



TESIS - KI142502

**Optimasi Strategi *Vertical Handover* pada Jaringan Nirkabel Heterogen untuk Menjaga Kualitas Layanan *Streaming Stored Multimedia***

Bagus Seta Inba Cipta  
NRP.5112201904

DOSEN PEMBIMBING  
Waskitho Wibisono, S.Kom., M.Eng., Ph.D

PROGRAM MAGISTER  
BIDANG KEAHLIAN KOMPUTASI BERBASIS JARINGAN  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2016





TESIS - KI142502

***OPTIMIZATION OF HANDOVER DECISION ALGORITHM IN  
HETEREGENOUS NETWORK TO MAINTAIN SERVICE  
STORED MULTIMEDIA STREAMING QUALITY***

Bagus Seta Inba Cipta  
NRP. 5112201904

SUPERVISOR  
Waskitho Wibisono, S.Kom., M.Eng., Ph.D

MAGISTER PROGRAMME  
INFORMATICS ENGINEERING DEPARTMENT  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2016

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrohmanirohim.

Alhamdulillahil'alamin, segala puji bagi Allah Subhanahu Wata'alla, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tesis yang berjudul "**Optimasi Strategi Vertical Handover pada Jaringan Nirkabel Heterogen untuk Menjaga Kualitas Layanan Streaming Stored Multimedia**" dengan baik.

Dalam pelaksanaan dan pembuatan Tesis ini tentunya sangat banyak bantuan-bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Kedua orang tua penulis, Bapak Bambang Mulyadi, Ibu Intyastuti, Adik Dessy Intan Anggun, dan Danil Tri Mulyadi yang telah memberikan dukungan moral, spiritual dan material, semangat, perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
3. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing, yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, perhatian, serta semua yang telah diberikan kepada penulis.
4. Teman-teman kontrakan pandugo, Andre, Sukma, Misbah, Sindung, Zarkasi, Dwi, Wisnu, yang sudah memberikan dukungan moril dan semangat.
5. Teman-teman seperjuangan bidang minat NCC terima kasih atas kebersamaannya selama ini.
6. Teman – teman admin NCC 2009 Arsyah, Cahya, Dedy, Anis, Sally, Ceri yang sudah memberikan dukungan semangat serta do'a.
7. Nuzulul Kurniawan I, yang sudah memberikan bantuan semangat serta do'a.
8. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tesis ini.

"Tiada Gading yang Tak Retak", begitu pula dengan Tesis ini. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Januari 2016

Penulis

**[Halaman ini sengaja dikosongkan]**

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

Bagus Seta Inba Cipta

Nrp. 5112201904

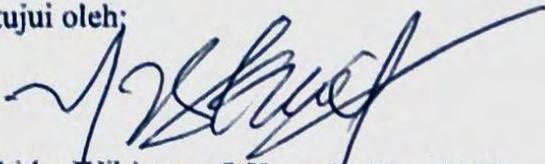
Dengan judul :

Optimasi Strategi Vertical Handover pada Jaringan Nirkabel Heterogen untuk Menjaga Kualitas Layanan Streaming Stored Multimedia

Tanggal Ujian : 19-1-2016

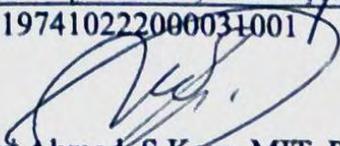
Periode Wisuda : 2015 Gasal

Disetujui oleh:



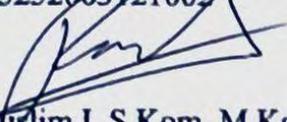
Waskitho Wibisono, S.Kom, M.Eng, Ph.D  
NIP. 197410222000031001

(Pembimbing 1)



Tohari Ahmad, S.Kom, MIT, Ph.D  
NIP. 197505252003121002

(Penguji 1)



Royyana Muslim I, S.Kom, M.Kom, Ph.D  
NIP. 197708242006041001

(Penguji 2)

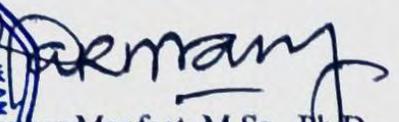


Baskoro Adi Pratomo, S.Kom, M.Kom  
NIP. 198702182014041001

(Penguji 3)



Direktur Program Pasca Sarjana,



Prof. H. Sunhar Manfaat, M.Sc., Ph.D.  
NIP. 196012021987011001

## **Optimasi Strategi Vertical Handover pada Jaringan Nirkabel Heterogen untuk Menjaga Kualitas Layanan Streaming Stored Multimedia**

Nama Mahasiswa : Bagus Seta Inba Cipta

NRP : 5112 201 904

Pembimbing : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

### **ABSTRAK**

Evolusi dari bidang komunikasi dengan adanya peningkatan pada pengembangan teknologi jaringan nirkabel membawa pada keberagaman solusi akses dari jaringan. Banyak dari teknologi nirkabel (2G, 3G, WLAN, WMAN, dll) dengan diikuti evolusi *mobile terminal* (MT) dengan lebih dari satu *network interface*, menjadikan pengguna dapat melakukan akses dari layanan hampir dari jaringan manapun yang terkoneksi dengan dirinya. Kemampuan untuk menyerahkan akses koneksi antar jaringan disebut *vertical handover*. *Vertical handover* (VH) biasanya berlangsung antara jaringan yang memiliki *bandwidth* yang besar, *data-rate* rendah akan tetapi memiliki jangkauan yang kecil, seperti WLAN dengan jaringan yang *always-on*, *data-rate* rendah dengan jangkauan area yang luas, seperti UMTS, HSDPA.

Salah satu tantangan utama pada proses VH adalah kepuasan dari pengguna terhadap kondisi jaringan yang dipilih. Salah satu aplikasi yang sering digunakan berkoneksi dengan jaringan saat ini adalah aplikasi *streaming* video. Kondisi jaringan yang baik tentu akan berpengaruh terhadap kondisi aplikasi *streaming*. Integrasi jaringan dengan menggunakan VH diharapkan mampu memberikan akses yang terbaik sesuai dengan kebutuhan pengguna.

Penulis ingin mengusulkan mekanisme perbaikan penentuan keputusan *vertical handover* dengan menggunakan parameter kualitas jaringan dan kondisi video *streaming stored*. Tujuan dari 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.

Penelitian menunjukkan bahwa mekanisme perbaikan yang diusulkan dapat meningkatkan performa *vertical handover*. Hal ini ditunjukkan dengan penurunan delay streaming sebesar 50%, penurunan jumlah *handover* yang terjadi rata-rata sebesar 50%. Namun terdapat kenaikan waktu *handover* dikarenakan proses pendapatan parameter tambahan yang diusulkan yaitu rata-rata 63%.

**Kata kunci:** *vertical handover, heterogenous wireless network, pemilihan jaringan, streaming video stored*

**[Halaman ini sengaja dikosongkan]**

**OPTIMIZATION OF HANDOVER DECISION ALGORITHM IN HETEREGENOUS  
NETWORK TO MAINTAIN SERVICE STORED MULTIMEDIA STREAMING QUALITY**

Student Name : Bagus Seta Inba Cipta  
NRP : 5112 201 904  
Supervisor : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

**ABSTRACT**

*The evolution of the field of communication by an increase in the development of wireless network technology bring to the diversity of network access solutions. Various wireless technologies ( 2G, 3G, WLAN, WMAN , etc.) followed by the evolution of the mobile terminal (MT) with more than one network interface, making the user can access from almost any network services connected with him. The ability to submit access connections between heterogeneous networks called vertical handover. Vertical handover (VH) usually lasts between networks that have a large bandwidth, low data-rate but has a small range, such as WLAN network always-on, low- data-rate with a wide range of areas, such as UMTS, HSDPA .*

*One of the major challenges in the process of VH is the satisfaction of the users with the network conditions selected. One of the application that is often used to connect to network nowadays is streaming video. Good network conditions would affect the conditions of video streaming application. Network integration by using VH expected to provide access to the needs of the user.*

*The author would like to propose a mechanism to improve vertical handover decision by using network quality parameters and conditions of application streaming stored video. The purpose of this research is to improve the effectiveness of resource usage in mobile terminals and maintain the quality of streaming video over the vertical handover decision in accordance with the conditions that the user wants.*

*The results show that the proposed mechanism can improve the performance of vertical handover. This is shown by a decrease in streaming delay by 50%, reduction in the number of handovers occurrence by 50%. However, there is an increase handover time on average 60% because the process parameters retrieval suggested.*

**Keywords:** *vertical handover, heterogenous wireless network, network selection, streaming stored video*

**[Halaman ini sengaja dikosongkan]**

## DAFTAR ISI

ABSTRAK .....	iii
ABSTRACT .....	v
KATA PENGANTAR .....	vii
DAFTAR ISI .....	ix
DAFTAR TABEL .....	xv
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	2
1.3. Tujuan .....	2
1.4. Manfaat .....	3
1.5. Kontribusi Penelitian .....	3
1.6. Batasan Masalah .....	3
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA .....	5
2.1. <i>Vertical Handover</i> (VHO) .....	5
2.2. Arsitektur Android Pada Jaringan Heterogen .....	7
2.3. <i>Android Service</i> .....	8
2.4. Streaming Stored Multimedia .....	9
2.5. Kondisi Jaringan : <i>Latency</i> .....	10
2.6. Power Condition .....	10
2.7. Kondisi Video Streaming (kondisi <i>buffer</i> ) .....	11
2.8. Metode <i>Vertical Handover</i> Berbasis RSS ( <i>Received Signal Strength</i> ) .....	12
BAB 3 METODOLOGI PENELITIAN .....	13
3.1. Perumusan Masalah .....	13
3.2. Studi Literatur .....	14
3.3. Desain Sistem .....	14
3.4. Desain Algoritma dan Implementasi .....	15
3.4.1. Inisiasi <i>Vertical Handover</i> .....	18
3.4.2. Proses mendapat data <i>resource</i> .....	20

3.4.3.	Proses mendapatkan data kondisi video .....	21
3.4.4.	<i>Horizontal Best First Handover</i> .....	24
3.5.	Pengujian dan Evaluasi.....	26
3.5.1.	Perancangan <i>Tesbed</i> .....	26
3.5.2.	Skenario Pengujian .....	28
3.5.1.1.	Uji Coba Fungsionalitas .....	28
3.5.1.2.	Uji Coba Performa.....	29
3.5.1.2.1.	<i>Delay Streaming</i> .....	29
3.5.1.2.2.	Jumlah Kejadian <i>Handover</i> .....	30
3.5.1.2.3.	Waktu <i>Handover</i> .....	30
3.5.1.2.4.	Penurunan Daya Baterai .....	30
3.6.	Analisis Hasil.....	31
3.7.	Penyusunan Buku Tesis.....	31
BAB 4 HASIL DAN PEMBAHASAN .....		33
4.1.	Tahapan Penelitian .....	33
4.2.	Implementasi Sistem .....	33
4.3.	Langkah-langkah Uji Coba.....	36
4.2.1.	Parameter Pengujian .....	36
4.2.2.	Skenario Pengujian .....	36
4.2.2.1.	Skenario Pengujian Performa .....	36
4.2.2.2.	Skenario Pengujian Fungsionalitas.....	38
4.4.	Hasil Uji Coba dan Analisis .....	40
4.3.1.	Hasil Uji Coba Performa .....	40
4.3.1.1.	Hasil Uji Coba Penghitungan <i>Delay Streaming</i> .....	40
4.3.1.2.	Hasil Uji Coba Penghitungan Jumlah Kejadian <i>Handover</i> .....	41
4.3.1.3.	Hasil Uji Coba Penghitungan Durasi <i>Handover</i> .....	42
4.3.1.4.	Hasil Uji Coba Penurunan Daya Baterai .....	43
4.3.2.	Hasil Uji Coba Fungsionalitas .....	45
4.3.2.1.	Hasil Uji Koefisien Menggunakan Regresi Linear.....	45
4.3.2.2.	Hasil Uji Coba Perubahan Kondisi <i>Latency Jaringan</i> .....	47
4.3.2.3.	Hasil Uji Coba Perubahan Persentase Baterai.....	47
4.3.2.4.	Hasil Uji Coba Perubahan Kondisi <i>Buffer Video</i> .....	48

BAB 5 KESIMPULAN DAN SARAN .....	51
5.1. Kesimpulan .....	51
5.2. Saran.....	52
DAFTAR PUSTAKA .....	53
LAMPIRAN.....	55
BIODATA PENULIS .....	57

**[Halaman ini sengaja dikosongkan]**

## DAFTAR GAMBAR

Gambar 2.1 Tahapan Pengambilan Keputusan Handover .....	6
Gambar 2.2 Latency and bandwidth (sumber: Grigorik, 2013).....	11
Gambar 3.1 Gambaran Algoritma Vertical Handover secara Umum .....	17
Gambar 3.2 Diagram Alir Proses Handover .....	19
Gambar 3.3 Proses mendapatkan <i>resource</i> .....	22
Gambar 3.4 Diagram alir proses mendapatkan kondisi aplikasi video.....	23
Gambar 3.5 Diagram alir <i>handover</i> secara horisontal .....	25
Gambar 3.6 Topologi Pengujian .....	27
Gambar 3.7 Lingkungan uji coba sistem.....	27
Gambar 3.8 Contoh studi kasus perpindahan pengguna pada jaringan .....	29
Gambar 4.1 Tampilan antarmuka aplikasi <i>handover</i> .....	34
Gambar 4.2 Tampilan notifikasi <i>service handover</i> berjalan .....	35
Gambar 4.3 Tampilan aplikasi video <i>streaming</i> .....	35
Gambar 4.4 Perbandingan durasi <i>handover</i> .....	43
Gambar 4.5 Hasil penurunan level baterai pada metode yang diusulkan .....	44
Gambar 4.6 Hasil penurunan level baterai pada metode Bussaneli.....	44
Gambar 4.7 Hasil analisa penentuan koefisien menggunakan regresi linear pada program excel 2007.....	45
Gambar 4.8 Hubungan variabel $W_1$ terhadap nilai residual .....	46
Gambar 4.9 Pengaruh <i>Latency</i> terhadap Nilai Kualitas (T).....	47
Gambar 4.10 Pengaruh Kondisi Baterai terhadap Nilai Kualitas(T) .....	48
Gambar 4.11 Kondisi <i>Buffer</i> Video dan <i>Latency</i> terhadap Nilai Kualitas (T).....	49

**[Halaman ini sengaja dikosongkan]**

## DAFTAR TABEL

Tabel 4.1 Perbandingan delay pada dua metode .....	41
Tabel 4.2 Perbandingan jumlah kejadian <i>handover</i> .....	42
Tabel 4.3 Hasil nilai <i>residual</i> antara $T$ dan <i>Predicted T</i> .....	46

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Evolusi dari bidang komunikasi dengan adanya peningkatan pada pengembangan teknologi jaringan nirkabel membawa pada keberagaman solusi akses dari jaringan (Cardoso, Neves and Ricardo, 2011 (Hong and Garcia, 2005)). Pengguna selalu menginginkan “*always best connected*” pada layanan jaringan dan internet, dan dengan adanya perkembangan layanan seperti *cloud computing*, *file sharing*, *video streaming* dapat menjelaskan semakin pentingnya akses dimana saja (Silva, Carvalano and Sousa, 2011). 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 (Yan, Y. and Narayanan, 2010).

