

VERTICAL HANDOVER BERBASIS ANDROID PADA APLIKASI STREAMING STORED VIDEO

Bagus Seta Inba C, Waskitho Wibisono
Jurusan Teknik Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember
Jl. Kampus ITS Keputih, Surabaya 60111, Jawa Timur, Indonesia

ABSTRACT

The development of mobile devices, including a device that has more than one interface so that it can choose networks connected to. Vertical handover is a handover ability to access the connection between different networks, to maintain the quality of the processes that are using the network connection. Vertical handover (VH) usually occur between a network that has a large bandwidth, low data-rate but has a small range, such as WLAN and always-on, low-data-rate with a broad range of areas, such as HSDPA. Android-based devices is one device that can perform the VH. One of the processes that require a stable connection is streaming video. In this paper, the authors propose an improved model of vertical handover Android-based decision-making on network latency conditions, the battery level and the condition of the buffer of video streaming. Results of the test showed that there is a decrease on handover occurrence and delay stream at 45-50%. But there is a rise 33-63% in the duration of the handover because the parameters obtaining suggested.

Keywords: vertical handover, mobile switching, video streaming, Android

ABSTRAK

Perkembangan perangkat mobile, diantaranya adalah adanya perangkat yang memiliki lebih dari satu antarmuka sehingga dapat melakukan koneksi dari jaringan yang terhubung dengan dirinya. Vertical handover merupakan kemampuan penyerahan akses koneksi antar jaringan yang berbeda, untuk mempertahankan kualitas dari proses yang sedang menggunakan koneksi jaringan. Perangkat berbasis Android merupakan salah satu perangkat yang bisa melakukan VH. Pada paper ini, penulis mengusulkan model perbaikan vertical handover berbasis Android dengan penentuan keputusan berdasarkan kondisi latency jaringan, level baterai dan kondisi buffer pada saat melakukan streaming video. Hasil dari pengujian menunjukkan bahwa terjadi penurunan delay streaming dan jumlah handover sebesar 45-50%. Namun terdapat kenaikan durasi proses handover sebesar 33-63%.

Kata Kunci: vertical handover, mobile switching, streaming video, Android

PENDAHULUAN

Peningkatan pada pengembangan teknologi jaringan nirkabel membawa pada keberagaman solusi akses dari jaringan [1]. Pengguna selalu menginginkan koneksi terbaik pada layanan jaringan dan internet, dan dengan adanya perkembangan layanan seperti *cloud computing*, *file sharing*, *video streaming* dapat menjelaskan semakin pentingnya akses dimana saja [2]. Banyak diantara perkembangan dari teknologi nirkabel (2G, 3G, WLAN, WMAN), dengan diikuti perkembangan *mobile terminal* (MT) yang pada umumnya memiliki lebih dari satu *network interface*, menjadikan pengguna dapat melakukan akses dari layanan hampir dari jaringan manapun yang terkoneksi dengan dirinya[3].

Kemampuan untuk menyerahkan akses koneksi antar jaringan yang beraneka ragam tanpa mengganggu jalannya percakapan (video atau suara) disebut *seamless vertical handover*. *Vertical handover* (VH) berlangsung antara jaringan dengan dua teknologi jaringan nirkabel yang berbeda, biasanya antara jaringan yang memiliki *bandwith* yang besar, *data-rate* rendah akan tetapi memiliki jangkauan yang kecil, seperti WLAN dengan jaringan yang *always-on*, *data-rate* rendah dengan jaringan yang memiliki jangkauan area yang luas, seperti UMTS, HSDPA[4].

Salah satu tantangan utama pada proses VH adalah kepuasan dari pengguna terhadap kondisi jaringan yang dipilih. Integrasi jaringan tersebut diharapkan mampu memberikan akses yang terbaik

sesuai dengan kebutuhan mereka. Algoritma VH yang diimplementasikan merupakan hal yang masih banyak dikembangkan oleh para peneliti. Penelitian sebelumnya yang dilakukan antara lain VH berbasis QOS pada perangkat Android [5] dan VH berbasis RSS menggunakan testbed nyata pada jaringan UMTS dan Wi-Fi [6]. Dari kedua penelitian ini penelitian pertama diusulkan sebuah *platform* NRASP yang bekerja pada layer transport dan mengatur terjadinya *handover* dengan memperhitungkan RSS (*Received signal strength*) dan QOS sebagai parameter. Akan tetapi terdapat masalah performa dan kesulitan dalam implementasi dari *platform* yang diusulkan. Penelitian kedua, melakukan eksperimen berdasarkan *testbed* nyata dengan Wi-Fi (Guglielmo) dan UMTS (Telecom Italia). Namun terdapat permasalahan yang dihadapi, yaitu lamanya proses *handover*. Penulis ingin mengusulkan perbaikan penentuan keputusan *vertical handover* dengan berdasarkan kedua penelitian tersebut dengan penentuan VH berdasarkan kualitas latency jaringan dan penyesuaian terhadap kondisi *buffer* pemutar video *streaming stored*, pada perangkat Android.

