PERBANDINGAN KINERJA PADA PROTOKOL OBEX DAN RFCOMM UNTUK RANCANG BANGUN SISI *SERVER* APLIKASI PSEUDO VIDEO *STREAMING* DENGAN TEKNOLOGI BLUETOOTH

SKRIPSI

Oleh

WISNU MURDIANTO
04 03 03 102 7

DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GANJIL 2007/2008

PERBANDINGAN KINERJA PROTOKOL OBEX DAN RFCOMM UNTUK RANCANG BANGUN SISI *SERVER* APLIKASI PSEUDO VIDEO *STREAMING* DENGAN TEKNOLOGI BLUETOOTH

SKRIPSI

Oleh

WISNU MURDIANTO
04 03 03 102 7

SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK

DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA

GANJIL 2007/2008

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

PERBANDINGAN KINERJA PROTOKOL OBEX DAN RFCOMM UNTUK RANCANG BANGUN SISI *SERVER* APLIKASI PSEUDO VIDEO *STREAMING* DENGAN TEKNOLOGI BLUETOOTH

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 7 Januari 2008

Wisnu Murdianto

NPM. 04 03 03 102 7

LEMBAR PENGESAHAN

Skripsi dengan judul :

PERBANDINGAN KINERJA PROTOKOL OBEX DAN RFCOMM UNTUK RANCANG BANGUN SISI *SERVER* APLIKASI PSEUDO VIDEO *STREAMING* DENGAN TEKNOLOGI BLUETOOTH

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian skripsi. Skripsi ini telah diujikan pada sidang ujian skripsi pada tanggal 4 Januari 2008 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 8 Januari 2008

Dosen Pembimbing

Dr. Ing.Ir. Kalamullah Ramli, M. Eng.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada :

Dr. Ing.Ir. Kalamullah Ramli, M. Eng

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.

Wisnu Murdianto
NPM 04 03 03 102 7
Departemen Teknik Elektro

Dosen Pembimbing
Dr. Ing.Ir. Kalamullah Ramli, M. Eng.

PERBANDINGAN KINERJA PROTOKOL OBEX DAN RFCOMM UNTUK RANCANG BANGUN SISI *SERVER* APLIKASI PSEUDO VIDEO *STREAMING* DENGAN TEKNOLOGI BLUETOOTH

ABSTRAK

Tuntutan akan kemudahan dan kepraktisan pengguna akan keberadaan sistem data dan informasi, khususnya media digital, yang dapat diakses secara *mobile* setiap saat memacu terus berlangsungnya pengembangan teknologi. Hal ini tampak dengan semakin mudahnya ditemui berbagai perangkat *mobile*, seperti telepon selular, PDA (*Personal Digital Assistant*), *tablet* PC *notebook,* pada kehidupan sehari-hari.

Aplikasi *video streaming* melalui teknologi Bluetooth adalah sebuah alternatif yang menarik untuk dikembangkan. Teknologi Bluetooth menawarkan fitur *wireless mobile* yang cukup baik dan efisien, baik dari sisi biaya maupun konsumsi daya, serta dapat menyediakan koneksi secara *real time* pada jarak jangkauan layanan hingga 100 meter. Teknologi ini beroperasi pada frekuensi 2,4 GHz. Teknologi ini telah banyak dijumpai di sebagian besar perangkat *mobile*.

Pada skripsi ini, dilakukan perbandingan kinerja obex dan RFCOMM untuk rancang bangun aplikasi dan sistem pada sisi PC server. Aplikasi dibangun berbasiskan Java dengan menggunakan J2SE (*Java Standar Edition*) dan IDE Netbeans 5.5.1. Protokol yang dipakai menggunakan RFCOMM dan Obex. Uji coba membandingkan kinerja kedua protokol tersebut berdasarkan kemampuan menemukan divais bluetooth lain disekitarnya dan kinerja keduanya dalam hal melakukan transfer file.

Hasil kinerja ini menunjukan protokol RFCOMM lebih baik daripada protokol Obex diukur berdasarkan waktu pencarian divais bluetooth lainnya. Sedangkan untuk transfer file kinerja Obex lebih baik ketimbang RFCOMM.

Kata Kunci : *video streaming*, bluetooth, J2ME, telepon selular, divais *mobile*

Wisnu Murdianto
NPM 04 03 03 102 7
Electrical Engineering Department

Counselor
Dr. Ing. Ir.Kalamullah Ramli, M. Eng.

A COMPARISON BETWEEN OBEX AND RFCOMM PROTOCOL FOR AN IMPLEMENTATION SYSTEM OF SERVER SIDE'S PSEUDO VIDEO STREAMING APPLICATION USING BLUETOOTH CONNECTION

ABSTRACT

The high increasing of computer's computing power supports the usage of very efficient video compression technologies therefore can be transmitted into network via several methode, including wireless streaming. Video streaming is a video access services which has been transmitted *real-time* and continuously, without must download a whole file before. Video encode, compression, and fragmentation are principal process that should be through.

Bluetooth is the promising wireless technology for electronic and mobile devices. Speed data rate up to 732 kbps allow not only transmission of the web and e-mail, but also audio/video conferencing dan streaming of live shows.There are not many research yet have been done about video streaming over Bluetooth, and for the further this is an opportunity and challenge for the researcher to develop it.

This final project focuses on implementation and communication systems of video streaming application using Bluetooth connection on client side, especially user interface at smartphone. This service is given by computer server to client (smartphone). Application is designed dan developed using Java 2 Micro Edition (J2ME) platform and Netbeans Mobility 5.5.1 as Integrated Development Environment (IDE). The protocol that is use for this project is RFCOMM and Obex. The comparison for the two protocol is based by their perfomance on service discovery and file transfer. The performance show that the RFCOMM is better than Obex on service discovery. But for transfer file data the Obex is much better that RFCOMM.

Keywords : video streaming, bluetooth, J2ME, cellular phone, mobile device

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

3GPP	3rd Generation Partnership Project
ACL	Asynchronous ConnectionLess
AMR	Adaptive Multirate
API	Application Programming Interface
Codec	Compression/decompression
GUI	Graphical User Interface
HCI	Host Controller Interface
IDE	Integrated Development Environment
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
ISDN	Integrated Services Digital Network
J2ME	Java 2nd Micro Edition
J2SE	Java 2nd Standard Edition
JABWT	Java APIs For Bluetooth Wireless Technology
L2CAP	Logical Link and Control Adaptation Protocol
MMAPI	Mobile Media API
MPEG	Motion Picture Expert Group
SCO	Synchronous Connection Oriented
QoS	Quality of Service

DAFTAR ISTILAH

Bluetooth	Teknologi wireless yang beroperasi pada frekuensi 2,4 GHz
Client	Divais bergerak yang dilayani oleh *server* dan dioperasikan oleh pengguna (*smartphone*, PDA)
Server	Komputer pusat yang mengatur sistem
Smartphone	Divais telepon yang dilengkapi sistem operasi terbuka (Symbian, Linux, Pocket PC, Palm) dan memori yang cukup besar sehingga mampu menjalankan beragam fungsi
Video Streaming	Video yang dijalankan secara terus menerus

BAB I
PENDAHULUAN

1.1 LATAR BELAKANG

Sebuah era baru dalam dunia komunikasi telah tiba. Perkembangan teknologi 3 G atau 3,5 G semakin pesat. Hal ini turut mempengaruhi berkembangnya perangkat lunak dan perangkat keras yang ada, sebagai contoh telepon selular yang dahulu hanya digunakan untuk melakukan panggilan dan mengirim pesan singkat, dengan teknologi yang ada sekarang turut menyertakan fitur-fitur tambahan seperti *pocket radio*, *cameraphone,* pemutar video, bahkan berkembangnya aplikasi-aplikasi berbasiskan java seperti java *game.*

Video streaming dapat didefiniskan sebagai layanan akses video yang dijalankan secara terus menerus, tanpa harus men-*download* file secara utuh terlebih dahulu. Contoh implementasinya yang kini dikembangkan yaitu *video streaming* pada *Mobile Learning.* Berkaitan dengan hal itu, penulis berupaya untuk membuat suatu aplikasi *video streaming* dimana penggunanya tidak perlu mengeluarkan biaya untuk menikmatinya, yakni dengan memanfaatkan teknologi yang sudah banyak terdapat pada telepon selular yaitu *bluetooth.*

Bluetooth adalah sebuah teknologi komunikasi tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* dengan jarak jangkauan layanan sampai dengan 100 m. Pada dasarnya *bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel di dalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *mobile wireless* dengan biaya yang relatif murah. Teknologi ini telah banyak tersedia di sebagian besar perangkat *mobile* (telepon selular, PDA, dll).

