

## OPTIMASI ADAPTASI KUALITAS KONTEN DENGAN DISTRIBUSI BEBAN UNTUK PENINGKATAN KINERJA LAYANAN MULTIMEDIA DALAM JARINGAN

Sofyan Noor Arief<sup>1)</sup>  
[sofyan.arief12@mhs.if.its.ac.id](mailto:sofyan.arief12@mhs.if.its.ac.id)  
 Institut Teknologi Sepuluh  
 Nopember

Waskitho Wibisono<sup>2)</sup>  
[waskwib@if.its.ac.id](mailto:waskwib@if.its.ac.id)  
 Institut Teknologi Sepuluh  
 Nopember

Royyana Muslim Ijtihadie<sup>3)</sup>  
[roy@if.its.ac.id](mailto:roy@if.its.ac.id)  
 Institut Teknologi Sepuluh  
 Nopember

### ABSTRAK

Akses internet saat ini tidak hanya dapat dinikmati oleh pengguna komputer personal saja, namun juga oleh pengguna perangkat bergerak. Namun, perangkat bergerak memiliki keterbatasan sumber daya karena sifatnya yang dimaksimalkan untuk mobilitas. Karena keterbatasan tersebut muncullah ide untuk mengubah halaman web agar dapat ditampilkan maksimal pada perangkat bergerak. Ide tersebut dikenal dengan adaptasi konten. Terdapat berbagai macam teknik adaptasi. Mulai dari adaptasi secara manual, adaptasi secara dinamis pada sisi *client* saja / sisi *server* saja/ sisi *intermediate* saja, sampai adaptasi secara dinamis dengan pemilihan lokasi pengadaptasian media antara *client*, *server* dan *intermediate*. Namun, adaptasi secara dinamis dengan pemilihan lokasi pengadaptasian medianya belum efisien, karena proses tersebut masih dilakukan pada satu sisi saja. Sehingga apabila media yang di adaptasi berupa data berukuran besar seperti video akan memberatkan sistem pada sisi yang memrosesnya. Pada penelitian ini diusulkan metode pengadaptasian media konten dengan mendistribusikan beban prosesnya pada sisi *server* dan *intermediate*. Pendistribusian beban diberikan sesuai dengan ketersediaan sumber daya yang dimiliki oleh masing-masing pekerja. Pendistribusian beban dilakukan dengan memecah video tersebut menjadi beberapa potong untuk kemudian disebar dan diproses (diadaptasikan) pada masing-masing pekerja. Kemudian potongan-potongan tersebut digabung kembali dan dikirimkan kepada *client*. Dengan menerapkan metode pendistribusian beban dalam proses pengadaptasian media konten, diharapkan akan terjadi peningkatan performa dari proses adaptasi konten yang dilakukan. Selain itu dengan metode ini diharapkan waktu respon dapat dipercepat dan meningkatkan performa sistem secara keseluruhan. Sehingga jaminan ketersediaan layanannya pun dapat dijamin.

**Kata Kunci:** adaptasi media konten, distribusi beban, perangkat bergerak, jaminan ketersediaan layanan.

#### I. Latar Belakang dan Penelitian Terkait

Perkembangan perangkat bergerak saat ini mengalami peningkatan yang cukup signifikan. Kemampuan komputasi dan penyimpanan data terus ditingkatkan. Bahkan saat ini telah dilengkapi dengan berbagai sensor dan kemampuan untuk mengakses internet layaknya perangkat tidak bergerak (seperti komputer personal). Keterbatasan dalam perangkat bergerak, seperti ukuran layar kapasitas baterai, dan kemampuan pemrosesan muncul seiring upaya untuk memaksimalkan aspek mobilitas. (Lum and Lau, 2002).

Dikarenakan adanya perbedaan fisik dan kemampuan, proses adaptasi untuk menyesuaikan keterbatasan perangkat tersebut sangat diperlukan. Telah banyak pendekatan dan metode yang diusulkan untuk melakukan adaptasi konten. Salah satunya, adalah penelitian Seunghwa dan Eunseok yang mengusulkan pendekatan baru

dengan cara melakukan distribusi adaptasi konten (Seunghwa and Eunseok, 2009). Dalam penelitian ini dibuat suatu modul adaptasi yang bertugas mengamati sumber daya yang dimiliki oleh *server*, *intermediate*, dan *client*. Hasil pengamatan dari setiap modul-modul tersebut digunakan sebagai alat pengambilan keputusan yang menentukan pada sisi manakah akan dilakukan proses adaptasi konten.

