

## Implementasi dan Analisa “Transcoder” Dengan Kemampuan Penyeimbang Beban Untuk Aplikasi Video Streaming

Kalamullah Ramli, Eko Purwanto dan Bagio Budiardjo  
Departemen Teknik Elektro, Fakultas Teknik - Universitas Indonesia  
Kampus UI, Depok 16424, Indonesia  
k.ramli@eng.ui.ac.id, bbudi@eng.ui.ac.id

### Abstrak

Beberapa tahun terakhir ini terjadi perkembangan yang sangat pesat dalam kebutuhan layanan multimedia. Penggunaan berbagai aplikasi multimedia secara kontinu dan terus-menerus menyebabkan peningkatan kebutuhan bandwidth. Di banyak negara, bandwidth masih merupakan hal yang langka, sehingga banyak pengguna jaringan yang menemui kesulitan menikmati aplikasi multimedia yang memang membutuhkan bandwidth yang besar. Sebuah solusi untuk ini adalah dengan melakukan perubahan format aplikasi multimedia secara dinamis dan berdasarkan kebutuhan, dari format yang besar ke format ke lebih efisien. Proses ini dikenal dengan nama transcoding. Namun karena proses transcoding membutuhkan sumber daya proses yang besar, perlu diterapkan teknik penyeimbangan beban diantara prosesor dalam satu jaringan. Teknik ini dapat meningkatkan kinerja dan efisiensi jaringan. Paper ini menjelaskan proses implementasi dan analisa kinerja transcoder dengan kemampuan penyeimbang beban dalam lingkungan aplikasi video streaming. Tiga algoritma penyeimbang beban dipertimbangkan dalam riset ini dan berbagai format video dan suara diuji dalam test yang sesungguhnya. Meskipun strategi penyeimbangan beban berhasil meningkatkan kinerja dan efisiensi jaringan, namun ditemukan bahwa bila jumlah proses transcoding meningkat, maka sistem kesulitan untuk menjaga kinerja dalam menghantarkan aplikasi multimedia. Dapat disimpulkan bahwa sistem yang dikembangkan masih memiliki masalah skalabilitas dan karenanya dibutuhkan kerja dan penelitian lanjut untuk memperbaikinya.

**Kata kunci:** Bandwidth, Transcoder, Keseimbangan beban dan Algoritma

### Abstract

In recent years, there is a significant increase on the demand of multimedia services. The use and delivery of various multimedia applications, collectively and continuously, creates increasing demand on network bandwidth. However, in most of the countries, the available bandwidth capacity remains very limited. Majority users living in these regions are having difficulties to access such applications. A solution to this problem is by embedding the capability to dynamically change the format of bandwidth-intensive multimedia applications to a more efficient one into the network. The process which alters multimedia application from one format to another format is known as Transcoding. For example, cinepak format requiring high bandwidth is transcoded into lower format such as example H.263. Since transcoding process is very demanding in terms computing resources, load balancing strategy in a network environment helps to share the load among the transcoder processor, and in turn, improves throughput and efficiency. This paper describes the implementation and analysis of applying transcoder with load balancing mechanism in a video streaming environment. Three load balancing algorithms are considered and real tests using various video and audio formats are performed in this environment. Eventhough the load balancing strategy improves the performance of the system, we still find that as the number processes increases the ability to deliver multimedia applications through transcoding process decreases. Moreover, we also conclude that our system still poses scalability problems and therefore further work is required.

**Keywords:** Bandwidth, Transcoder, Load balancing and Algorithm.

## 1. Pendahuluan.

Dalam dekade terakhir terjadi peningkatan yang sangat signifikan terhadap kebutuhan layanan multimedia. Pengiriman dan penggunaan aplikasi-aplikasi multimedia secara bersama menyebabkan peningkatan kebutuhan *bandwidth* pada jaringan, dengan kapasitas *bandwidth* yang tetap terbatas. Kecendrungan lain adalah semakin banyaknya tipe peralatan komputer, seperti *desktop PC*, *notebook PC*, *PDA*, *mobile phone* yang mempunyai kemampuan berbeda, baik dari sisi kekuatan komputasi dan kemampuan tampilan (*display*).

Hal-hal tersebut diatas menyebabkan *server* multimedia sebagai penyedia layanan dituntut harus mampu menyediakan aplikasi-aplikasi multimedia yang membutuhkan *bandwidth* kecil, sehingga dapat diakses oleh pengguna yang memiliki *bandwidth* kecil serta pengguna dengan berbagai peralatan komputernya.

Sebuah solusi untuk permasalahan di atas adalah dengan mengubah aplikasi-aplikasi multimedia tersebut dari format yang memiliki jangkauan *bandwidth* lebar menjadi format yang memiliki jangkauan *bandwidth* kecil. Proses ini dikenal sebagai *transcoding*, yang berarti pengubahan satu format aplikasi ke format yang lain. Proses *transcoding* membutuhkan sebuah *server* yang disebut *transcoder*.

Aplikasi *transcoder* dengan kemampuan penyeimbang beban memungkinkan seorang pengguna jaringan mengirimkan sebuah permintaan *streaming* multimedia, kemudian *transcoder* mengkonversi ke bentuk format yang sesuai dengan kemampuan yang dimiliki pengguna, seperti kemampuan peralatan dan jangkauan *bandwidth*. Sistem yang diteliti dan dikembangkan pada penelitian ini juga dapat menentukan *transcoder* yang sesuai untuk pengguna, dan sekaligus dapat mendistribusikan beban kerja secara seimbang pada setiap *transcoder* dan akhirnya dapat meningkatkan *throughput* jaringan.