1.2 PERUMUSAN MASALAH

Penelitian ini akan membahas upaya pembuatan aplikasi *video streaming* pada sisi server dengan memanfaatkan teknologi yang sudah banyak terdapat pada telepon

selular yaitu *bluetooth*. Aplikasi tersebut diimplementasikan dengan bahasa pemrograman berorientasi objek J2SE (*Java2 Standard Edition*).

Pembangunan tahapan koneksi Bluetooth dengan menggunakan J2SE menjadi suatu tantangan tersendiri yang harus dihadapi.

1.3 TUJUAN PENELITIAN

Skripsi ini ditujukan untuk mengimplementasikan aplikasi *video streaming* server berbasis J2SE pada PC (*Personal Computer*) dengan menggunakan koneksi teknologi Bluetooth. *Streaming* dilakukan dari PC, sebagai *server,* menuju ke telepon selular, sebagai *client.*

1.4 BATASAN MASALAH

Pada skripsi ini akan dibahas mengenai konsep dan desain pembuatan aplikasi *video streaming* pada PC sebagai *server* melalui koneksi Bluetooth berbasiskan J2SE.

Implementasi *video streaming* yang dilakukan mempunyai sumber *streaming server* berupa PC, dan tujuan *streaming client* berupa telepon selular. Aplikasi pada telepon selular yang dibuat dapat memainkan video tanpa dapat melakukan proses *editing* maupun *rewind*, adapun video tersebut telah direkam sebelumnya dan memiliki format .3gpp. *Server* yang dibuat hanya dapat menangani sebuah *client.*

1.5 METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan yaitu :

a) Studi literatur

   Mempelajari informasi dari berbagai sumber literatur, seperti: jurnal, buku, dan artikel-artikel yang berkaitan dengan aplikasi yang dibuat.

b) Desain dan pembangunan aplikasi *video streaming* melalui Bluetooth pada sisi server, dalam hal ini PC, dengan menggunakan metode berorientasi objek. Tahapan yang tercakup didalamnya adalah analisa dan identifikasi kebutuhan sistem, perancangan sistem, dan dilanjutkan dengan pengimplementasiannya pada bahasa pemrograman J2ME.

c) Penarikan kesimpulan.

1.6 SISTEMATIKA PENULISAN

Penulisan skripsi ini dibagi menjadi 5 (lima) bab, dengan sistematika penulisan sebagai berikut :

BAB 1 PENDAHULUAN

Membahas mengenai latar belakang penulisan, perumusan masalah, tujuan penelitian, pembatasan masalah, metodologi pembahasan, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Berisi tentang konsep *video streaming*, Bluetooth, serta hal-hal lainnya yang kaan digunakan dalam pembangunan aplikasi.

BAB 3 PERANCANGAN SISTEM dan APLIKASI

Menjelaskan tentang analisa perancangan sistem server dan aplikasinya dengan menggunakan metode *object oriented,* dengan tercakup di dalamnya diagram alir (*flowchart*).

BAB 4 ANALISA IMPLEMENTASI pada SISTEM

Membahas tentang implementasi dan konfigurasi sistem yang akan dipergunakan bagi penyelenggaran aplikasi.

BAB 5 PENUTUP

Berisi mengenai kesimpulan atas data dan analisa yang telah dihasilkan dalam penelitian.

BAB II
LANDASAN TEORI

2.1 *BLUETOOTH*

2.1.1 Sejarah

Bluetooth adalah suatu spesifikasi standard terbuka untuk suatu frekwensi radio (RF)-based, teknologi connectivas jarak pendek yang memberikan harapan untuk merubah wajah komputasi dan komunikasi wireless saat ini. Bluetooth dirancang untuk menjadi suatu sistem networking murah, system jaringan wireless untuk semua kelas alat portable, seperti laptop, PDA (personal digital assistants), dan handphone atau mobile phone. Bluetooth juga akan memungkinkan koneksi tanpa kawat untuk komputer desktop, membuat koneksi antara monitor, printer, keyboard, dan CPU itu tanpa kabel.

Gagasan untuk suatu cable-free, atau wireless, teknologi ini pada awalnya dipahami oleh Ericsson di tahun 1994, manakala perusahaan Ericsson memulai suatu studi untuk menyelidiki kelayakan dari suatu teknologi yang hemat energi, hemat biaya untuk menghubungkan antara telepon gengam dan asesoris mereka. Tujuan utama dari perusahaan Ericsson ini adalah menghapuskan kebutuhan akan kabel (cable-free).

Gagasan awal perusahaan Ericsson adalah menciptakan suatu chip radio kecil, murah, yang bisa digunakan pada laptop, printer, handphone, dan seterusnya untuk memancarkan atau tukar menukar data antar Bluetooth. Dengan Chip radio, Bluetooth mampu menggantikan fungsi kabel. Biaya pembuatan chip di perkirakan sekitar $ 5, dan chip itu harus dapat digunakan pada low power sehingga bisa dipakai pada alat yang bersandar pada baterei.

Ketika gagasan ini berkembang, suatu kelompok pengembagan khusus **SIG** special interest group) dibentuk untuk menciptakan suatu standard untuk teknologi ini.

SIG yang pertama kali dibentuk pada tahun 1998, terdiri dari lima perusahaan:

- Ericsson
- IBM
- Intel
- Nokia

- Toshiba

Empat perusahaan utama lainnya (Microsoft, 3Com, Lucent, dan Motorola) kemudian bergabung dengan **SIG** dan mendirikan Bluetooth Promoter Group. Banyak lagi perusahaan lain yang sejak itu telah menjadi bagian dari revolusi Bluetooth, mengembangkan visi yang asli, dan membantu mengarahkan pengembangan dari teknologi baru ini.

Nama Bluetooth berasal dari sejarah orang Denmark. Harald Blatand, yang biasa disebut Bluetooth, adalah putra Raja Gorm The Old, yang menguasai Jutland, semenanjung Denmark yang utama. Pada saat Harald menjadi raja, ia adalah prajurit Viking terampil. Maka, manakala saudarinya yang meminta bantuan untuk mengamankan Norwegia setelah suaminya meninggal, Harald dengan cepat menangkap kesempatan ini untuk mempersatukan negara-negara dan memperluas kerajaannya. Pada tahun 960 menurut cerita itu, Harald pada puncak kekuasaannya, dan menguasai baik Norwegia dan Denmark.

Walaupun menurut kepercayaaan yang terkenal bahwa Raja itu Harald mempunyai suatu gigi biru, dan berbagai cerita lainnya yang menjelaskan bagaimana ini terjadi, ini lebih mungkin bahwa nama Bluetooth adalah Bahasa Inggris yang diserap dari kata Viking asli, Blâtand. Nama Bluetooth dipilih untuk teknologi tanpa kawat atau wireless ini sebab harapan dari penyelenggara dan pengembangnya akan dapat mempersatukan dunia mobile, sama halnya dengan Raja Harald yang mempersatukan dunianya.

2.1.2 Contoh Pemakaian Bluetooth

Pemanfaatan Bluetooth dapat sangat luas, namun yang sudah berjalan sekarang ini di antaranya:

Wireless headset

Istilah dan manfaat dari Bluetooth paling sering kita temui dalam dunia handphone. Hampir semua produsen handphone sekarang ini sudah mengeluarkan versi handphone yang dilengkapi Bluetooth. Hal ini memungkinkan penggunaan wireless headset bersamaan dengan handphone tersebut. Wireless headset memungkinkan seseorang menggunakan handphone-nya walaupun handphone itu tersimpan di dalam tas atau koper.

Internet bridge

Teknologi wireless Bluetooth juga memungkinkan handphone untuk digunakan sebagaimana modem dengan memanfaatkan salah satu profil dari Bluetooth yaiut Dial-Up Networking yang ada pada PC. Memungkinkan kita untuk terhubung ke Internet tanpa koneksi fisik ke line telepon.

File exchange

Bluetooth dapat membentuk suatu jarimgan komunikasi data tanpa membutuhkan infrastruktur tambahan lainnya. Dengan cara demikian untuk melakukan pertukaran data atau informasi antar computer atau telepon gengam cukup dengan mengaktifkan Bluetooth pada alatnya masing-masing.

