BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Menurut jurnal yang berjudul "Rancang Bangun *Server Learning Management System* menggunakan *Load Balancer* dan *Reverse Proxy*" yang dibuat oleh Sirajuddin, Dkk, menyebutkan

Internet yang semakin mudah dijangkau masyarakat ditandai dengan jumlah pengguna yang semakin besar. Terjadinya lonjakan trafik yang tidak mampu diatasi oleh *web server* menyebabkan *web server* tersebut mengalami kelebihan beban (*overload*) dan membuat *website* menjadi tidak dapat diakses. Masalah tersebut juga terjadi pada *server* situs *e-learning*. *Web Server* sebagai komponen utama sebuah situs pembelajaran *online* (*e-learning)* seringkali mendapat beban yang semakin berat seiring dengan bertambahnya *user* yang mengaksesnya. Dibutuhkan *web server* yang handal untuk mengatasi peningkatan jumlah *user* dan kompleksitas konten *e-learning* seperti penambahan bahan berupa kuis dan konten *video* yang semakin banyak dijumpai.

Ketika pengakses situs *e-learning* secara bersama-sama sedikit maka cukup dipakai satu *web server* saja, tetapi ketika jumlah pengakses bertambah besar maka ini menyebabkan terjadinya peningkatan *request* dari pengakses (klien) yang tidak mampu ditangani oleh *web server* sehingga mengakibatkan *web server* mengalami kelebihan beban.

UIB Repository(c)2015

Permasalahan tersebut umumnya dihadapi oleh *web server* yang hanya memiliki mesin tunggal atau single *web server*. Salah satu solusi untuk permasalahan tersebut adalah dengan menggunakan *load balancer. Load balancing* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi.

2.1.2 Menurut jurnal yang berjudul "Analisa Implementasi *Reverse Proxy* pada *Web Server* Menggunakan *Squid* dan *Lighttpd*" oleh Moehammad Iqbal, dkk, menyebutkan

Web adalah sistem pengiriman dokumen tersebar yang berjalan di internet. Umumnya sebuah *website* terdiri dari banyak halaman *web*. Halaman *web* itu sendiri disusun dari rangkaian teks, gambar, gambar bergerak, atau mungkin suara, dan efek - efek visual lain. Seluruh halaman *web* dan rangkaian *file* pendukungnya harus diletakkan pada sebuah *web server* sehingga pengguna internet di seluruh dunia dapat mengaksesnya. Pada saat mengakses internet *user* mengharapkan kecepatan akses yang memadai sehingga kebutuhan mereka akan suatu dokumen dapat segera terpenuhi. Kestabilan dalam pengaksesan juga sangat diperlukan dalam kenyamanan waktu *download file* dari internet. Banyaknya permintaan akses ke sebuah *website* dapat membebani sebuah *web server* sehingga terkadang sebuah *web server* terasa sangat lambat dalam memberikan

UIB Repository(c)2015

respon. Bahkan saat permintaan telah mencapai batasan limit yang dapat ditampung maka sebuah *web server* tidak akan memproses permintaan tersebut.

Secara garis besar jalur akses internet dengan *proxy* akan melewatkan semua *request* pada *proxy* terlebih dahulu. *Proxy* akan mengevaluasi *request* tersebut dan menyimpannya pada *cache.* Dan jika ada *user* lain yang mengirimkan *request* dan pada *cache* telah terdapat informasi tersebut maka *proxy* akan langsung mengirimkan kepada *user* tanpa harus *download* dari *web server*. Salah satu metode untuk meningkatkan akselerasi dari sebuah *web server* adalah dengan menggunakan *reverse proxy*. Oleh karena itu dibutuhkan suatu konfigurasi tertentu agar fungsi dari sebuah *web server* dalam melayani *request* dapat berjalan dengan maksimal.

2.1.3 Menurut jurnal yang berjudul "Analisa Perbandingan Unjuk Kerja Sistem Penyeimbang Beban *Web Server* dengan *Haproxy* dan *Pound Links*" oleh Dite Ardhian, menyebutkan

Perkembangan teknologi *web* harus didukung pula dengan infrastruktur yang baik dan berkecepatan tinggi agar berbagai kebutuhan tersebut dapat terpenuhi dengan baik. Peningkatan permintaan pada situs, menyebabkan *web server* sibuk menjawab permintaan klien dan terkadang pula *web server* mengalami *server* mati atau kegagalan *server* jika terlalu banyak permintaan sehingga *web server* tidak dapat menanganinya. Hal ini

UIB Repository(c)2015

akan merugikan pihak yang mempercayakan situsnya pada suatu *web server*, karena situs-situs tersebut tidak dapat diakses untuk sementara waktu. Parameter seperti waktu tanggap, kemerataan beban dan keberhasilan pelayanan merupakan parameter yang menentukan kinerja suatu *web server*.

Salah satu mekanisme untuk lebih mengoptimalkan penggunaan sumber daya yang ada adalah dengan mekanisme *load balancing* yang akan menyeimbangkan beban dari seluruh *server* yang ada pada *cluster*. Melalui penggunaan mekanisme penyeimbang beban, permintaan koneksi HTTP yang datang pada *web server* akan ditangani oleh *server* dengan kondisi yang paling baik, dimana pemilihan tersebut dilakukan dengan sistem prioritas *web server*.

Ada beberapa metode penyeimbang beban *web server*. Salah satu metode penyeimbang beban *web* dapat menggunakan *reverse proxy*, yaitu *proxy* yang memiliki maksud yang berkebalikan dengan *forward proxy* yaitu menjadi perantara *user-user* di internet terhadap akses ke *web-web server* yang berada pada LAN, sehingga seolah-olah *user* di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya *user* di internet mengakses *web-web server* yang terdapat di LAN melalui *reverse proxy* tersebut. Tentunya komputer yang bertindak sebagai *reverse proxy* tersebut memiliki IP publik sehingga dapat diakses dari Internet. Salah satu contoh perangkat lunak *opensource* yang menggunakan metode *reverse proxy* adalah *Haproxy* dan *Pound Links*, kedua perangkat lunak

UIB Repository(c)2015

tersebut dapat digunakan sebagai penyeimbang beban *web server* dan *caching web server* sehingga akses ke *web* menjadi cepat dan beban *web server* menjadi seimbang.