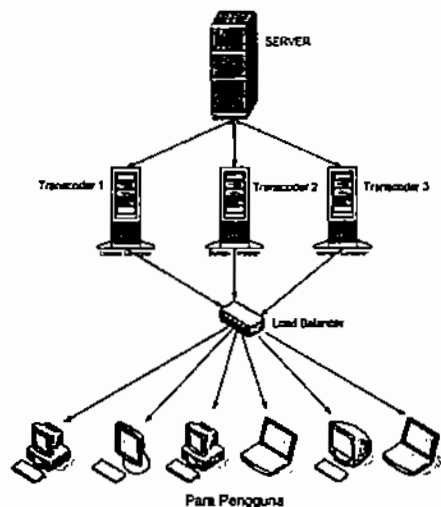
## 2. Infrastruktur *Transcoding*

Ada dua pendekatan yang berbeda pada infrastruktur *transcoding*, yaitu pendekatan *server-side* dan pendekatan *network-side*. Pendekatan *server-side* menyediakan beberapa format media yang berbeda pada *server*, yang dapat dipilih oleh pengguna. Sedangkan pada pendekatan *network-side*, hanya tersedia satu format media pada *server* dan jaringan melakukan *transcode* dengan bantuan elemen jaringan yang dikenal dengan nama *transcoder*.

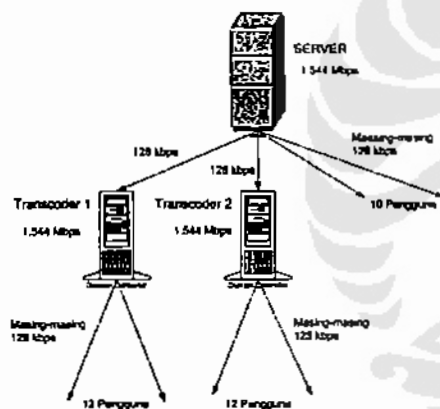
### 2.1. Kegunaan *Transcoder*

Tugas utama *transcoder* adalah mengubah aliran media ke dalam bentuk format yang sesuai dengan jangkauan *bandwidth* atau peralatan yang dimiliki oleh pengguna. Pada Gambar 1 diperlihatkan infrastruktur aplikasi *transcoder* yang sederhana pada sebuah jaringan. Media *server* mempunyai *bandwidth* 128 Kbps yang kemudian diubah oleh *transcoder* ke dalam tiga bentuk format berbeda dan untuk tiga pengguna yang berbeda melalui koneksi yang berbeda. Aliran data yang telah diubah mempunyai *bandwidth* lebih kecil dibandingkan dengan yang asli, dan berturut-turut memerlukan 64 Kbps, 32 Kbps dan 13,2 Kbps. Lebih jauh, format *DVI* dengan *bandwidth* 32 Kbps dan format *GSM* dengan *bandwidth* 13,2 Kbps memerlukan daya komputasi pada prosesor yang lebih kecil dibandingkan *MPEG/Audio* dengan *bandwidth* 64 Kbps. Dalam hal kualitas suara, *MPEG/Audio* tentunya akan lebih baik dari *DVI* dan *GSM*. Bagaimanapun, aliran data dari hasil proses pengubahan memiliki kualitas yang tidak sebaik aslinya, namun masih memnuhi standar kualitas pelayanan.

Perancangan *transcoder* pada penelitian ini dilengkapi dengan kemampuan untuk berbagi beban antara *transcoder* dan kemampuan untuk mendukung peningkatan jumlah pengguna (skalabilitas). Pembagian beban antar *transcoder* memerlukan strategi yang dikenal sebagai algoritma penyeimbang beban (*load balancing*).



Gambar 1.  
Infrastruktur *Transcoder* dengan *Load Balancing*



Gambar 2.  
Infrastruktur *Cascading transcoder*.

Sebagai contoh, sebuah *server* menawarkan aliran media *MPEG/Audio* 128 Kbps, dan dihubungkan ke jaringan dengan kapasitas *bandwidth* 1,554 Mbps. Jumlah maksimum *transcoder* yang dapat dilayani oleh *server* secara serentak adalah 12 *transcoder*. Bila setiap *transcoder* yang dihubungkan ke jaringan 1,554 Mbps dapat melayani 12 pengguna, jumlah maksimum pengguna adalah 144 individu. Gambar 2 menunjukkan infrastruktur aplikasi *transcoder* yang mampu melayani banyak pengguna.

*Transcoder* dapat melaksanakan fungsi lain sebagai tempat sementara untuk media

yang paling banyak diakses, sehingga *transcoder* tidak harus mengakses ulang media ketika pengguna yang lain meminta yang sama. Kekurangan dari skenario ini adalah karena hanya dapat digunakan pada aplikasi *on-demand*, bukan untuk aplikasi percakapan dan aplikasi terdistribusi.

## 2.2. *Directory Service* dan *Service Broker*.

Ada tiga komponen dasar dalam infrastruktur *transcoder* yaitu *server*, pengguna, dan *transcoder*. Tiga komponen perlu dilengkapi dengan komponen lain untuk menangani isu-isu tambahan yang terkait.

Isu pertama adalah *service brokering*, yaitu teknik pemilihan *transcoder* untuk pengguna berdasarkan format media yang telah ditentukan. Pada penelitian ini, untuk proses *service brokering*, digunakan suatu komponen yang disebut layanan perantara (*service broker*). *Service broker* bertindak sebagai *black-box* yang menggunakan algoritma tertentu untuk memilih *transcoder* terbaik untuk pengguna.

Isu kedua adalah bagaimana *service broker* mengetahui jenis *transcoder* yang tersedia pada jaringan. Metoda umum untuk memecahkan masalah ini adalah dengan mengirimkan pesan *multicast* ke jaringan untuk menemukan *transcoder* tersebut. Dengan dua komponen tambahan ini, infrastruktur *transcoder* terdiri dari lima elemen, yaitu *server*, pengguna, *service broker* dan *directory service* seperti yang ditunjukkan pada Gambar 3.