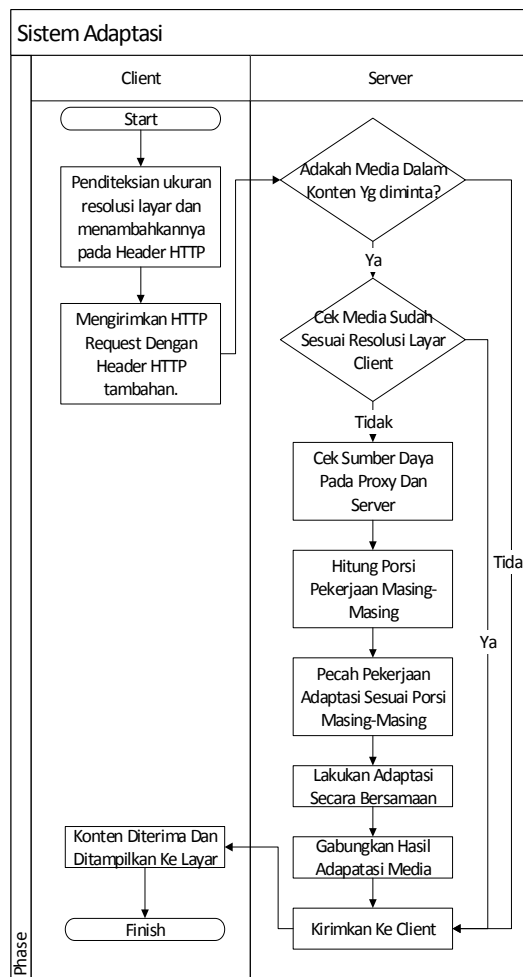
Dari percobaan tersebut, terbukti bahwa pendekatan tersebut berhasil memaksimalkan proses pengadaptasian dengan melakukan proses adaptasi pada sisi yang tidak sibuk. Sehingga kinerja dari masing-masing sisi menjadi lebih ringan, karena tugas pengadaptasian tidak menumpuk pada satu sisi saja. Proses pengadaptasian bisa dilakukan pada sisi *server*, *intermediate*, dan *client*. Namun satu tugas pengadaptasian media tetap dilakukan di satu sisi saja. Sehingga jika konten

yang akan di adaptasi besar (video misalnya), maka sisi yang diberi tugas akan terasa berat, dan dapat mengakibatkan penurunan performa adaptasi konten secara keseluruhan. Namun, pada percobaan yang dilakukan tidak terlalu nampak, karena media konten yang di adaptasi berupa gambar yang ukurannya relatif kecil.

Dalam penelitian ini, diajukan penyelesaian masalah tersebut dengan menerapkan pembagian komputasi pengadaptasian media dari konten yang akan di adaptasi menggunakan metode distribusi beban (*load distribution*). Dengan distribusi beban ini, proses pengadaptasian media dari konten yang akan di adaptasi akan ditanggung oleh *server* dan *intermediate* dengan porsi yang berbeda tergantung pada ketersediaan sumber daya yang dimilikinya. Dengan menerapkan metode distribusi beban dalam proses pengadaptasian media konten, maka akan terjadi peningkatan performa dari proses adaptasi konten yang dilakukan.