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 yang dua teknologi jaringan nirkabel yang berbeda, biasanya antara jaringan yang memiliki *bandwidth* 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 (Ma et al., 2004).

Salah satu tantangan utama pada proses VH adalah kepuasan dari pengguna terhadap kondisi jaringan yang dipilih (Kassar, Kervella and Pujolle, 2008). Integrasi jaringan tersebut diharapkan mampu memberikan akses yang terbaik sesuai dengan kebutuhan mereka (Akyildiz, xie and Mohanty, 2004). Langkah-langkah menuju tujuan ini, telah diwujudkan dalam munculnya standart IEEE 802.21 dalam suatu kerangka kerja (*framework*) protokol yang mendukung adanya VH (K. Taniuchi, 2009). Standart IEEE 802.21 hanya menyediakan kerangka kerja global. 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 (S.P and Anand , 2011) dan VH berbasis RSS menggunakan testbed nyata pada jaringan UMTS dan Wi-Fi (Busanelli et al., 2011). Dari kedua penelitian ini penelitian pertama (S.P and Anand , 2011) 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 (Cardoso, Neves and Ricardo, 2011). Penelitian kedua, (Busanelli et al., 2011) melakukan eksperimen berdasarkan pelaksanaan *testbed* nyata dengan Wi-Fi (Guglielmo) dan UMTS (Telecom Italia) digunakan. Namun terdapat permasalahan yang dihadapi, yaitu lamanya proses *handover*. Oleh karena itu penulis ingin mengusulkan optimasi penentuan keputusan *vertical handover* dengan berdasar kualitas jaringan dan penyesuaian aplikasi video *streaming stored*, pada *platform* yang relatif mudah diimplementasikan yaitu perangkat Android.

## 1.2. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah:

1. Bagaimana meningkatkan efektifitas terjadinya *handover* pada saat terjadi *streaming* multimedia?
2. Bagaimanakah meminimalisasi efek proses *handover* terhadap proses *streaming* multimedia yang berlangsung?
3. Bagaimana melakukan integrasi pada proses *handover* dengan mempertimbangkan kondisi jaringan dan *resource* dari *client*?

## 1.3. Tujuan

Tujuan yang akan dicapai dalam pembuatan tesis ini adalah mengembangkan strategi untuk pengaturan proses *handover* dengan menggunakan parameter kondisi jaringan heterogen, kondisi baterai dan parameter kondisi *streaming* multimedia.

#### **1.4. Manfaat**

Manfaat dari 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.

#### **1.5. Kontribusi Penelitian**

Kontribusi yang diharapkan dari penelitian ini adalah mengembangkan optimasi strategi pada proses pengambilan keputusan terjadinya *vertical handover* berdasarkan kualitas jaringan heterogen dan kondisi *streaming* multimedia.

#### **1.6. Batasan Masalah**

Batasan masalah pada penelitian ini adalah:

1. Penelitian dilakukan dengan uji coba langsung pada *real device* menggunakan perangkat Android
2. *Vertical Handover* menggunakan skenario *no-coupling* (tidak adanya campur tangan antara pengguna dan operator jaringan)
3. Prosedur autentikasi jaringan tidak dilakukan optimasi, tetap dijalankan sesuai dengan yang sudah ada.
4. *Streaming* multimedia yang dimaksud adalah *streaming stored video*.
5. Penelitian ini berfokus pada mekanisme keputusan pemilihan jaringan yang dilakukan *handover*.
6. Studi kasus yang digunakan adalah perpindahan perangkat *mobile* dan *streaming* video.

**[Halaman ini sengaja dikosongkan]**

## BAB 2

### DASAR TEORI DAN KAJIAN PUSTAKA

Pada bagian ini akan dijelaskan hasil dari pencaian referensi pembelajaran yang bersumber dari buku, jurnal, atau artikel. Referensi yang terkait dengan topik ini diantaranya adalah referensi yang membahas tentang penelitian terkait *vertical handover*, *Android*, *Android service*, kualitas jaringan, *streaming stored video*. Selanjutnya pada subbab berikutnya akan dijelaskan poin – poin yang perlu dijelaskan.

#### **2.1. Vertical Handover (VHO)**

*Vertical handoff* adalah perubahan tipe konektivitas dari titik jaringan yang digunakan untuk mengakses infrastuktur pendukungnya, biasanya untuk mendukung pergerakan. 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. (Hong and Garcia, 2005)

Handover merupakan proses pemeliharaan *session* dari pengguna ketika perangkat *mobile* berganti titik koneksi untuk mengakses jaringan (*point of attachment*). *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 (Yan, Y. and Narayanan, 2010).

Secara umum tujuan *handover* (Akyildiz, xie and Mohanty, 2004) 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 celuller* 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*, sesuai dengan Gambar 2.1. *Handover measurement* dilakukan untuk mengumpulkan informasi untuk identifikasi dibutuhkan *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 dan dengan alokasi sumber daya pada *channel* (Kassar, Kervella and Pujolle, 2008).



**Gambar 2.1 Tahapan Pengambilan Keputusan Handover**

Tanpa memperhatikan tipe *handover*, kontrol terhadap proses *handover* atau mekanisme pengambilan keputusan *handover* dapat berada pada entitas jaringan atau pada perangkat jaringan itu sendiri. Pengambilan keputusan, biasanya membutuhkan beberapa pengukuran dan pengumpulan informasi tentang kapan dan dimana untuk melakukan *handover*. Pada *network controlled handover* (NCHO), entitas jaringan yang mempunyai kontrol terhadap jalannya *handover*. Sedangkan, pada *mobile controlled handover* (MCHO), perangkat tersebut sendiri yang harus mengatur dan memutuskan terjadinya *handover*. Ketika, informasi dan pengukuran yang dilakukan oleh perangkat pada jaringan dan digunakan oleh perangkat *mobile* disebut *network assisted handover* (NAHO). Sebaliknya, ketika informasi dan pengukuran yang dilakukan oleh perangkat *mobile* dan digunakan oleh perangkat jaringan

dikategorikan *mobile assisted handover* (MAHO) (Kassar, Kervella and Pujolle, 2008).

## 2.2. Arsitektur Android Pada Jaringan Heterogen

Android adalah platform open-source untuk perangkat mobile berbasis sistem operasi Linux (*Android Open Source Project*), yang dikembangkan oleh Google dan Open Handset Alliance (OHA). Kebanyakan perangkat *mobile* Android saat ini sudah dilengkapi dengan minimal dua buah perangkat Antarmuka untuk berkomunikasi data, yaitu antarmuka Wi-Fi dan seluler.

Dalam hal konektifitasnya, fungsi Wi-Fi dalam sistem Android didukung oleh tiga komponen: driver, wpa-suppllicant dan WifiManager. *Driver* adalah modul yang berjalan di kernel. Wpa-suppllicant adalah inti komponen yang mengelola Wi-Fi, menyediakan API untuk *platform* Android dan menerapkan mekanisme autentikasi. WifiManager adalah layanan yang berjalan dalam kerangka aplikasi dan menyediakan API untuk mengelola antarmuka Wi-Fi (melalui wpa-suppllicant tersebut) (Silva, Carvalano and Sousa, 2011).

Konektifitas UMTS pada *platform* Android terdiri dari tiga komponen: TelephonyManager, Radio Antarmuka Radio Lapisan Daemon (RILD) dan *driver*. *Driver* adalah implementasi milik dari produsen dan mengimplementasikan secara spesifik API ditentukan oleh *platform* Android. RILD adalah layanan asli yang menyediakan API antara *platform* Android dan *driver*. TelephonyManager adalah layanan yang berjalan dalam kerangka aplikasi dan menyediakan API. Namun, API yang sudah disebutkan ini tidak memiliki fungsi utama untuk mengontrol komunikasi UMTS, misalnya perintah seperti mematikan antarmuka tidak tersedia. Jenis perintah control komunikasi hanya tersedia melalui layanan internal menggunakan TelephonyManager, tetapi layanan ini tidak menyediakan semua perintahnya (Apache 2.0, 2015).

Konektifitas UMTS memang tidak banyak menyediakan fungsi kontrol terhadap antarmukanya, kebanyakan adalah untuk mendapatkan info saja. Akan tetapi, aktif dan tidaknya konektifitas UMTS, berhubungan langsung dengan kondisi dari konektifitas Wi-Fi pada perangkat tersebut. Ada beberapa kondisi yang mempengaruhi sehingga masih bisa dimanfaatkan untuk pengaplikasian *Vertical Handover* pada Android, yang nantinya akan digunakan sebagai *testbed*.

### 2.3. Android Service

*Service* atau layanan adalah komponen aplikasi yang dapat melakukan operasi yang lama berjalan di *background* dan tidak diberikan antarmuka pengguna. Komponen aplikasi lain dapat memulai layanan dan dapat seterusnya berjalan di latar belakang bahkan jika pengguna beralih ke aplikasi lain. Selain itu, komponen dapat mengikat ke layanan untuk berinteraksi dengan itu dan bahkan melakukan komunikasi interprocess (IPC). Sebagai contoh, sebuah layanan mungkin menangani transaksi jaringan, bermain musik, melakukan file I/O, atau berinteraksi dengan penyedia konten, semua dari latar belakang (Apache 2.0, 2015).

Sebuah *service* dasarnya dapat mengambil dua bentuk:

- *started*

*Service* "dimulai" ketika komponen aplikasi (seperti *activity*) dimulai dengan memanggil `startService()`. Setelah dimulai, *service* dapat berjalan di latar belakang tanpa batas, bahkan jika komponen yang memulai dihancurkan. Biasanya, *service* ini melakukan suatu operasi dan tidak mengembalikan hasilnya ke pemanggil. Sebagai contoh, mungkin download atau upload berkas melalui jaringan. Ketika operasi tersebut sudah dilakukan, *service* akan berhenti sendiri.

- *bound*

*Service* "terikat" ketika komponen aplikasi berikatan dengan *service* dengan memanggil `bindService()`. Sebuah *service* terikat menawarkan antarmuka client-server yang memungkinkan komponen untuk berinteraksi dengan *service*, mengirim

permintaan, mendapatkan hasil, dan bahkan melakukannya di proses dengan komunikasi *interprocess* (IPC). Sebuah *service* terikat berjalan hanya selama komponen aplikasi lain terikat dengan *service* tersebut. Beberapa komponen dapat mengikat ke satu *service* sekaligus, tetapi ketika mereka semua melepaskan ikatannya, layanan ini hancur.

#### **2.4. Streaming Stored Multimedia**

*Streaming* adalah sebuah teknologi untuk memainkan file video atau audio secara langsung ataupun dengan pre-recorder dari sebuah mesin server (*webserver*). Dengan kata lain, berkas video ataupun audio yang terletak dalam sebuah server dapat secara langsung pada *browser* saat proses *buffering* mulai berjalan

Aplikasi multimedia *streaming* diklasifikasikan ke dalam tiga kategori: (i) *streaming stored audio/video*, (ii) suara *voice/ video-over-IP*, dan (iii) *live streaming audio/video* (Kurose and Ross, 2013). Masing-masing dari kategori aplikasi memiliki kebutuhan dan desain yang berbeda. Akan tetapi, pada bagian ini yang dijelaskan adalah kategori *streaming stored audio/video*.

Aplikasi dengan kategori *streaming stored audio/video*, media yang mendasari adalah video rekaman, seperti film, acara televisi, acara olahraga, atau video buatan pengguna rekaman (seperti yang biasa terlihat di YouTube). Video yang direkam sebelumnya ini ditempatkan pada server, dan pengguna mengirim permintaan ke server untuk melihat video sesuai permintaan / *request*. *Streaming stored* memiliki tiga fitur kunci yang membedakan:

1. *Streaming*. Dalam *streaming* yang tersimpan aplikasi video, pada sisi pengguna video biasanya mulai dimainkan dalam beberapa detik setelah video mulai diterima dari server. Hal ini menunjukkan bahwa dari sisi pengguna akan memainkan audio/video dari satu lokasi bagian video, sementara pada waktu yang bersamaan menerima bagian video selanjutnya dari dari server. Teknik semacam ini dikenal sebagai proses *streaming*, yaitu menghindari harus men-

download file video secara keseluruhan (potensi delay yang lama) sebelum video *playout* dimulai.

2. *Interactivity*. Karena media direkam sebelumnya (*prerecorded*), pengguna dapat melakukan *pause*, reposisi ke depan, reposisi mundur, *fast-forward*, dan sebagainya melalui konten video.
3. *Continuous playout*. Setelah *playout* video dimulai, harus dilanjutkan sesuai dengan waktu asli dari rekaman. Oleh karena itu, data harus diterima dari server dalam waktu untuk *playout* tersebut pada klien, jika tidak, pengguna akan mengalami *freeze frame* video (ketika klien menunggu *frame* yang tertunda) atau *frame skipping* (ketika klien melompati / *skipping frame* yang tertunda)

## 2.5. Kondisi Jaringan : *Latency*

*Latency* adalah waktu yang dibutuhkan dari sumber mengirimkan paket ke tujuan yang menerimanya (Grigorik, 2013). *Bandwidth* adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi. Seperti diperlihatkan pada Gambar 2.2. Itulah sebabnya kenapa ada orang yang langganan internet dengan *bandwidth* besar, tapi untuk melihat video di YouTube kadang masing buruk. Karena, meski *bandwidth*-nya besar, tapi *latency*-nya terhambat.