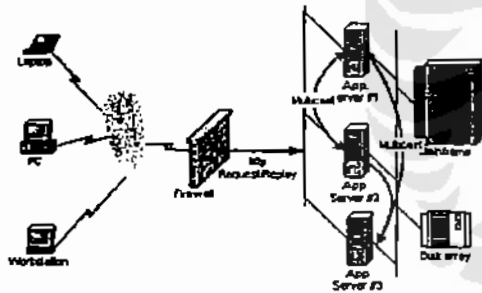
### 2.3.1. Metoda-metoda Penyeimbang Beban.

Ada dua metoda yang digunakan untuk menerapkan penyeimbang beban, yaitu metoda *web server* dan metoda penjadwalan bersama. Pada metoda penjadwalan bersama, aplikasi *server* diijinkan untuk memutuskan antara mereka sendiri bagaimana cara mendistribusikan beban kerja. Sedangkan pada metoda *web server*, *web server* akan melaksanakan tugas sebagai penyeimbang beban untuk aplikasi *server* [1].



Gambar 3.  
Arsitektur transcoder multimedia

Metoda penyeimbang beban menggunakan metoda penjadwalan bersama ditunjukkan pada Gambar 4. Pada metoda ini, server berbagi informasi beban secara periodik. Pengguna mengirimkan request HTTP melalui firewall secara langsung ke server tertentu. Server akan memilih transcoding server yang cocok untuk memproses permintaan, dan jika dibutuhkan menyelesaikan permintaan tersebut. Jika tidak, server yang terpilih akan memproses permintaan dan hasilnya dikembalikan kepada pengguna.



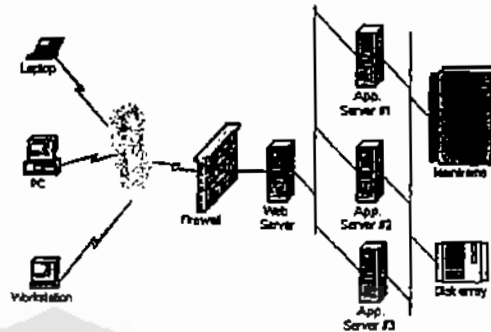
Gambar 4  
Infrastruktur penyeimbang beban dengan metoda penjadwalan bersama [1]

Metoda penyeimbang beban dengan menggunakan metoda web server diperlihatkan pada Gambar 5. Pada metoda ini, pengguna mengirimkan sebuah permintaan HTTP melalui firewall ke web server. Web server akan memilih server tertentu yang dapat mengirimkan permintaan tersebut. Permintaan diproses dan hasilnya dikembalikan kepada pengguna.

### 2.3.2. Algoritma Penyeimbang Beban.

Algoritma penyeimbang beban digunakan oleh service broker untuk

menentukan transcoder yang memiliki beban kerja paling kecil yang akan memproses permintaan aliran media dari pengguna. Algoritma penyeimbang beban yang umum digunakan dan dipertimbangkan dalam pengembangan penelitian ini antara lain [2]:



Gambar 5.  
Infrastruktur penyeimbang beban dengan metoda web server [1].

#### a. Round robin.

Algoritma yang mendistribusikan setiap koneksi baru kepada transcoder berikutnya yang tersedia.

#### b. Weighted round robin with respon time.

Adalah peningkatan dari algoritma round robin dimana waktu tanggap (respon-time) untuk setiap transcoder dalam layanan virtual diukur secara tetap untuk menentukan transcoder mana yang akan mengambil koneksi berikutnya.

#### c. Fewest Connection with limits.

Sebuah algoritma yang menentukan transcoder mana yang mendapatkan koneksi berikutnya dengan memegang sebuah catatan berapa banyak koneksi setiap transcoder yang tersedia saat ini. Transcoder dengan koneksi lebih kecil akan mendapatkan permintaan berikutnya.

Algoritma round robin lebih efektif untuk distribusi antar transcoder dengan kapasitas pemrosesan yang sama. Jika transcoder memiliki kapasitas pemrosesan berbeda, penggunaan waktu tanggap atau jumlah koneksi yang aktif sebagai kriteria pilihan dapat mengoptimalkan waktu tanggap pengguna.

### 3. Penentuan Transcoder dan Format Media

Tugas utama *service broker* adalah menentukan salah satu *transcoder* yang sesuai dengan format yang diinginkan oleh pengguna. Gagasan dasarnya adalah mendapatkan daftar *transcoder* yang mampu melayani pengguna dan kemudian menentukan satu yang terbaik. Pada Penelitian ini, tugas *service broker* tidak hanya menentukan *transcoder* untuk pengguna, tetapi juga harus mampu mendistribusikan beban kerja yang seimbang kepada setiap *transcoder*.

#### 3.1. Format Sumber dan Format Tujuan

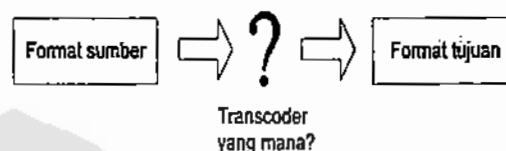
Sebelum melayani permintaan dari pengguna *service broker* perlu mengetahui format media yang didukung oleh *server* sebagai format sumber, dan format yang didukung oleh pengguna sebagai format tujuan. Penentuan format tujuan dapat dilakukan berdasarkan informasi yang diterima *service broker* dari pengguna. *Service broker* menginformasikan pengguna format apa saja yang didukung oleh *server*. Pengguna kemudian secara manual dapat memilih salah satu format yang disediakan oleh *server*.

