



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 OSI Reference Model

Open System Interconnection (OSI) model adalah sebuah permodelan yang dibuat sebagai referensi untuk membantu orang-orang memahami komunikasi data di antara dua buah sistem yang terhubung melalui jaringan komputer. Dalam penggambaran *OSI reference model*, jaringan komputer dibagi menjadi tujuh lapisan (*layers*) yang memiliki fungsi masing-masing dan berperan untuk menopang lapisan-lapisan yang berada di atasnya (Simoneau, 2006).

Tabel 2.1 *Seven Layers of OSI Reference Model*

<i>Application Layer</i>	Memberikan dukungan langsung kepada aplikasi-aplikasi dan proses yang aktivitasnya terdistribusi dalam suatu sistem yang menggunakan beberapa komputer dalam suatu jaringan.
<i>Presentation Layer</i>	Melakukan transformasi dan penyesuaian format data agar sesuai dengan format yang dapat dibaca oleh aplikasi yang digunakan.
<i>Session Layer</i>	Melakukan sinkronisasi dan membuat komunikasi di antara <i>host-host</i> yang terhubung menjadi lebih terstruktur.
<i>Transport Layer</i>	Mentransfer data secara <i>end-to-end</i> , dengan menyembunyikan properti dari jaringan fisiknya.
<i>Network Layer</i>	Melakukan transfer data di dalam jaringan dengan menggunakan gambaran topologi <i>logic</i> yang menggunakan pengalamatan dengan <i>IP address</i> .
<i>Data-Link Layer</i>	Melakukan transfer data di antara <i>device-device</i> yang terhubung secara fisik, menggunakan sistem pengalamatan dengan <i>MAC address</i> untuk mengidentifikasi <i>device</i> .

Tabel 2.1 Seven Layers of OSI Reference Model (Lanjutan)

Physical Layer	Mentransfer data secara fisik, berupa bit-bit yang dikirimkan dalam bentuk sinyal melalui media fisik seperti kabel atau gelombang radio.
-----------------------	---

Penomoran *layer* pada *OSI reference model* dimulai dari *physical layer* sebagai *layer* pertama sampai dengan *application layer* sebagai *layer* ketujuh. *OSI reference model* tidak hanya memberikan aturan mengenai arsitektur standar mengenai jaringan komputer yang terbagi menjadi lapisan-lapisan, tetapi juga mendefinisikan lapisan-lapisan konseptual yang umum terdapat di dalam suatu sistem komunikasi (Sharp, 2008).

2.2 Link-State Routing

2.2.1 Routing Protocol

Teknik *routing* dapat dibagi menjadi dua, yaitu *static routing* dan *dynamic routing*. *Static routing* adalah teknik konfigurasi *routing* di mana *network administrator* perlu melakukan konfigurasi *router* secara manual, dengan menginputkan satu per satu *subnet* yang terdapat di dalam jaringan tersebut dan juga alamat *next hop* yang harus dilalui agar data sampai ke *host* tujuan. Sementara itu, jika menggunakan teknik konfigurasi *dynamic routing*, *network administrator* hanya perlu memasukkan data *subnet* yang terhubung langsung dengan *router* yang dikonfigurasi dan parameter-parameter lain yang dapat digunakan oleh *routing protocol* untuk memilih rute yang terbaik (Lammle, 2000).

Pada *OSI Reference Model*, proses *routing* berada pada *layer* ketiga, yaitu *network layer* (Simoneau, 2006). *Router* yang dikonfigurasi menggunakan metode *dynamic routing* akan berkomunikasi dengan *router* lainnya dengan menggunakan *routing protocol*, yang secara umum dapat dikelompokkan menjadi *distance vector routing protocol* dan *link-state routing protocol* (Johnson, 2008).

2.2.2 Link-State Routing Protocol

Link-state routing protocol dapat didefinisikan sebagai sebuah *routing protocol* yang menggunakan perhitungan rute berdasarkan *link-state*. Pada setiap *node* yang terdapat di dalam sekumpulan *router* yang berhubungan melalui suatu *link-state routing protocol*, terdapat informasi *link-state* dari setiap *link* di antara *router-router* yang terhubung (Kim, Ko, & Vaidya, 2008).