Streaming video adalah sebuah teknologi untuk memainkan berkas video secara langsung ataupun dengan *pre-recorder* dari sebuah mesin server (*webserver*). Dengan kata lain, berkas video yang terletak dalam sebuah server dapat secara langsung dimainkan pada *browser* saat proses *buffering* mulai berjalan. Sistem akan membaca informasi dari *buffer* dan tetap melakukan proses unduhan berkas, sehingga proses *streaming* tetap berlangsung ke komputer *client*. Hal ini tentunya menyebabkan adanya penggunaan koneksi secara kontinyu, yang tentu saja kualitasnya dipengaruhi oleh proses *handover*.

Permasalahan yang diangkat dalam penelitian ini adalah terjadinya *vertical handover* pada saat melakukan *streaming* video yang sebenarnya tidak dibutuhkan. Terdapat kondisi – kondisi *handover* pada saat *streaming* yang bisa dihilangkan, sehingga penggunaan *resource* yang tidak diperlukan sebenarnya dapat dihindari. Sehingga penggunaan *resource* yang tidak diperlukan sebenarnya dapat dihindari. Untuk mengatasi masalah ini, penulis mengusulkan penentuan keputusan *vertical handover* dengan berdasar kondisi jaringan dengan diintegrasikan pada penyesuaian aplikasi video *streaming*.

Manfaat yang ingin dicapai dengan diusulkannya penelitian ini adalah meningkatkan efektifitas penggunaan sumber daya pada *mobile terminal* dan menjaga kualitas *streaming* video melalui keputusan terjadinya *vertical handover* sesuai dengan kondisi yang diinginkan pengguna.

TINJAUAN PUSTAKA

Vertical handoff adalah perubahan tipe konektivitas dari titik jaringan yang digunakan untuk mengakses infrastruktur pendukungnya, biasanya untuk mendukung pergerakan. Algoritma VH adalah proses yang dilakukan oleh perangkat untuk membuat keputusan *vertical handover* dan memilih jaringan target dari jaringan kandidat [7]. Sebuah Laptop kemungkinan bisa menggunakan *wireless* LAN dan teknologi seluler untuk akses internet. *Vertical handover* merupakan perpindahan dari satu teknologi ke teknologi lain untuk mempertahankan komunikasi.

Handover berdasarkan titik akses jaringan tersebut berasal, dapat dibedakan menjadi horisontal atau vertikal. *Horizontal handover* terjadi antara dua titik dengan dukungan teknologi yang sama, contohnya antara dua *base station* yang bertetangga pada jaringan seluler. Sedangkan, *vertical handover* terjadi antara dua titik dengan teknologi jaringan yang berbeda, contohnya antara *access point* IEEE 802.11 dengan *base station* jaringan seluler [3]. Secara umum tujuan *handover* [8] adalah:

1. Menjamin keberlangsungan layanan ketika perangkat bergerak melintasi batas *cell*.
2. Mempertahankan *Quality of Service*
3. Meminimalisir level interferensi dari keseluruhan sistem dengan cara menjaga *mobile celluler* agar tetap terhubung ke satu atau lebih *base station* yang lebih baik dari *base station* yang lain
4. *Roaming* antar jaringan yang berbeda

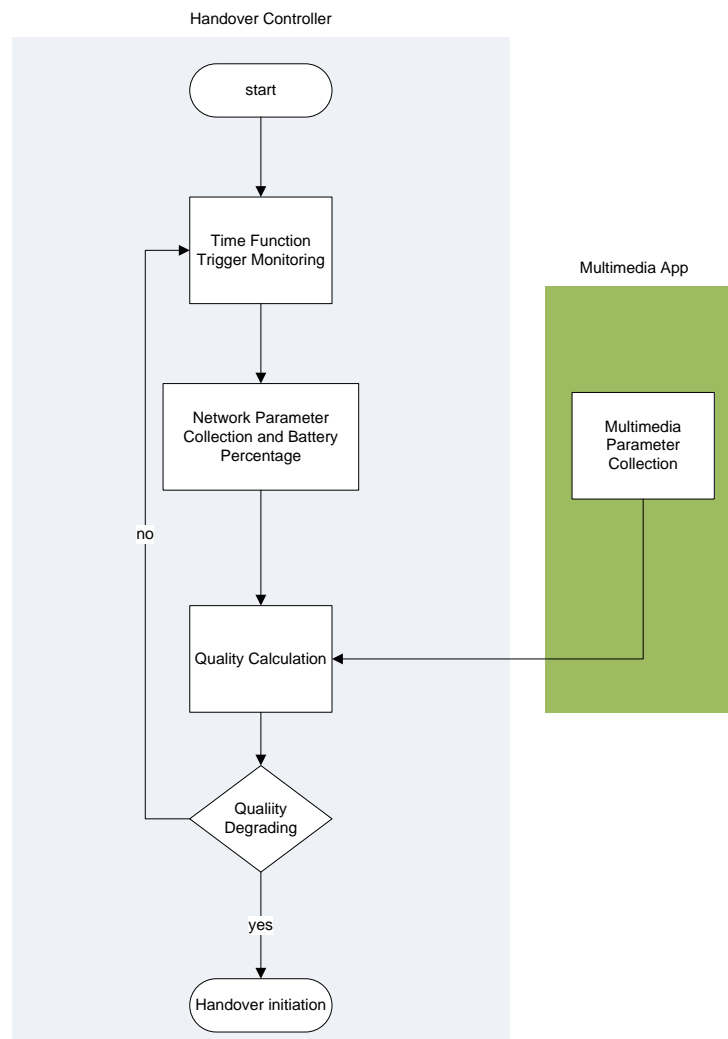
Proses *handover* terdiri dari tiga tahap, yaitu: *handover measurement*, penentuan *handover*, dan *channel assignment*. *Handover measurement* dilakukan untuk mengumpulkan informasi untuk identifikasi dibutuhkannya *handover*. Penentuan *handover* meliputi pemilihan titik koneksi dengan