Sinkronisasi

Bluetooth memungkinkan sinkronisasi antar-piranti. Sebagai contoh komputer desktop dapat secara wireless mensinkronisasi daftar alamat, informasi tugas, kalender., dan lainnya dengan handphone, PDA, ataupun laptop.

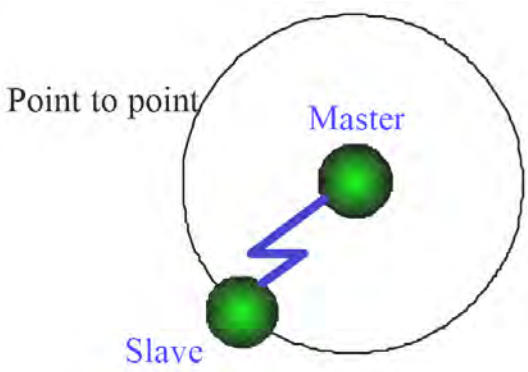
Printing

Printer dan notebook yang sudah dilengkapi dengan Bluetooth sudah tersedia. Komputer atau PDA yang Bluetoothenabled dapat langsung mengenali printer yang juga Bluetooth-enabled dan dapat mencetak tanpa harus repot-repot men-setup printer tersebut.

2.1.3 Cara Kerja Bluetooth

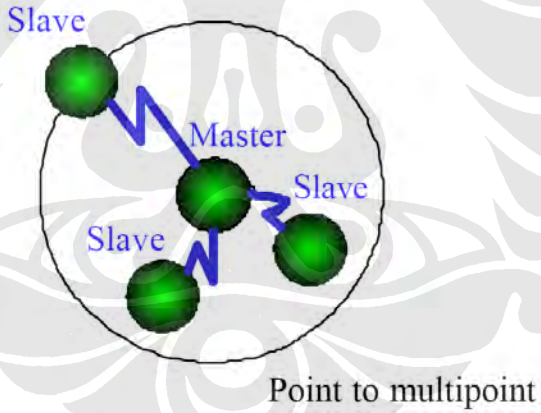
Suatu piranti Bluetooth dapat berperan sebagai *master* atau sebagai *slave*, tergantung pada skenario penerapannya. Bluetooth menerapkan FHSS (frequency hopping spread spectrum) untuk berkomunikasi. Jadi agar beberapa piranti Bluetooth dapat mengenali satu sama lain, semuanya harus bersinkronisasi pada *hopping sequence* yang sama. Master men-set dan mengaktifkan *hopping sequence* dan para slave mensinkronisasi dirinya dengan master. Master unit adalah unit yang memulai transmisi, dan slave unit adalah unit yang yang menjawab

Teknologi Bluetooth menyediakan baik suatu point-to-point koneksi dan suatu point-to-multipoint koneksi. Di point-to-multipoint koneksi, saluran yang ada dipakai bersama oleh beberapa unit Bluetooth. Di point-to-point koneksi, hanya dua unit berbagi koneksi itu, seperti yang tampak pada gambar 2.1.



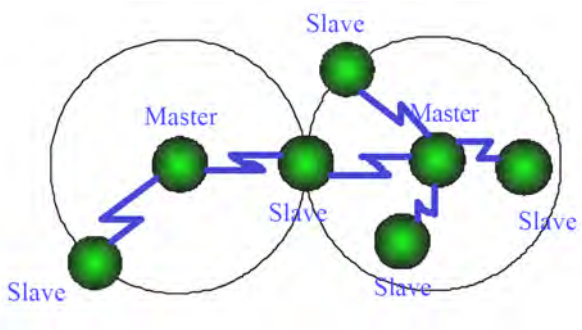
Gambar 2.1 Koneksi point to point

Suatu *master* dengan beberapa *slave* (satu sampai tujuh) membentuk yang disebut *piconet* seperti yang terliaht pada gambar2.2. Slave dalam suatu *piconet* hanya berkomunikasi dengan *master*-nya. Dua atau lebih *piconet* dapat dihubungkan membentuk suatu *scatter net* seperti yang terliaht pada gambar 2.3. Apabila suatu piranti berada dalam lebih dari satu *piconet*, maka piranti itu harus berpindah- pindah dalam sinkronisasinya dengan *master* dari *piconet* yang dikomunikasikannya.



Gambar 2.2 Piconet

Jika beberapa piconets melakukan overlap(tumpang tindih) dalam suatu area phisik, dan beberapa anggota piconets berkomunikasi dengan satu sama lain, unit yang baru ini, dengan jaringan lebih besar dikenal sebagai suatu scatternet. Unit manapun di dalam satu piconet dapat komunikasi segera dengan piconet kedua selama ia bertondak selaku master untuk hanya satu piconet serentak.



Gambar 2.3 Scatternet

Perbedaan yang utama antara Bluetooth dan arsitektur radio selular adalah Bluetooth itu memungkinkan networking khusus (ad hoc networking). Dibandingkan bergantung pada suatu broadband sistem, yang bersandar pada terminal dan base station untuk mempertahankan koneksinya kepada jaringan via radio link, Bluetooth mengembangkan peer-to-peer connectivas— tidak ada base station atau terminal dilibatkan.

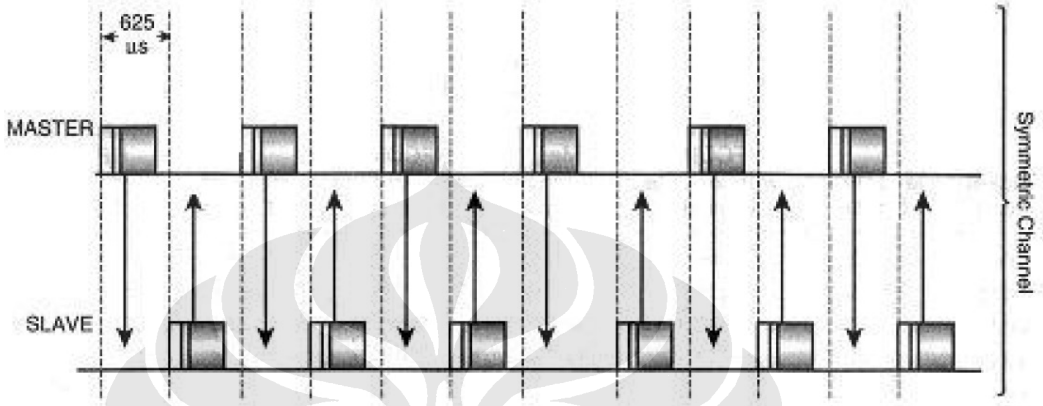
Dengan menggunakan peer-to-peer connectivas, teknologi Bluetooth menyederhanakan area pribadi koneksi tanpa kawat, memungkinkan semua alat digital untuk berkomunikasi secara spontan. Awal aplikasi diharapkan untuk meliputi penggantian kabel untuk laptop, PDA, telepon gengam, dan kamera digital. Karena Bluetooth mendukung transmisi suara, headsets juga akan menjadi tanpa kawat. Teknologi Bluetooth menawarkan keuntungan berikut :

- Voice/Data akses point akan dapat dipakai, sebagai contoh, telephone/Internet koneksi mobile.
- Kabel digantikan oleh suatu Bluetooth chip yang memancarkan informasi pada suatu frekwensi radio khusus kepada receveir Bluetooth chip.
- Networking khusus (ad hoc networking) memungkinkan alat pribadi ke secara otomatis menukar informasi dan mensinkronkan satu sama lain. Sebagai contoh, janji pertemuan dalam calendar PDA secara otomatis nampak pada calender desktop juga.

2.1.4 ACL dan SCO Link

Links dan channels digunakan untuk memancarkan data antar unit Bluetooth. Pertama, link dibentuk. Bluetooth teknologi mendukung dua jenis link: *synchronous conections-oriented* (SCO) dan *asynchronous conectionless* (ACL) link. Link SCO

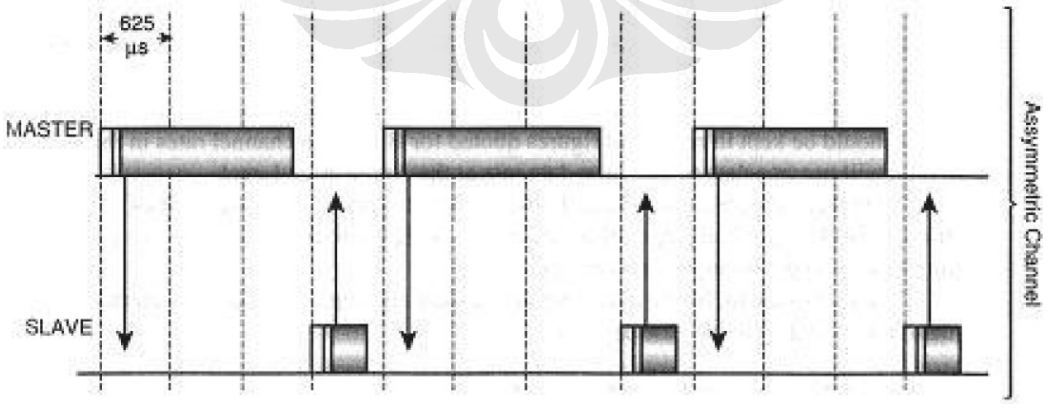
digunakan terutama untuk komunikasi suara seperti yang terlihat pada gambar 2.4. Link ACL digunakan untuk paket data seperti yang tampak pada gambar 2.5. Bluetooth dapat menggunakan link yang manapun dan dapat merubah link selama transmisi, walaupun suatu link ACL harus dibentuk sebelum suatu link SCO dapat digunakan.