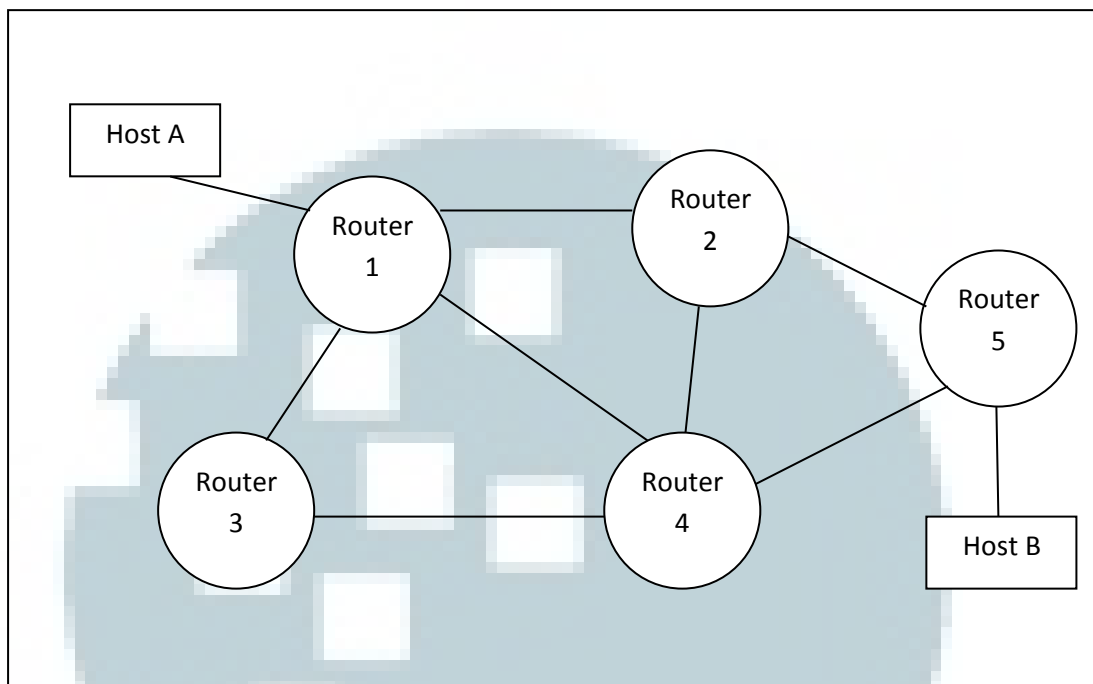
Link-state routing protocol memberikan informasi *link-state* kepada *router* lainnya melalui LSA (*Link-State Advertisements*). Dengan melakukan *broadcast LSA*, *router-router* yang menerima informasi tersebut dapat membuat pemetaan topologi jaringan secara keseluruhan, sehingga memberikan mereka kemampuan untuk membuat *routing table* yang mencakup seluruh jaringan berdasarkan informasi yang diterima (Shaikh, Isett, Greenberg, Roughan, & Gottlieb, 2002).

Salah satu contoh *routing protocol* yang menggunakan *link-state* adalah *Open Shortest Path First* (OSPF). Pada saat penelitian ini dijalankan, OSPF adalah salah satu *link-state routing protocol* yang paling banyak digunakan (Shaikh, Wang, Li, Yates, & Kalmanek, 2003). Berikut ini adalah langkah-

langkah yang dilakukan oleh *routing protocol* OSPF pada saat melakukan perhitungan rute untuk membuat *routing table* (Nortel Networks, 2002).