memilih sambungan yang sesuai dengan kriteria yang diinginkan. *Channel assignment* berkaitan dengan pembentukan hubungan dengan titik koneksi yang baru

METODE

Algoritma *handover* yang dirancang ini merupakan algoritma *vertical handover* menggunakan penyesuaian berdasarkan kualitas jaringan yang diwakili oleh *latency* dari jaringan, *power consumption*, dan kondisi *buffer streaming* video dengan berbasiskan *platform* Android. Gambaran umum algoritma secara umum dapat dilihat pada Gambar 1. Algoritma ini dimulai ketika pengguna memulai *service* atau layanan *handover*, fungsi *handover* akan dijalankan setiap fungsi waktu terpicu, setiap waktu tunggu yang ditentukan. Kemudian, akan dilakukan pembacaan kondisi jaringan dan kondisi baterai pada saat itu, serta dari luar layanan *handover* atau dari pemain video *streaming* didapatkan kondisi *buffer* video yang dicapainya saat ini. Data – data kondisi ini akan dihitung sesuai dengan persamaan perubahan yang dijelaskan pada persamaan (1), kemudian akan didapatkan nilai kualitas keadaan saat ini.

Melalui proses penghitungan akan didapatkan kualitas kondisi yang digunakan untuk menentukan apakah kondisi saat ini menurun. Jika kondisi saat ini menurun, maka akan dilakukan *handover* ke jaringan yang lain.



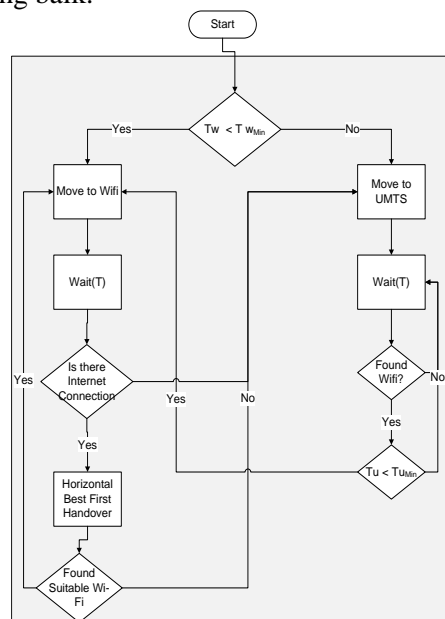
Gambar 1. Desain algoritma secara umum

$$T_i = w_l \frac{l}{l_{max}} + w_c(1 - c) + w_v(1 - \frac{V}{V_{max}}) \quad \dots\dots (1)$$

dimana :

- T_i = Parameter Kondisi ke-i
- w_l = Bobot parameter l
- l = *latency* jaringan
- l_{max} = *latency* maximal
- w_c = Bobot parameter c
- c = *power percentage*
- w_v = Bobot parameter v
- V_{max} = nilai maksimum kondisi *buffer* video
- V = nilai kondisi *buffer* video
- $w_l + w_c + w_v = 100$