2.1.4 Menurut jurnal yang berjudul "Perancangan dan Implementasi *Proxy Server* dan Manajemen *Bandwidth* Menggunakan *Linux Ubuntu Server*" oleh Sandy Arjuni, menyebutkan

Internet banyak diterapkan di perkantoran, namun kadang terdapat *users* yang mengakses situs terlarang dan menghabiskan *bandwidth* dengan melakukan *download* maupun *streaming* sehingga koneksi pengguna lain akan menjadi lambat. Selain itu, seringkali *users* melakukan *browsing* dengan mengunjungi situs yang sama berulang-ulang. Hal ini tentunya akan menjadikan penggunaan *bandwidth* yang tidak efektif dan diperlukan solusi untuk masalah ini.

Proxy server ini merupakan sebuah sistem yang dibangun sebagai solusi dan dapat berfungsi untuk menangani *caching, bandwidth management* dan pemblokiran situs secara bersamaan. Dengan menyimpan data yang relatif statis dalam *cache* dan mengambilnya dari *cache* bila diperlukan, maka kita dapat menghemat waktu yang diperlukan dalam menghasilkan data.*(Wawan Bahtiar,2007). Proxy* memiliki fungsi *filtering* yang tentunya dapat berguna untuk memblokir *content* dan situs tertentu. Selain itu, *Proxy Server* juga dimanfatkan untuk mengatur pembagian

UIB Repository(c)2015

bandwidth agar tidak terjadi ketimpangan kecepatan koneksi Internet antara satu *client* dengan *client* lainya.

2.1.5 Menurut jurnal yang berjudul "Rancang Bangun *Internet Sharing* di Jaringan Lokal pada Amik Sigma" oleh Molavi Arman, menyebutkan

Internet diperlukan guna menunjang kebutuhan individu, masyarakat, organisasi untuk mendapatkan informasi terkini dan membantu dalam menunjang fasilitas pekerjaan. Pada saat ini untuk mendapatkan layanan Internet dibutuhkan berlangganan dengan *Internet Service Provider* dengan melakukan penyewaan sebuah nomor *IP Address* Publik, untuk penyewaan *IP Address* Publik cukuplah membebani dari sisi ekonomis karena harga yang cukup mahal,dengan begitu untuk medapatkan efisiensi cukup menyewa 1 *IP Address* Publik yang akan digunakan sebagai *Proxy Server* yang akan menjembati ke komputer *workstation* atau klien *Proxy server* dapat digunakan didalam berbagai macam sistem operasi, seperti *Windows* dan *Linux*.

Jika *Proxy* digunakan dalam sistem operasi *Windows* maka akan memiliki biaya yang cukup mahal baik dari segi perangkat keras maupun perangkat lunak. Sehingga jika digunakan di skala jaringan yang kecil cukup membebani dari sisi ekonomis. Salah satu solusinya dengan merancang *server proxy*, *server proxy* memiliki biaya yang lebih rendah bila dilihat dari segi penggunaan perangkat keras dan perangkat lunak. Ini dikarenakan *server proxy* adalah *proxy* yang berbasiskan komputer *PC*

UIB Repository(c)2015

pribadi yang mempunyai kemampuan untuk melakukan proses dengan dukungan dariperangkat lunak yang ada didalam Sistem Operasi *Linux* dari komputer tersebut. Salah satu dari sistem operasi yang mendukung proses tersebut adalah sitem operasi *Linux*, dengan menggunakan sistem operasi *Linux* yang mendukung protokol-protokol serta aplikasi-aplikasi jaringan seperti aplikasi *proxy* itu sendiri. Selain itu pula, *Linux* merupakan sistem operasi yang *free* atau bebas sehingga akan menekan dari segi biaya pengeluaran dalam membangun perancangan dan penerapan didalam jaringan di AMIK SIGMA.

.

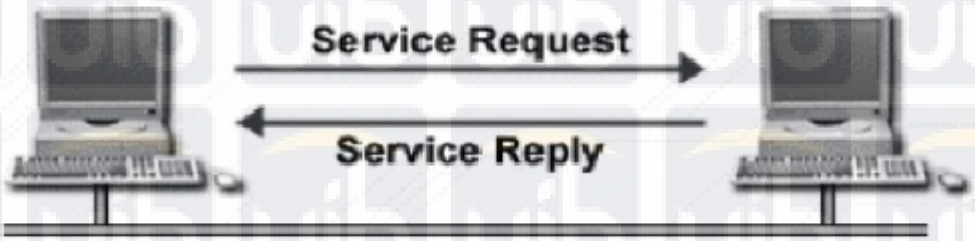
2.2 Landasan Teori

2.2.1 Pengertian Jaringan Komputer

Jaringan komputer merupakan sekumpulan komputer berjumlah banyak yang terpisah-pisah akan tetapi saling berhubungan dalam melaksanakan tugasnya *(Umi Proboyekti).*

Hingga saat ini jaringan komputer sudah menjadi kebutuhan umum masyarakat, dan karena itu pemahaman dasar tentang jaringan komputer diperlukan, terutama bagi orang-orang yang berkecimpung dalam dunia teknologi informasi.