**II. Algoritma Yang Diusulkan**

Secara umum, alur dari metode yang diusulkan dapat digambarkan sebagai berikut:



Gambar 1. Alur Metode Yang Diusulkan

Sebuah gagasan baru diajukan untuk mengatasi permasalahan penurunan *response time* dan tingkat kegagalan sistem yang di akibatkan oleh tingginya *load* atau komputasi yang berjalan pada sisi pelaku adaptasi media dari konten yang di adaptasi. Tingkat kesibukan tersebut sebanding dengan jumlah *client* yang meminta sebuah halaman web untuk diadaptasikan olehnya. Gagasan yang diajukan dalam penelitian ini menggunakan metode komputasi terdistribusi. Komputasi secara terdistribusi tersebut akan dilaksanakan pada sisi *server* dan *proxy*. Di mana pembagian bebannya disesuaikan dengan porsi tertentu. Porsi tersebut ditentukan dengan menghitung jumlah ketersediaan sumber daya melalui rumus berikut (Seunghwa and Eunseok, 2009):

$$Cur_{CPU} = (1 - CPU_{load}) \frac{CPU_{speed}}{CPU_{min}} \quad (1)$$

$$Cur_{RAM} = (1 - ) (1 - RAM_{usage}) \frac{RAM_{size}}{RAM_{min}} \quad (2)$$

$$= \frac{W_{CPU}}{W_{CPU} + W_{RAM}} \quad (3)$$

Di sini, *W* merupakan beban yang dialokasikan kepada CPU dan RAM. Alokasi tersebut ditentukan dari jenis adaptasi apa yang akan dilakukan. Beban (*W*) tersebut jika mempunyai nilai maksimal 10. Hal ini dilakukan untuk lebih memudahkan dalam menghitungnya. Misal jika kasus media adaptasinya merupakan adaptasi video, maka *W<sub>CPU</sub>* dapat dimisalkan 7 dan *W<sub>RAM</sub>* dapat dimisalkan menjadi 3 mengingat proses konversi media membutuhkan porsi CPU lebih banyak daripada RAM. Dalam kasus ini nilai dari masing-masing adalah 0,7 dan 0,3. *CPU<sub>min</sub>* dan *RAM<sub>min</sub>* adalah kapasitas minimum untuk memproses sebuah pekerjaan. *CPU<sub>speed</sub>* dan *RAM<sub>size</sub>* merupakan kapasitas sumber daya dari sistem yang statis. *CPU<sub>load</sub>* dan *RAM<sub>usage</sub>* adalah sumber daya sistem yang terpakai saat ini. Setelah sumber daya dari CPU dan RAM dihitung, maka keseluruhan sumber daya sistem yang dipunyai dihitung dengan persamaan (Seunghwa and Eunseok, 2009):

$$Cur_{resources} = Cur_{CPU} + Cur_{RAM} \quad (4)$$

Setelah diketahui sumber daya yang tersedia pada masing-masing komputernya, pembagian dari tugas pengadaptasian pun diberikan berdasarkan jumlah sumber daya yang ada tersebut. Perhitungan tersebut dihitung berdasarkan persamaan berikut:

$$Job\ to\ handle = \frac{Cur_{res}}{sys_{res}} \times video_{duration} \quad (5)$$

Misal pada suatu sistem dipergunakan untuk mengkonversi suatu video dengan durasi 4 menit (240 detik). Sistem tersebut terdiri dari 5 buah *server* sebagai *worker* dengan jumlah

sumber daya yang tersedia masing-masing 4, 6, 8, 3, dan 6. Maka porsi pekerjaan yang akan dilakukan di masing-masing *worker* adalah:

$$\text{server}_1 = \frac{4}{27} \times 240 = 35,56 = 36 \text{ second}$$

$$\text{server}_2 = \frac{6}{27} \times 240 = 53,33 = 53 \text{ second}$$

$$\text{server}_3 = \frac{8}{27} \times 240 = 71,11 = 71 \text{ second}$$

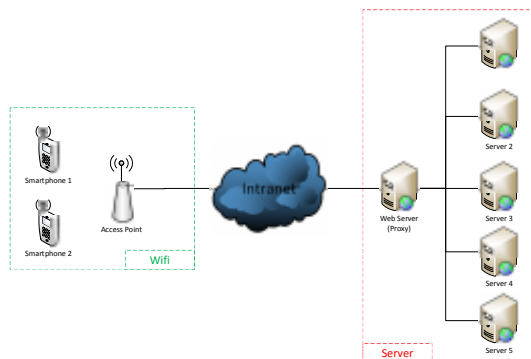
$$\text{server}_4 = \frac{3}{27} \times 240 = 26,67 = 27 \text{ second}$$

$$\text{server}_5 = \frac{6}{27} \times 240 = 53,33 = 53 \text{ second}$$

Setelah pembagian tugas dihitung, maka video tersebut diproses sesuai porsinya pada masing-masing *worker* dalam sistem. Setelah itu, video yang telah diproses pada masing-masing *worker* digabung kembali oleh *proxy*. Setelah video tersebut digabungkan, barulah video tersebut dikirimkan kepada *client* yang memintanya.