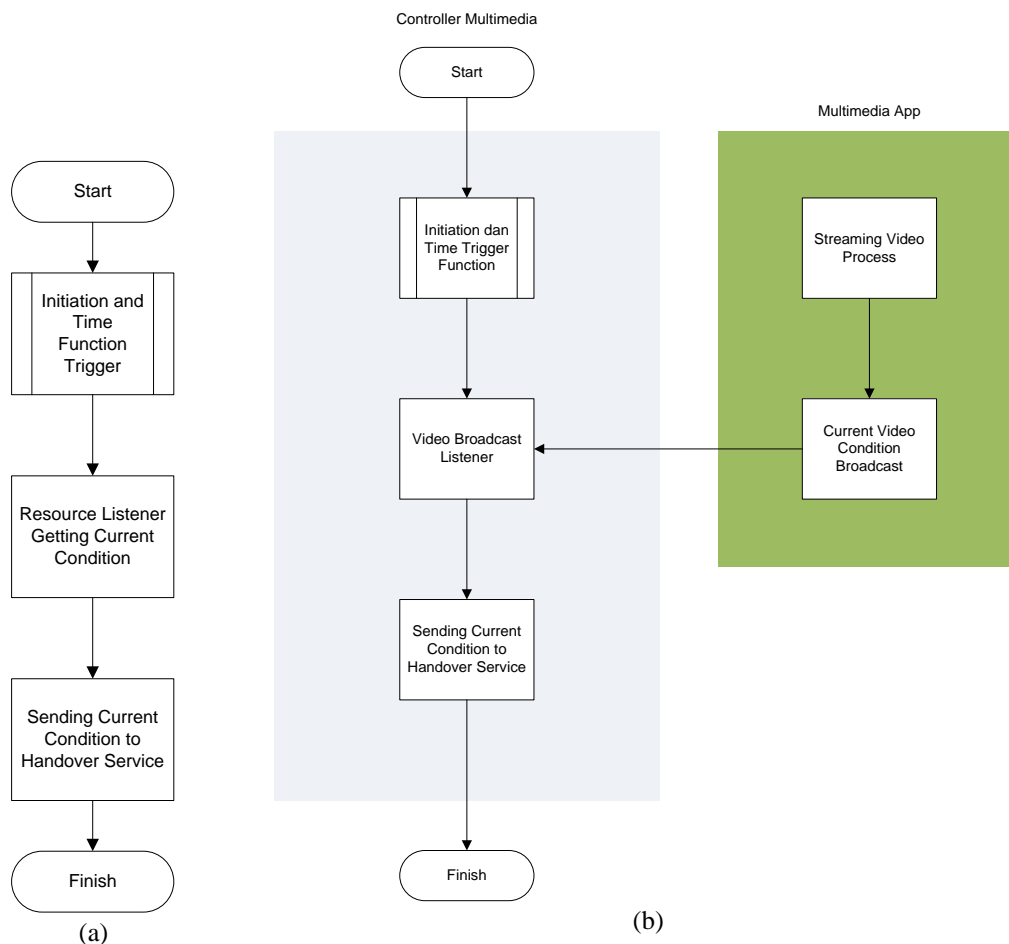
Secara detail proses perpindahan jaringan dijelaskan pada Gambar 2. Sebelum melakukan *handover* (inisiasi) dari satu jaringan ke jaringan lain, terdapat proses mendapatkan nilai dari kondisi jaringan dan aplikasi video dari luar modul *controller vertical handover*. Nilai – nilai tersebut dihitung sesuai dengan persamaan (1), dan akan didapatkan nilai kondisi (T_i) yang akan digunakan sebagai penentu terjadinya *handover*.. Pertama, inisiasi dari algoritma dianggap bahwa koneksi berada pada kondisi seluler, kemudian apabila terdapat koneksi Wi-Fi yang ditemukan maka dilihat nilai ambang batas “kondisi” dari Wi-Fi sudah memenuhi syarat minimal (T_{wMin}), jika sudah maka akan pindah ke Wi-Fi tersebut. Proses pindah ke Wi-Fi ini melalui proses selanjutnya (*horizontal best handover*), yaitu proses pemilihan koneksi Wi-Fi terbaik, apabila ada beberapa pilihan koneksi Wi-Fi. Jika sudah selesai waktu pengecekan atau waktu tunggu (t) yang dapat disesuaikan dengan keinginan pengguna. Untuk mengurangi kompleksitas dari algoritma pengecekan kondisi jaringan, dimanfaatkan nilai boolean dari *class connectivityManager.getActiveNetworkInfo().isConnected()* yang merupakan salah satu kelas yang ada pada Android, untuk pengecekan koneksi jaringan sudah terkoneksi dengan jaringan internet atau belum. Jika sudah terkoneksi, maka akan masuk ke tahapan berikutnya. Jika tidak maka akan pindah ke koneksi UMTS dan men-*disable* jaringan tersebut selama D waktu. Apabila T_{wMin} dan T_{uMin} tidak terpenuhi kedua duanya, maka akan dibandingkan antara keduanya, manakah yang paling baik.



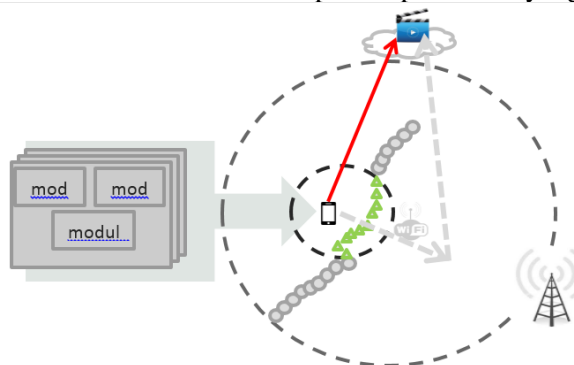
Gambar 2. Desain algoritma yang diusulkan

Metode Pengambilan Data

Proses mendapatkan data parameter yang digunakan sebagai pertimbangan terjadinya handover dijelaskan melalui Gambar 3. Dalam penelitian ini, data yang didapatkan adalah *latency*, prosentase penggunaan baterai dan kondisi *buffer* dari pemutar *streaming* video dari perangkat saat ini. Proses untuk mendapatkan kondisi *latency* jaringan, prosentase baterai dan *buffer* video ini tidak bisa dijadikan satu dengan *service* dari *handover*. *Service handover* hanya memproses data dan mengontrol proses *handover*. *Service* pada Gambar 3(a) menerima data *latency* dan baterai dari *activity* lain, melalui *listener* yang mempunyai wewenang untuk memperoleh data tersebut. Data lain yang perlu didapatkan untuk digunakan sebagai perhitungan *vertical handover* adalah kondisi *buffer* aplikasi *streaming* video, yang ditunjukkan pada Gambar 3(b). *Service* pada Gambar 3(b) menerima data *buffer* melalui *listener* dari aplikasi luar (pemutar video *streaming*) berupa data *broadcast*.