Gambar 2.4 Link Simetris

Channel simestris digunakan untuk tipe paket yang memiliki ukuran yang sama pada kedua channel. SCO bkerja dengan bandwith 64 Kb/s dan mampu menangani koneksi 3 data suara sekaligus. Kualitas suara yang dihasilkan hamper sama dengan *handphone* GSM.

ACl digunakan pada komunikasi data. ACL menyediakan transmisi bebas error, yang berarti mengirimkan kembali data yang hilang atau error. Maksimun data yang bisa ditransmisi adalah sekitar 650 Kb/s.



Gambar 2.5 Link Asimetris

2.1.5 Profil Power Level dan Daya Jangkau

Tabel 2.1 Kelas dan Jangkaun

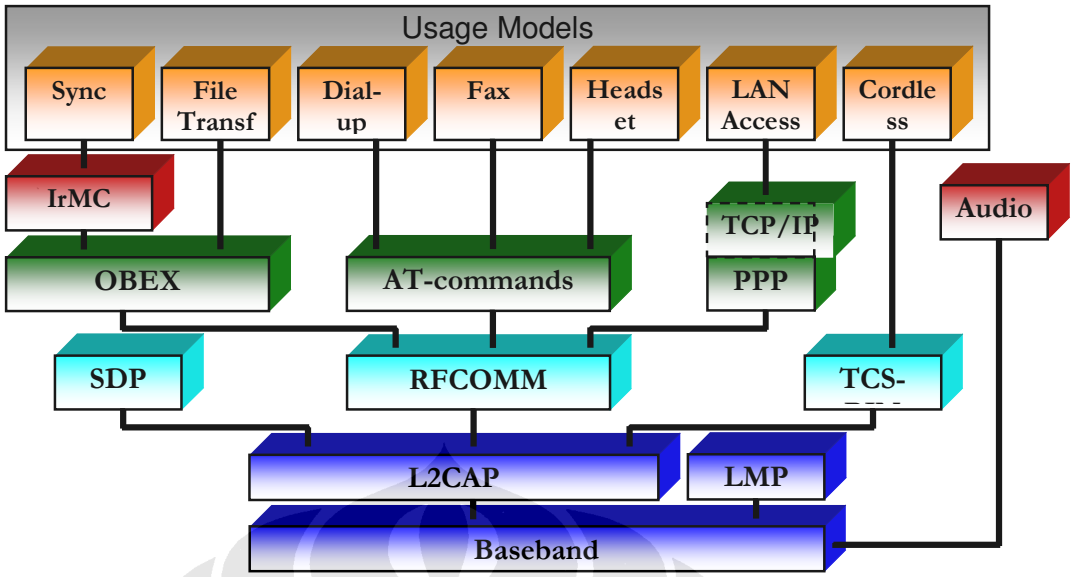
Kelas	Daya Output Maksimun/mW	Jangkauan/m
1	100	100+
2	2.5	10
3	1	1

Berdasarkan table 2.1, Bluetooth bergantung pada baterai untuk sumber dayanya, dan ditetapkan sebagai piranti kelas 3 yang dirancang dan beroperasi pada power level 0 dBm (1mW), yang mempunyai daya jangkau sampai 1 meter.

Piranti kelas 2 dapat memanfaatkan output power 4 dBm (2,5 mW), dan piranti kelas 1 dapat memanfaatkan output power sampai 20 dBm (100 mW). Piranti kelas 1 mempunyai jangkauan sampai 100 meter.

Piranti Bluetooth kelas 2 dan 3 secara opsional dapat mengimplementasikan adaptive power control. Mekanisme ini memungkinkan suatu Bluetooth radio menurunkan dayanya sampai ke suatu level minimum sekedar untuk mempertahankan link, yang berarti menghemat daya dan mengurangi potensi interferensi dengan network lain yang berdekatan.

2.1.6 Bluetooth stack



Gambar 2.6 Protokol Stack Bluetooth

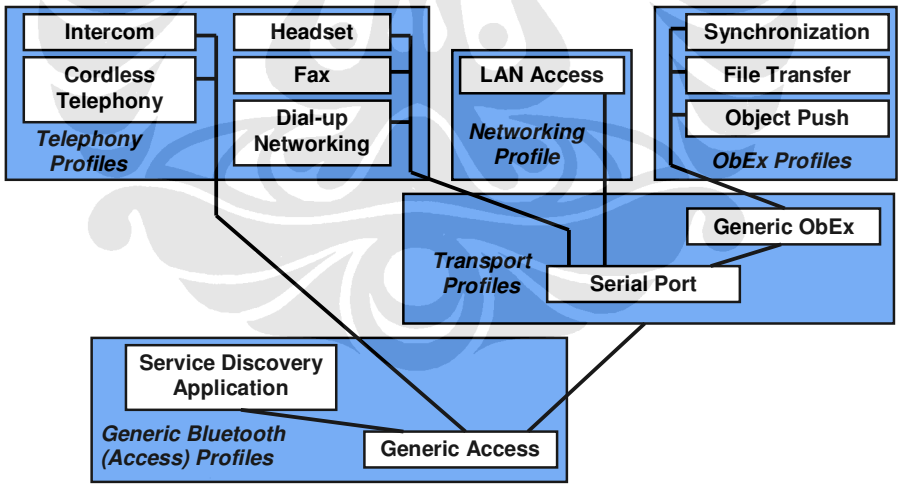
Bluetoooh memiliki beberapa stack yang mempunyai kegunaan tertentu. Pada gambar2.6 diilustrasikan gambar stack pada Bluetooth dan pada tabel 2.2 dideskripsikan layanan pada masing-masing layer pada bluetooth

Tabel 2.2 Deskripsi komponen *protocol stack* Bluetooth

Layer	Deskripsi
Applications	*Profile* bluetooth yang memberikan petunjuk bagi *developer* dalam menentukan bagaimana sebuah aplikasi harus menggunakan protokol *stack*
Telephony Control System (TCS)	Menyediakan *Telephony service*
Service Discovery Protocol (SDP)	Menyediakan fungsi aplikasi Bluetooth untuk menemukan servis dan karakteristik dari servis yang tersedia khusus ke Bluetooth.
WAP and OBEX	Menyediakan *interface* pada bagian layer yang lebih tinggi pada protokol komunikasi yang lain
RFCOMM	Menyediakan keterhubungan melalui bluetooth seperti yang digunakan pada standard serial (COM) port.
L2CAP	Merupakan layer yang menangani *packet segmentation and reassembly* (SAR). *Multiplexes* data dari layer yang lebih tinggi dan merubah merubah antar ukuran packet yang berbeda.
Host Control Interface (HCI)	Menangani komunikasi antara *host* dan bluetooth *module*

Link Manager Protocol	Mengontrol dan konfigurasi *link* kepada *device* lain
Baseband and Link Controller	Mengontrol akses dan mengirim paket data ke media fisik yang melewati bluetooth radio. Layanan-layanan yang disediakan pada layer ini antara lain: prosedur inquiry dalam menemukan *device*, fungsionalitas *device* dalam sinkronisasi *clock* dan membangun *link* pada frekuensi *hopping* yang tepat, *error correction*, *data whitening*, *flow control*, fungsi untuk men-*generate* kunci enkripsi
Radio	*Modulate* dan *demodulate* data dari transmisi dan menerima di udara. Bluetooth Radio merupakan *layer* fisik yang menggunakan *frequency hopping spead spectrum* untuk memperkecil interferensi dari device lain yang menggunakan band ini. Bluetooth membagi frekuensi band 2.4 GHz menjadi 79 *channel* pada 1 MHz (dari 2.402 sampai 2.480 GHz) dan frekuensi hop 1600 kali per detik.