Untuk menentukan format sumber beberapa metoda dapat digunakan. Metoda pertama adalah menyediakan database dalam *service broker* atau dalam komputer lain yang berisi format-format dari alamat-alamat yang diketahui. Sebuah media data mempunyai beberapa bentuk format yang tersimpan dalam database. Contoh: "http://192.168.10.117:1220/ultramanJPEG.mov"- JPEG-320 x 240 pixels 24 fps. Kerugian metoda ini adalah administrator jaringan perlu menyediakan database yang besar untuk menyimpan format-format tersebut.

Metoda kedua adalah dengan mengirimkan permintaan kepada *server*. *Server* akan memberitahukan format media apa saja yang didukung kepada *service broker*. Kerugian dari metoda ini adalah tidak semua *server* dapat mendukung metode ini. Satu contoh *server* yang dapat

menangani permintaan seperti ini adalah *RTSP server*. *Service broker* dapat mengirimkan permintaan kepada *RTSP server* untuk meminta format media yang didukung.

Setelah mengetahui format sumber dan format tujuan *service broker* perlu mendapatkan semua format *transcoder* yang mampu melayani pengguna, kemudian menentukan satu yang terbaik untuk pengguna. Gambar 6 menunjukkan tugas *service broker* menentukan *transcoder* yang sesuai untuk pengguna.



Gambar 6.  
Bagaimana menentukan *transcoder* yang sesuai untuk pengguna

Untuk mendapatkan semua format *transcoder*, *service broker* menggunakan direktori yang disimpan di dalam *directory service*. Sedangkan untuk menentukan salah satu *transcoder* yang terbaik untuk pengguna, *service broker* memerlukan sebuah algoritma penyeimbang beban (*load balancing*).

#### 3.2. Kualitas Format *Transcoding*.

*Service broker* memberikan tingkat kualitas untuk setiap format *transcoding*. Ini dibutuhkan karena suatu format mempunyai kualitas yang lebih baik dari format yang lain. Tingkat kualitas diberikan untuk memberitahukan kepada pengguna bahwa tingkat tertinggi mempunyai kualitas yang paling baik. Sehingga akan menjadi pertimbangan bagi pengguna untuk menentukan format apa yang diinginkan.

Tabel 1 menunjukkan perbandingan kualitas format suara dan video. Format-format tersebut digunakan dalam perancangan dan penelitian *transcoder* dalam penelitian ini.

Tabel 1.  
Kualitas format pada service broker [3]

Kua Litas	Format	Bit rates (Kbps)	Kompu Tasi
1	MPEG 352 x 288, 30 fps MP3 44 KHz, 16 bits, stereo	428	Sangat tinggi
2	H.263 352 x 288, 30 fps MP3 44 KHz, 16 bits, stereo	278	Rendah
3	MPEG 352 x 288, 30 fps DVI 22 KHz, 4 bits, mono	364	Tinggi
4	H.263 352 x 288, 30 fps DVI 22 KHz, 4 bits, mono	215	Sangat rendah

#### 4. Service Chaining.

Service chaining adalah protokol yang digunakan untuk membangun path dari server ke pengguna melalui transcoder. Service broker mencari sebuah direktori pada directory service untuk memilih sebuah transcoder dan kemudian memberikan informasi ini ke pengguna. Selanjutnya pengguna akan mendengar aliran media dari transcoder yang dipilih oleh service broker.

##### 4.1. Pendaftaran Layanan ke Directory Service

Ketika transcoder atau service broker dipasang pada jaringan, komponen ini harus mendaftar ke directory service. Pendaftaran komponen atau layanan ke directory service, dilakukan dengan menggunakan kotak dialog, seperti pengaturan proxy pada web browser, atau menggunakan konfigurasi file.

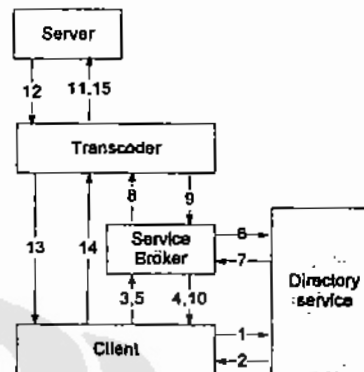
##### 4.2. Permintaan Layanan Transcoder.

Pada tahap ini service broker dan semua transcoder telah mendaftarkan layanannya ke directory service. Ketika seorang pengguna meminta sebuah aliran media, infrastruktur transcoding menentukan transcoder yang terbaik dengan menggunakan algoritma service brokering dan algoritma penyeimbang beban. Selanjutnya transcoder mengubah aliran media tersebut sesuai kebutuhan pengguna.

##### 4.2.1. Permintaan atas Inisiatif Pengguna.

Permintaan atas inisiatif pengguna berarti pengguna berinisiatif mengajukan

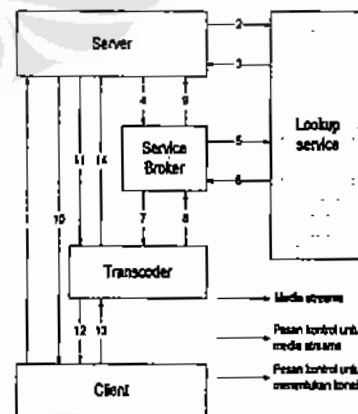
permintaan ke service broker untuk mendapatkan transcoder terbaik. Gambar 7 memperlihatkan aliran media dan kontrol pesan antara komponen pada aplikasi transcoding. Nomor panah merepresentasikan langkah layanan protokol chaining yang diimplementasikan dalam penelitian ini.



Gambar 7.  
Aliran media dan kontrol aliran pada permintaan atas inisiatif pengguna.