### III. Testbed

Metode yang diajukan dalam penelitian ini akan diujikan dalam lingkungan sistem yang sesungguhnya. Adapun detail lingkungan pengujian/*testbed*-nya dapat digambarkan pada Gambar 2.



Gambar 2. Topologi Jaringan *Testbed*

Dua buah ponsel pintar yang digunakan dalam penelitian ini adalah ASUS Zenfone 5 (A500CG) yang memiliki resolusi layar 720x1280 pixel dan Samsung Galaxy Grand (i9082) yang memiliki resolusi layar 480x800 pixel.

Setiap *server* dalam lingkungan pengujian ini merupakan *server* virtual yang dibuat menggunakan aplikasi virtualisasi VMware vSphere Hypervisor 5.5 Update 2. Setiap *server* dalam lingkungan pengujian ini memiliki 1 inti *processor* virtual, 1GB *memory*, 30 GB *hardisk*, dan menjalankan sistem operasi Debian 7 32-bit. *Server* virtual tersebut berjalan di atas *hardware* fisik sebuah komputer yang memiliki 8 inti *processor* dengan kecepatan

4,0GHz (AMD FX-8350), 8GB *memory*, dan 500GB *harddisk*.

### IV. Pengujian

Dalam pengujian ini kasus yang akan diujikan adalah pengadaptasian sebuah halaman web yang berisi sebuah video dengan durasi 5 menit dengan resolusi 1080p. Halaman tersebut akan di akses menggunakan dua buah ponsel pintar sesuai dengan *testbed*.

Masing-masing *server* dan *proxy* dalam lingkungan pengujian ini di beri beban secara manual untuk mensimulasikan jumlah sumber daya yang terpakai di pada masing-masing komputer tersebut. Adapun beban dari masing-masing *server* dan *proxy* ditunjukkan pada Tabel 1.

Tabel 1. Simulasi Beban Lingkungan Pengujian

Komputer	Kecepatan CPU (GHz)	Beban CPU (%)	Ukuran RAM (MB)	RAM Terpakai (%)
Proxy	4,0	30	1024	20
Server 1		60		50
Server 2		40		30
Server 3		80		70
Server 4		20		10
Server 5		50		40

Yang akan diukur adalah waktu respon dari sistem terhadap permintaan *client*. Pengukuran waktu respon dihitung dengan membandingkan waktu ketika permintaan dari *client* mulai dikirim dengan waktu ketika permintaan dikirimkan kepada *client*.

### V. Hasil Pengujian

Pengujian dengan dua resolusi layar yang berbeda ditujukan untuk mengetahui sejauh mana proses adaptasi menggunakan algoritma yang diusulkan, dapat berjalan sesuai dengan kriteria yang dimiliki oleh *client*. Gambar 3 menunjukkan *header* yang diterima oleh *server* pada setiap HTTP *request* yang dikirimkan oleh *client*.

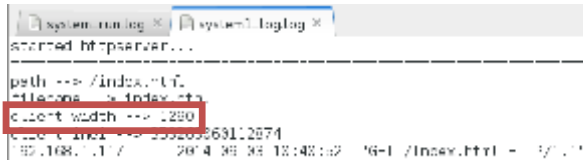
```

system running * | system: loglog *
started httpserver...
-----
path --> /index.html
hostname --> index.html
client width --> 1280
client local --> 353263063112074
10.108.1.1/ 2014-05-05 12:48:02 16-1 /index.html - 1/1 298

```

Gambar 3. *Header* yang diterima oleh *server* pada setiap HTTP *request*

Pada Gambar 3, terdapat *header* berupa resolusi dari layar yang dimiliki oleh *client*. *Header* tersebut dipakai oleh *server* untuk melakukan proses adaptasi video. Pada Gambar 4 terlihat bahwa *client* (yang merupakan perangkat bergerak pertama) memiliki resolusi layar dengan lebar 1280 *pixel*.