2.1.7 Profil Dasar Bluetooth



Gambar 2.7 Bluetooth Profile

Bluetooth mempunyai berbagai profil dasar yang saling berkaitan seperti yang diilustrasikan pada gambar 2.7, beberapa profil dijelaskan sebagai berikut:

General Access Profile (GAP)

Profil ini diperlukan oleh semua model penggunaan dan mendefinisikan cara men-*discover* adanya piranti Bluetooth dan menghubungkan satu dengan lainnya, selain juga mendefinisikan *security protocol.*

Service Discovery Application Profile (SDAP)

SDAP menggunakan bagian dari GAP untuk mendefinisikan *discovery* dari service-service untuk piranti-piranti Bluetooth.

Serial Port Profile

Profil ini mendefinisikan cara setup dan koneksi port-port serial maya antara dua piranti. Emulasi kabel serial ini kemudian dapat digunakan untuk tugas-tugas seperti transfer data dan pencetakan.

Generic Object Exchange Profile (GOEP)

GOEP bersifat dependen terhadap Serial Port Profile dan digunakan oleh aplikasi-aplikasi untuk menangani pertukaran obyek. Pada gilirannya, kemampuan ini dimanfaatkan oleh profil-profil lain untuk menjalankan fungsi-fungsi seperti Object Push, File Transfer, dan Synchrorization.

Object Push

Profil ini digunakan untuk pertukaran obyek-obyek kecil seperti misalnya *electronic calling card.*

File Transfer

Profil ini digunakan untuk mentransfer data-data antara dua piranti Bluetooth.

Synchorization

Profil ini digunakan untuk mensinkronisasi kalender dan informasi alamat diantara piranti-piranti.

2.1.8 Bluetooth Security

Bluetooth Security ditangani oleh General Access Profile. Ada tiga security mode:

Mode 1

Non-secure. Otentikasi bersifat opsional

Mode 2

Service-level enforced security. Service yang diberikan oleh aplikasi ini memutuskan apakah memang otentikasi atau enkripsi dibutuhkan.

Mode 3

Link-level enforced security. Kedua piranti harus mengimplementasi security procedure agar dapat membentuk koneksi. Sebagai tambahan dari mode-mode di atas, suatu piranti dapat dikonfigurasi untuk tidak merespons, sehingga piranti lain tidak dapat terkoneksi padanya. Atau dapat dikonfigurasi sedemikian agar hanya piranti-piranti yang mengetahui address-nya yang dapat terhubung padanya.

2.2 JAVA

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi, yang dapat digunakan untuk membuat aplikasi desktop, web, mobile dan lainnya. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai system operasi dan bersifat *open source.*

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut :

1. Sederhana

Java memiliki kemiripan sintaks dengan C++ sehingga memudah sebagian programmer untuk memahaminya. Namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection* untuk menghindari kebocoran memori.

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular sehingga dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Dapat didistribusi dengan mudah

Java adalah bahasa pemograman yang *portable* dan arsitektur pemograman yang netral menyediakan pemograman yang atraktif dan sederhana untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries* networking yang terintegrasi pada Java.

4. Interpreter

Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* dapat dibuat dengan lebih cepat tanpa harus menunggu *compiler* atau siklus pemograman lainnya.

5. Robust

Java mempunyai reliabilitas yang tinggi menghasilkan solusi pemograman dengan kualitas tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

6. Aman

Salah satu fitur keamanan Java adalah *memory allocation*, dimana penggunaan memory ditentukan pada saat *runtime* dan bukan oleh Java programmer ataupun compiler. Maksudnya adalah penentuan memory yang akan dipakai lebih bergantung pada saat program itu dijlankan di piranti keras mana dan *Operating System* apa.

7. Architecture Neutral

Solusi baru ditawarkan oleh JAVA adalah "*binary code format"* yang infepent terhadap piranti keras ataupun *operating system.* Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan *Java Virtual Machine*.

8. Portabel

Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

9. Performance

Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers* (JIT).

10. Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa menggangu program yang menggunakan *class* tersebut.

2.3 *BLUETOOTH FOR JAVA*

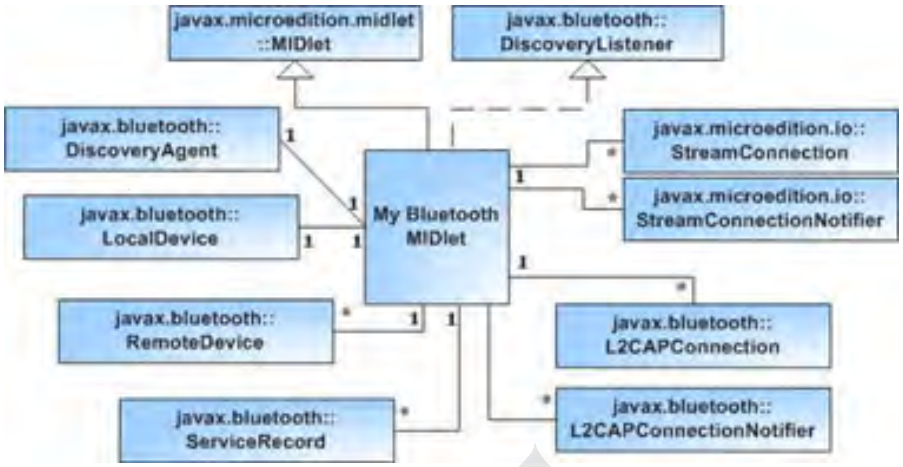
JCP (Java Community Process) merupalkan sebuah prosedur formal untuk mendapatkan sebuah gagasan dari menggabungkan beberapa konsep sederhana emnjadi sebuah standar bahsa pemograman Java. Proses ini didukung oleh para developer dan ahli-ahli dari dunia industri untuk membuat berbagai standar bahasa pemograman Java. API (Application Programable Interface) yang terkenal seperti

Java USB, Java real Time, Java Printing, Java New I/O, J2ME MIDP1.0, J2ME MIDP 2.0, JDBC 3.0, EJB 2.0, dan bahkan JDK 1.4 semuanya melalui JCP. Jika ingin menambahkan beberapa fungsi baru dalam bahasa pemograman Java, atau anda menyarankan untuk membuat API baru atau menguslkan beberapa kelas baru dalam paket java.* atau javax.* maka anda bisa melalui JCP.

JSR adalah singakatan dari Java Special Request dalam JCP. JSR-82 adalah sebutan formal yang diberikan untuk Java API Bluetooth. Ketika JSR disetujui, sebuah kelompok ahli dibentuk. Yang bertanggung jawab untuk kelompok ahli tadi adalah Motorola bersama kelompok ahli lainnya, mereka menemukan secara resmi Java Bluetooth API. Perusahaan lain yang tergabung dalam JSR-82 adalah :

Menurut JCP, kelompok ahli tadi bertanggung jawab untuk membentuk RI (Reference Implementation) dan juga TCK (Technology Compatibility Kit). RI pada dasarnya merupakan jaminan dari konsep untuk memberikan bukti dari spesifikasi yang ditenukan bisa diimplementasikan. Perusahaan lain bisa secara bebas mengimplementsaikan JSR-82, dan untuk menjamin bahwa produk perusahaan mereka sesuai pada standar JSR-82, produk perusahaan mereka harus lulus uji TCK.

Spesifikasi dari JSR-82 pada kenyataanya memiliki 2 RI dan TCK. Ini dikarenakan SIG Bluetooth mengadopsi beberapa protocol yang sudah ada dalam spesifikasi Bluetooth, yang dinamakan OBEX. Protokol OBEX umumnya digunakan oleh teknologi inframerah untuk melakukan transmisi objek, jauh sebelun teknologi Bluetooth dikenalkan. Para desainer dari Java Bluetooth memutuskan untuk tidak memasukannya ke dalam Bluetooth . JSR-82 mengandung 2 paket independent javax.bluetooth (13 kelas dan interface untuk komunikasi dengan protocol Bluetooth) javax.obex (8 kelas untuk mengirimkan objek antar piranti seperti yang digambarkan pada gambar 2.8.



Gambar 2.8 Kelas dan Interface dalam JABWT

Pada tabel 2.3 dan tabel 2.4 dijelaskan dan dideskripsikan kelas dan interface dalam paket javax Bluetooth dan javax Obex.