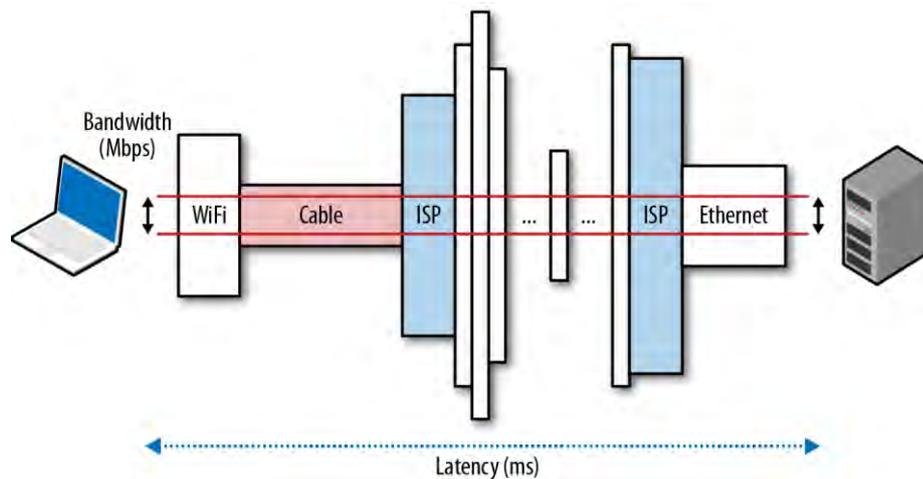
Sebab koneksi jaringan yang mempunyai *latency* lambat antara lain;

- Buruknya kualitas saluran, contohnya jika menggunakan media kabel, kemungkinan terdapat kabel yang rusak. Bila menggunakan media wireless, kemungkinan terdapat gangguan sinyal (distorsi sinyal atau interferensi).
- Jarak yang terlalu jauh, dan tidak ada penguat sinyal.

## 2.6. Power Condition

Dalam jaringan, perlu dicari cara untuk meningkatkan efisiensi energi. Daya tidak hanya dikonsumsi oleh terminal pengguna, tetapi juga dikaitkan dengan

peralatan base station. Daya juga dikonsumsi selama mobile switching atau *handoffs*. Selama *handoff*, seringkali aktivasi antarmuka dapat menyebabkan cukup pembuangan baterai. Isu penghematan daya juga muncul dalam pencarian jaringan karena aktivasi antarmuka yang tidak perlu dapat meningkatkan konsumsi daya. Faktor konsumsi daya juga penting untuk dimasukkan, saat keputusan *handoff* (Gondara and Kadam, 2011).



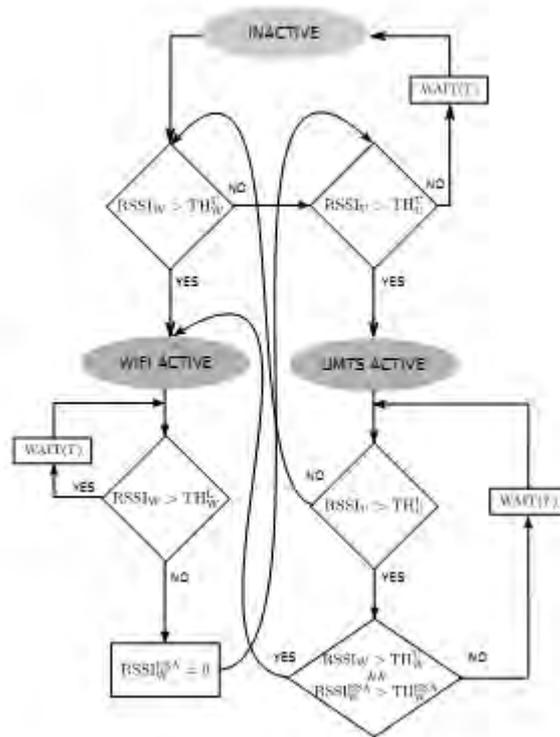
Gambar 2.2 Latency and bandwidth (sumber: Grigorik, 2013)

## 2.7. Kondisi Video Streaming (kondisi *buffer*)

Saat dilakukan *streaming* pada sebuah video, maka *browser* menyimpan dan mengolah kode-kode video dalam *buffer*, sehingga tidak harus terus menerus tersambung ke server. Jika koneksi terputus untuk sementara, maka server akan mengetahui sampai sejauh mana *file* berhasil terunduh dalam *buffer*, untuk kemudian server akan mengirimkan kode-kode selanjutnya untuk memutar video. *Browser* mendownload dan menyimpan kode-kode video dalam *buffer* dengan secara terus menerus hingga video siap dimainkan di layar. (Grigorik, 2013)

## 2.8. Metode *Vertical Handover* Berbasis RSS (*Received Signal Strength*)

Pada subbab ini akan dibahas metode sebelumnya yang nantinya pada pengujian akan digunakan sebagai pembandingan. Metode yang dibahas adalah metode yang diusulkan oleh Bussaneli (Busanelli et al., 2011) yang menggunakan RSS sebagai parameter penentu untuk melakukan *handoff*, ditunjukkan pada Gambar 2.3. Algoritma ini menggunakan ambang batas atas dan ambang batas bawah dari RSSI untuk menentukan perpindahan *state*-nya. Pada algoritma ini dibedakan antara ambang batas yang ada pada Wi-Fi dan UMTS disimbolkan dengan huruf yang merepresentasikannya.



Gambar 2.3 Diagram alir algoritma *low-complexity RSS-based*

## **BAB 3**

### **METODOLOGI PENELITIAN**

Langkah-langkah penelitian yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. Perumusan Masalah
2. Studi Literatur
3. Desain Sistem
4. Desain Algoritma dan Implementasi
5. Pengujian dan Evaluasi
6. Analisis Hasil
7. Penyusunan Buku Tesis

#### **3.1. Perumusan Masalah**

*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. Untuk mengatasi masalah ini, penulis mengusulkan penentuan keputusan *vertical handover* dengan berdasar kondisi jaringan dengan diintegrasikan pada penyesuaian aplikasi video *streaming*.

Dari sini, permasalahan berkembang ke penentuan karakteristik jaringan yang bagaimana, dijadikan acuan untuk melakukan *handover* dari satu jaringan ke jaringan lain. Selain itu, kondisi aplikasi *streaming* video apa saja yang seharusnya bisa mengurangi penentuan *vertical handover* yang tidak dibutuhkan.

### **3.2. Studi Literatur**

Tahapan ini merupakan tahapan pencarian referensi pembelajaran yang dapat dipertanggungjawabkan seperti referensi yang bersumber dari buku, jurnal atau artikel. Pencarian referensi berguna untuk menambah wawasan dari penelitian terdahulu yang terkait sebagai penunjang untuk membuat suatu penelitian yang dapat bermanfaat bagi penelitian. Topik-topik referensi yang terkait dengan penelitian ini diantaranya mengenai *vertical handover*, *android*, *android service* kaitannya dengan *handover*, kualitas jaringan, *power consumption*, *streaming stored video*, dan *video quality streaming handover decision*. Dari studi literatur yang telah dilakukan maka diperoleh informasi yang berkaitan dengan penelitian yang dilakukan ini, seperti berikut:

1. Tipe dari algoritma *vertical handover*, karakteristik dan macam - macamnya.
2. Isu dari penelitian – penelitian terkait dengan penentuan keputusan *handover*.
3. Penelitian – penelitian yang sudah ada, terutama berkaitan dengan perkembangan teknologi komunikasi yang ada saat ini.
4. Protokol dan arsitektur yang mendukung digunakannya *handover*.

### **3.3. Desain Sistem**

Dalam mendesain sistem penelitian ini, ada beberapa tahapan yang dilakukan, yaitu:

1. Penentuan lingkungan uji coba

Lingkungan uji coba yang digunakan dalam penelitian ini meliputi aplikasi *vertical handover* dan aplikasi *streaming stored video* pada perangkat *mobile* yang digunakan untuk menjalankan *vertical handover*.

## 2. Penentuan parameter uji coba

Proses penentuan parameter uji coba merupakan proses untuk menentukan parameter apa saja yang digunakan dalam penelitian serta apa saja yang diukur/dihitung. Parameter uji coba yang digunakan dalam penelitian ini adalah:

- a. *Latency* pada jaringan
- b. *Resource* pada *mobile device*
- c. Kondisi pada aplikasi video

## 3. Pemilihan skenario uji coba

Proses pemilihan skenario uji coba merupakan proses memilih konsep uji coba yang sesuai dengan tujuan penelitian sebagai pembuktian terhadap kebenaran hipotesis yang diajukan.

## 4. Pengukuran hasil uji coba

Pengukuran hasil uji coba dilakukan untuk mengukur data yang dihasilkan dari skenario uji coba yang telah dilakukan

## 5. Analisis hasil uji coba

Analisis hasil uji coba bertujuan sebagai bahan pertimbangan terhadap kebenaran hipotesis yang diajukan.

## 6. Evaluasi

Evaluasi dilakukan untuk me-review hasil pengujian, kelemahan-kelemahan yang ada serta saran dan masukan untuk penelitian selanjutnya.

### **3.4. Desain Algoritma dan Implementasi**

Permasalahan yang diangkat dari penelitian ini, yaitu proses *handover* kaitannya dengan penggunaan *resource* pada studi kasus proses *streaming video stored* pada perangkat seluler Android. Penelitian ini berfokus pada pengoptimalan penggunaan *resource* serta penyesuaian algoritma *vertical handover*. Untuk membuktikan penelitian ini tersebut perlu adanya suatu desain atau rancangan algoritma yang diajukan.

Algoritma *handover* yang dirancang ini merupakan algoritma *vertical handover* menggunakan penyesuaian berdasarkan kualitas jaringan yang diwakili oleh *latency* dari jaringan, *power consumption*, dan kondisi *streaming* video. Gambaran umum algoritma secara umum dapat dilihat pada Gambar 3.1. Algoritma ini dimulai ketika pengguna memulai *service* atau layanan *handover*, fungsi *handover* akan dijalankan setiap fungsi waktu memicu, setiap waktu tunggu yang ditentukan. Kemudian, akan dilakukan pembacaan kondisi jaringan dan kondisi baterai pada saat itu, serta dari luar dari layanan *handover* atau dari aplikasi video *streaming* didapatkan kondisi video yang dicapainya saat ini. Data – data kondisi ini akan dihitung sesuai dengan persamaan perubahan kualitas yang dijelaskan pada persamaan 3.1, 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. Perhitungan nilai kualitas dijelaskan pada persamaan 3.1. Persamaan ini dikembangkan berdasarkan fungsi bobot yang diusulkan oleh Tawil (R., G. and O., 2008).

$$T_i = w_l \frac{l}{l \max} + w_c(1 - c) + w_v(1 - \frac{V}{V \max}) \quad (3.1)$$

Keterangan:

$T_i$  = Parameter Kondisi ke-i

$W_l$  = Bobot parameter l

l = *latency* jaringan

l max = *latency* maximal

$w_c$  = Bobot parameter c

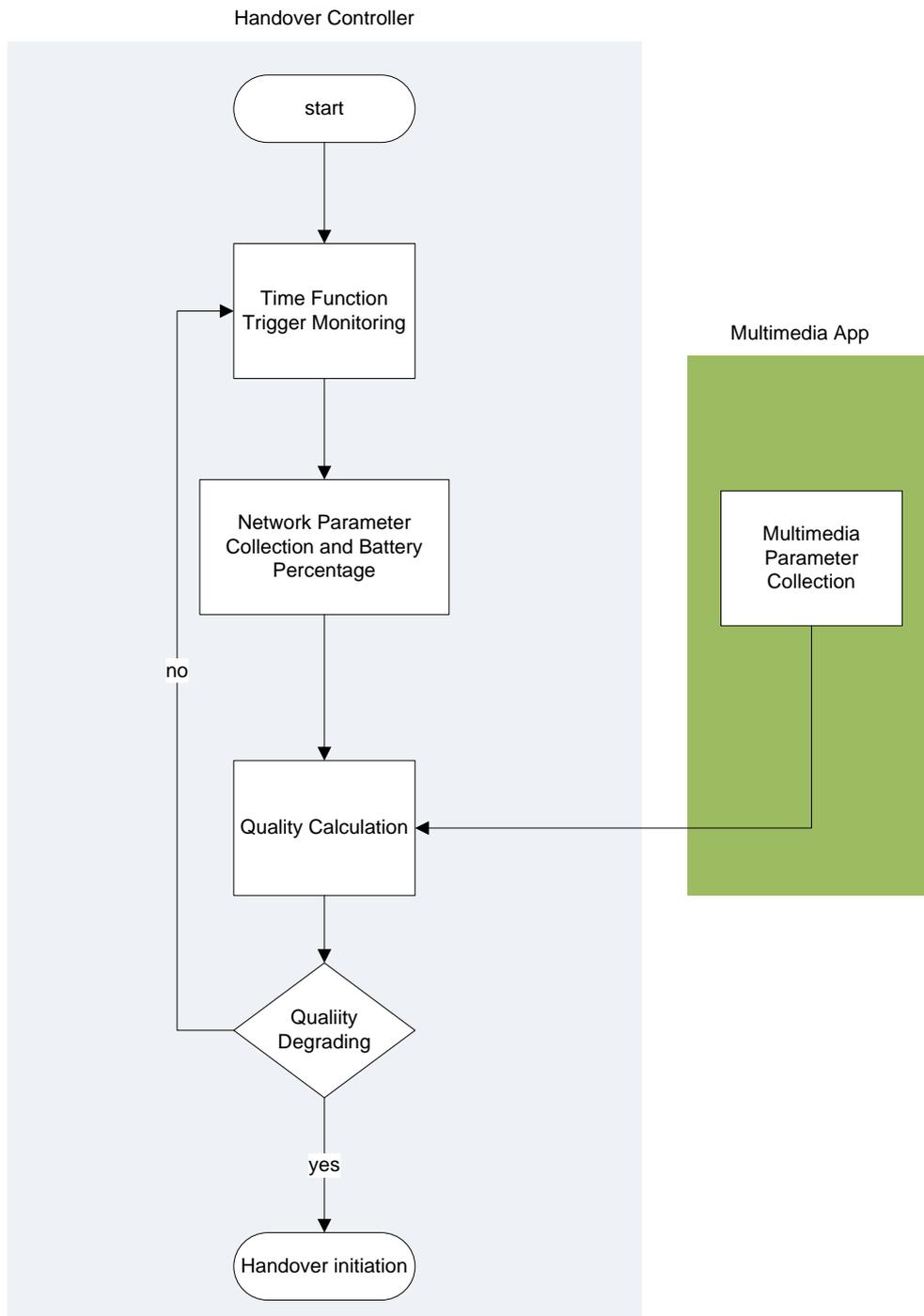
c = *power percentage*

$W_v$  = Bobot parameter v

V max = nilai maximum kondisi video

V = nilai kondisi video

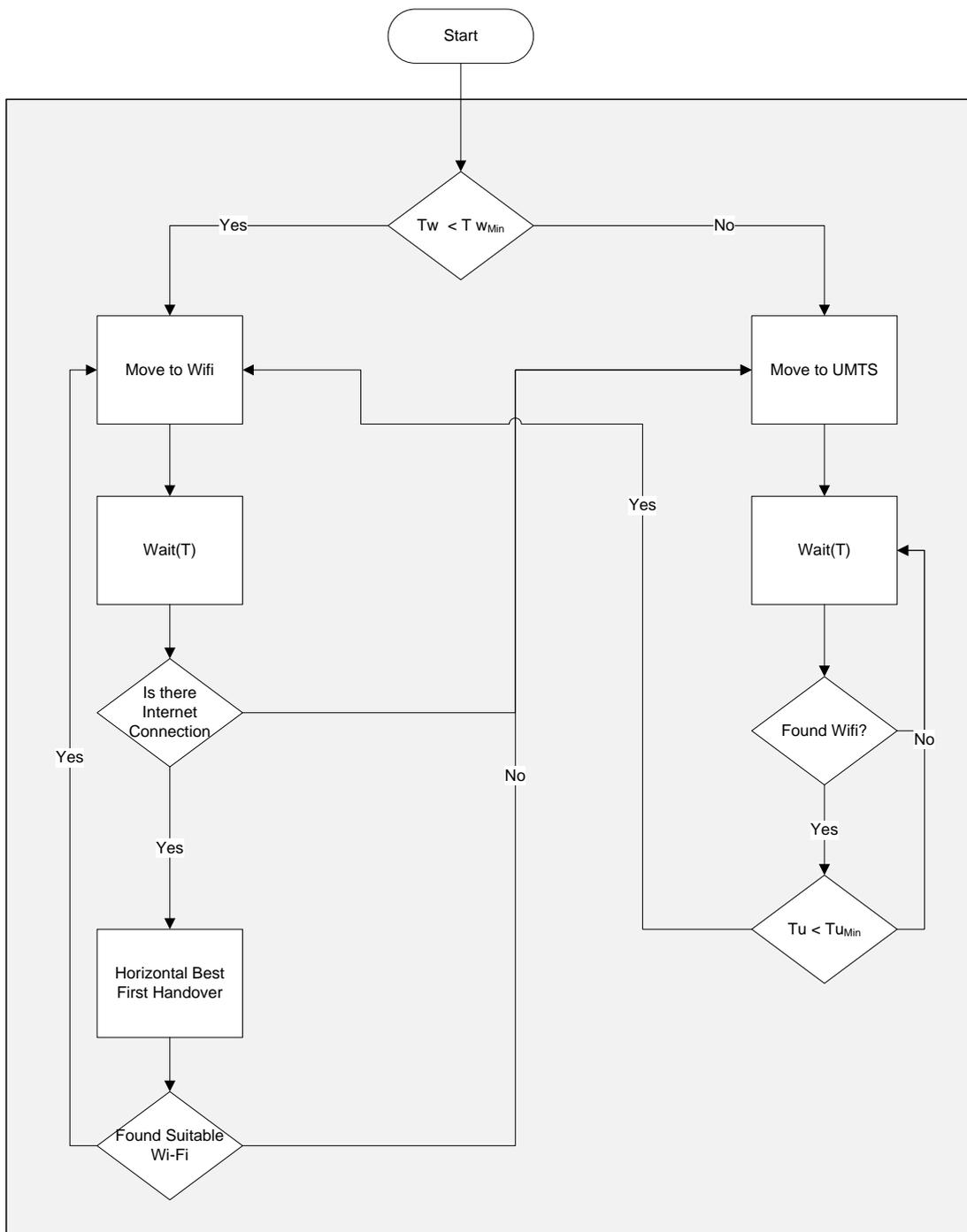
$W_l + W_c + W_v = 100$



**Gambar 3.1** Gambaran Algoritma Vertical Handover secara Umum

### 3.4.1. Inisiasi *Vertical Handover*

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 3.1, dan akan didapatkan nilai kondisi ( $T_i$ ) yang akan digunakan sebagai penentu terjadinya *handover*, seperti dijelaskan pada Gambar 3.2. 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 ( $T_{W_{Min}}$ ), jika sudah maka akan pindah ke Wi-Fi tersebut. Proses pindah ke Wi-Fi ini melalui proses selanjutnya (*horizontal best handover*) 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()` 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. Untuk lebih jelasnya akan dijelaskan melalui Pseudocode 3.1. Apabila  $T_{W_{Min}}$  dan  $T_{u_{Min}}$  tidak terpenuhi kedua duanya, maka akan dibandingkan antara keduanya, manakah yang paling baik.



**Gambar 3.2 Diagram Alir Proses Handover**

1	<code>public class MyService extends Service {</code>
2	<code>mywifimanager = (WifiManager) getSystemService(Context.WIFI_SERVICE);</code>
3	<code>mNM=(NotificationManager) getSystemService(NOTIFICATION_SERVICE);</code>
4	<code>TimerTask timerTask = new TimerTask() {</code>
5	<code>if(Connectivity.isConnectedMobile(getApplicationContext())) {</code>
6	<code>    if(!adaOpenWifi()){ //fungsi pengecekan pada list discovery wifi</code>
7	<code>    mywifimanager.setWifiEnabled(false);</code>
8	<code>    automatic_change_to_mobile();</code>
9	<code>    }else{</code>
10	<code>        Tu = b1*(LAT_MAX/lat)+b2*batteryPct+b3*(VID_MAX/paramVid)</code>
11	<code>        Tu_Min = Calculate_Treshold_from_preference_activity();</code>
12	<code>        If(Connectivity.getNetworkInfo().isConnected() &amp;&amp; Tu &lt;=</code> <code>Tu_Min)</code>
13	<code>            mywifimanager.setWifiEnabled(true);</code>
14	<code>            masuk_algo_hbfh()</code>
15	<code>        Else</code>
16	<code>            mywifimanager.setWifiEnabled(false);</code>
17	<code>            disable_wifi_bssid(DISABLE_INTERVAL, netId);</code>
18	<code>    } timer = new Timer("myTimer");</code>
19	<code>timer.schedule(timerTask, DELAY_INTERVAL, UPDATE_INTERVAL);</code>

**Pseudocode 3.1** Proses Vertical Handover

### 3.4.2. Proses mendapat data *resource*

. Tahapan - tahapan vertical handover pada perangkat, salah satunya adalah pengukuran. Proses pengukuran membutuhkan data dari kondisi yang dipertimbangkan. Dalam penelitian ini, data yang didapatkan adalah *latency* dan prosentase penggunaan baterai dari perangkat saat ini. Proses untuk mendapatkan kondisi jaringan dan prosentase baterai ini tidak bisa dijadikan satu dengan *service* dari *handover*. Karena *service* hanya memproses data dan mengontrol proses *handover*. *Service* menerima kondisi *resource* ini melalui *listener* dari *activity* lain

yang mempunyai wewenang untuk memperoleh data tersebut. Tahapan – tahapan proses ini, untuk lebih detail dapat dilihat pada Gambar 3.3 dan Pseudocode 3.2.

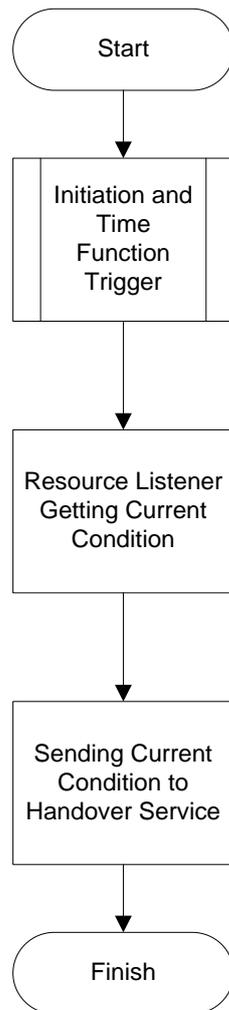
1	<code>float batteryPct;</code>
2	<code>int ping;</code>
3	<code>infoReceiver myInfoReceiver;</code>
4	<code>IntentFilter filter = new IntentFilter();</code>
5	<code>filter.addAction(Intent.ACTION_BATTERY_CHANGED);</code>
6	<code>registerReceiver(myInfoReceiver, filter);</code>
7	<code>private class infoReceiver extends BroadcastReceiver{</code>
8	<code>int level = intent.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);</code>
9	<code>int scale = intent.getIntExtra(BatteryManager.EXTRA_SCALE, -1);</code>
10	<code>batteryPct = level / (float)scale;</code>
11	<code>ping = fungsi_get_icmp();</code>
12	<code>}</code>

**Pseudocode 3.2 Pseudocode proses mendapatkan data *resource***

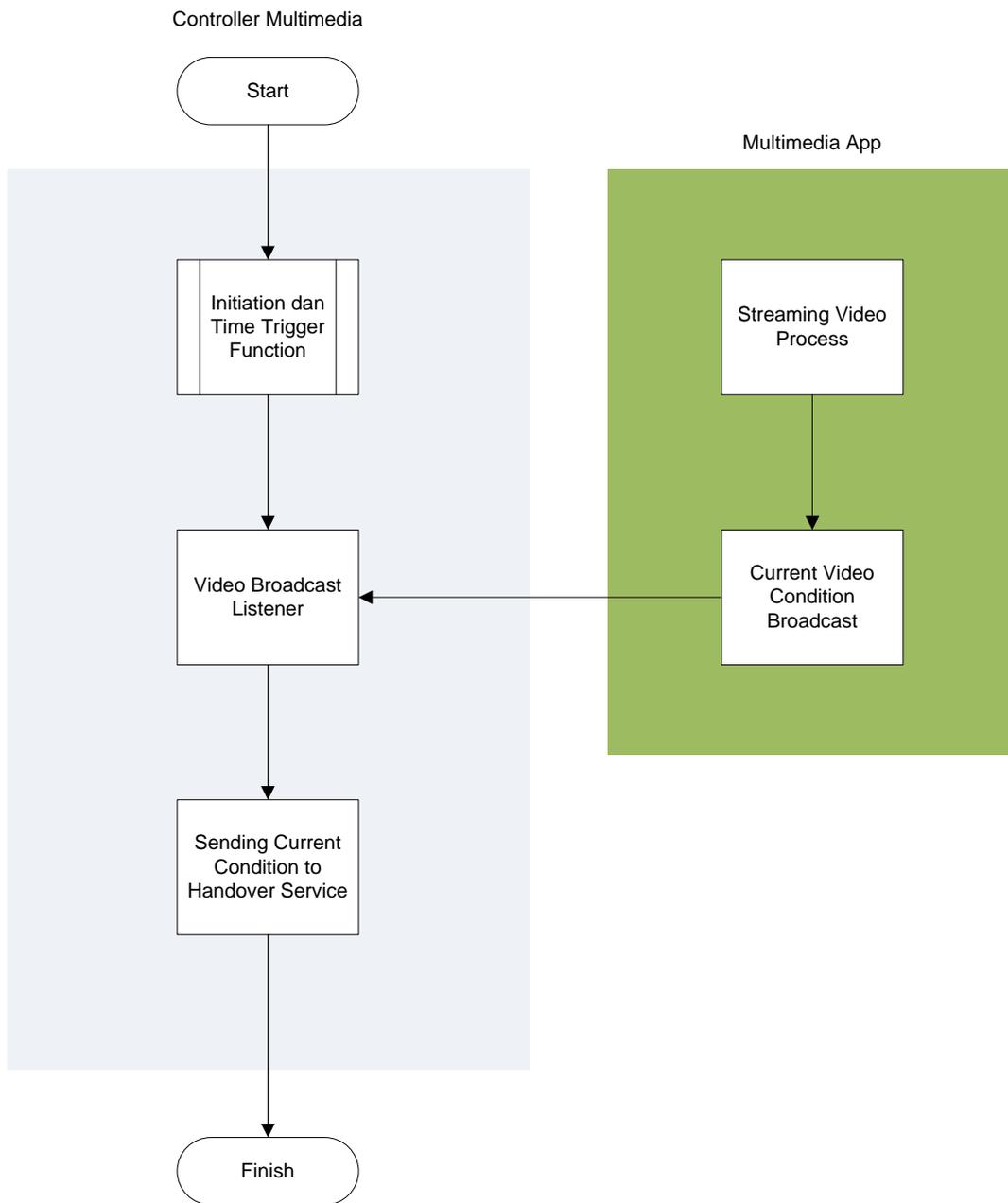
### 3.4.3. Proses mendapatkan data kondisi video

Data lain yang perlu didapatkan untuk digunakan sebagai perhitungan *vertical handover* adalah kondisi aplikasi *streaming* video. Pada saat melakukan *streaming* video, pengguna juga menggunakan koneksi, dan akan terasa bagaimanakah kondisi jaringan pada saat itu, baik atau buruk. Video yang dimainkan lewat proses *streaming* sebenarnya tidak secara utuh dimainkan, akan tetapi sesuai dengan *buffer* penyimpanan yang sudah berhasil diunduh. Berkaitan dengan *buffer* ini penulis menambahkan kondisi video yang diambil dari aplikasi multimedia *streaming*. Jadi, aplikasi *handover* berjalan *dibackground* dan mengontrol jalannya koneksi, sementara aplikasi multimedia juga berjalan. Pada saat aplikasi multimedia ini berjalan, terdapat juga *service* yang mengirim data pada *service handover* dan

digunakan sebagai data tambahan. Proses pengiriman dan penerimaan data kondisi aplikasi pemutar video ini ditunjukkan pada Gambar 3.4, Pseudocode 3.3 dan Pseudocode 3.4



**Gambar 3.3 Proses mendapatkan *resource***



**Gambar 3.4 Diagram alir proses mendapatkan kondisi aplikasi video**

1	<code>vidView = (VideoView) findViewById(R.id.videoView);</code>
2	<code>String vidAddress=</code> <code>"http://www.android.com/AndroidCommercial.3gp";</code>
3	<code>vidView.setVideoURI(vidUri);</code>
4	<code>vidView.start();</code>

5	
6	<code>private void sendParamBroadcast(Intent intent){</code>
7	<code>Intent intent = new Intent(MainActivity.VIDEO_PARAM);</code>
8	<code>Intent.putExtra("vid_duration", vidView.getDuration());</code>
9	<code>Intent.putExtra("vid_buffPct", vidView.getBufferPercentage());</code>
10	<code>sendBroadcast(intent);}</code>