UIB Repository(c)2015



Gambar 2.1 Jaringan Komputer

Peer to Peer merupakan jenis jaringan yang tidak memerlukan *server* secara khusus, karena komputer yang terhubung pada jaringan dapat bertindak sebagai *server* ataupun *client.* Contoh: hubungan komputer ke komputer. (Wahidin : Hal 1 : 2007)

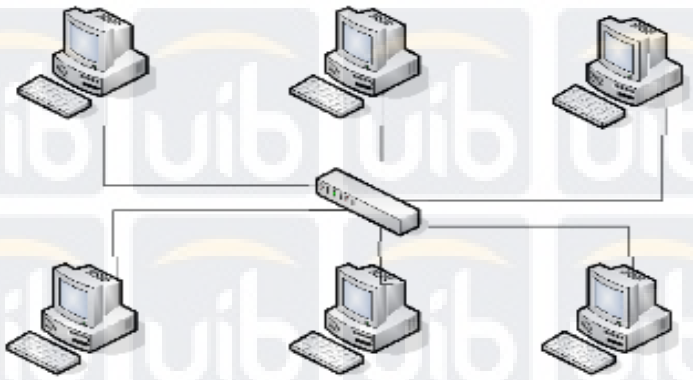
2.2.2 Jenis – Jenis Jaringan Komputer

2.2.2.1 Local Area Network (LAN)

LAN adalah sebuah jaringan komputer dengan jangkauan area yang terbatas dan hubungan fisik antar komputer saling berdekatan. Misalkan Jaringan komputer di sebuah kantor, jaringan komputer di sebuah ruangan kerja (laboratorium).

Menurut Syafrizal (2005), *Local Area Network* (LAN), merupakan jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan, seperti sebuah kantor pada sebuah gedung, atau tiap - tiap ruang pada sebuah sekolah maupun universitas.

UIB Repository(c)2015



Gambar 2.2 Topologi Jaringan LAN

Keuntungan LAN :

1. Pertukaran *File*, data antar komputer dapat dilakukan dengan mudah (*file sharing).*
2. Penggunaan printer dapat dilakukan oleh semua *user (printer sharing)*
3. Proses *backup data* menjadi lebih mudah dan cepat.

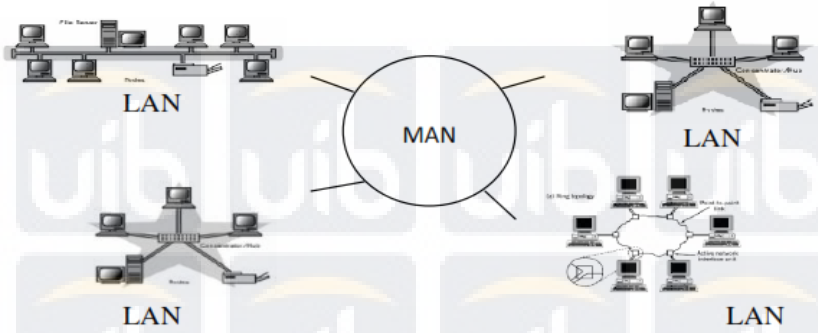
Kerugian LAN :

1. Area jaringan hanya sebatas satu gedung atau ruangan
2. *IP Address* komputer LAN tidak bisa di akses dari jaringan WAN

2.2.2.2 Metropolitan Area Network (MAN)

MAN adalah sekumpulan jaringan komputer menghubungkan beberapa buah jaringan – jaringan LAN kedalam lingkungan area yang lebih besar, sebagai contoh : jaringan pada BANK (*system online Perbankan*). Setiap bank memiliki kantor pusat dan kantor cabang.

UIB Repository(c)2015



Gambar 2.3 Topologi Jaringan MAN

Keuntungan MAN :

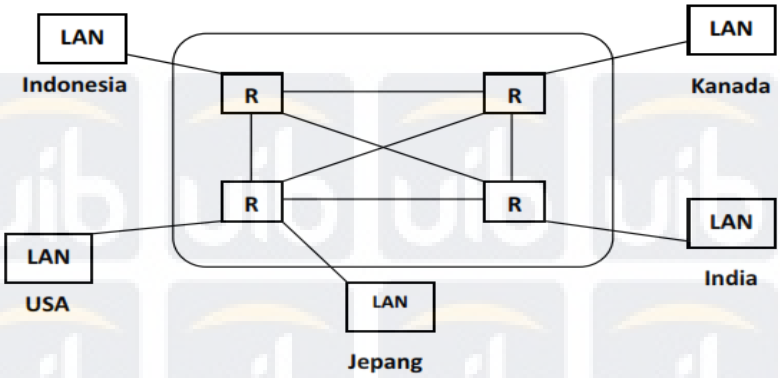
1. *Server* kantor pusat dapat berfungsi sebagai pusat data dari kantor cabang
2. Transaksi yang *Real Time* (data di *server* pusat di update saat itu juga, contoh ATM Bank untuk wilayah Nasional)

Kerugian MAN

1. Installasi Infrastrukturnya tidak mudah
2. Rumit jika terjadi *trouble* jaringan (*Network trouble shooting*)

2.2.2.3 Wide Area Network (WAN)

WAN adalah jaringan komputer dengan jangkauan area geografi yang paling luas, antar Negara, antar benua bahkan keluar angkasa (sebagai contoh jaringan internet yang menggunakan sistem koneksi satelit).



ambar 2.4 Topologi Jaringan WAN

Keuntungan WAN :

1. Penggunaan kartu kredit di seluruh dunia
2. Pengambilan uang dengan jaringan internasional (ATM Internasional)
3. *Pooling data* dan *up date data* antar kantor dapat dilakukan setiap hari pada waktu yang ditentukan

Kerugian WAN :

1. Biaya operasional mahal
2. Susah dalam Installasi insfratrukturnya

2.2.3 Topologi Jaringan

Topologi jaringan dalam telekomunikasi adalah suatu cara menghubungkan perangkat telekomunikasi yang satu dengan yang lainnya sehingga dapat membentuk suatu jaringan.