Tabel 2.3 Kelas dalam javax.bluetooth

Kelas	**Deskripsi**
DiscoveryListener	Interface ini menyediakan sebuah aplikasi agar dapat menerima device discovery dan service discovery events
L2CAPConnection	Interface ini mewakili koneksi oriented melalui L2CAP
L2CAPConnectionNotifier	Interface ini menyediakan pemberitahuan untuk koneksi L2CAP
ServiceRecord	Interface ini menjelaskan karakteristik dari servis di bluetooth
DataElement	Kelas ini mendefinisikan berbagai macam tipe atribut dari servis Bluetooth
DeviceClass	Kelas ini mewakili record dari Class of Device (CoD) yang didefiniskan di bluetooth
DiscoveryAgent	Kelas ini menyediakan methode untk DeviceDiscovery dan ServiceDiscovery.
LocalDevice	Kelas ini mereprenstasikan piranti local dari Bluetooth.
RemoteDevice	Kelas ini mereprensentasikan piranti remote dari Bluetooth.
UUID	Kelas ini mendefiniskan bilangan universal yang unik.
BluetoothConnectionException	Exception ini di lempar ketika terjadi koneksi

	Bluetooth (L2CAP,RFCOMM,OBEX) ketika bermasalah
BluetoothStateException	Exception ini dilempar ketika permintaan yang dibuat kepada suatu system Bluetooth tidak didukung.
ServiceRegistrationException	Exception ini dilempar ketika terjadi kegagalan untuk menambahkan service record pada local SDDB (Service Discovery Database)

Tabel 2.4 Kelas dalam javax.obex

Kelas	Deskripsi
Authenticator	Interface ini menyediakan cara untuk memberikan respon untuk authentifikasi
ClientSession	Interface ini menyediakan metode untuk request OBEX
HeaderSet	Interface ini menjelaskan methode untuk mendapatkan dan memberi nilai untuk OBEX header.
Operation	Interface ini menyediakan cara untuk memanipulasi operasi OBEX GET dan PUT
SessionNotifier	Interface ini menjelaskan pemberitahuan mengenai koneksi pada sisi server dalam koneksi OBEX.
PasswordAuthentication	Kelas ini bertanggung jawab untuk kombinasi nama user dan sandi.
ReponseCodes	Kelas ini mengandung beberapa kode respon yang benar, yang dikirmkan server ke client.
ServerRequestHandler	Kelas yang menjelaskan event listener yang akan memberikan respon pada OBEX request untuk server

Ada dua kelebihan menggunakan Java Bluetooth API dibanding API yang berbasiskan C :

Independent atau tidak tergantung pada stack dan radio

Java Bluetooth API atau JSR-82 tidak tergantung pada stack dan piranti keras dari Bluetooth. Ini memudahkan anda untuk dapat membuat sebuah aplikasi tanpa perlu memiliki pengetahuan tentang piranti keras bluetooth atau stack. Dan dengan Java pula kita bisa menjalankan pada hardware dan OS dengan sedikit modifikasi.

API Bluetooth yang terstandarisasi

Jika anda memilih Bluetooth SDK bebasiskan bahasa C, sama saja anda mengharapkan kemurahan hati dari vendor, karena tidak ada standar untuk Bluetooth SDK bebasiskan C, karena dengan begitu masing-masing vendor akan memberi nama secara bebas pada fungsi dan metode yang mereka pilih. Sebagai ilustrasi, vendor A memiliki 5 profil dalam SDK dan Vendor B memiliki 3 profil. Jika ingin mengganti piranti Bluetooth dan stack, maka anda harus menulis aplikasi pemograman anda dan mengganti fungsinya. Karena JSR-82 merupakan Java API yang telah ditetapkan, semua vendor harus menyertakan core layer dan profil mereka dalam Bluetooth SDK

JSR-82 Bluetooth stack harus mengikutsertakan beberapa layer:

- Host Controller Interface (HCI)
- Logical Link Control and Adaptation Protocol (L2CAP)
- Service Discovery Protocol (SDP)
- RFCOMM
- These profiles are also required:
- Generic Access Profile
- Service Discovery Application Profile
- Serial Port Profile
- Generic Object Exchange Profile

2.4 VIDEO STREAMING

2.4.1 Video digital

Video digital terdiri *array* tiga-dimensi dari piksel berwarna. Dua dimensi sebagai arah horizontal dan vertikal dari gambar yang bergerak, dan satu dimensi mewakilkan domain waktu. Sebuah rangkaian video digital berisi sejumlah *frame* atau gambar yang dimainkan pada kecepatan yang tetap. Semakin tinggi kecepatan *frame*, semakin tinggi pula *bandwidth* yang dibutuhkan untuk transmisi sinyal video. Karenanya untuk memungkinkan penggunaan *bandwidth* yang kecil dan untuk menghasilkan gerakan yang halus, paling sedikit digunakan kecepatan 25 *frame* per detik.

2.4.2 Konsep streaming

Pada transmisi *stored video* umumnya dikenal dalam dua metode, yaitu metode *download* dan metode *streaming.* Pada metode *download*, seluruh file video harus di-*download* terlebih dahulu, setelah itu video dapat dimainkan kembali. Proses

transfer file tersebut akan berlangsung cukup lama sehingga menyebabkan *transfer time* melonjak naik. Sedangkan, pada metode *streaming*, file video akan dimainkan setelah sebagian kandungan file diterima dan dikodekan, tanpa harus men-*download*-nya secara keseluruhan dahulu.

Secara bahasa, *stream* berasal dari bahasa Inggris yang berarti pengaliran atau mengalirkan. Teknologi *streaming* adalah teknik pengiriman data kontinu secara terus-menerus yang dilakukan melalui internet. Sebuah teknlogi yang memungkinkan distribusi data audio, video dan multimedia secara *real-time* melalui Internet . Paket-paket data dikompresi terlebih dahulu untuk memudahkan pengirimannya melalui internet. Sedangkan pada sisi pengguna, *streaming* memungkinkan suatu *file* dapat segera dijalankan secara langsung tanpa harus menunggu selesai di-*download* seluruhnya atau tanpa perlu men-*download*-nya ke dalam *harddisk.*

Konsep *streaming* memiliki perbedaan mendasar dengan konsep *download and play*. Pada *download and play* suatu *file* di-*download* terlebih dahulu seluruhnya untuk kemudian dimainkan. Metode ini memang memberikan akses yang cepat untuk melihat bagian *file* yang berbeda. Tetapi diperlukan waktu untuk men-*download* keseluruhan *file* sebelum dapat dilihat isinya. Apabila ukuran *file*-nya kecil, maka hal ini tidak menjadi masalah. Tetapi jika ukuran *file*-nya besar maka dibutuhkan waktu yang cukup lama untuk dapat melihatnya. Proses untuk menjalankan *file* dengan metode ini sepenuhnya berlangsung di sisi *client*. Konsep *download and play* tidak mampu menyajikan *content* secara *real time.*

Streaming dapat didefinisikan sebagai metode pengiriman data secara kontinu dan stabil, dengan disertai proses kompresi atau penyusutan ukuran file audio dan video agar lebih efisien. Pen*transfer*an file audio dan video tersebut dilakukan secara terus-menerus. Metode ini dilakukan dari *server* ke *client* setelah mengalami paketisasi, yakni data pada file *streaming* dibagi-bagi ke dalam beberapa paket kecil. Paket-paket tersebut dikirim ke sebuah aliran secara terus menerus dan berurutan ke pengguna akhir serta memungkinkan penerima melakukan *decode* dan *playback* video tanpa harus menunggu seluruh video terkirim.

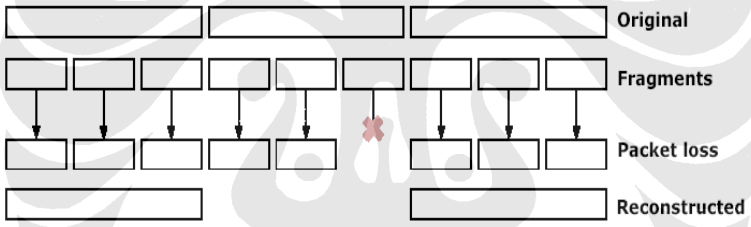
2.4.3 *Encode / Decode*

Codec adalah suatu metode atau algoritma yang digunakan untuk melakukan proses kompresi dan dekompresi *file* media. Proses kompresi pada *file* media bergantung pada standar *codec* yang digunakan. *Codec* sangat menentukan kualitas

dan tingkat kerapatan kompresi. Prinsip kerja *codec* adalah menghilangkan bagian-bagian video dan audio yang tidak signifikan dengan tujuan untuk mengurangi ukuran data.