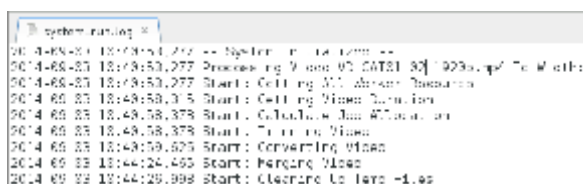
```

system running * system1.log.log *
started httpserver...
-----
path --> /index.html
filesize --> index.html
client width == 1280
client height == 720
2014-09-03 18:43:09.69112674
80.168.1.177 2014-09-03 18:43:02 '6-1 /index.html - 1/1'

```

Gambar 4. *Header* resolusi layar pada perangkat bergerak pertama

Kemudian proses adaptasi video dilakukan oleh *server* berdasarkan resolusi layar tersebut. Proses adaptasi tersebut dimulai dengan mendapatkan sumber daya yang ada pada setiap *worker* proses adaptasi dan mendapatkan total durasi dari video yang akan diadaptasi. Kemudian setelah itu akan dilakukan proses perhitungan porsi pekerjaan untuk masing-masing *worker*-nya. Setelah semua terhitung video akan dipotong oleh *server* sesuai dengan porsi dari masing-masing *worker*. Setelah itu proses konversi video akan dilakukan/dijalankan pada setiap *worker*-nya. Setelah setiap *worker* selesai mengkonversi, *server* akan menggabungkan video tersebut menjadi 1 buah video utuh untuk kemudian siap dikirimkan kepada *client*. Catatan sistem (*log*) dari proses-proses tersebut dapat dilihat pada Gambar 5.



```

system running *
2014-09-03 18:43:01.272 -- System: P...
2014-09-03 18:43:01.277 Processing Video #0 [DATA: 30] [920x640] To Width:
2014-09-03 18:43:01.277 Start: Getting Video Resources
2014-09-03 18:43:01.375 Start: Getting Video Duration
2014-09-03 18:43:01.378 Start: Calculating Video Allocation
2014-09-03 18:43:01.378 Start: Getting Video
2014-09-03 18:43:01.405 Start: Converting Video
2014-09-03 18:43:01.495 Start: Merging Video
2014-09-03 18:43:01.998 Start: Clearing Jobs - Done

```

Gambar 5. *Log* proses konversi video untuk perangkat bergerak pertama

Setelah proses konversi dilakukan, video hasil pengadaptasiannya akan dikirimkan oleh *server* kepada *client* dan disajikan oleh *client* seperti pada Gambar 6.



Gambar 6. Hasil yang ditampilkan pada perangkat bergerak pertama

Hal yang sama juga berlaku pada *request* yang dilakukan oleh perangkat bergerak kedua. *Header* berupa resolusi dari layar yang dimiliki oleh *client*. tersebut dipakai oleh *server* untuk melakukan proses adaptasi video. Pada Gambar 7 terlihat bahwa *client* (yang merupakan perangkat bergerak kedua) memiliki resolusi layar dengan lebar 800 *pixel*. Dan proses adaptasi videonya dapat dilihat pada catatan sistem (*log*) pada Gambar 8. Hasil dari proses pengadaptasian tersebut akan dikirimkan oleh *server* kepada *client* dan disajikan oleh *client* seperti pada Gambar 9.

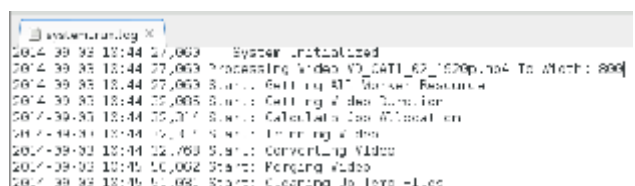


```

system running * system1.log.log *
started httpserver...
-----
path --> /index.html
filesize --> index.html
client width == 800
client height == 480
2014-09-03 18:43:09.69112674
80.168.1.177 2014-09-03 18:43:02 '6-1 /index.html - 1/1'