1. Setiap *router* akan saling bertukar informasi *link-state* dengan *router-router* lain yang bertetangga dengannya.
2. Setelah mendapatkan informasi *link-state*, *router* akan menghitung jarak di antara *router* yang satu dengan *router* yang lainnya dengan menggunakan algoritma Dijkstra untuk memetakan topologi jaringan.
3. Kemudian *router* akan mengirim paket-paket data ke *router-router* lainnya secara periodik untuk melakukan *testing* dan memastikan apakah keadaan *link-state* sebenarnya sama dengan informasi yang dimiliki oleh *router* tersebut.
4. Jika terdapat perbedaan antara informasi *link-state* yang tersimpan di dalamnya dengan informasi *link-state* sebenarnya, maka *router* akan melakukan *update* terhadap *routing table* yang dimilikinya.

U
M
N

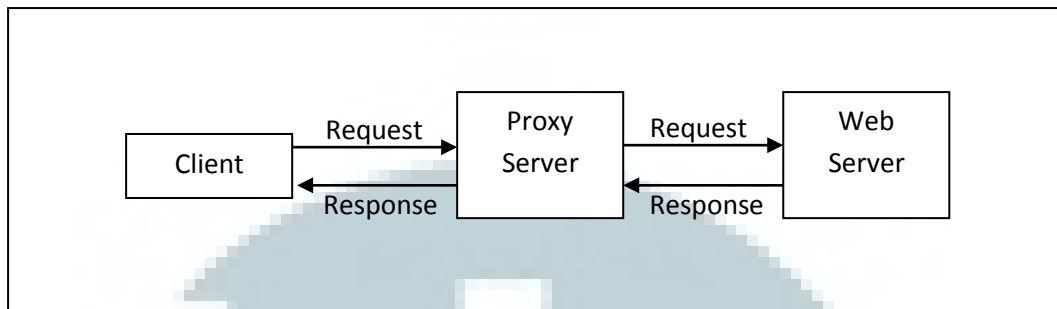


Gambar 2.1 Contoh Topologi *Logic* untuk *Routing*

2.3 Proxy Selector

2.3.1 Proxy Server

Proxy server adalah sebuah *server* yang berperan sebagai perantara yang meneruskan *request* dari *host* yang satu ke *host* lainnya yang dituju oleh *request* tersebut, dan kemudian akan meneruskan *response* dari *host* tujuan kembali ke *host* pengirim *request*. Jika digunakan dalam konteks *HTTP proxy*, maka *proxy server* adalah sebuah *server* yang berperan sebagai perantara dalam komunikasi antara *host client* dengan *web server* (Saini, 2011).