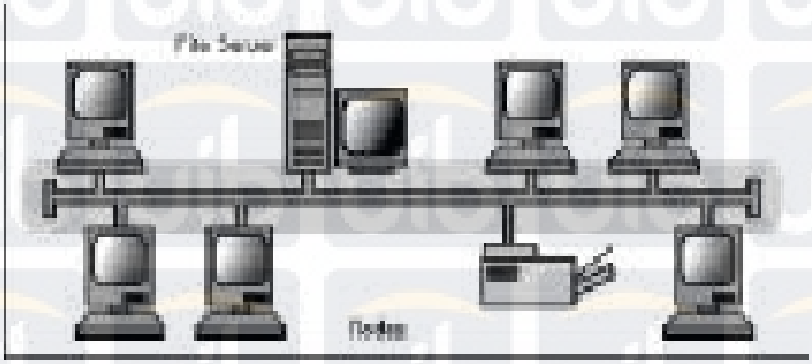
Topologi jaringan terbagi lagi menjadi dua, yaitu topologi secara fisik (*physical topology*) dan topologi secara logika (*logical topology*). Topologi secara fisik menjelaskan bagaimana susunan dari kabel dan komputer dan lokasi dari semua komponen jaringan. Sedangkan topologi secara logika menetapkan

bagaimana informasi atau aliran data dalam jaringan. (Yudhistira Adi Nugraha P, dkk : 2008)

Jenis – jenis topologi dapat dibagi sebagai berikut:

2.2.3.1 Topologi BUS

Topologi ini mempunyai bentuk, satu kabel utama menghubungkan ke tiap saluran tunggal komputer (membentuk huruf T), kecuali simpul disalah satu ujung kabel utama, yang hanya terhubung ke saluran. Komputer yang ingin terhubung kejaringan dapat mengkaitkan dirinya dengan mentap *Ethernetnya* sepanjang kabel.



Gambar 2.5 Topologi BUS

Kelebihan Topologi BUS

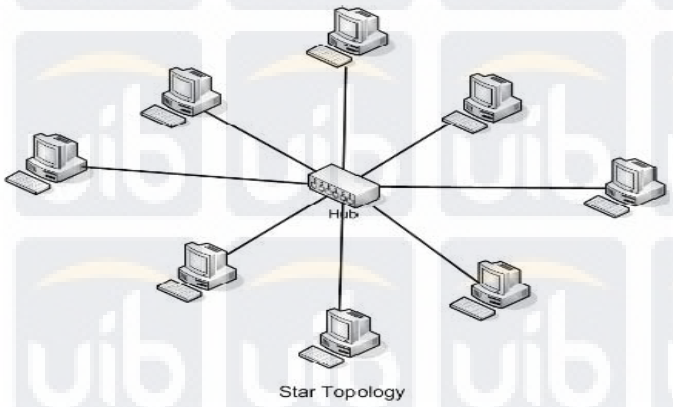
1. Topologi sederhana
2. Kecepatan pengiriman data lebih cepat, karena data berjalan searah.
3. Lebih mudah dan murah jika ingin menambah atau mengurangi jumlah *node*, karena yang dibutuhkan hanya kabel dan konektornya saja

Kelemahan Topologi BUS

1. Jika lalulintas data yang diolah terlalu besar dapat mengakibatkan kemacetan.
2. Jika salah satu *node* mengalami kerusakan, maka jaringan tidak dapat beroperasi.

2.2.3.2 Topologi STAR

Topologi ini merupakan bentuk topologi jaringan yang berupa *konvergensi* dari *node* tengah ke setiap *node* atau pengguna dengan menggunakan bantuan alat lain untuk mengkoneksikan jaringan computer. Contoh alat yang di pakai disini adalah *hub*, *switch*, dll.



Gambar 2.6 Topologi *Star*

Kelebihan Topologi Star

1. Jika salah satu terminal rusak, maka terminal lainnya tidak mengalami gangguan.
2. Arus lalu lintas informasi data lebih optimal.
3. Memudahkan admin dalam mengelola jaringan

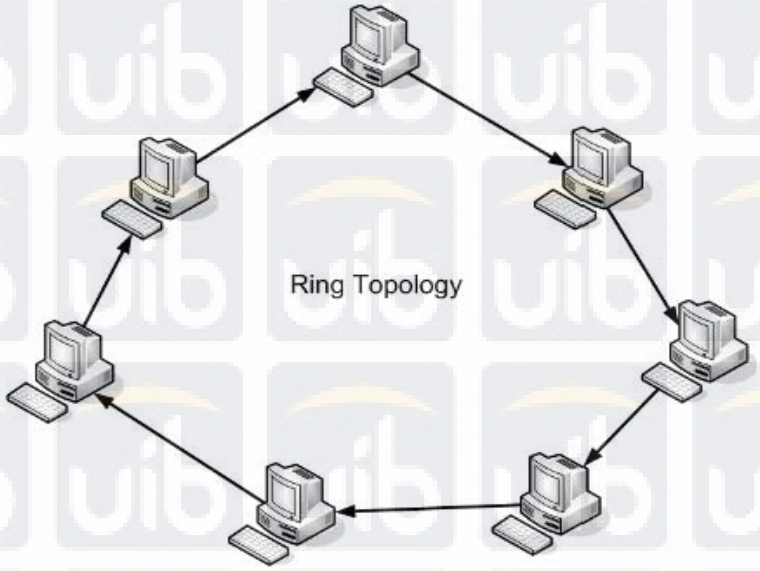
UIB Repository(c)2015

Kekurangan Topologi Star

- Jumlah terminal terbatas, tergantung dari *port* yang ada pada hubungan lalulintas data yang padat dapat menyebabkan jaringan bekerja lebih lambat.

2.2.3.3 Topologi RING

Topologi *Ring* atau yang disebut juga dengan topologi cincin adalah topologi jaringan dimana setiap komputer yang terhubung membuat lingkaran.



Gambar 2.7 Topologi Ring

Kelebihan Topologi RING

1. Aliran data mengalir lebih cepat karena dapat melayani data dari kiri atau kanan dari *server*.
2. Data mengalir dalam satu arah sehingga terjadinya *collision* dapat di hindarkan.