##### 4.2.2. Permintaan atas Inisiatif Server.

Permintaan atas inisiatif server berarti bahwa permintaan ke service broker dilakukan oleh server. Gambar 8 memperlihatkan aliran media dan pesan kontrol antara komponen dari permintaan atas inisiatif server

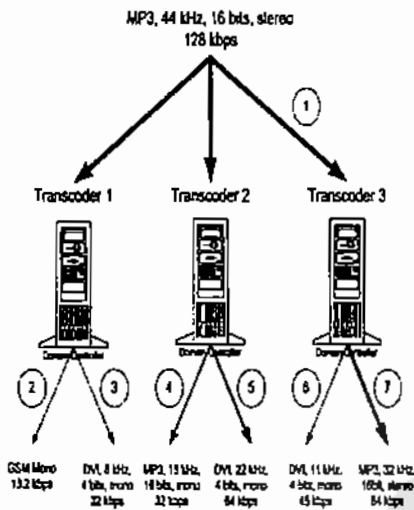


Gambar 8.  
Aliran media dan kontrol aliran pada permintaan atas inisiatif server.

##### 4.3. Konfigurasi Transcoder

Ada dua metoda yang digunakan untuk menempatkan transcoder pada jaringan,

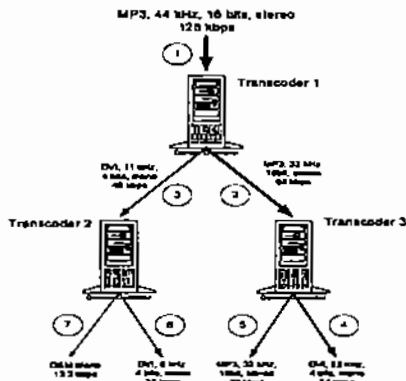
yaitu secara flat dan hirarki [4]. Gambar 9 adalah konfigurasi *transcoder* secara flat, sedangkan Gambar 10 memperlihatkan konfigurasi *transcoder* secara hirarki.



Gambar 9: Infrastruktur flat *transcoding* [3].

Fitur utama konfigurasi secara hirarki adalah penghematan *bandwidth*. Contoh, ada enam pengguna meminta enam format berbeda. Pada konfigurasi flat, terdapat 384 Kbps (3 x 128 Kbps) trafik dari *server* ke jaringan. Pada konfigurasi hirarki, hanya ada 128 Kbps trafik.

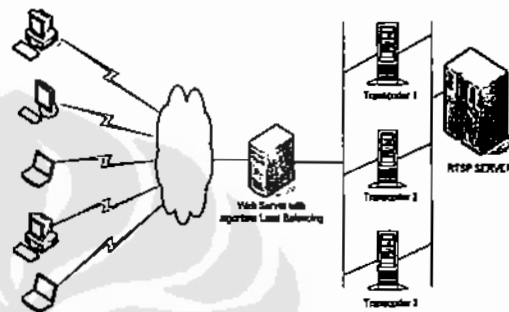
Konfigurasi hirarki mempunyai kekurangan berupa kegagalan titik tunggal. Bila satu *transcoder* rusak, maka semua pengguna tidak dapat menerima aliran. Hal ini tidak dijumpai pada konfigurasi secara flat, karena hanya pengguna yang meminta saja yang tidak dapat menerima aliran media.



Gambar 10. Infrastruktur hirarki *transcoding* [3]

#### 4.4. *Transcoder* dengan Penyeimbang Beban

Pada aplikasi ini, setiap *transcoder* memiliki format yang sama, sehingga konfigurasi *transcoder* yang paling cocok digunakan adalah konfigurasi secara flat. Permasalahan dari konfigurasi *transcoder* tersebut adalah bagaimana mendistribusikan beban kerja kepada setiap *transcoder* secara seimbang. Gambar 11 memperlihatkan sebuah infrastruktur flat *transcoding* dengan menggunakan algoritma penyeimbang beban.



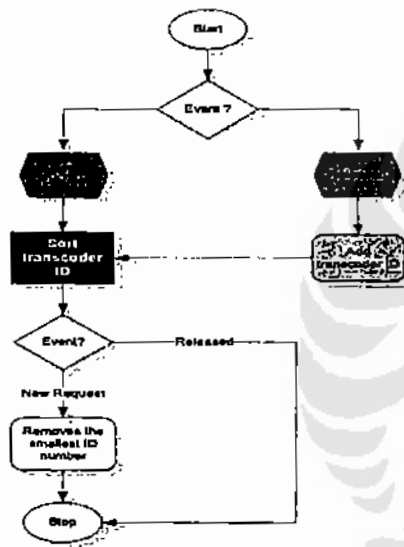
Gambar 11. Infrastruktur flat *transcoding* menggunakan algoritma penyeimbang beban

Ketika pengguna pada jaringan meminta aliran media, *service broker* akan memberikan informasi format yang didukung *server* untuk pengguna. Pengguna memilih format yang sesuai dengan kemampuan yang dimilikinya. Menggunakan algoritma *service brokering* dan algoritma penyeimbang beban, *service broker* menentukan *transcoder* terbaik dan selanjutnya memproses permintaan pengguna.

Layanan *transcoder* diberikan oleh *service broker* kepada pengguna berdasarkan *ID transcoder* yang paling kecil. *ID transcoder* diurutkan baik pada saat ada permintaan aliran media ataupun saat permintaan selesai dari pengguna. Apabila ada permintaan aliran media dari pengguna lain, maka layanan akan diberikan kepada *transcoder* yang mempunyai *ID* yang paling kecil.