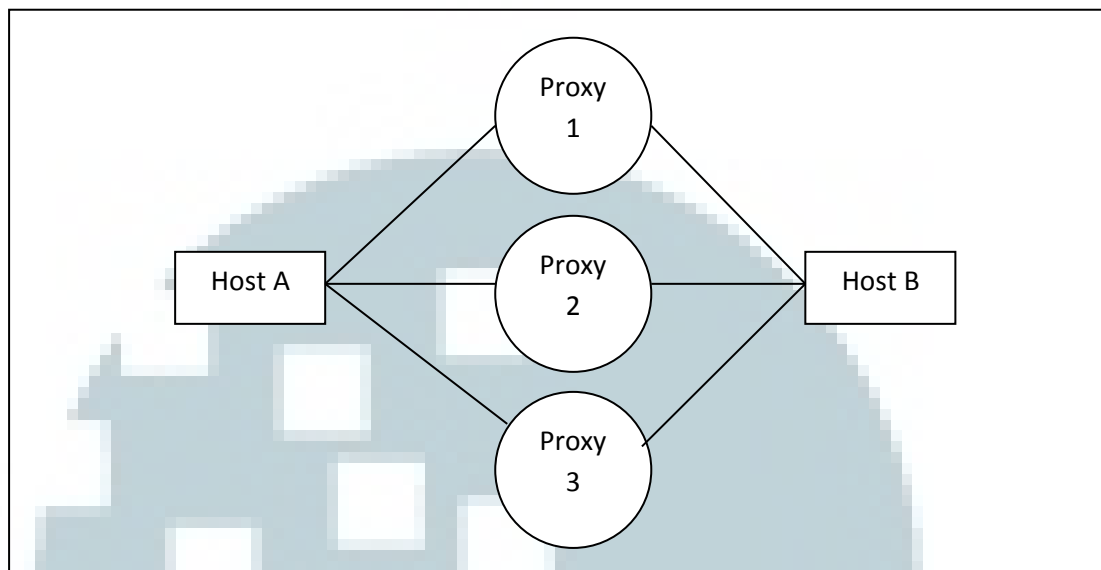


Gambar 2.2 Proxy Server

Performance dari sebuah *proxy server* dapat diukur dengan menggunakan *throughput* dari *proxy server* tersebut ke masing-masing *client*, probabilitas besar *delay* antara *client request* dan *server response*, dan penggunaan *resource* dari *server* di mana *proxy server* tersebut berjalan (Gautam, 2002).

Dalam *OSI reference model*, *user client* dan *server* terdapat dalam *layer* ketujuh, yaitu *application layer* (Simoneau, 2006). Karena pada dasarnya *proxy server* adalah sebuah *server*, maka *proxy server* termasuk dalam *layer* ketujuh dalam *OSI reference model*.

U
M
N



Gambar 2.3 Contoh Topologi Logic untuk Proxy Server

2.3.2 Pemilihan Server

Pada *content delivery network* (CDN) dan *peer-to-peer communication* (P2P), konten seringkali direplikasi dan diletakkan pada beberapa *server* atau *peer* sehingga tercipta kondisi di mana terdapat banyak *server* atau *peer* yang memiliki konten tersebut. Konten-konten direplikasi disebarkan dengan tujuan agar *client* dapat memilih untuk mengambil konten yang diinginkan dari *server* atau *peer* yang dapat memberikan pelayanan terbaik untuk *client* tersebut. Teknik pemilihan *server* digunakan dalam proses pemilihan *server* atau *peer* secara otomatis oleh program (Starobinski & Wu, 2005).

Dalam pemilihan *server* untuk *load-balancing*, teknik pemilihan *server* dengan probabilistik dapat digunakan untuk melakukan perhitungan kemungkinan-kemungkinan pemilihan *server* oleh *client* dan mengantisipasi agar *request* dari *client* tidak dikirimkan ke *server* yang sama sehingga menyebabkan

server tersebut *overload* (Bartolini, Bongiovanni, & Silvestri, 2007). Teknik dengan probabilistik ini juga dapat digunakan dalam jaringan komunikasi P2P agar suatu *client* dapat secara otomatis memilih konten yang diperkirakan akan dibutuhkan untuk proses selanjutnya dari *peer* yang telah dipilih berdasarkan perkiraan *performance* dari *peer* tersebut yang dilihat dari *performance*-nya pada beberapa operasi terakhir (D'Acunto, Chiluka, Vinkó, & Sips, 2013).

2.4 Moving Average

Moving average (MA) adalah salah satu teknik probabilistik yang umum untuk digunakan dalam menganalisa tren dan kecenderungan suatu nilai untuk bertambah, berkurang, atau tetap konstan. *Moving average* umum digunakan dalam analisa pasar dan saham karena hasil perhitungan *moving average* dapat digunakan untuk melakukan prediksi perubahan dari suatu nilai. Hasil perhitungan *moving average* dari suatu deretan nilai A dapat dibandingkan dengan hasil perhitungan *moving average* dengan interval yang sama dari deretan nilai B untuk mendapatkan prediksi dari keduanya dan menentukan pilihan yang lebih baik antara A dan B (Hwa, 2007).

$$SMA = \frac{p_M + p_{M-1} + \dots + p_{M-(n-1)}}{n}$$

...Rumus 2.1 Simple Moving Average

Ketepatan hasil prediksi yang telah dibuat dengan menggunakan *moving average* dapat diukur dengan menggunakan perhitungan *mean squared error* (MSE).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2.$$

...Rumus 2.2 Mean Squared Error

Prediksi yang baik akan menghasilkan nilai yang rendah pada saat perhitungan MSE (Boken, 2000). Dalam beberapa percobaan perhitungan *standard moving average* dengan menggunakan interval tertentu, panjang interval yang terbaik adalah panjang interval di mana nilai MSE-nya mencapai titik paling rendah.

UMMN