**Pseudocode 3.3** Pseudocode proses pengiriman kondisi video pada aplikasi multimedia

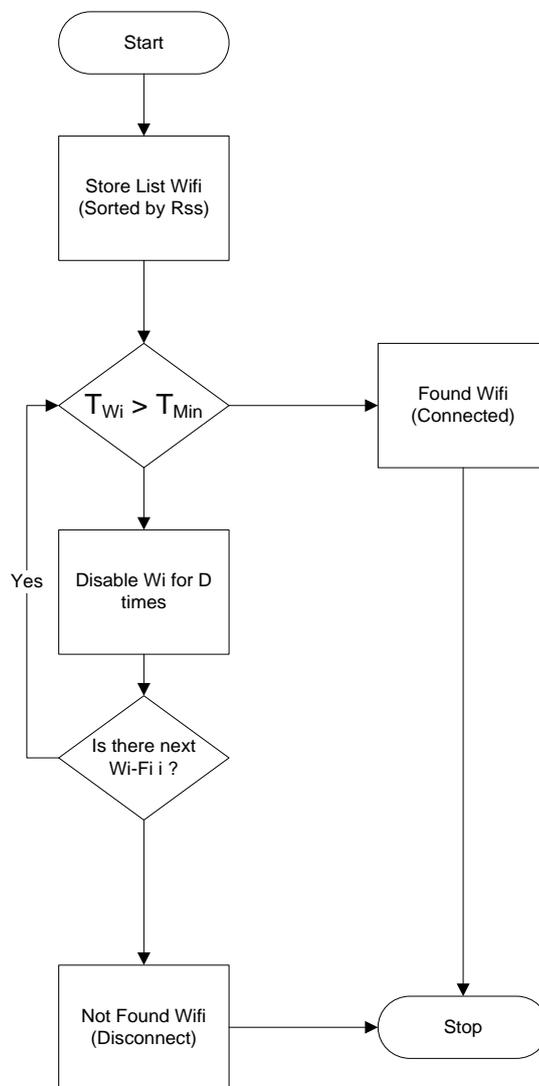
1	<code>public static final String VIDEO_PARAM = "com.example.basic.sample1.MainActivity.VIDEO_PARAM";</code>
2	
3	<code>IntentFilter filter = new IntentFilter();</code>
4	<code>filter.addAction(VIDEO_PARAM);</code>
5	<code>registerReceiver(videoReceiver, filter);</code>
6	
7	<code>private class videoReceiver extends BroadcastReceiver{</code>
8	<code>if(intent.getAction().equals(VIDEO_PARAM) {</code>
9	<code>String temp = intent.getStringExtra("vid_duration");</code>
10	<code>String temp2 = intent.getStringExtra("vid_buffPct");</code>
11	<code>}</code>
12	<code>}</code>

**Pseudocode 3.4** Pseudocode penerimaan kondisi video pada *service handover*

#### 3.4.4. Horizontal Best First Handover

Kemungkinan terdapat beberapa Wi-Fi atau *access point* yang berdekatan dalam satu daerah. Biasanya terdapat Wi-Fi dengan kondisi sinyal, kepadatan jaringan yang berbeda. Untuk memilih yang terbaik, penulis mengusulkan tambahan penggunaan algoritma untuk menangani masalah ini. Algoritma ini memilih berdasarkan urutan RSS yang didapatkan saat proses *discovery* jaringan Wi-Fi, mencoba satu persatu dan membuat koneksi dengan yang ditemukan lebih dulu dengan kondisi yang memenuhi ambang batas. Proses *handover* secara horisontal ini

ditunjukkan dengan diagram alir pada Gambar 3.5. Gambar 3.5 Diagram alir *handover* secara horisontal dan Pseudocode 3.5.



Gambar 3.5 Diagram alir *handover* secara horisontal

1	List<ScanResult> networkList = mywifimaneger.getScanResults();
2	if(networkList != null) {
3	for (ScanResult network : networkList) {
4	int cek = getSecurity(result);

5	<code>if(cek == SECURITY_NONE) {</code>
6	<code>  boolean bagus = Cek_treshold_jaringan(result);</code>
7	<code>  if(bagus) return FOUND_WIFI;</code>
8	<code>}</code>
9	<code>Return NOT_FOUND_WIFI;</code>

**Pseudocode 3.5 Pseudocode *horizontal handover***

### 3.5. Pengujian dan Evaluasi

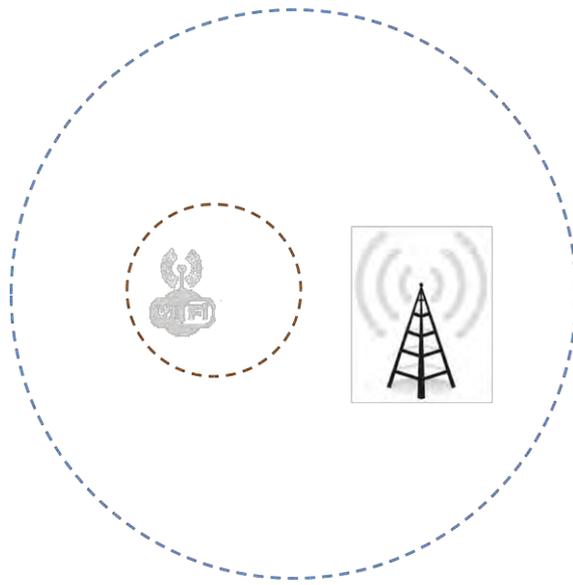
Lingkungan uji coba dijelaskan untuk mengetahui cakupan area modul dari skenario uji coba yang akan dilakukan. Pada sistem ini, uji coba dilakukan untuk mengetahui seberapa besar optimasi atau perbaikan yang ditingkatkan dibandingkan dengan metode yang ada sebelumnya

#### 3.5.1. Perancangan *Tesbed*

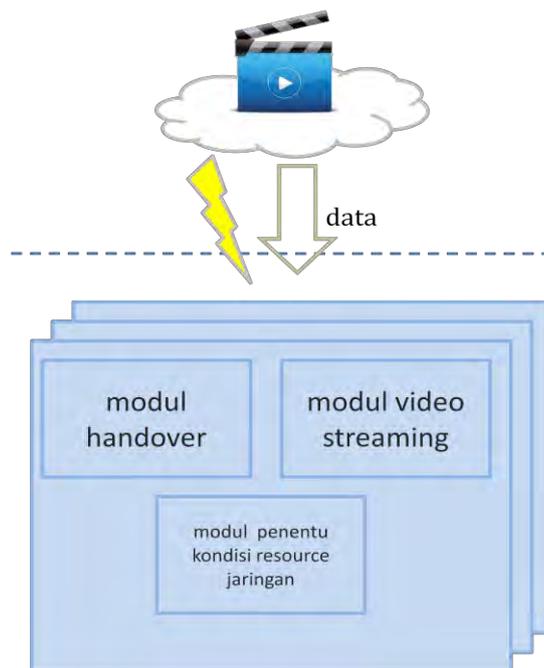
Desain topologi *testbed* yang digunakan sebagai topologi pengujian pada penelitian ini dapat dilihat pada Gambar 3.6. Topologi *testbed* ini dibuat memanfaatkan jaringan yang sudah ada, menggunakan skenario *no-coupling* (tidak adanya campur tangan antara pengguna dan operator jaringan). Perangkat *mobile* sendiri yang mengatur terjadinya *handover* pada koneksinya. Wi-Fi yang digunakan berlokasi di rumah menggunakan koneksi IndiHome 10Mbps dan berjarak kurang lebih 500 m dari *base station* provider Telkomsel *up to* 7.2Mbps.

Perangkat *mobile* yang digunakan berupa *smartphone* Android. Adapun spesifikasi dari *smartphone* yang digunakan adalah sebagai berikut:

- a. Processor : ARMv7 Processor rev 2 (v7l)
- b. RAM : 384 MB
- c. Hardisk : 2GB
- d. Sistem Operasi : Android 4.1.2



**Gambar 3.6 Topologi Pengujian**



**Gambar 3.7 Lingkungan uji coba sistem**

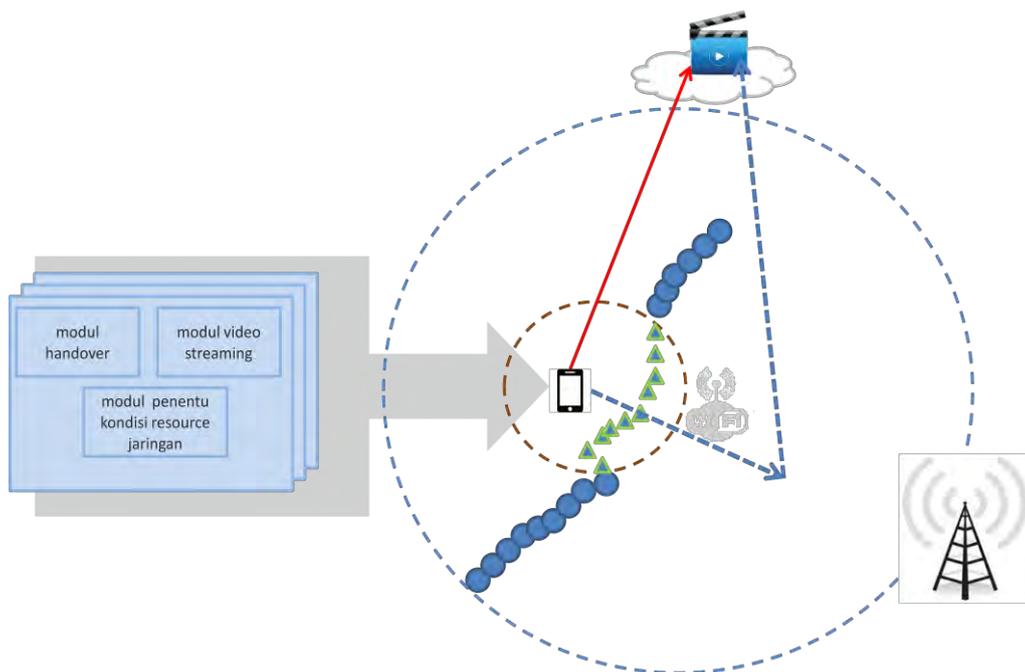
Optimasi *vertical handover* dilakukan ketika pengguna melakukan proses *streaming* video, dengan mempertimbangkan kondisi aplikasi *video player*, sehingga bisa mengurangi proses *handoff* yang tidak diperlukan saat itu. Lingkungan uji coba yang digunakan pada penelitian dijelaskan melalui Gambar 3.7. meliputi beberapa modul yang akan dijalankan pada perangkat *mobile*. Modul tersebut adalah modul *handover* yang mengatur terjadinya proses *handover*, modul *video streaming*, dan modul penentu kondisi *resource* dan jaringan.

### **3.5.2. Skenario Pengujian**

Skenario pengujian yang digunakan untuk pada penelitian ini meliputi skenario uji coba fungsionalitas dan skenario uji coba performa. Uji coba fungsionalitas menguji kesesuaian data yang dihasilkan dengan rancangan aturan sistem yang dibuat. Uji coba performa digunakan untuk menguji kehandalan dari metode yang diusulkan. Berikut ini adalah beberapa skenario uji coba yang dilakukan pada penelitian ini.

#### **3.5.1.1. Uji Coba Fungsionalitas**

Skenario uji coba fungsionalitas merupakan skenario pengujian yang dilakukan untuk melihat kesesuaian data yang dihasilkan dengan rancangan aturan sistem yang dibuat. Uji coba fungsionalitas akan dilakukan dengan skenario sesuai dengan Gambar 3.8. Melalui topologi yang sudah direncanakan perangkat *mobile* akan bergerak diantara base station dan menyesuaikan kondisi dari fungsionalitas yang akan diuji. Fungsionalitas dari sistem yang akan diuji yaitu pengujian berdasarkan perubahan kondisi jaringan, kondisi aplikasi *video streaming* dan kondisi baterai terhadap pergerakan perangkat *mobile*.



Gambar 3.8 Contoh studi kasus perpindahan pengguna pada jaringan

### 3.5.1.2. Uji Coba Performa

Sedangkan skenario uji coba performa merupakan skenario pengujian yang dilakukan untuk melihat seberapa besar perbaikan, penghematan *resource* dan kualitas video *streaming* yang dihasilkan. Selain itu, skenario uji coba performa juga dilakukan untuk melihat pengaruh beberapa variabel yang berbeda. Uji coba performa akan dilakukan untuk mengukur keberhasilan metode terhadap tolok ukur sebagai berikut.

#### 3.5.1.2.1. *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). Uji coba dilakukan dengan membandingkan dua mekanisme *handover*, yaitu menggunakan metode Bussaneli dan metode ini. Studi kasus yang digunakan adalah penggunaan kedua metode

dengan video *streaming* yang sama. Ketika kedua mekanisme dibandingkan, melalui *log* masing-masing metode saat dijalankan.

#### **3.5.1.2.2. Jumlah Kejadian Handover**

Jumlah *handoff* yang terjadi mempengaruhi juga terhadap keberhasilan metode *handover*. Melalui proses *handoff*, seringkali aktivasi antarmuka, *mobile switching* yang dapat menyebabkan cukup pembuangan baterai. Studi kasus yang digunakan adalah penggunaan kedua metode dengan N kali percobaan. Ketika kedua mekanisme dibandingkan, melalui *log* masing-masing metode saat dijalankan, maka dapat diperoleh perbedaan jumlah *handoff* yang terjadi.

#### **3.5.1.2.3. Waktu 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. Studi kasus yang digunakan adalah penggunaan kedua metode dengan N kali percobaan. Ketika kedua mekanisme dibandingkan, melalui *log* masing-masing metode saat dijalankan, maka dapat diperoleh perbedaan waktu *handover* pada masing – masing metode.

#### **3.5.1.2.4. Penurunan Daya Baterai**

Uji coba performa 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. Uji coba dilakukan dengan mengukur penurunan daya baterai ketika sistem dijalankan. Uji coba dilakukan dengan

membandingkan dua mekanisme *handover*, yaitu menggunakan metode Bussaneli dan metode ini. Studi kasus yang digunakan adalah lama penggunaan kedua metode. Ketika kedua mekanisme dibandingkan, maka dapat diperoleh perbedaan penurunan daya baterai pada masing – masing metode.

### **3.6. Analisis Hasil**

Analisis hasil dilakukan untuk melihat apakah hasil pengujian dari hipotesis yang diajukan sesuai dengan tujuan penelitian. Analisis hasil dilakukan terhadap skenario pengujian yang telah dilakukan.

### **3.7. Penyusunan Buku Tesis**

Penyusunan buku tesis dilakukan sebagai dokumentasi terhadap serangkaian penelitian yang dikerjakan agar dapat dijadikan sebagai bahan pembelajaran, referensi maupun sebuah perbaikan penelitian di masa depan yang berhubungan dengan perbaikan mekanisme *update* informasi.

**[Halaman ini sengaja dikosongkan]**

## **BAB 4**

### **HASIL DAN PEMBAHASAN**

Pada bab 4 ini dijelaskan tahapan penelitian mengenai perbaikan mekanisme *reporting*, tahapan pengujian yang dilakukan dan analisis terhadap hasil uji coba yang diberikan.

