penyimpanan data sementara (RAM) untuk menyimpan data selama proses runtime berlangsung. Selain memory, *embedded system* membutuhkan *input* dan *output*. [Heryana, 2012]

*Embedded system* ini didedikasikan untuk perintah spesifik, seperti rancangan desain untuk mengoptimasi mesin, pengurangan ukuran dan biaya produk, atau meningkatkan performa kerja. Sesungguhnya banyak produk massal yang merupakan hasil dari *embedded system*. Sebut saja MP3 Player, hingga peranti instalasi skala besar seperti lampu lalu lintas. Bahkan juga sistem pengontrol pembangkit listrik nuklir.

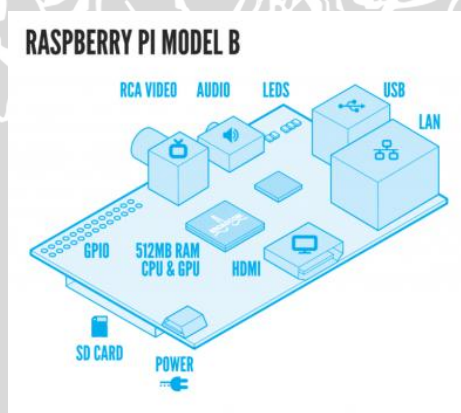


*Embedded system* adalah sistem dengan ciri-ciri sebagai berikut [Magdalena, 2007]:

- Mempunyai computing power. Dengan kata lain dilengkapi dengan sebuah processor.
- Bekerja di lingkungan luar ruangan IT. Jadi kemungkinan besar tidak dilengkapi dengan AC dan menghadapi gangguan dari luar seperti getaran dan debu.
- Memiliki tugas yang spesifik. Beda dengan PC atau Server yang relatif lebih multi purpose.

### Raspberry Pi

Raspberry Pi adalah sistem komputer yang besarnya seperti kartu kredit yang dapat dihubungkan dengan perangkat display dan keyboard. Benda kecil ini dapat bekerja layaknya sebuah desktop PC. Juga mampu menampilkan video dengan resolusi cukup tinggi.



Gambar 2.3 Raspberry Pi Model B  
Sumber : raspberrypi

Spesifikasi Raspberry Pi Model B sebagai berikut:

Tabel 2.5 Spesifikasi Raspberry Pi Model B

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, I <sup>2</sup> C bus, SPI bus with two chip selects, +3.3 V, +5 V, ground
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 g (1.6 oz)
Operating Systems	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS

Sumber : raspberrypi