Gambar 3. Proses mendapatkan parameter yang diukur



Gambar 4. Skenario pengujian

Pengujian dilakukan melalui topologi pada Gambar 4, perangkat *mobile* tersambung dengan Wi-Fi dan UMTS. Pengujian direncanakan dengan perangkat *mobile* akan bergerak diantara *base station* dan disesuaikan dengan kondisi dari yang akan diujikan. Pemutar video akan melakukan *streaming* video dengan URL tertentu dan aplikasi *vertical handover* akan berjalan pada *background*. Pada saat aplikasi berjalan, akan dicatat pada berkas *log*, kondisi *latency*, kondisi *buffer* dan nilai T yang didapatkan. Dari berkas ini akan dilakukan analisa.

Topologi ini dibuat memanfaatkan jaringan yang sudah ada. *Vertical handover* menggunakan skenario *no-coupling* (tidak adanya campur tangan antara pengguna dan operator jaringan). Perangkat *mobile* sendiri yang mengatur terjadinya *handover* pada koneksinya.

Uji coba dilakukan dengan membandingkan dua mekanisme *handover*, yaitu menggunakan metode Bussaneli dan metode ini. Uji coba akan dilakukan untuk mengukur keberhasilan metode terhadap tolak ukur sebagai berikut:

1. *Delay Streaming*

Streaming Delay adalah durasi waktu jeda berhenti yang terjadi pada proses *streaming* video yang disebabkan adanya proses *handover* pada jaringan. Satuan yang digunakan adalah *millisecond* (ms).

2. Jumlah Kejadian *Handover*

Jumlah *handoff* yang terjadi mempengaruhi juga terhadap keberhasilan metode *handover*. Melalui proses *handoff*, seringnya aktivasi antarmuka, *mobile switching* yang dapat menyebabkan cukup pembuangan baterai.

3. Waktu Proses *Handover*

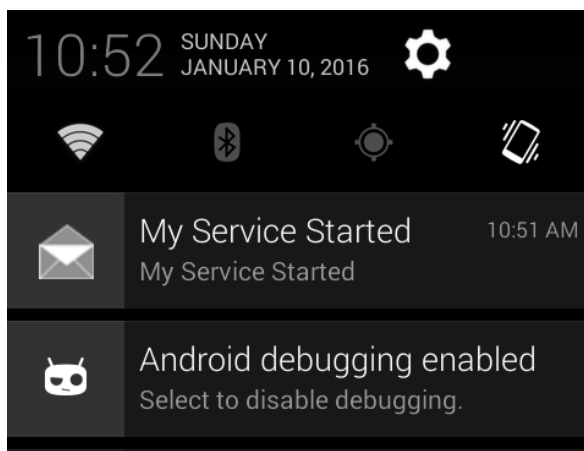
Waktu *handover* adalah durasi antara inisiasi dan waktu penyelesaian *handover*. Inisiasi berhubungan dengan pada saat koneksi akan berpindah dari satu koneksi ke koneksi lain. Waktu penyelesaian dihitung pada saat koneksi sudah berhasil berganti ke koneksi lain.

4. Penurunan Daya Baterai

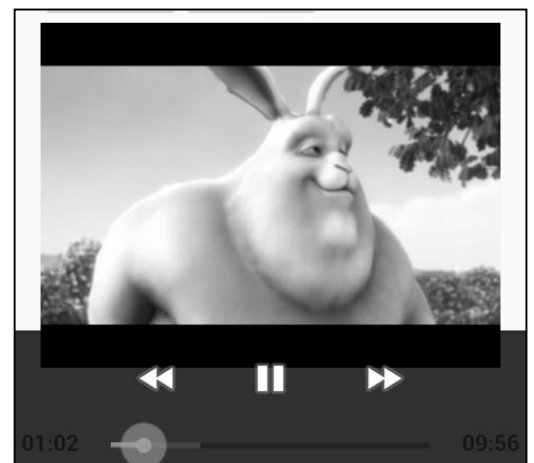
Uji coba pada parameter penurunan daya baterai merupakan representasi besaran presentase daya baterai yang digunakan oleh sistem dan diperlukan sebagai parameter output untuk mengetahui besar persentase penghematan daya baterai pada penelitian ini.

HASIL DAN PEMBAHASAN

Sistem diimplementasikan kedalam dua bagian aplikasi yaitu aplikasi yang bertugas untuk mengatur proses *handover* dan aplikasi pemutar video *streaming*. Implementasi algoritma pada aplikasi Android ditunjukkan melalui *screenshot* pada Gambar 5. Gambar 5 (a) merupakan *screenshot* ketika *service* pada aplikasi *handover* berjalan. Gambar 5(b) merupakan *screenshot* aplikasi pemutar video *streaming* berjalan. Pembagian kedalam dua bagian ini bertujuan agar sistem berjalan sesuai dengan keadaan sebenarnya.