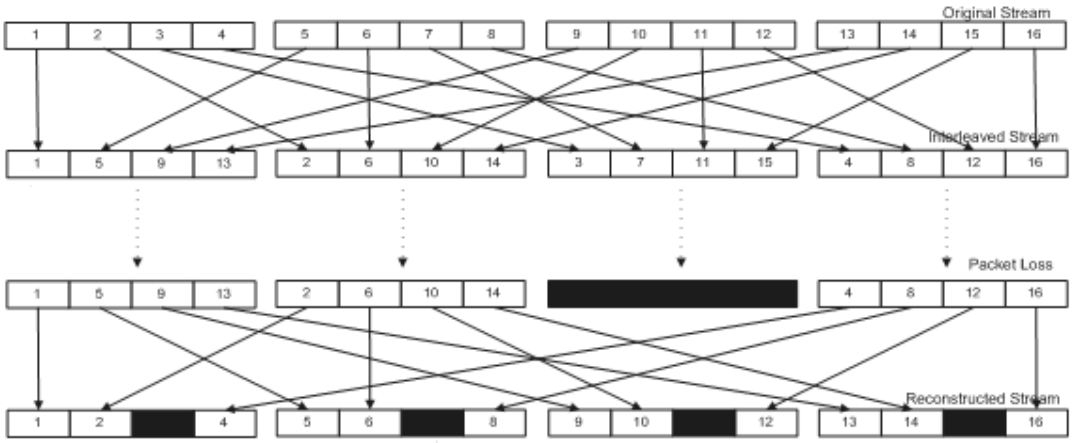
Fungsi *codec* adalah untuk mengkompresi *file* media agar ukurannya dapat diperkecil sehingga dapat mengoptimasi penggunaan *bandwidth* jaringan. *File* tersebut kemudian didekompresi ke ukuran semula untuk dapat dijalankan. Ilmu *codec* adalah sebuah seni digital. Banyak hal yang harus dipertimbangkan jika kita ingin melakukan proses kompresi-dekompresi *file* media. Pertimbangan tersebut adalah untuk mengimbangi antara kompresi yang dilakukan dengan kualitas video yang dihasilkan agar tidak mengalami degradasi.

Untuk melakukan pengiriman secara *streaming,* data video terlebih dahulu dilakukan kompresi dan fragmentasi, yakni membagi paket-paket besar menjadi beberapa *slice* yang lebih kecil untuk kemudian dikirimkan melalui jaringan. Hal itu dimaksudkan agar paket dapat di*transmisi*kan secara efisien melalui jaringan yang memiliki keterbatasan *bandwidth*.



Gambar 2.9 Proses Fragmentasi

Pada proses fragmentasi yang ditunjukan pada gambar 2.9, jika salah satu *slice* dalam satu paket mengalami kerusakan, maka paket tersebut akan dibuang sehingga akan dianggap sebagai *packet loss*. Untuk mengatasi masalah tersebut, maka dilakukan modifikasi fragmentasi guna memaksimalkan penanganan terhadap *packet loss. Original stream* dibagi menjadi beberapa *chunk* (potongan) yang lebih kecil, kemudian *chunk* tersebut di-*interleaved* (diselipkan) pada paket-paket yang terpisah, sehingga jika terjadi *packet loss*, masih terdapat *chunk* lain yang dapat digunakan untuk rekonstruksi data. Hal tersebut diilustrasikan seperti pada gambar 2.10 berikut.



Gambar 2.10 Fragmentasi modifikasi

2.4.4 Jenis-jenis video streaming

Penerapan *video streaming* dapat dilakukan dalam dua bentuk, yaitu *video streaming on-demand* dan *live video streaming*.

2.4.4.1 Video streaming on-demand

Pada video streaming o*n-demand* pengguna dapat melihat *file* video kapanpun, juga dapat mempercepat, menghentikan, dan meminta *server* untuk mengirim bagian yang dikehendaki. Video jenis ini merupakan klip yang direkam terlebih dahulu dulu (*pre-recorded*) sehingga tersedia dalam bentuk *file video streaming.* Pengguna dapat melihat isi *file* tersebut bersamaan dengan proses *download* file. Untuk memainkannya pengguna tidak perlu menunggu keseluruhan *file* selesai di-*download.*

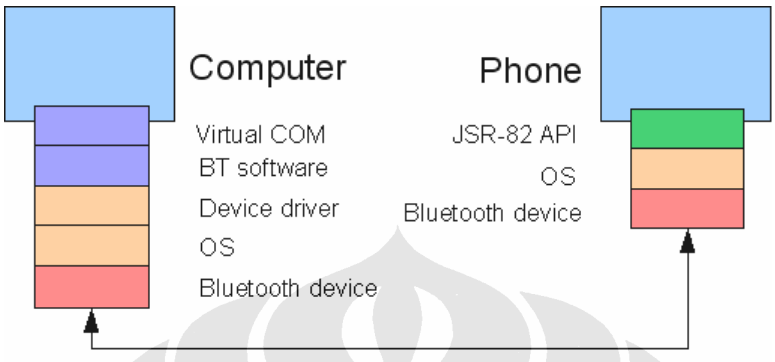
Pada proses *streaming*, jika data diterima dalam kecepatan yang lebih cepat dari seharusnya, maka diperlukan suatu *buffer* sebagai tempat penyimpanan sementara data yang berlebih. Sedangkan jika data tidak ditransmisikan dengan kecepatan yang cukup, maka tampilan data tersebut tidak akan berjalan lancar seperti yang diinginkan. Pada saat menjalankan *file* media *streaming* akan terjadi proses *buffering* yang memberikan jeda waktu selama beberapa detik untuk men-*download* sebagian dari awal klip video. Kemudian ketika bagian itu diputar, *media player* akan men-*download* lanjutannya sehingga video dapat dijalankan secara kontinu.

2.4.4.2 *Live video streaming*

Live video streaming adalah proses *streaming video* yang *content*-nya langsung di-*capture* dari suatu kamera dan disajikan secara *real time* pada *media streaming player*. Tampilan video yang diterima secara *live streaming* merupakan hasil *capture* pada waktu yang bersamaan. Proses kompresi dan dekompresi juga dilakukan secara langsung. *Live video streaming* terbagi menjadi dua, yaitu satu arah dan dua arah. Penerapan *live video streaming* satu arah bersifat pasif, pengguna hanya dapat menyaksikan saja tanpa ada interaksi. Sedangkan pada *live video streaming* dua arah memungkinkan terjadinya komunikasi audio dan video dua arah secara *real time*.

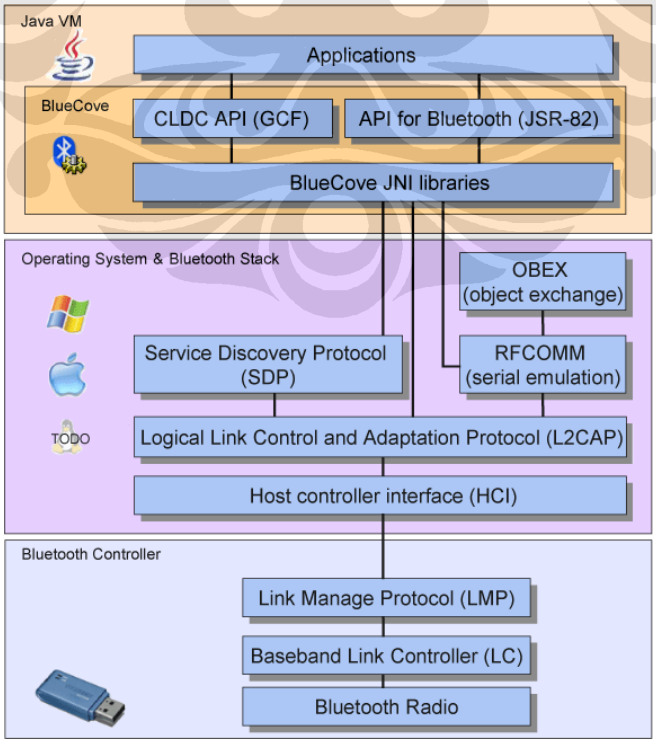
BAB III
ANALISA DAN PERANCANGAN

3.1 ANALISA SISTEM SERVER



Gambar 3.1 Arsitektur Sistem Client-Server

Arsitektur sistem *video streaming over Bluetooth* terdiri dari dua komponen utama, yaitu *client* dan komputer *server*. Komunikasi antara keduanya diselenggarakan menggunakan koneksi serial Bluetooth. Gambar 3.1 menampilkan skema arsitektur sistem yang dimaksud. Pada gambar 3.2 diperlihatkan arsitektur umum dari server yang terdiri dari Bluetooth dongle, komputer dengan system opersai yang mendukung Bluetooth stack dan Java API.