```

Gambar 7. *Header* resolusi layar pada perangkat bergerak kedua



```

system running *
2014-09-03 18:44 27,069 System: Initializ...
2014-09-03 18:44 27,069 Processing Video #0 [DATA: 60] [640x480] To Width: 800
2014-09-03 18:44 27,069 Start: Getting Video Resources
2014-09-03 18:44 27,069 Start: Getting Video Duration
2014-09-03 18:44 27,077 Start: Calculating Video Allocation
2014-09-03 18:44 27,114 Start: Getting Video
2014-09-03 18:44 27,269 Start: Converting Video
2014-09-03 18:44 27,069 Start: Merging Video
2014-09-03 18:44 27,069 Start: Clearing Jobs - Done

```

Gambar 8. *Log* proses konversi video untuk perangkat bergerak kedua



Gambar 9. Hasil yang ditampilkan pada perangkat bergerak kedua

## VI. Kesimpulan

Perangkat bergerak sekarang in tidak lagi menjadi kebutuhan sekunder, melainkan telah menjadi kebutuhan primer dalam mengakses sebuah informasi. Lebih dari 80% websitediakses menggunakan perangkat bergerak. Keterbatasan resolusi layar yang dimiliki oleh perangkat bergerak merupakan salah satu kendala pengguna untuk mengakses informasi, karena tidak semua website dengan konten video dapat ditampilkan dengan baik oleh semua perangkat. Dengan penelitian ini, telah dibuktikan bahwa konten video dapat di adaptasi sesuai dengan resolusi layar yang dimiliki oleh *client*. Proses adaptasi tersebut dapat dilakukan dengan metode distribusi beban pada setiap pekerjaanya.

Saran pengembangan selanjutnya adalah untuk menyisipkan informasi lain pada HTTP *header* yang dapat memaksimalkan proses adaptasi agar dapat ditampilkan dengan lebih baik oleh *client* seperti kapasitas memori yang tidak terpakai, daya baterai yang tersisa dan lain sebagainya.

## VII. Referensi

- [1] Andrews, G.R., 1999. Foundations of Multithreaded, Parallel, and Distributed Programming, 1 Edition. ed. Addison-Wesley.
- [2] Baolin, L., Shuai, X., Liang, A., Li, T., 2007. Handheld Device Oriented Adaptation Disposal for Web Access, in: IFIP International Conference on

- Network and Parallel Computing Workshops, 2007. NPC Workshops. Presented at the IFIP International Conference on Network and Parallel Computing Workshops, 2007. NPC Workshops, pp. 721–726. doi:10.1109/NPC.2007.94
- [3] Barton, J.J., Zhai, S., Cousins, S.B., 2006. Mobile Phones Will Become The Primary Personal Computing Devices, in: 7th IEEE Workshop on Mobile Computing Systems and Applications, 2006. WMCSA '06. Proceedings. Presented at the 7th IEEE Workshop on Mobile Computing Systems and Applications, 2006. WMCSA '06. Proceedings, pp. 3–9. doi:10.1109/WMCSA.2006.17
- [4] CBSNewsCBS/APFebruary, 2010. Number of Cell Phones Worldwide Hits 4.6B [WWW Document]. URL <http://www.cbsnews.com/news/number-of-cell-phones-worldwide-hits-46b/> (accessed 3.13.14).
- [5] Laakko, T., Hiltunen, T., 2005. Adapting Web Content to Mobile User Agents. IEEE Internet Comput. 9, 46–53. doi:10.1109/MIC.2005.29
- [6] Lum, W.Y., Lau, F.C.M., 2002. A Context-Aware Decision Engine for Content Adaptation. IEEE Pervasive Comput. 1, 41–49. doi:10.1109/MPRV.2002.1037721
- [7] Schmiedl, G., Seidl, M., Temper, K., 2009. Mobile Phone Web Browsing: A Study on Usage and Usability of the Mobile Web, in: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '09. ACM, New York, NY, USA, pp. 70:1–70:2. doi:10.1145/1613858.1613942
- [8] Seunghwa, L., Eunseok, L., 2009. Distributed Adaptation System for Quality Assurance of Web Service in Mobile Environment. J. Inf. Sci. Eng. 25.