(a)



(b)

Gambar 5. Implementasi algoritma pada aplikasi Android

Hasil Uji Coba Penghitungan *Delay Streaming*

Uji coba ini dilakukan untuk mengetahui *delay* yang terjadi pada proses pengunduhan data video ke *buffer* pada saat *streaming*, yang dikarenakan adanya proses *handoff* yang terjadi. Proses pengukuran ini memanfaatkan listener dari class Android yang diimplementasi *VideoView* salah satunya adalah listener dari *MediaPlayer* yaitu *setOnError-Listener* untuk dilihat jika terjadi *error* dengan nilai konstanta *MEDIA_ERROR_TIMED_OUT* terhadap waktu pengunduhan pada saat *streaming*, kemudian disimpan ke dalam berkas *log*. Uji coba dilakukan dengan membandingkan dua mekanisme *handover*, yaitu menggunakan metode Bussanelli dan metode ini. Ketika kedua mekanisme dibandingkan, melalui *log* masing-masing metode saat dijalankan, maka dapat diperoleh perbedaan *delay streaming* pada masing – masing metode. Video yang diputar pada aplikasi pemutar merupakan video *sample* dengan durasi kurang lebih 10 menit. Video diputar beberapa kali untuk diketahui perbandingan dari masing – masing metode.

Tabel 1. Perbandingan *delay* pada dua metode

| N | total <i>delay</i> metode Bussanelli (s) | total <i>delay</i> metode yg diusulkan (s) |
|-----------|--|--|
| 1 | 4 | 2 |
| 2 | 2 | 2 |
| 3 | 2 | 0 |
| 4 | 5 | 0 |
| 5 | 1 | 0 |
| rata-rata | 2.8 | 0.16 |

Perbandingan hasil uji coba pengukuran *delay* ditunjukkan pada Tabel 1. Dari metode Bussanelli diperoleh rata – rata *delay streaming* adalah 2.67 detik lebih banyak dibandingkan dengan metode yang diusulkan dengan rata – rata *delay* diperoleh 1.33 detik. Hal ini dikarenakan pada metode yang diusulkan, pada kondisi jaringan dilihat dari kondisi sebenarnya bukan hanya dilihat dari *signal strength* yang belum tentu merepresentasikan kondisi jaringan. Terkadang terdapat kondisi jaringan dengan RSS tinggi akan tetapi dalam kondisi yang padat, maka akan mengganggu jalannya *streaming*.

Hasil Uji Coba Penghitungan Jumlah Kejadian *Handover*

Uji coba ini bertujuan untuk menghitung jumlah *handoff* yang terjadi. Melalui proses *handoff*, seringkali aktivasi antarmuka, *mobile switching* yang dapat menyebabkan cukup pembuangan baterai. Skenario pengujian yang digunakan untuk menghitung keberhasilan metode ini adalah penggunaan kedua metode terhadap waktu percobaan. Pada saat dijalankan akan ditambahkan variabel *counter* yang akan bertambah setiap kali *handoff* berlangsung. Kedua mekanisme dibandingkan melalui *log* masing-masing metode saat dijalankan, maka dapat diperoleh perbedaan jumlah *handoff* yang terjadi.

Tabel 2. Perbandingan jumlah kejadian *handover*

| N | Kejadian <i>Handover</i> | |
|-----------|--------------------------|-------------------------|
| | metode Bussanelli (s) | metode yg diusulkan (s) |
| 1 | 7 | 4 |
| 2 | 7 | 3 |
| 3 | 6 | 3 |
| 4 | 8 | 4 |
| 5 | 7 | 4 |
| rata-rata | 7 | 3.6 |

Perbandingan hasil uji coba pengukuran jumlah kejadian *handover* dapat dilihat pada Tabel 2, dapat dilihat bahwa jumlah kejadian *handover* pada metode Bussaneli memiliki jumlah rata – rata 7 kali *handover*. Metode yang diusulkan memiliki jumlah *handover* yang lebih baik yaitu 3.6 kali sehingga perpindahan atau *mobile switching* lebih jarang terjadi. Hal ini dikarenakan pergerakan *mobile terminal* pada jaringan sehingga memicu perubahan kekuatan *signal*, sehingga menyebabkan seringnya terjadi *handoff* pada metode Bussaneli.

Hasil Uji Coba Penghitungan Durasi *Handover*

Uji coba ini dilakukan untuk mengetahui pengaruh mekanisme perbaikan *vertical handover* ini terhadap durasi waktu *handoff*. Pengujian yang digunakan untuk mengukur parameter ini adalah penggunaan metode Bussaneli dan metode yang diusulkan dengan N kali percobaan. Ketika kedua mekanisme dijalankan durasi pada saat terjadi *handoff* disimpan ke dalam berkas *log* masing-masing metode. Sehingga dapat diperoleh hasil durasi *handoff* dari kedua metode.