*Flowchart* algoritma penyeimbang beban yang digunakan oleh *service broker*

ditunjukkan pada Gambar 12. Pada langkah pertama, *service broker* akan memberikan *ID transcoder* berdasarkan nomor urut *transcoder*. Transcoder#1 mendapatkan ID transcoder 1, Transcoder#2 mendapatkan ID transcoder 2 dan Transcoder#3 mendapatkan ID transcoder 3. Layanan *transcoder* akan diberikan berdasarkan algoritma *round robin*. Pengguna 1 akan dilayani oleh Transcoder#1, pengguna 2 akan dilayani oleh Transcoder#2 dan pengguna 3 akan dilayani oleh Transcoder#3. Algoritma *round robin* digunakan sampai ada salah satu pengguna mengirimkan pesan selesai.

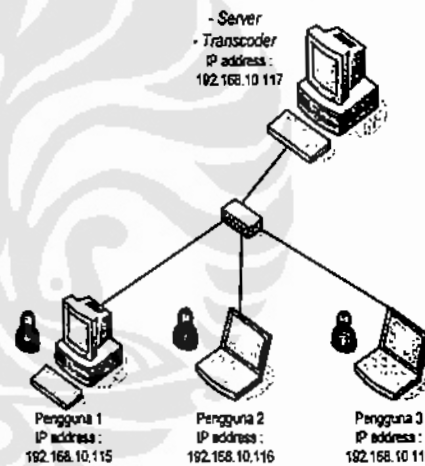


Gambar 12. Algoritma penyeimbang beban

Apabila ada seorang pengguna menghentikan aliran media, misalnya pengguna 2, maka transcoder#2 akan menghentikan aliran media dan menambahkan *ID transcoder* kepada *service broker*. Kemudian *service broker* akan menggantikan ID transcoder#2 dengan *ID transcoder* baru. *ID transcoder* yang baru tersebut adalah *ID transcoder* yang paling kecil yang dimiliki oleh *service broker*, yaitu *ID transcoder* 1. Algoritma penyeimbang beban yang semula adalah algoritma *round robin*, akan digantikan oleh algoritma *fewest connection with limit*. Bila ada pengguna lain, misalnya pengguna 4, meminta aliran media maka layanan akan diberikan kepada transcoder#4.

## 5. Integrasi Sistem.

Tiga komponen dasar (*server*, pengguna, dan *transcoder*) diintegrasikan menjadi sebuah sistem dengan menggunakan empat komputer. Satu komputer sebagai *server* dan tiga komputer lain sebagai pengguna. *Transcoder* terintegrasi dalam komputer *server*, sehingga komputer *server* akan mempunyai peran ganda, yaitu sebagai *server* yang mengirimkan permintaan aliran media dan sebagai *transcoder* untuk memproses media sesuai format yang diinginkan oleh pengguna. Keempat komputer terintegrasi dalam sebuah *Local Area Network (LAN)* yang dihubungkan oleh sebuah hub. Gambar 13 menunjukkan integrasi komponen infrastruktur *transcoder* pada *Local Area Network*.



Gambar 13. Integrasi Komponen pada Infrastruktur Transcoder (LAN)

### 5.1. Penyedia Layanan (Server).

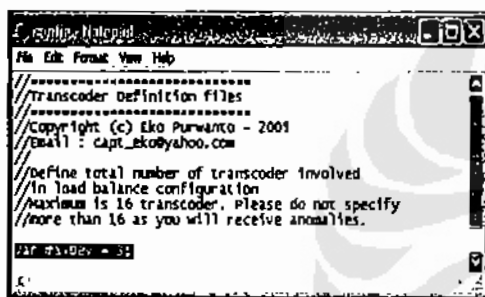
Komputer *server* adalah *VMware-workstation version 5.0.0-13* dengan spesifikasi CPU Pentium 4/2,4 GHz, RAM 512 MB, *NIC VMware Virtual Ethernet Adapter for VMnet0*. Sistem Operasi yang digunakan adalah *Windows XP professional* dengan *J2SE version 1.4.0\_01*, *JMF version 2.1.1e* dan *web server Internet Information Services (IIS) version 5.1* dengan nomor *port* 1220. *Web server* digunakan untuk menyimpan file video dan suara, sehingga untuk meminta sebuah aliran



media seorang pengguna harus mengirimkan permintaannya ke <http://192.168.10.117:1220/MTtransbalancing.html>.

### 5.2. Transcoder.

*Transcoder* mempunyai dua tugas utama, yaitu registrasi layanannya ke *directory service* dan mengubah aliran media dari *server* ke pengguna. *Transcoder* berperan sebagai pengguna untuk *server* dan sebagai *server* untuk pengguna. Registrasi sebuah *transcoder* ke *directory service* dilakukan dengan memasukkan secara manual pada sebuah kotak dialog. Gambar 14 memperlihatkan kotak dialog untuk pengaturan jumlah *transcoder*.