#### **4.1. Tahapan Penelitian**

Penelitian ini dilakukan pada lingkungan uji coba yang real. Secara umum, penelitian perbaikan mekanisme *vertical handover* dilakukan melalui beberapa tahapan sebagai berikut:

1. Tahap pertama, merupakan tahapan perancangan dan implementasi sistem.
2. Tahap kedua, merupakan tahapan perancangan skenario uji coba meliputi skenario uji coba fungsionalitas dan performa untuk melakukan pengujian fungsionalitas maupun performa dari penelitian perbaikan mekanisme *vertical handover*. Skenario pengujian fungsionalitas meliputi fungsionalitas perubahan kondisi jaringan, kondisi aplikasi video *streaming* dan kondisi baterai terhadap pergerakan perangkat *mobile*. Sedangkan skenario pengujian performa meliputi beberapa parameter, yaitu *delay streaming*, jumlah kejadian *handover*, waktu *handover*, dan penurunan daya baterai.
3. Tahap ketiga, merupakan tahapan dilakukannya uji coba fungsionalitas dan performa secara keseluruhan pada lingkungan *real device* sesuai dengan skenario pengujian yang telah dirancang sebelumnya.

#### **4.2. Implementasi Sistem**

Pada tahapan ini, sistem diimplementasikan dalam kondisi sebenarnya. Sistem diimplementasikan pada perangkat *smartphone* dengan spesifikasi sebagai berikut:

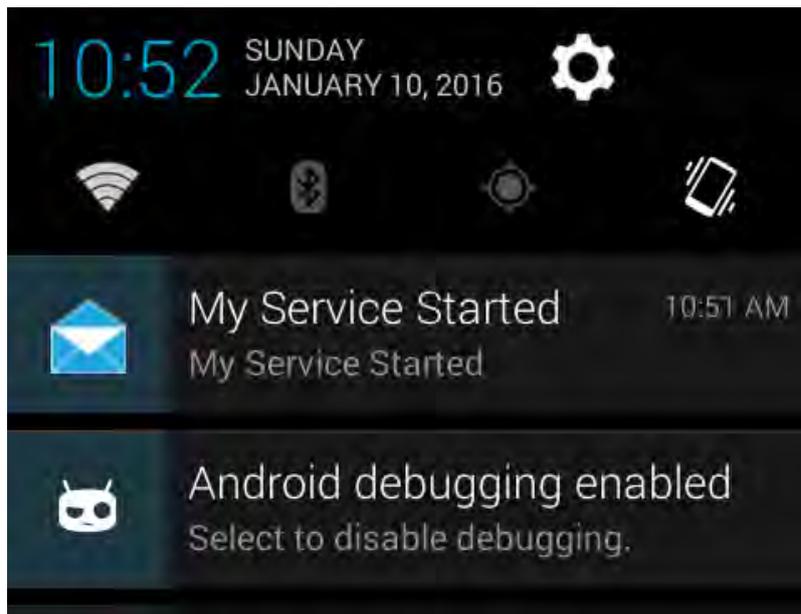
- a. Processor : ARMv7 Processor rev 2 (v7l)
- b. RAM : 384 MB

- c. Hardisk : 2GB
- d. Sistem Operasi : Android 4.1.2

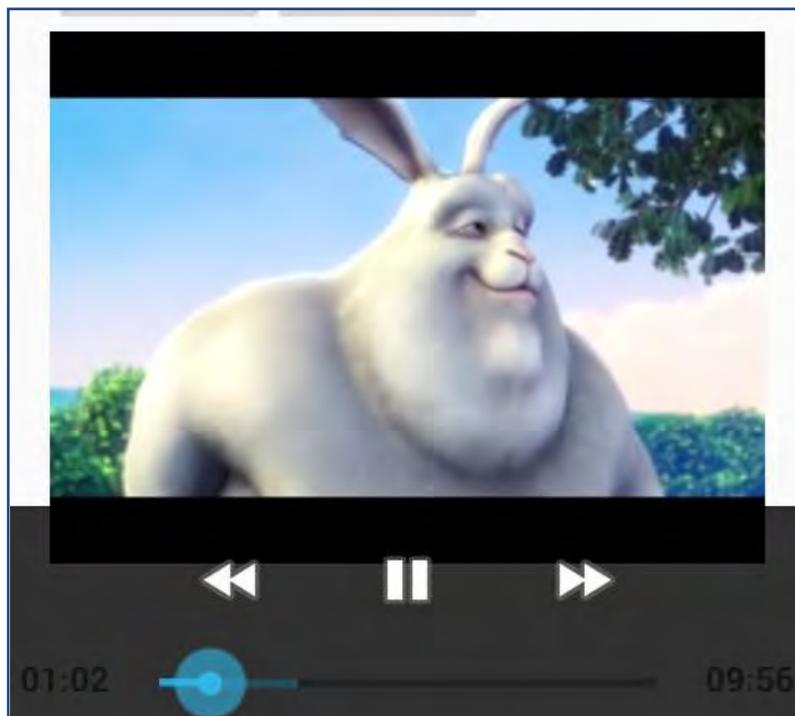
Sistem diimplementasikan kedalam dua bagian aplikasi yaitu aplikasi yang bertugas untuk mengatur proses *handover* dan aplikasi video *streaming*. Pembagian kedalam dua bagian ini bertujuan agar sistem berjalan sesuai dengan keadaan sebenarnya. Gambar 4.1, Gambar 4.2, dan Gambar 4.3 merupakan gambar antarmuka hasil *screenshot* dari aplikasi yang sudah diimplementasikan pada perangkat Android. Gambar 4.1 merupakan tampilan dari fitur *controller handover*, ketika tombol “START” ditekan maka *service* dari *handover* akan berjalan dan akan muncul notifikasi pada Gambar 4.2. Notifikasi pada *service* merupakan hal yang dibutuhkan jika ingin menjalankan *service foreground* atau *servicei* yang berjalan lama di *background*. Jika tidak dijadikan *service foreground* maka sebuah *service* akan dihapus keluar dari memori dalam waktu tertentu, yang merupakan fitur dari Android. *Service* berjalan berkomunikasi dengan aplikasi *video streaming* melalui mekanisme IPC (*interprocess communication*) karena berada dalam proses atau *thread* yang berbeda. Gambar 4.3 merupakan tampilan antarmuka aplikasi video *streaming* ketika dijalankan.



Gambar 4.1 Tampilan antarmuka aplikasi *handover*



Gambar 4.2 Tampilan notifikasi *service handover* berjalan



Gambar 4.3 Tampilan aplikasi *video streaming*

### **4.3. Langkah-langkah Uji Coba**

Tahapan uji coba pada penelitian ini meliputi penentuan parameter pengujian dan dilanjutkan dengan penentuan beberapa skenario pengujian. Tahap pertama, ditentukan parameter apa yang sesuai dengan tujuan penelitian. Tujuan penelitian ini adalah efektifitas penggunaan sumber daya pada *mobile terminal* dan menjaga kualitas *streaming* video melalui keputusan terjadinya *vertical handover*. Setelah menentukan parameter pengujian, tahapan selanjutnya adalah menentukan beberapa jenis skenario pengujian dan melakukan analisis terhadap hasil uji coba untuk mengukur tingkat keberhasilan sistem.

#### **4.2.1. Parameter Pengujian**

Beberapa parameter pengujian yang digunakan pada penelitian ini adalah:

1. Nilai *latency* jaringan
2. Kondisi *buffer* video pada aplikasi video *player*
3. Penurunan daya baterai

#### **4.2.2. Skenario Pengujian**

Skenario pengujian terdiri dari skenario pengujian performa dan fungsionalitas. Skenario pengujian yang dilakukan akan dijelaskan sebagai berikut.

##### **4.2.2.1. Skenario Pengujian Performa**

Skenario pengujian performa dilakukan untuk melihat tingkat keberhasilan sistem terhadap tujuan penelitian yang ingin dicapai. Beberapa skenario pengujian performa adalah sebagai berikut.

#### 4.2.2.1.1. Uji Coba Pengukuran *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 `setOnErrorListener` 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 Bussaneli dan metode yang diusulkan. Studi kasus yang digunakan adalah penggunaan kedua metode dengan video streaming yang sama. Ketika kedua mekanisme dibandingkan, melalui *log* masing-masing metode saat dijalankan, maka dapat diperoleh perbedaan *delay streaming* pada masing – masing metode. Pengujian ini dilakukan sesuai dengan variabel waktu menyesuaikan durasi dari video yang dilakukan *streaming*.

#### 4.2.2.1.2. Uji Coba Penghitungan Jumlah Kejadian *Handover*

Jumlah *handoff* yang terjadi mempengaruhi juga terhadap keberhasilan metode *handover*. 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 dengan N kali 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. variabel waktu untuk masing-masing mekanisme.

#### **4.2.2.1.3. Uji Coba Pengukuran Durasi *Handover***

Uji coba ini dilakukan untuk mengetahui pengaruh mekanisme perbaikan *vertical handover* ini terhadap durasi waktu *handoff*. Durasi 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. Pengujian yang digunakan untuk mengukur parameter ini adalah penggunaan dua metode 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.

#### **4.2.2.1.4. 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 selama  $\pm 360$  detik untuk masing-masing mekanisme *vertical handover*.

#### **4.2.2.2. Skenario Pengujian Fungsionalitas**

Pengujian fungsionalitas dilakukan untuk mengetahui sistem berjalan sesuai dengan aturan yang telah dibuat. Beberapa pengujian fungsionalitas dilakukan sebagai berikut.

##### **4.2.2.2.1. Uji Coba Perubahan Kondisi *Latency* Jaringan**

Uji coba ini dilakukan untuk mengetahui pengaruh kondisi jaringan (*latency*) terhadap pengambilan keputusan terjadinya *vertical handoff*. Pengujian akan

dilakukan sesuai dengan skenario yang telah dijelaskan pada subbab 3.5.1.1. Pengujian akan dilakukan sesuai terhadap variabel waktu yaitu selama  $\pm 360$  detik. Pada saat ini akan disimpan data *log* kondisi jaringan nilai T yang dihasilkan, serta keputusan *handoff* yang diambil. *Latency* pada saat pengujian akan dibuat bervariasi.

#### **4.2.2.2.2. Uji Coba Perubahan Kondisi *Buffer* Aplikasi Video Streaming**

Pengujian ini berkaitan dengan proses *streaming* oleh aplikasi lain yang sudah diimplementasikan. Pengujian ini dilakukan untuk mengetahui pengaruh kondisi video *streaming* terhadap pengambilan keputusan terjadinya *vertical handoff* apakah sudah sesuai dengan aturan yang sudah dibuat. Pengujian akan dilakukan sesuai dengan skenario yang telah dijelaskan pada subbab 3.5.1.1. Pengujian akan dilakukan sesuai terhadap variabel bilangan N kali melakukan *streaming*. Pada saat ini yang dicatat adalah nilai kondisi *streaming* dan T yang terbentuk serta keputusan *handoff* yang diambil.

#### **4.2.2.2.3. Uji Coba Perubahan Kondisi Level Baterai**

Pengujian ini dilakukan untuk mengetahui tentang kondisi baterai dan pengaruhnya terhadap pengambilan keputusan *handoff*. Kesesuaian antara kondisi baterai dengan aturan yang sudah dibuat. Uji coba dilakukan sesuai dengan skenario yang direncanakan pada subbab 3.5.1.1. Pengujian akan dilakukan selama  $\pm 360$  detik dengan *level* baterai yang berbeda.

#### **4.4. Hasil Uji Coba dan Analisis**

Pada penelitian ini, uji coba dilakukan dengan data masukan data real dari perangkat *mobile*. Analisis hasil uji coba untuk memperoleh suatu kesimpulan terhadap serangkaian penelitian yang telah dilakukan.

##### **4.3.1. Hasil Uji Coba Performa**

Hasil uji coba performa merupakan output dari uji coba performa sistem yang disajikan dalam bentuk gambar, grafik maupun tabel untuk mempermudah pemahaman mengenai hasil uji coba performa. Uji coba performa yang dilakukan pada penelitian ini meliputi *delay streaming*, jumlah kejadian *handover*, waktu *handover*, dan penurunan daya baterai. Hasil uji coba performa secara rinci akan dijelaskan pada subbab berikut.

###### **4.3.1.1. 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 `setOnErrorListener` 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 Bussaneli 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* yang bisa digunakan dengan akses melalui URL: [http://www.sample-videos.com/video/mp4/240/big\\_buck\\_bunny\\_240p\\_50mb.mp4](http://www.sample-videos.com/video/mp4/240/big_buck_bunny_240p_50mb.mp4) dengan durasi kurang

lebih 10 menit. Video diputar beberapa kali untuk diketahui perbandingan dari masing – masing metode.

**Tabel 4.1 Perbandingan delay pada dua metode**

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

Perbandingan hasil uji coba pengukuran *delay* ditunjukkan pada Tabel 4.1, dari metode Busanelli 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*.

#### 4.3.1.2. 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 4.2 Perbandingan jumlah kejadian *handover***

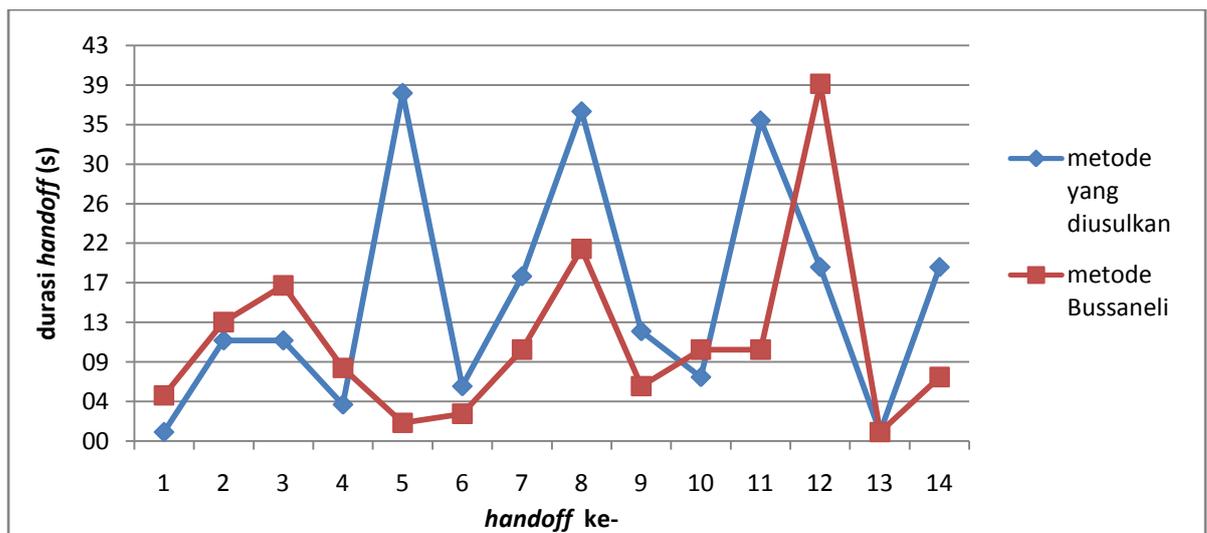
N	Kejadian <i>Handover</i>	
	metode Bussaneli(s)	metode yg diusulkan(s)
1	7	4
2	7	3
3	6	3
4	8	4
5	7	4
<b>rata-rata</b>	7	3.6

Perbandingan hasil uji coba pengukuran jumlah kejadian *handover* dapat dilihat pada Tabel 4.2, 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.