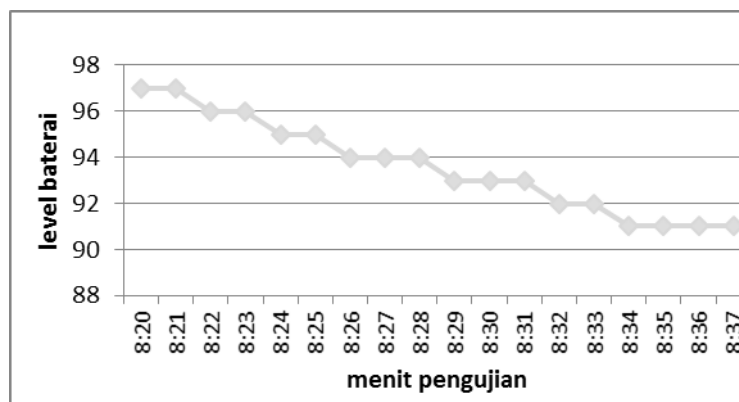
Dapat dilihat pada Gambar 6, menunjukkan bahwa perbandingan durasi *handover* pada kedua metode. Rata – rata durasi metode Bussaneli yaitu 11 s, sedangkan metode yang diusulkan 18 s. Metode yang diusulkan memiliki durasi yang lama ketika berpindah dari UMTS ke Wi-Fi, hal ini dikarenakan perlunya pengalokasian alamat DHCP yang menambah waktu, dan juga penghitungan kondisi *latency*.

Gambar 6. Perbandingan durasi *handover*

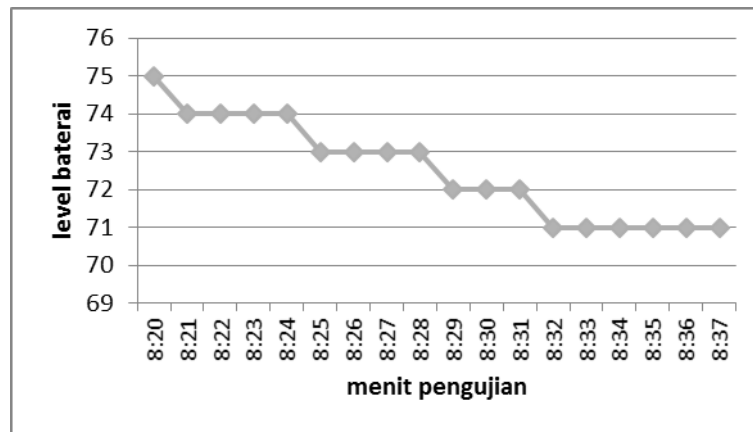
Hasil Uji Coba Penurunan Daya Baterai

Uji coba ini dilakukan untuk mengetahui pengaruh mekanisme perbaikan *vertical handover* terhadap penurunan daya baterai dan penghematan daya baterai yang dapat dicapai dengan adanya sistem perbaikan mekanisme *vertical handover*. Untuk mengetahui penurunan daya baterai dan penghematan yang dapat dicapai, dilakukan pengujian performa. Pengujian ini dilakukan terhadap variabel waktu untuk masing-masing mekanisme *vertical handover*.

Dapat dilihat pada Gambar 7 dan Gambar 8 perbandingan penurunan baterai diperoleh rata - rata yaitu penurunan baterai pada metode yang diusulkan yaitu 0.5% per menit, sedangkan metode Bussaneli 0.25% per menit. Hal ini disebabkan karena metode yang diusulkan melakukan koneksi terlebih dahulu dengan *access point* maupun BTS untuk mengetahui parameter – parameter yang dibutuhkan. Sedangkan untuk metode Bussaneli hanya membutuhkan kekuatan sinyal yang tidak memerlukan koneksi bisa didapatkan. Selain itu pengaruh jenis koneksi yang digunakan juga memengaruhi penurunan baterai, apabila lebih sering tersambung dengan Wi-Fi juga menyebabkan baterai akan lebih tahan lama, dibandingkan jika tersambung dengan koneksi UMTS.



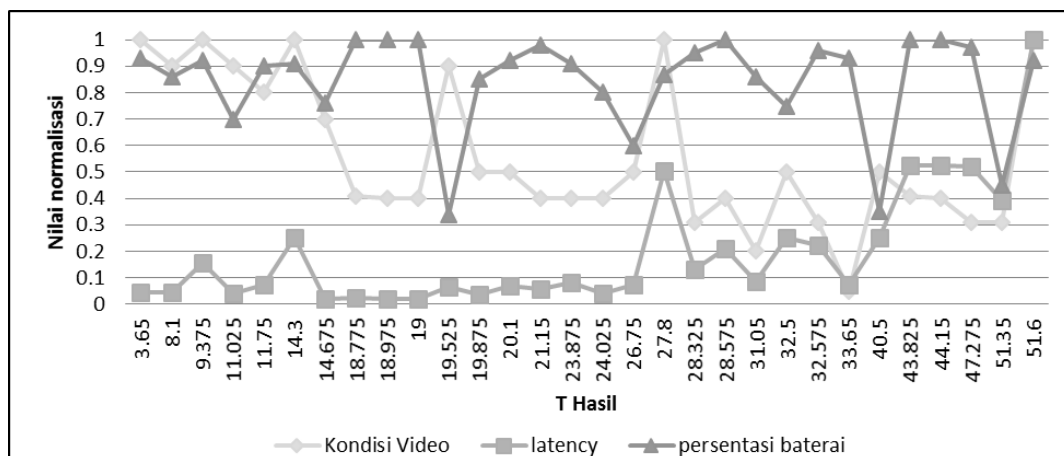
Gambar 7. Hasil penurunan level baterai pada metode yang diusulkan