UIB Repository(c)2015

Kekurangan Topologi RING

1. Penambahan terminal/*node* menjadi lebih sulit bila *port* sudah habis.
2. Sulit untuk melakukan konfigurasi ulang.

2.2.4 Media Transmisi Data

2.2.4.1 Kabel UTP

Jaringan komputer sekarang menggunakan kabel UTP (*Unshielded Twisted Pair*) dengan standar 100 *base - TX Fast Ethernet*. Kabel UTP terdiri dari 4 pasang kawat berulir (*twistwed Pair Wire*) sehingga pada kabel itu semuanya terdapat 8 kawat. Setiap pasang kawat ini di beri kode warna, yang pertama warna penuh (Biru, Jingga, Hijau atau cokelat) dan pasangannya yang berulir seputar yang utama tadi dengan warna putih dan strip warna yang sesuai dengan pasangannya.

Gambar 2.8 Kabel UTP

2.2.4.2 Kabel Koaksial

Kabel ini mempunyai sifat-sifat sebagai berikut:

- Paling populer digunakan pada *Local Area Network* (LAN)

- Memiliki bandwidth yang lebar, sehingga bisa digunakan untuk komunikasi *broadband* (*multiple channel*)
- Ada bermacam-macam jenis kabel coax seperti kabel TV, thick, ARCnet, dan thin coax.

Gambar 2.9 Kabel Koaksial

2.2.5 Hardware

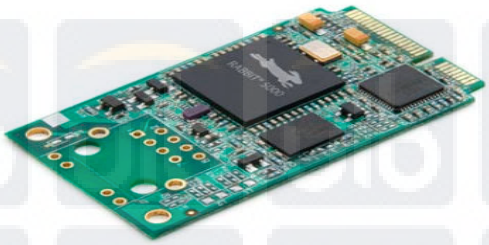
2.2.5.1 Server

Secara umum *server* adalah komputer program yang menyediakan *service* kepada komputer program lain yang sama ataupun berbeda. Dalam model programming *client/server*, *server* adalah program yang menunggu dan memenuhi permintaan dari *client* program yang sama atau berbeda.

2.2.5.2 NIC (Network Interface Card)

NIC Adalah computer *circuit board or card* yang dipasang dalam komputer (*server* maupun *workstation*) sehingga komputer dapat dihubungkan kedalam jaringan. Dilihat dari jenis *interfacenya* umumnya terbagi dua yaitu PCI dan ISA. Terdapat juga beberapa *card* diperuntukkan khusus untuk laptop atau *notebook* dengan *socket* PCMCIA. Sering disebut dengan *ethernet card*, *Network card.*

UIB Repository(c)2015



Gambar 2.10 Network Interface Card

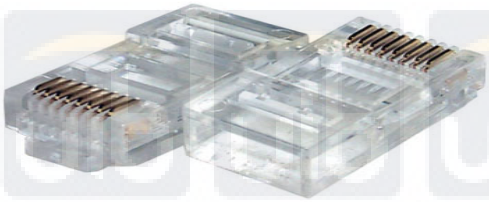
2.2.5.3 Hub dan Switch

Hub dan *Switch* Merupakan konsentrator yang membagi sinyal data bagi *network interface card* (NIC).

Gambar 2.11 Hub dan Switch

2.2.5.4 RJ-45

Pada ujung-ujung kabel CAT 5 ini dipasangkan konektor yang dikenal sebagai konektor RJ-45 (RJ dari kata *'Registered Jack')*. Konektor RJ-45 ini mirip dengan konektor pada kabel telepon (RJ-11). Bila pada kabel telepon menggunakan tiga pasang kawat, maka kabel network ini empat pasang. Untuk melihat urutan kawat-kawat yang dipasang pada konektor RJ-45, harus melihat dengan memegang 'klip' konektor ini di bagian bawah, agar 'lubang'-nya (tempat memasukkan kabel) menghadap Anda.



Gambar 2.12 RJ 45

2.2.5.5 Crimp tool

Crimp tool / Crimping tool adalah alat untuk memasang kabel UTP ke konektor RJ-45/RJ-11. Bentuknya macam-macam ada yang besar dengan berbagai fungsi, seperti bisa memotong kabel, mengupas dan lain sebagainya. Ada juga yang hanya diperuntukan untuk crimp RJ-45 atau RJ-11 saja.

Gambar 2.13 Crimp Tool

2.2.6 Software

2.2.6.1 Ubuntu 12.04

Ubuntu adalah sistem operasi turunan dari distro Linux jenis Debian unstable (sid), Ubuntu merupakan *project* untuk komunitas, yang bertujuan untuk menciptakan sebuah sistem operasi beserta dengan paket aplikasinya yang bersifat *free* dan *open source*, karena Ubuntu mempunyai prinsip untuk selamanya bersifat gratis *(free of charge)* dan tidak ada tambahan untuk versi

enterprise edition. Ubuntu memiliki berbagai kelebihan distribusi debian diantaranya adalah:

a. Pemaketan (*Packaging*)
b. Pemilihan aplikasi yang luas (*Application choice*)
c. Siklus pembaharuan dilakukan secara rutin (*Updates*)
d. Dikenal stabilitas dan kualitasnya terutama di sisi *Server* (*Stability and quality*).