#### **4.3.1.3. 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 4.4 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* jaringan memerlukan waktu tersendiri pula.



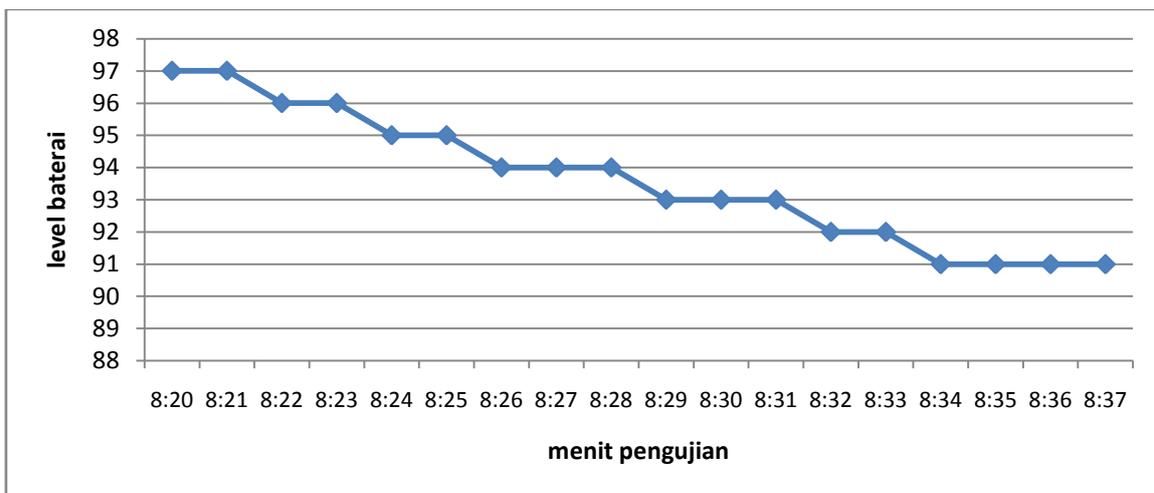
Gambar 4.4 Perbandingan durasi *handover*

#### 4.3.1.4. Hasil Uji Coba Penurunan Daya Baterai

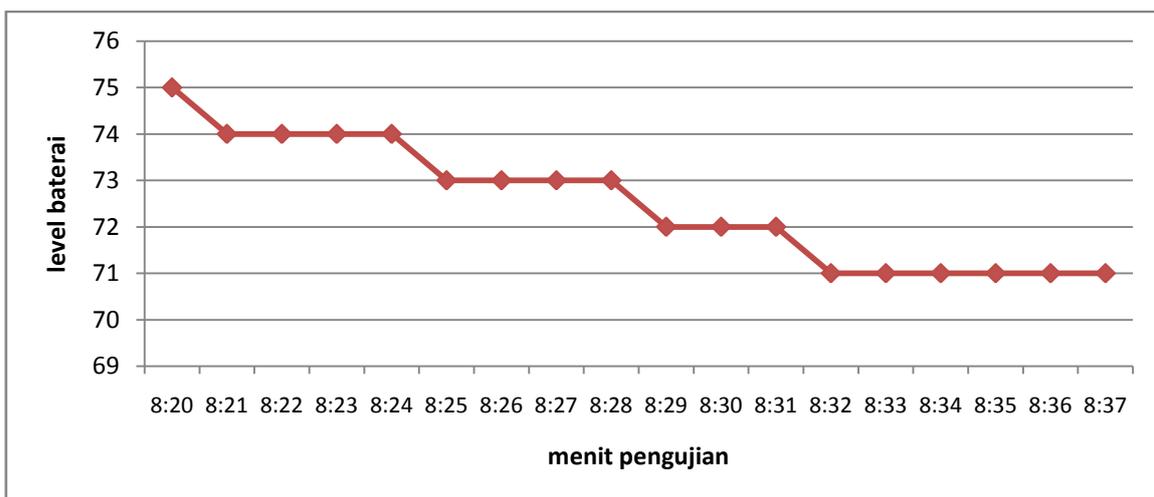
Uji coba ini dilakukan untuk mengetahui pengaruh mekanisme perbaikan *vertical handover* terhadap penurunan daya baterai dan penghematan daya baterai yang dapat dicapai dengan adanya sistem perbaikan mekanisme *vertical handover*. Untuk mengetahui penurunan daya baterai dan penghematan yang dapat dicapai,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

Dapat dilihat pada Gambar 4.5 dan Gambar 4.6 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 4.5 Hasil penurunan level baterai pada metode yang diusulkan



Gambar 4.6 Hasil penurunan level baterai pada metode Bussaneli

### 4.3.2. Hasil Uji Coba Fungsionalitas

Uji coba fungsionalitas dilakukan untuk mengetahui bahwa sistem berjalan sesuai dengan aturan yang telah dirancang. Uji coba dilakukan pada beberapa variabel pengujian, yaitu *vertical handover* berdasarkan prediksi perubahan kondisi jaringan, *vertical handover* berdasarkan prediksi perubahan persentase baterai dan *vertical handover* berdasarkan perubahan kondisi *video streaming*. Uji coba fungsionalitas yang dilakukan menggunakan konstanta masing – masing  $W_1, W_c, W_v$ , adalah 50, 20, dan 30.

#### 4.3.2.1. Hasil Uji Koefisien Menggunakan Regresi Linear

Penentuan konstanta diperhitungkan dengan menggunakan regresi linear, menggunakan program excel, ditunjukkan pada Gambar 4.7. Dari Gambar 4.7 dapat dilihat bahwa nilai statistik regresi, nilai *R square* menunjukkan korelasi koefisien memiliki korelasi linear yang kuat. Koefisien bernilai positif artinya terjadi hubungan positif antara nilai variabel  $W_1$  dengan nilai T yang dijelaskan pada persamaan 3.1, semakin tinggi nilai  $W_1$  maka nilai T akan semakin tinggi pula. Sedangkan nilai negatif pada variabel menunjukkan hubungan sebaliknya.

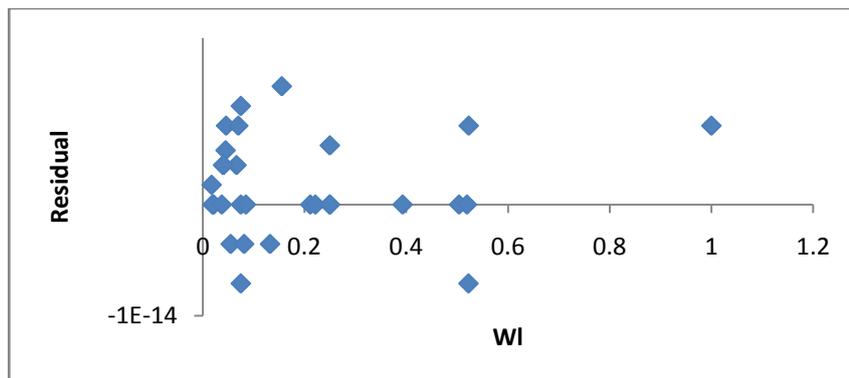
SUMMARY OUTPUT								
<i>Regression Statistics</i>								
Multiple R	1							
R Square	1							
Adjusted R S	1							
Standard Error	2.72E-15							
Observation	30							
<i>ANOVA</i>								
	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>Significance F</i>			
Regression	3	4878.758417	1626.252806	2.2E+32	0			
Residual	26	1.92575E-28	7.40671E-30					
Total	29	4878.758417						
	<i>Coefficients</i>	<i>Standard Error</i>	<i>t Stat</i>	<i>P-value</i>	<i>Lower 95%</i>	<i>Upper 95%</i>	<i>Lower 95.0%</i>	<i>Upper 95.0%</i>
Intercept	50	2.73132E-15	1.83061E+16	0	50	50	50	50
W1	50	2.27354E-15	2.19921E+16	0	50	50	50	50
Wb	-20	2.76541E-15	-7.23221E+15	0	-20	-20	-20	-20
Wv	-30	1.8508E-15	-1.62092E+16	0	-30	-30	-30	-30

Gambar 4.7 Hasil analisa penentuan koefisien menggunakan regresi linear pada program excel 2007

Selisih dari nilai T dan nilai prediksi T yang dihasilkan dari proses regresi ditunjukkan pada Tabel 4.3 disebut nilai *residual*. Grafik nilai residual terhadap variabel bebas digunakan untuk menentukan apakah model persamaan termasuk dalam model linear atau non linear. Pada Gambar 4.8 ditampilkan hubungan antara nilai *residual* dan variable  $W_1$ , menunjukkan pola yang cukup acak, sehingga dapat disimpulkan bahwa model persamaan merupakan model yang cocok sebagai model linear.

**Tabel 4.3 Hasil nilai *residual* antara T dan Predicted T**

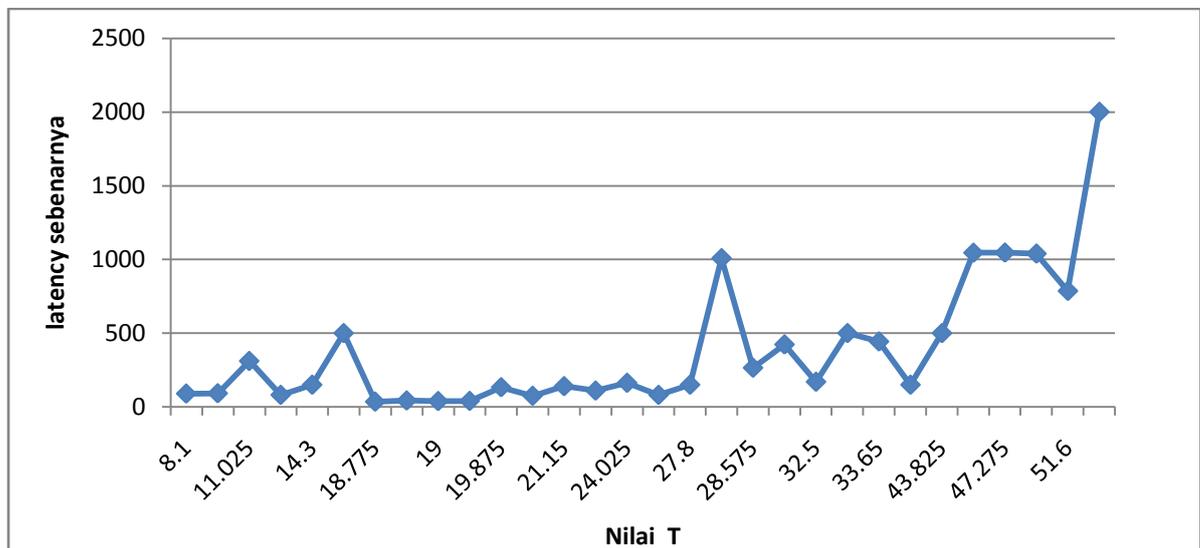
<i>Observation</i>	<i>Predicted T</i>	<i>Residuals</i>	<i>Observation</i>	<i>Predicted T</i>	<i>Residuals</i>
<b>1</b>	3.65	4.88E-15	<b>16</b>	24.025	3.55E-15
<b>2</b>	8.1	7.11E-15	<b>17</b>	26.75	0
<b>3</b>	9.375	1.07E-14	<b>18</b>	27.8	0
<b>4</b>	11.025	3.55E-15	<b>19</b>	28.325	-3.6E-15
<b>5</b>	11.75	8.88E-15	<b>20</b>	28.575	0
<b>6</b>	14.3	5.33E-15	<b>21</b>	31.05	0
<b>7</b>	14.675	1.78E-15	<b>22</b>	32.5	0
<b>8</b>	18.775	0	<b>23</b>	32.575	0
<b>9</b>	18.975	0	<b>24</b>	33.65	-7.1E-15
<b>10</b>	19	0	<b>25</b>	40.5	0
<b>11</b>	19.525	3.55E-15	<b>26</b>	43.825	-7.1E-15
<b>12</b>	19.875	0	<b>27</b>	44.15	7.11E-15
<b>13</b>	20.1	7.11E-15	<b>28</b>	47.275	0
<b>14</b>	21.15	-3.55E-15	<b>29</b>	51.35	0
<b>15</b>	23.875	-3.55E-15	<b>30</b>	51.6	7.11E-15



**Gambar 4.8 Hubungan variabel  $W_1$  terhadap nilai residual**

#### 4.3.2.2. Hasil Uji Coba Perubahan Kondisi *Latency* Jaringan

Uji coba ini dilakukan untuk mengetahui pengaruh kondisi jaringan (*latency*) terhadap pengambilan keputusan terjadinya *vertical handoff*. Pengujian akan dilakukan sesuai dengan skenario yang telah dijelaskan pada subbab 3.5.1.1. Pengujian akan dilakukan sesuai terhadap variabel N kali *handoff* terjadi yaitu sebanyak 30 kali. Dari *log* yang dihasilkan pada percobaan yang sudah dijalankan didapatkan hubungan nilai *latency* terhadap nilai T yang ditunjukkan pada Gambar 4.9. Dilihat dari Gambar 4.9 hubungan *latency* dengan nilai T sudah sesuai, yaitu linear. Jika *latency* tinggi maka nilai T akan naik juga, terkecuali jika ada pengaruh dari parameter yang lain.