Gambar 8. Hasil penurunan level baterai pada metode Bussaneli

Hasil Uji Coba Variasi Data Parameter *Latency*, *Level Baterai* dan *Buffer Streaming Video*

Pengujian ini berkaitan dengan proses *streaming* oleh aplikasi lain yang sudah diimplementasikan. Pengujian ini dilakukan untuk mengetahui pengaruh kondisi *latency* jaringan, baterai serta *buffer* video *streaming* terhadap pengambilan keputusan terjadinya *vertical handoff* apakah sudah sesuai dengan aturan yang sudah dibuat. Pengujian dilakukan sesuai dengan metode pengujian sebanyak 30 kali melakukan *streaming*. Hasil dari pengujian ditunjukkan pada Gambar 9. Uji coba yang dilakukan menggunakan konstanta masing – masing W_l, W_c, W_v , adalah 50, 20, dan 30. Hasil menunjukkan bahwa aplikasi yang dibuat sudah sesuai dengan aturan yang sudah dibuat.



Gambar 9. Hasil pengujian variasi parameter terhadap nilai T

KESIMPULAN

Pengujian dan analisis yang telah dilakukan menghasilkan beberapa kesimpulan penelitian, yaitu:

1. Perbaikan strategi *vertical handover* pada jaringan nirkabel heterogen berdasarkan parameter *latency* jaringan, level baterai dan kondisi *buffer* aplikasi *streaming* video dapat meningkatkan efektifitas penggunaan sumber daya pada *mobile terminal* dan menjaga kualitas *streaming* video. Hal ini dibuktikan dari keberhasilan mekanisme yang diusulkan dalam pengambilan keputusan untuk *handoff*
2. Hasil uji coba pengukuran *delay* menunjukkan bahwa metode Bussaneli diperoleh rata – rata *delay streaming* adalah 2.8 detik, dibandingkan dengan metode yang diusulkan dengan rata – rata *delay* diperoleh 0.16 detik, terjadi penurunan *delay streaming* sebesar 50%.

3. Hasil uji coba pengukuran jumlah kejadian *handover* menunjukkan bahwa jumlah kejadian *handover* pada metode Bussaneli memiliki jumlah rata – rata 7 kali kejadian *handover*. Metode yang diusulkan memiliki jumlah *handover* yang lebih baik yaitu 3.6 kali atau lebih sedikit 50% jika dibandingkan dengan metode Bussaneli.
4. Perbandingan durasi *handover* pada kedua metode menunjukkan bahwa rata – rata durasi metode Bussaneli yaitu 11 s, sedangkan metode yang diusulkan 18 s. Hal yang wajar jika terdapat kenaikan durasi dikarenakan kompleksitas dari metode yang diusulkan lebih jika dibandingkan dengan metode Bussaneli.
5. Uji coba penurunan baterai menunjukkan bahwa rata - rata penurunan baterai pada metode yang diusulkan yaitu 0.5% per menit, sedangkan metode Bussaneli 0.25% per menit.

DAFTAR PUSTAKA

- [1] Cardoso , Neves Ta, Ricardo Pa. 2011. Media independent handover management in heterogeneous access networks-an empirical evaluation. IEEE. 73rd: p. 1-5.
- [2] Silva R, Carvalano , Sousa P. 2011. Enabling Heterogeneous Mobility in Android Devices. Springer Science. Juni. p. 518-528.
- [3] Yan X, Y. AS, Narayanan S. 2010. A survey of vertical handover decision algorithm in Fourth Generation heterogeneous wireless networks. elsevier. 54: p. 1848-1863.
- [4] Ma L, Yu F, Leung VC, Randhawa T. 2004. A new method to support UMTS/WLAN vertical handover using SCTP. Wireless Communications, IEEE. 11: p. 44-51.
- [5] S.P , Anand S. 2011. A Novel Scalable Software Platform on Android for efficient QoS on Android. Wireless Telecommunications Symposium IEEE. p. 1-6.
- [6] Busanelli S, Martalo M, F G, S G. 2011. Vertical Handover between WiFi and UMTS Networks: Experimental Performance Analysis. International Journal of Energy, Information and Communications. Februari; 2(1).
- [7] Liu, Li, Sun, Lingfen., I, Emmanuel. 2015. A QOE-Driven Vertical Handover Algorithm Based On Media Independent Handover Framework. Eight International Workshop on Selected Topics in Mobile and Wireless Computing.
- [8] Akyildiz IF, xie J, Mohanty. 2004. A survey of mobility management in next-generation all-IP-based wireless sytems. IEEE. 11: p. 16-28.