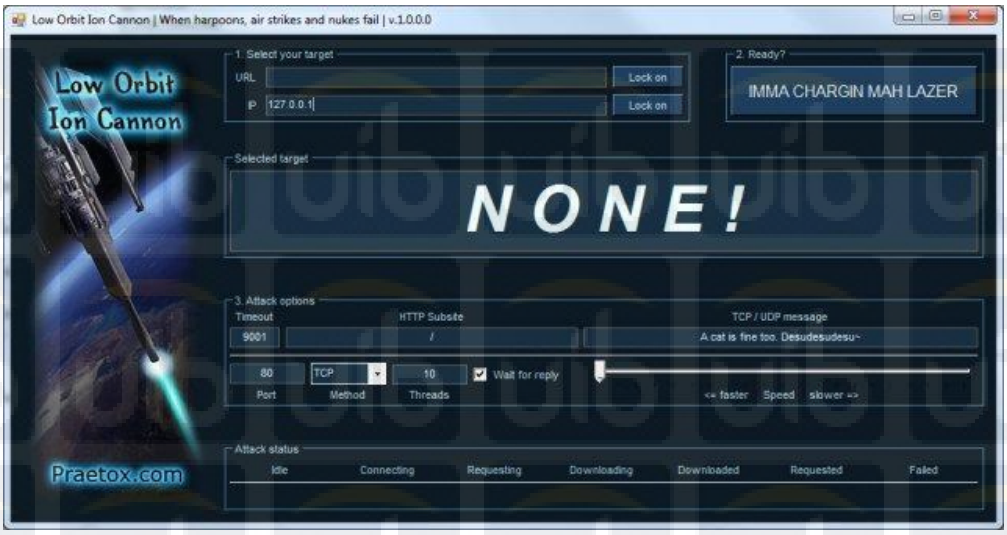
2.2.6.2 Apache Benchmark

Apache Benchmark (ab) adalah sebuah program komputer yang berisi perintah *single-threaded* untuk mengukur kinerja dari HTTP *server web*. Yang awalnya hanya dirancang untuk menguji Apache HTTP *Server*, yang cukup umum digunakan untuk menguji *web server*.

2.2.6.3 Low Orbit Ion Cannon

Low Orbit Ion Cannon (Loic) adalah sebuah *software open source* yang digunakan untuk pengujian jaringan rumit dan aplikasi *Denial-of-service* (DoS), yang ditulis dalah C#. Loic pada awalnya dikembangkan oleh *Praetox Technologies*, namun kemudian di *release* kedalam *domain public* dan sekarang sudah dapat *host* dibeberapa *platform open source.*

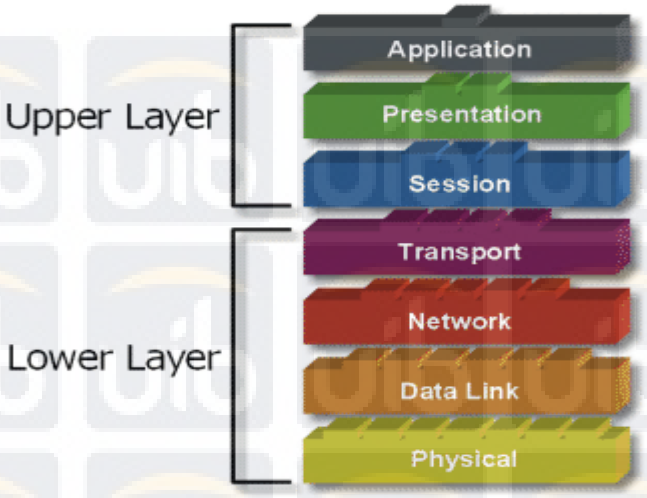
UIB Repository(c)2015



Gambar 2.14 Tampilan Low Orbit Canon

2.2.7 Model Layar OSI

OSI memberikan pandangan yang abstrak dari arsitektur jaringan yang di bagi dalam ketujuh lapisan. Model ini di ciptakan berdasarkan sebuah proposal yang di buat oleh *International Standard Organization* (ISO) sebagian langkah awal menuju standarisasi protokol internasional yang di gunakan pada berbagai layer. Ketika data di *transfer* melalui jaringan, sebelum data tersebut harus melewati ketujuh *layer* dari satu terminal, mulai dari *layer* Aplikasi sampai *layer physical*, kemudian di sisi si penerima, data tersebut melewati layer *physical* sampai *application*.



Gambar 2.15 Model OSI Layer

Model OSI terdiri atas 7 *layer*, yaitu sebagai berikut:

a. *Application*
b. *Presentation*
c. *Session*
d. *Transport*
e. *Network*
f. *Data Link*
g. *Physical*

Fungsi masing-masing dari tiap layer pada OSI adalah sebagai berikut:

1. Application

Application layer menyediakan jasa untuk aplikasi pengguna, *layer* ini bertanggung jawab atas pertukaran informasi antara program komputer, seperti program *e-mail* dan *servis* lain yang berjalan di jaringan seperti *server printer* atau aplikasi komputer lainnya.

Contoh Protokol yang berada dalam lapisan ini adalah **HTTP, FTP, SMTP, dan NFS.**

2. *Presentation*

Presentation layer bertanggung jawab bagaimana data dikonversi dan di format untuk *transfer data*. Contoh konversi format text ASCII untuk dokumen, .GIF dan .JPG, untuk gambar layer ini membentuk kode konversi, translasi data, *enkripsi* dan *konversi.*

3. *Sesion*

Session layer menentukan bagaimna dua terminal menjaga, memelihara dan mengatur koneksi. Bagaimana mereka saling berhubungan satu sama lain. Contoh:

a. *NFS (Network File System)*
b. *SQL (Structured Query Language)*

4. *Transport*

Transport layer bertanggung jawab membagi data menjadi segmen, menjaga koneksi logika *"end – to _ end"* antar *terminal*, dan menyediakan penanganan *error (error handling).*

Contoh:

a. *Service-point addressing*
b. *Segmentasion dan reassembly*

5. *Network*

Network layer bertanggung jawab menentukan alamat jaringan, menentukan *rute* yang harus diambil selama perjalanan, menjaga antrian tafik di jaringan.

Contoh dimana *Network* memiliki dua tugas pokok diantaranya:

a. Logical addressing

b. Routing

6. *Datalink*

Data link layer menyediakan *link* untuk *data*. Memaketkannya menjadi *frame* yang berhubungan dengan "*hardware*" kemudian diangkut melalui media komunikasinya dengan kartu jaringan, mengatur komunikasi *layer physical* antara sistem koneksi dengan penaganan *error*.