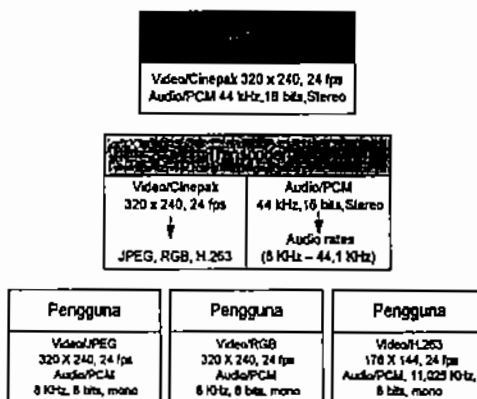


Gambar 14. Dialog Pengaturan jumlah *transcoder*

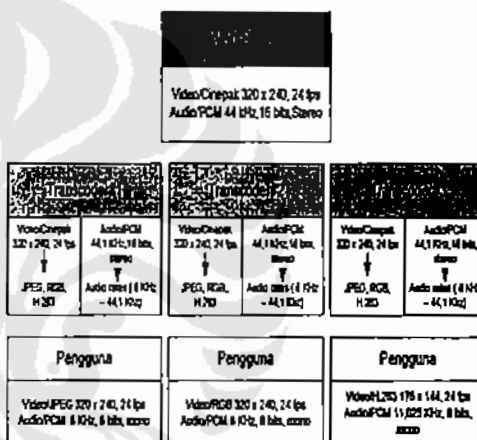
Gambar 15 memperlihatkan konfigurasi satu *transcoder* pada jaringan. Seluruh permintaan aliran media dari pengguna hanya akan dilayani oleh satu *transcoder*. Karenanya, *service broker* algoritma *load balancing* untuk mendistribusikan beban kerja tidak diperlukan. Hanya ada satu format sumber yaitu video *cinepak*-320 x 240, 24 fps dan suara *PCM*-44,1 KHz, 16 bits, *stereo*. *Transcoder* mengubah format asal ke format *JPEG*-320 x 240, 24 fps dan suara *PCM*-8 KHz, 8 bits, *mono*; *RGB*-320 x 240, 24 fps dan suara *PCM*-8 KHz, 8 bits, *mono*; *H.263*-176 x 144, 24 fps dan suara *PCM*-11.025 KHz, 8 bits, *mono*.

Gambar 16 memperlihatkan konfigurasi dengan tiga *transcoder* dan setiap *transcoder* mempunyai format sama. Karena terdapat lebih dari dua *transcoder*, maka *service broker* perlu menggunakan algoritma penyeimbang beban untuk

mendistribusikan beban kerja secara seimbang kepada setiap *transcoder*.



Gambar 15. Konfigurasi dengan satu *transcoder*



Gambar 16. Konfigurasi dengan tiga *transcoder*

### 5.3. Pengguna

Semua komputer pengguna menggunakan Sistem operasi *Windows XP professional* dan harus diinstalasi *J2SE version 1.4.x\_xx* dan *JMF version 2.1.1e*. Spesifikasi perangkat keras komputer pengguna adalah sebagai berikut:

- Pengguna 1: PC Pentium 4/2,4 GHz, RAM 256 MB, *NIC VIA Rhine II Fast Ethernet Adapter*.
- Pengguna 2: *Notebook* Pentium 2,8 GHz, RAM 256 MB, *NIC Realtek RT8139/810x Family Fast Ethernet*.
- Pengguna 3: *VMware* pada *Notebook* Pentium 2,8 GHz, RAM 256 MB, *NIC VMware Virtual Ethernet Adapter for VMnet0*.

## 6. Pengujian dan Analisa

Pengujian dilakukan dengan mengirimkan format video kepada alamat tujuan yang berbeda. Hasil uji dianalisa dengan membandingkan kualitas video yang diterima oleh setiap pengguna.

Pada paper ini, kualitas video yang diterima pengguna akan dilihat berdasarkan besarnya paket *loss* dan *delay jitter* dalam jaringan ketika jaringan tidak membawa trafik apapun. Sehingga tidak ada antrian paket dan prosesor 100% tersedia untuk melaksanakan proses *transcoding*.

### 6.1. Uji Coba untuk Satu Alamat Tujuan

Pada uji coba pertama, seorang pengguna dengan *IP address* 192.168.10.116 meminta aliran media dengan format *JPEG-320 x 240, 24 fps* dan suara *8 KHz, 8 bits, mono*.

Gambar 17 memperlihatkan grafik trafik *RTP* pada pengiriman format video *JPEG* ke *IP address* 192.168.10.116 dan tidak terlihat adanya *dropped packet*. Kualitas video pada penerima terlihat mulus karena paket *loss* dan *delay jitter* masih dalam batas toleransi.

### 6.2. Uji Coba untuk Dua Alamat Tujuan.

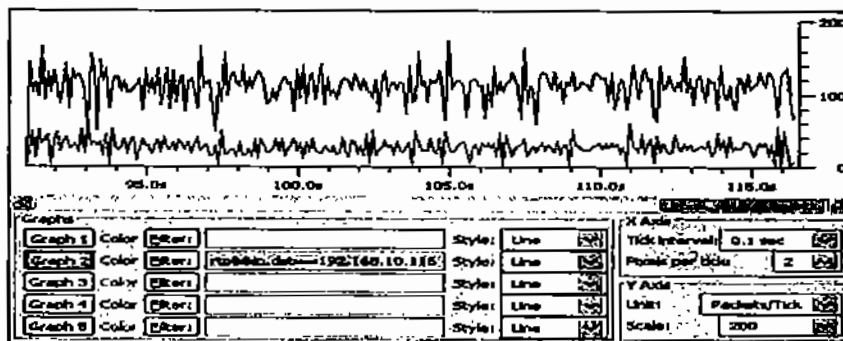
Pada uji coba kedua, dua pengguna dengan *IP address* 192.168.10.116 dan ke *IP address* 192.168.10.118 secara bersamaan meminta aliran media dengan format *RGB-320 x 240, 24 fps* dan suara *8 KHz, 8 bits, mono*.

Gambar 18 menunjukkan trafik *RTP* pada pengiriman format video *RGB* ke *IP address* 192.168.10.116 dan 192.168.10.118 dan pada kedua grafik terjadi *dropped packets* secara bergantian untuk beberapa saat sampai prosesor selesai melaksanakan proses *transcoding*. Sehingga pada uji coba ini, kualitas video pada kedua penerima terlihat mulus ketika prosesor selesai melaksanakan proses *transcoding*.