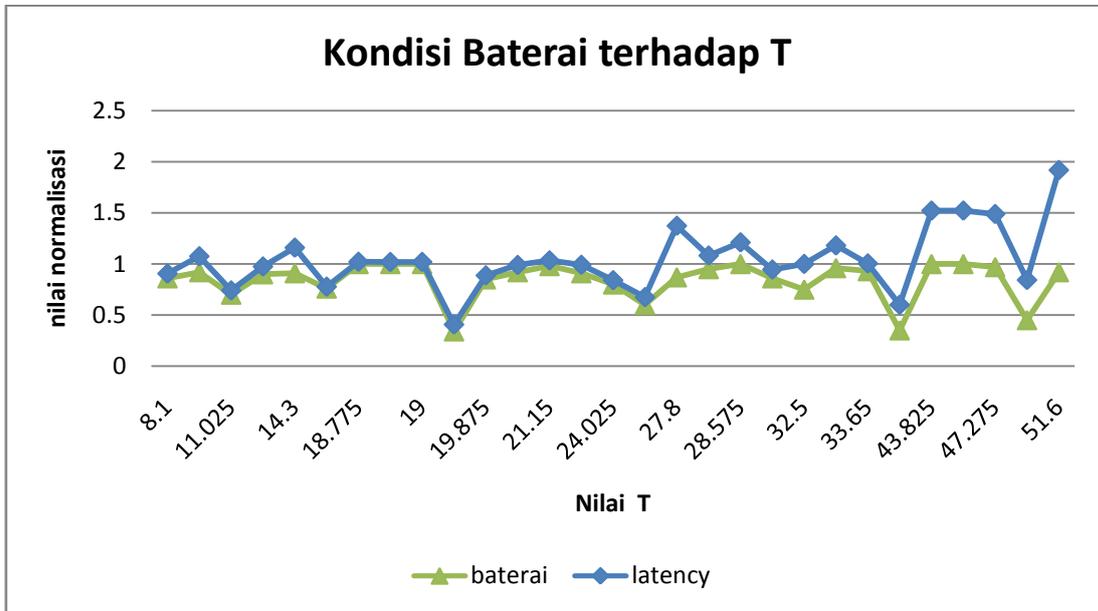


Gambar 4.9 Pengaruh *Latency* terhadap Nilai Kualitas (T)

#### 4.3.2.3. Hasil Uji Coba Perubahan Persentase Baterai

Pengujian ini dilakukan untuk mengetahui tentang kondisi baterai dan pengaruhnya terhadap pengambilan keputusan *handoff*. Kesesuaian antara kondisi baterai dengan aturan yang sudah dibuat. Uji coba dilakukan sesuai dengan skenario yang direncanakan pada subbab 3.5.1.1. Pengujian akan dilakukan sebanyak 30 kali dengan *level* baterai yang berbeda. Dapat dilihat pada Gambar 4.10 pengaruh kondisi

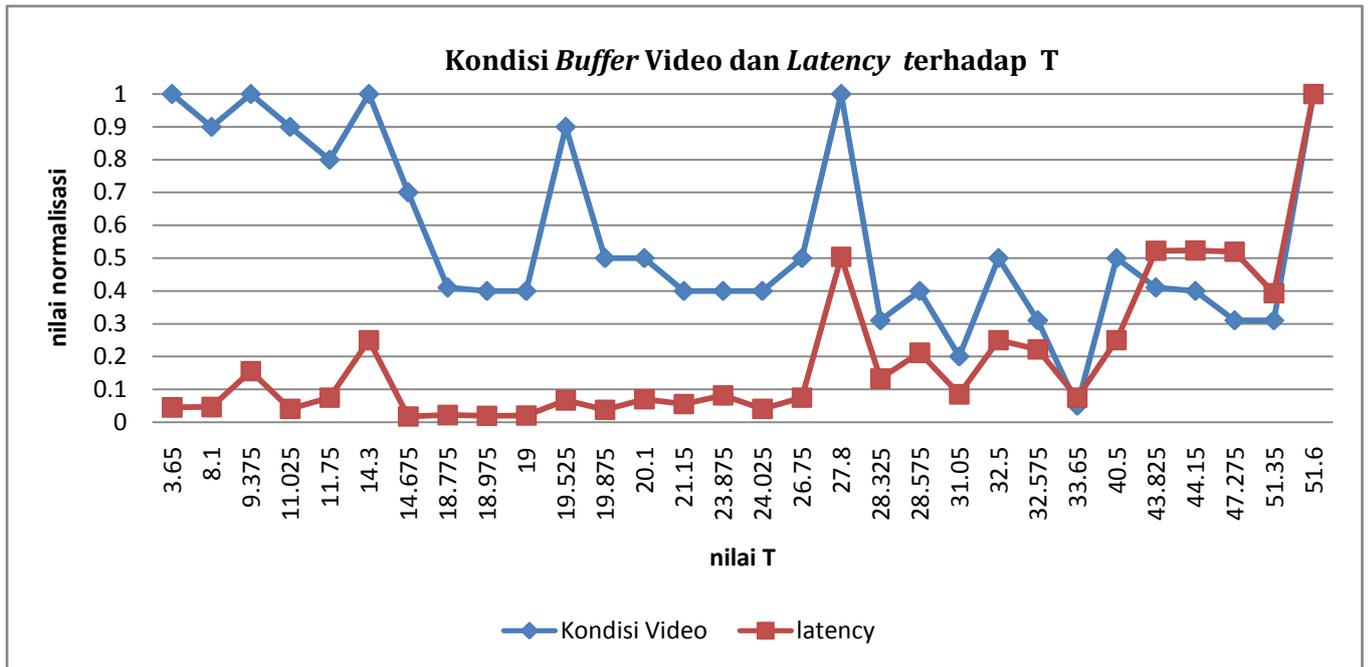
baterai terhadap nilai T yang dihasilkan. Gambar 4.10 menunjukkan bahwa nilai linear dari baterai tidak terlalu memengaruhi nilai T, disini dikarenakan konstanta yang ditetapkan untuk pengujian, yaitu 20.



Gambar 4.10 Pengaruh Kondisi Baterai terhadap Nilai Kualitas(T)

#### 4.3.2.4. Hasil Uji Coba Perubahan Kondisi *Buffer Video*

Pengujian ini berkaitan dengan proses *streaming* oleh aplikasi lain yang sudah diimplementasikan. Pengujian ini dilakukan untuk mengetahui pengaruh kondisi video *streaming* terhadap pengambilan keputusan terjadinya *vertical handoff* apakah sudah sesuai dengan aturan yang sudah dibuat. Pengujian akan dilakukan sesuai dengan skenario yang telah dijelaskan pada subbab 3.5.1.1. Pengujian akan dilakukan sesuai terhadap variabel bilangan N kali melakukan *streaming*. Hasil dari pengujian ditunjukkan melalui Gambar 4.11. Gambar 4.11 menunjukkan bahwa nilai linear dari video tidak terlalu memengaruhi nilai T, disini dikarenakan konstanta yang ditetapkan untuk pengujian, yaitu 30.



Gambar 4.11 Kondisi Buffer Video dan Latency terhadap Nilai Kualitas (T)

**[Halaman ini sengaja dikosongkan]**

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari serangkaian penelitian yang telah dilakukan dan saran yang dapat diambil untuk pengembangan penelitian

#### **5.1. Kesimpulan**

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

1. Perbaikan strategi *vertical handover* pada jaringan nirkabel heterogen berdasarkan parameter jaringan dan kondisi 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 2.67 detik, dibandingkan dengan metode yang diusulkan dengan rata – rata *delay* diperoleh 1.33 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 6.67 kali kejadian *handover*. Metode yang diusulkan memiliki jumlah *handover* yang lebih baik yaitu 3.33 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.

## **5.2. Saran**

Pada sistem perbaikan mekanisme *vertical handover* berdasarkan parameter jaringan dan kondisi aplikasi *streaming* video ini juga masih terdapat kekurangan. Metode *vertical handover* ini masih menjadi penelitian yang menarik untuk dikembangkan. Salah satu pengembangannya adalah penelitian mengenai sistem perbaikan mekanisme *vertical handover* dengan mempertimbangkan parameter yang lain atau penyesuaian yang lebih spesifik, seperti *cost data*, *resource*, dan lain lain dengan kondisi yang berbeda.

## LAMPIRAN

### Contoh log parameter dan nilai T

<jenis koneksi><baterai><video><latency><t min><t><waktu>

```
wifi prmB 1.0 prm vid : 51 lat : 89.6 32.5 17.54 08-01-2016_08:20:06
wifi prmB 1.0 prm vid : 51 lat : 1046.082 32.5 41.45205 08-01-2016_08:20:12
mobile no wifi prmB 1.0 prm vid : 40 lat : 2000 08-01-2016_08:20:18
mobile no wifi prmB 1.0 prm vid : 40 lat : 2000 08-01-2016_08:20:20
mobile prmB 0.98 prm vid : 9 lat : 2000 08-01-2016_09:24:47
mobile no wifi prmB 0.97 prm vid : 21 lat : 487.488 08-01-2016_09:33:55
mobile prmB 0.97 prm vid : 26 lat : 2000 08-01-2016_09:34:17
```

### Contoh log metode Bussaneli

<jenis koneksi><RSS><waktu>

```
wifi rssi 4 08-01-2016_11:40:35
wifi rssi 4 08-01-2016_11:40:51
mobile ada wifi 08-01-2016_11:41:35
mobile ada wifi 08-01-2016_11:41:35
mobile ada wifi 08-01-2016_11:42:36
```

### Contoh log deteksi delay streaming

<jenis error><waktu error>

```
error timed out 10-01-2016_11:05:14
error timed out 10-01-2016_11:05:33
error timed out 10-01-2016_11:31:19
error timed out 10-01-2016_12:21:48
error timed out 10-01-2016_13:33:33
error unknown 10-01-2016_13:36:00
error unknown 10-01-2016_16:16:14
error timed out 10-01-2016_16:58:57
error timed out 10-01-2016_17:17:49
```

**[Halaman ini sengaja dikosongkan]**

## DAFTAR PUSTAKA

- Akyildiz, I.F., xie, J. and Mohanty, (2004) 'A survey of mobility management in next-generation all-IP-based wireless sytems', *IEEE*, vol. 11, pp. 16-28.
- Apache 2.0 (2015) *Android Developer*, Desember, [Online], Available: <http://developer.android.com/reference/android/telephony/TelephonyManager.html> [Desember 2015].
- Balasubramaniam, S. and J., I. (2004) 'Vertical Handover Supporting Pervasive Computing inFuture Wireless Networks', *Computer Communications Elsevier*, vol. 27, no. 8, pp. 708-719.
- Busanelli, S., Martalo, M., F, G. and S, G. (2011) 'Vertical Handover between WiFi and UMTS Networks: Experimental Performance Analysis', *International Journal of Energy, Information and Communications* , vol. 2, no. 1, Februari.
- Cardoso, Neves, T.a. and Ricardo, P.a. (2011) 'Media independent handover management in heterogeneous access networks-an empirical evaluation', *IEEE*, vol. 73rd, Mei, pp. 1-5.
- Gondara, M.K. and Kadam, S. (2011) 'REQUIREMENTS OF VERTICAL HANDOFF MECHANISM IN 4G WIRELESS NETWORKS', *International Journal of Wireless & Mobile Networks (IJWMN)*, vol. Vol. 3, no. No. 2, April.
- Grigorik, I. (2013) *High Performance Browser Networking*, O'Reilly Atlas.
- Hong, J.W.-k. and Garcia, A.L. (2005) 'Requirements for the Operations and Management of 4G networks', *International Conference on Performance Challenges for Efficient Next Generation Networks*, vol. 19th, pp. 981-990.
- K. Taniuchi, Y.O.V.F.S.D.M.T.Y.-H.C.A. (2009) "'IEEE 802.21: Media independent handover: Features, applicability, and realization', *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 112-120.

- Kassar, , Kervella, and Pujolle, (2008) 'An overview of vertical handover decision strategies in heterogeneous wireless networks', *Computer Communication Elsevier*, vol. 31, Januari, pp. 2607–2620.
- Kurose, F. and Ross, W. (2013) *Computer Networking: A Top-Down Approach*, 6<sup>th</sup> edition, Pearson Education.
- Ma, L., Yu, F., Leung, V.C. and Randhawa, a.T. (2004) 'A new method to support UMTS/WLAN vertical handover using SCTP', *Wireless Communications, IEEE*, vol. 11, Agustus, pp. 44-51.
- Ong, L. (2002) 'An introduction to the stream control transmission protocol', *IETF Document*.
- Petander, H. (2009) 'Energy-aware network selection using traffic estimation', Proceedings of the 1st ACM workshop on Mobile internet through cellular networks, 55-60.
- R., T., G., P. and O., S. (2008) 'A vertical handoff decision scheme in heterogeneous wireless systems', Vehicular Technology Conference (VTC'08 - Spring), Marina Bay - Singapore, 2626-2630.
- S.P, , and Anand , S. (2011) 'A Novel Scalable Software Platform on Android for efficient QoS on Android', *Wireless Telecommunications Symposium IEEE*, pp. 1-6.
- Silva, R., Carvalano, and Sousa, P. (2011) 'Enabling Heterogeneous Mobility in Android Devices', *Springer Science*, Juni, pp. 518-528.
- Soderman, P..G.K..C.G..I.Y..&.B.A. (2012) 'Sub-Second Transport Layer Vertical Handover Using mSCTP in Android Mobile Devices', *IEEE*, pp. 661-665.
- Yan, X., Y., A.S. and Narayanan, S. (2010) 'A survey of vertical handover decision algorithm in Fourth Generation heterogeneous wireless networks', *elsevier*, vol. 54, Februari, pp. 1848-1863.

## BIODATA PENULIS



Bagus Seta Inba Cipta, biasa dipanggil Bagus, dilahirkan di kota Malang pada hari sumpah pemuda tahun 1991. Penulis adalah anak sulung dari tiga bersaudara dan dibesarkan di kota kelahiran Malang, Jawa Timur. Penulis menempuh pendidikan SDI Dewi Mayithoh Gondanglegi (1997-2003), MTsN Malang 3 (2003-2006), SMA Negeri 1 Kepanjen (2006-2009). Pada tahun 2009, penulis mengikuti SNMPTN dan diterima di strata satu Jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan 2009. Tahun 2012 penulis menempuh strata dua di Jurusan Teknik Informatika ini, penulis mengambil bidang minat Net Centric Computing (NCC). Selama menempuh kuliah, penulis juga aktif sebagai anggota departemen dalam negeri di Himpunan Mahasiswa Teknik Computer (HMTC) C-19. Penulis juga aktif sebagai administrator lab NCC selama masa perkuliahannya. Pada beberapa acara kampus, penulis juga beberapa kali aktif menjadi panitia, baik sebagai anggota maupun koordinator. Selain itu, penulis beberapa kali menjadi asisten, diantaranya Asisten mata kuliah Pemrograman Terstruktur, Asisten Sistem Operasi, Asisten mata kuliah PIKTI ITS, dan Asisten Praktikum Sistem Operasi dan Jaringan Komputer. Penulis dapat dihubungi melalui alamat e-mail di [bagus28seta@gmail.com](mailto:bagus28seta@gmail.com) atau [basic147@yahoo.co.id](mailto:basic147@yahoo.co.id)