Contohya :

a. Topologi bus

b. Topologi ring

7. *Physical*

Physical layer bertyanggung jawab atas proses data menjadi bit dan mentransfernya melalui media (seperti kabel) dan menjaga koneksi fisik antar sistem. Contoh topologi jaringan dan pengkabelan.

UIB Repository(c)2015

2.2.8 *Internet Protocol* (IP)

Alamat IP (*Internet Protocol Address* atau sering disingkat IP) adalah deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer *host* dalam jaringan Internet. Agar komputer bisa berkomunikasi dengan komputer lainnya, maka menurut aturan TCP/IP, komputer tersebut harus memiliki suatu *address* yang unik. Alamat tersebut dinamakan IP *address*. IP *Address* memiliki format sbb: *aaa.bbb.ccc.ddd*. Contohnya: 167.205.19.33.

2.2.8.1 Kelas IP Address

a. **Kelas A** , pada kelas A 8 bit pertama adalah *network Id*, dan 24 bit selanjutnya adalah *host Id*, kelas A memiliki *network Id* dari 0 sampai 127.

b. **Kelas B** , pada kelas B 16 bit pertama adalah *network Id*, dan 16 bit selanjutnya adalah *host Id*, kelas B memiliki *network id* dari 128 sampai 191.

c. **Kelas C**, pada kelas C 24 bit pertama adalah *network Id*, dan 8 bit selanjutnya adalah *host Id*, kelas C memiliki *network id* dari 192 sampai 223.

d. **Kelas D**, IP kelas D digunakan untuk *multicasting*, yaitu penggunaan aplikasi secara bersama-sama oleh beberapa komputer, dan IP yang bisa digunakan adalah 224.0.0.0 – 239.255.255.255.

e. **Kelas E**, memiliki *range* dari 240.0.0.0 – 254.255.255.255, IP ini digunakan untuk eksperimen yang dipersiapkan untuk penggunaan IP address di masa yang akan datang.

UIB Repository(c)2015

Agar lebih jelas, silakan lihat tabel di bawah ini:

Tabel 2.1 *Range* dari masing – masing kelas IP Address

Kelas	Dari	Sampai
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.255

Selain pengelompokan alamat diatas, alamat IP juga dibagi atas *Private IP* dan *Public IP*. *Private IP* adalah alamat yang digunakan untuk pengalamatan LAN *(Local Area Network*) dan tidak dikenal oleh internet, sedangkan *Public IP* adalah alamat yang digunakan untuk pengalamatan internet. Adapun *range* dari Private IP pada setiap kelas adalah seperti pada tabel di bawah ini:

Tabel 2.2 *Range* IP Private

Kelas	Dari	Sampai
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.32.255.255
C	192.168.0.0	192.168.255.255

2.2.7.2 Pengalokasian IP Address

IP Address terdiri atas dua bagian yaitu *network ID* dan *host ID*. *Network ID* menunjukkan nomor *network*, sedangkan *host ID* mengidentifkasikan *host* dalam satu *network*. Pengalokasian *IP address* pada

UIB Repository(c)2015

dasarnya ialah proses memilih *network ID* dan *host ID* yang tepat untuk suatu jaringan. Tepat atau tidaknya konfigurasi ini tergantung dari tujuan yang hendak dicapai, yaitu mengalokasikan IP address se-efisien mungkin.

2.2.9 Server

Server yang akan dibahas dibawah ini hanya mencakup penelitian yang digunakan oleh penulis, adapun diantaranya:

2.2.9.1 Webserver

Web server merupakan *software* yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman *web* yang umumnya berbentuk dokumen HTML (Tyan Suhesti, 2008).

Cara kerja dari *web server* sebenarnya sangat mudah kita pahami. Contoh paling mudah seperti berikut, kita akan membuka sebuah halaman *website*, yang biasanya berupa URL http://www.google.com. Kita akan mengetikkan URL tersebut di *browser* kemudian menekan tombol enter, tanpa kita ketahui proses yang terjadi di belakang layar atau di dalam *browser* itu sendiri, maka akan muncul halaman *website* di layar monitor kita.

Proses yang akan terjadi pada *browser* adalah *browser* akan membentuk koneksi dengan *web server*, meminta halaman *website* dan menerimanya. *Web server* kemudian mengecek permintaan tersebut apakah tersedia atau tidak. Apabila tersedia, maka *web server* akan mengirimkan *data* kepada *browser*. Apabila permintaan tidak ditemukan atau terjadi *error* maka *web server* akan

UIB Repository(c)2015

mengirimkan pesan *error* kepada *browser*. Pembentukan koneksi, permintaan data, penerimaan data dari *browser* ke *web server* diatur dalam sebuah kode **RFC2616**. RFC2616 mencantumkan status *web server* dalam bentuk kombinasi tiga angka yang memiliki arti berbeda-beda. Status ini muncul di peramban saat kita mengakses *web server* tertentu.

- **Apache**

Apache merupakan Aplikasi *Web Server* yang paling banyak digunakan saat ini, selain dapat berjalan di banyak *Operating System* seperti Windows, Linux dan lainya, Apache juga bersifat *Open Source*. Secara *default* Apache menggunakan Script PHP dan menggunakan MySQL sebagai Data Base nya dan semua itu dapat di gunakan secara gratis.

Ada beberapa Aplikasi Apache yang dapat berjalan di *Windows* yaitu :

- ✓ Apache2 triad
- ✓ XAMPP
- ✓ WAMPP
- ✓ DLL

a. Kelebihan Web Server Apache

- Apache termasuk dalam kategori *freeware*.
- Apache mudah sekali proses instalasinya jika dibanding *web server* lainnya seperti NCSA, IIS, dan lain-lain.
- Mampu beroperasi pada berbagai *platform* sistem operasi.
- Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.