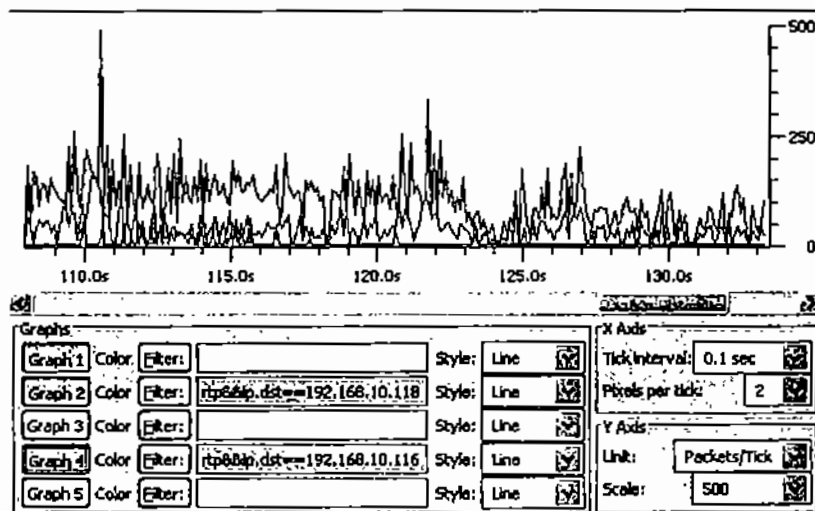
### 6.3. Uji Coba untuk Tiga Alamat Tujuan.

Pada uji coba ketiga ini, tiga pengguna dengan *IP address* 192.168.10.115, *IP address* 192.168.10.116 dan *IP address* 192.168.10.118 secara bersamaan meminta aliran media. Pada permintaan pertama, ketiga pengguna meminta aliran media dengan format yang sama yaitu video *H.263-176 x 144, 24 fps* dan suara *11,025 KHz, 8 bits, mono*.

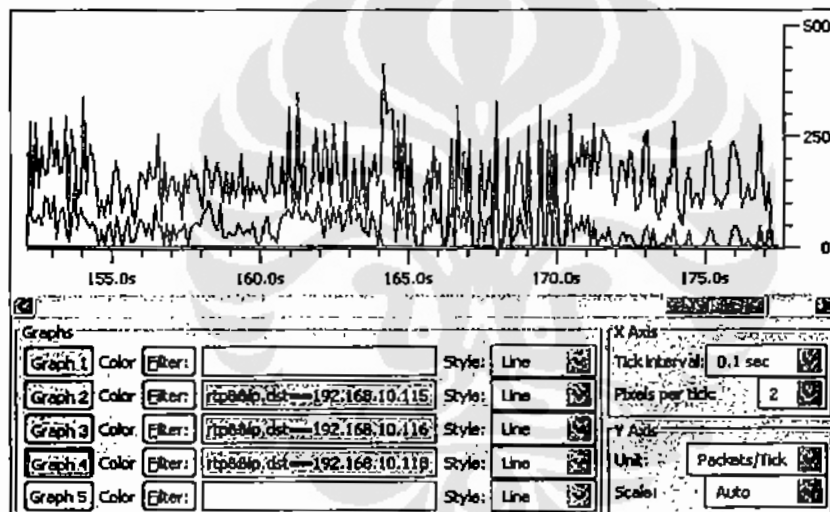
Pada Gambar 19 diperlihatkan trafik *RTP* pada pengiriman format video *H.263-176 x 144, 24 fps* dan suara *11,025 KHz, 8 bits, mono* ke *IP address* 192.168.10.115, 192.168.10.116 dan 192.168.10.118. Dari gambar tampak pengguna dengan *IP address* 192.168.10.116 tidak ada paket yang diterima, sehingga tidak ada tampilan video pada penerima. Pada pengguna dengan *IP address* 192.168.10.118 terlihat *dropped packets*. Adanya *dropped packets* ini menyebabkan kualitas video tidak mulus untuk beberapa saat. Sedangkan pengguna dengan *IP address* 192.168.10.115 mengalami *dropped packets*, sehingga kualitas video terlihat mulus.



Gambar 17.  
Grafik trafik *RTP* pada pengiriman format video *JPEG*.



Gambar 18.  
Grafik trafik RTP pada pengiriman format video RGB.



Gambar 19.  
Grafik trafik RTP pada pengiriman format video H.263.

## 7. Kesimpulan

Kualitas video pada saat pengiriman video kepada satu alamat tujuan terlihat mulus. Pada pengiriman video kepada dua atau lebih alamat tujuan kualitas video terjadi penurunan kualitas gambar selama beberapa saat sampai prosesor selesai melaksanakan proses *transcoding*. Penyebabnya adalah kemampuan prosesor yang terbatas untuk melaksanakan proses *transcoding*. Lebih jauh lagi, Keterlambatan pemrosesan ini telah menyebabkan prosesor tidak dapat

mengirimkan paket secara konstan ke jaringan.

Meskipun strategi penyeimbangan beban berhasil meningkatkan kinerja dan efisiensi jaringan, namun ditemukan bahwa bila jumlah proses *transcoding* meningkat, maka sistem kesulitan untuk menjaga kinerja dalam menghantarkan aplikasi multimedia. Dapat disimpulkan bahwa sistem yang dikembangkan masih memiliki masalah skalabilitas dan karenanya dibutuhkan kerja lanjutan untuk memperbaikinya.

## References

1. Edward J. Rhodes. *An Introduction to Load Balancing with the iPlanet Application Server*. Technology Evangelist, Sun Microsystems, Inc, 2004
2. TechBrief Extreme Network. *Server Load Balancing*. Diakses 12 Januari 2005.
3. Antony Pranata. *Development of Network Service Infrastruktur For Transcoding Multimedia Streams*. University of Stuttgart, Master Thesis, May 2002.
4. K. Jonas, M. Kretschmer and J. J. Modeker, *Get a KISS – Communicatio Infrastructure for Streaming Services in a Heterogeneous Environment*. ACM Multimedia 98, Bristol, UK, September 1998.