UIB Repository(c)2015

b. Kekurangan Web Server Apache

- *Web server* Apache tidak memiliki kemampuan mengatur *load* seperti IIS, sehingga akan terus mem-fork proses baru hingga nilai MaxClients tercapai atau hingga batas yang diizinkan oleh OS. Ini tentunya menguntungkan penyerang karena habisnya RAM akan lebih cepat tercapai.
- Apache tidak memproses karakter kutip dalam string Referrer dan User-Agent yang dikirimkan oleh *Client.* Ini berarti *Client* dapat memformulasi inputnya secara hati hati untuk merusak format baris *log akses*.
- Terganggunya proses *upload* data, yang bisa menyebabkan *software* salah dalam menerjemahkan ukuran data yang masuk.

2.2.9.2 Proxy Server

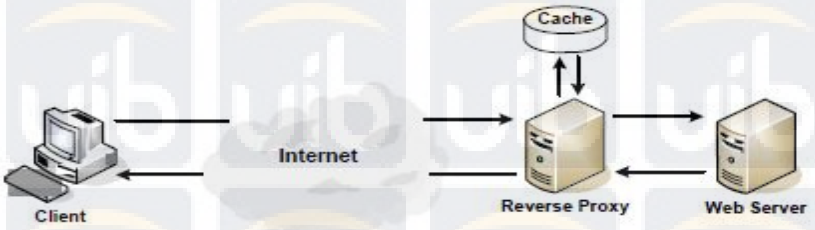
Proxy Server adalah sebuah komputer *server* atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan *request* terhadap *content* dari internet atau intranet. *Proxy Server* bertindak sebagai *gateway* terhadap dunia internet untuk setiap komputer klien. *Web server* yang menerima *request* dari *proxy server* akan menginterpretasikan *request-request* tersebut seolah-olah *request* itu datang secara langsung dari komputer *client*, bukan dari *proxy server* (Yudha Purwanto, dkk). *Proxy server* digunakan untuk melakukan *accounting* dan *caching*. *Accounting* berfungsi untuk membatasi *user* yang dapat mengakses internet sementara *caching* berfungsi untuk mengakumulasi dan menyimpan *file* yang paling banyak di *request* oleh *user* dari internet.

UIB Repository(c)2015

Dengan adanya *proxy server* maka penggunaan koneksi internet akan lebih efisien karena hanya *user-user* tertentu saja yang dapat mengakses internet dan tidak terjadi pengulangan *download file* yang sama. Hal tersebut dikarenakan *cache* pada *proxy server* mungkin telah menyimpan *file* yang di *request* oleh user sehingga *proxy* akan segera mengirimkan langsung kepada *user*. *File* yang disimpan pada *cache* akan disegmentasi pada beberapa tingkatan berdasarkan ukuran *file* serta frekuensi *akses*.

- **Reverse Proxy**

Reverse proxy adalah salah satu metode untuk mengkonfigurasi *proxy server* yang fungsinya mengurangi beban pada *web server*, daripada mengurangi *bandwidth* dihulu jaringan pada sisi *client*. *Reverse proxy* adalah *proxy server* yang dipasang di antara Internet dan situs *web* dan menangani semua lalu lintas sebelum dapat mencapai *server web*. Biasanya, *reverse proxy* digunakan di depan *web server*. Semua koneksi yang datang dari internet dialamatkan ke *web server* akan dialirkan melalui *proxy server*, yang dapat menangani *request* dari diri sendiri atau seluruh *request* atau sebagian ke *main web server*.



G

Gambar 2.16 Cara Kerja Reverse Proxy

2.2.9.3 Client-Server

Jenis jaringan ini terdapat komputer yang bertindak sebagai *server* dan komputer yang berperan sebagai *client (workstation).* Secara normal komputer *server* dapat mengontrol sepenuhnya komputer *client. Server* biasanya mempunyai kemampuan yang lebih besar dari pada *client.* (Wahidin : Hal 1 : 2007).

2.2.10 Denial of Service (DoS)

Denial of Service merupakan serangan yang dilancarkan melalui paket-paket tertentu. Bentuk serangan yang digunakan dalam pengujian adalah *ping attack* dan *syn attack*. Menurut Kurose (2010), *Denial of Service* merupakan serangan yang dapat membuat informasi atau data menjadi tidak tersedia untuk *user* yang *valid*. *Web server*, *e-mail server*, *DNS server*, dan jaringan institusi merupakan tujuan dari penyerangan DOS.

Berikut adalah beberapa efek yang akan ditimbulkan apabila jaringan terkena *DOS Attack*, diantaranya:

1. *Traffic Floading*. Teknik penyerangan yang ditimbulkan oleh jenis serangan ini adalah dengan membanjiri lalu lintas jaringan dengan banyak data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk kedalam sistem jaringan.

UIB Repository(c)2015

2. *Request Floading*. Teknik penyerangan yang ditimbulkan oleh jenis serangan ini adalah dengan Membanjiri jaringan dengan banyak *request* terhadap sebuah layanan jaringan yang disedakan oleh sebuah *host* sehingga *request* yang datang dari pengguna terdaftar tidak dapat dilayani oleh layanan tersebut.

3. Mengganggu komunikasi antara sebuah *host* dan kliennya yang terdaftar dengan menggunakan banyak cara, termasuk dengan mengubah informasi konfigurasi sistem atau bahkan perusakan fisik terhadap komponen dan server.

Beberapa contoh Serangan DoS lainnya adalah:

1. Serangan *Buffer Overflow*, mengirimkan data yang melebihi kapasitas sistem, misalnya paket ICMP yang berukuran sangat besar.
2. Serangan SYN, mengirimkan data TCP SYN dengan alamat palsu.
3. Serangan *Teardrop*, mengirimkan paket IP dengan nilai *offset*yang membingungkan.
4. Serangan *Smurf*, mengirimkan paket ICMP bervolume besar dengan alamat *host* lain.
5. *ICMP Flooding*.

UIB Repository(c)2015