Gambar 3.2 Arsitektur Server

3.1.1 Piranti Bluetooth

Gambar 3.3 Bluetooth dongle

1 buah Bluetooth dongle atau perangkat Bluetooth yang ditanamkan pada komputer server seperti pada gambar 3.3.

3.1.2 Bluetooth Host

Gambar 3.4 PC dan SmartPhone sebagai host

Adalah sebuah computer yang secara fisik terhubung langsung dengan Bluetooth dongle. Anda bisa menggunakan PC atau laptop seperti pada gambar 3.4, PDA atau smart phone, PC/ laptop bertindak sebagai server dan media penyimpanan video yang akan dimainkan. Sedangkan PDA/smartphone bertindak client, PDA/smartphone yang memiliki fitur bluetooth dan system operasi Symbian serta mendukung untuk menjalankan video. Untuk spesifikasi *handphone*, minimal *support* MIDP 2.0 dan CLDC 1.0. Koneksinya pada PC/laptop umumnya menggunakan port USB atau RS-232. Pada PC/laptop ini sebaiknya didukung dengan memori RAM sebesar 512k supaya bisa menjalankan JVM (Java Virtual Machine).

3.1.3 Bluetooth Stack

Gambar 3.5 Logo Bluetooth Stack

Bluetooth stack diperlukan agar Bluetooth Host (PC) bisa berkomunikasi dengan piranti Bluetooth. Pada diagram Bluetooth stack pada layer bawah terdapat HCI (Host Controler Interface), HCI ini adalah piranti lunak yang dibutuhkan sebgai interface antara host dan piranti Bluetooth. Bluetooth stack juga bisa diartikan sebuah driver yang disediakan oleh vendor piranti Bluetooth. Seperti Broadcomm, Microsoft (Winsock) dan Bluesoeil yang merek dagangnya dapat dilihat pada gambar3.5.

3.1.4 Bluetooth API

Terakhir, yang kita perlukan adalah sebuah libaries sebagai interface dengan Bluetooth stack. Seperti yang kita ketahui bahwa JSR-82 hanya bisa dijalankan pada fitur J2ME. JSR-82 tidak bisa dikembangkan pada J2SE, pada J2SE sedang dikembangkan JSR-197 Generic Connection Framework, jadi kita memakai standar development kit dari vendor tertentu seperti yang dipaparkan pada tabel 3.1.

Tabel 3.1 Java API SDK

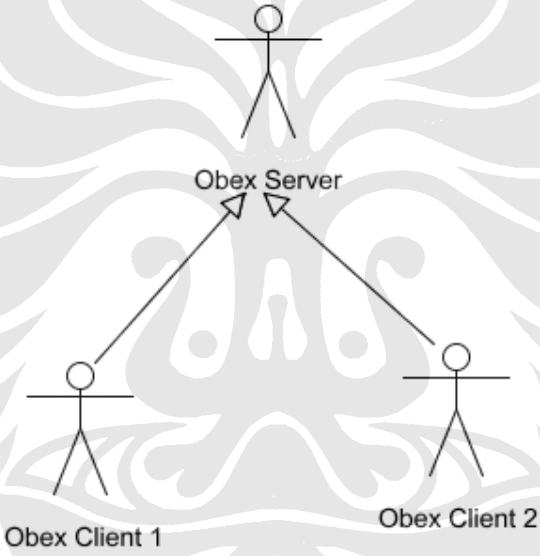
Nama Pengembang	JAVAX.BLUETOOTH	JAVAX. OBEX	Platform	OS
Atinav	Ya	Ya	J2ME, J2SE	Win-32, Linux, Pocket PC
Avetana	Ya	Ya	J2SE	Win-32, Mac OS X, Linux, Pocket PC
Blue Cove	Ya	Ya	J2SE	Win-32, Pocket PC
Electric Blue	Ya	Ya	J2SE	WinXP SP2
Harald	Tidak	Tidak	Yang mendukung **javax.comm**	banyak
JavaBluetooth.org	Ya	Tidak	Yang mendukung **javax.comm**	banyak
Rococo	Ya	Ya	J2ME, J2SE	Linux, Palm OS

Yang kita gunakan dalam system ini adalah Bluecove. Kita *download* BlueCove dari situs blucove.org. Kemudian kita buka paketnya dan kita tempatkan *file "intelbth.dl"* pada `C:\WINDOWS\SYSTEM32'. Kemudian tambahakan bluecove.jar pada *classpath.*

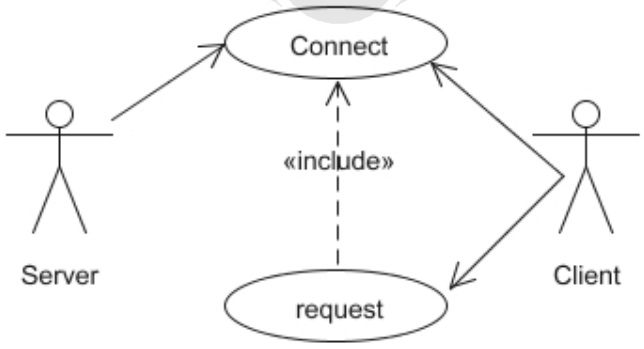
3.2 PERANCANGAN SISTEM

Pada tahap perancangan akan digunakan notasi UML dengan *metode object oriented.* UML adalah bahasa atau alat bantu untuk memvisualisasikan, menspesifikasikan, mengkonstruksikan, dan mendokumentasikan artifak-artifak sistem perangkat lunak [7].

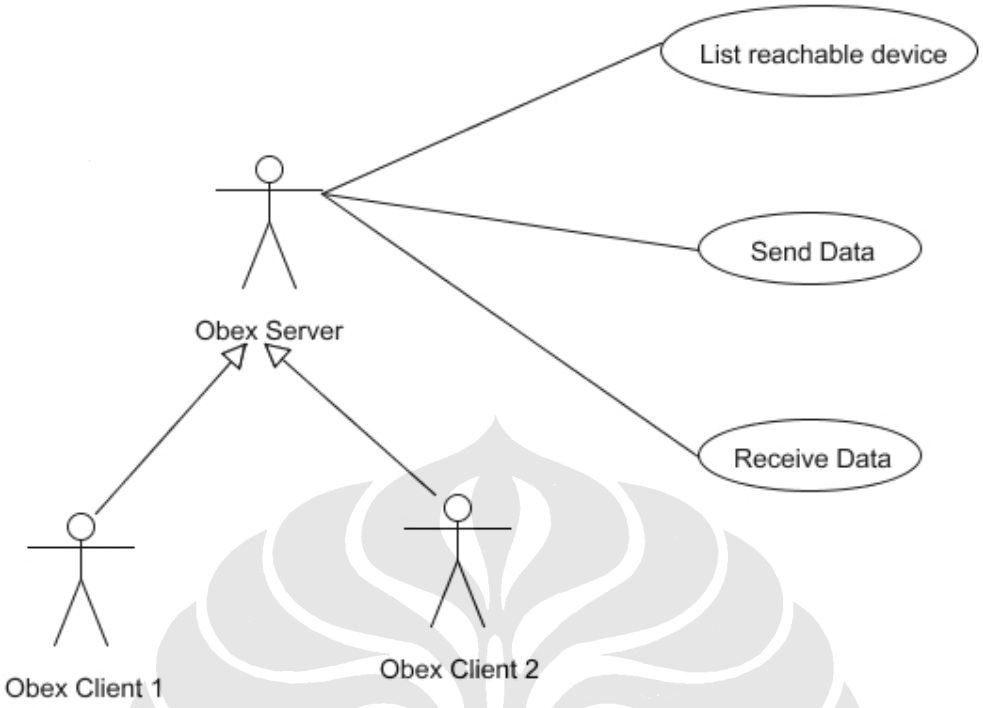
Use Case Diagram



Gambar 3.6 Diagram Use Case User Obex Server



Gambar 3.7 Diagram Use Case User RFCOMM



Gambar 3.8 Diagram Use Case Kirim File

3.2.1.1 Skenario Use Case OBEX SERVER dan RFCOMM

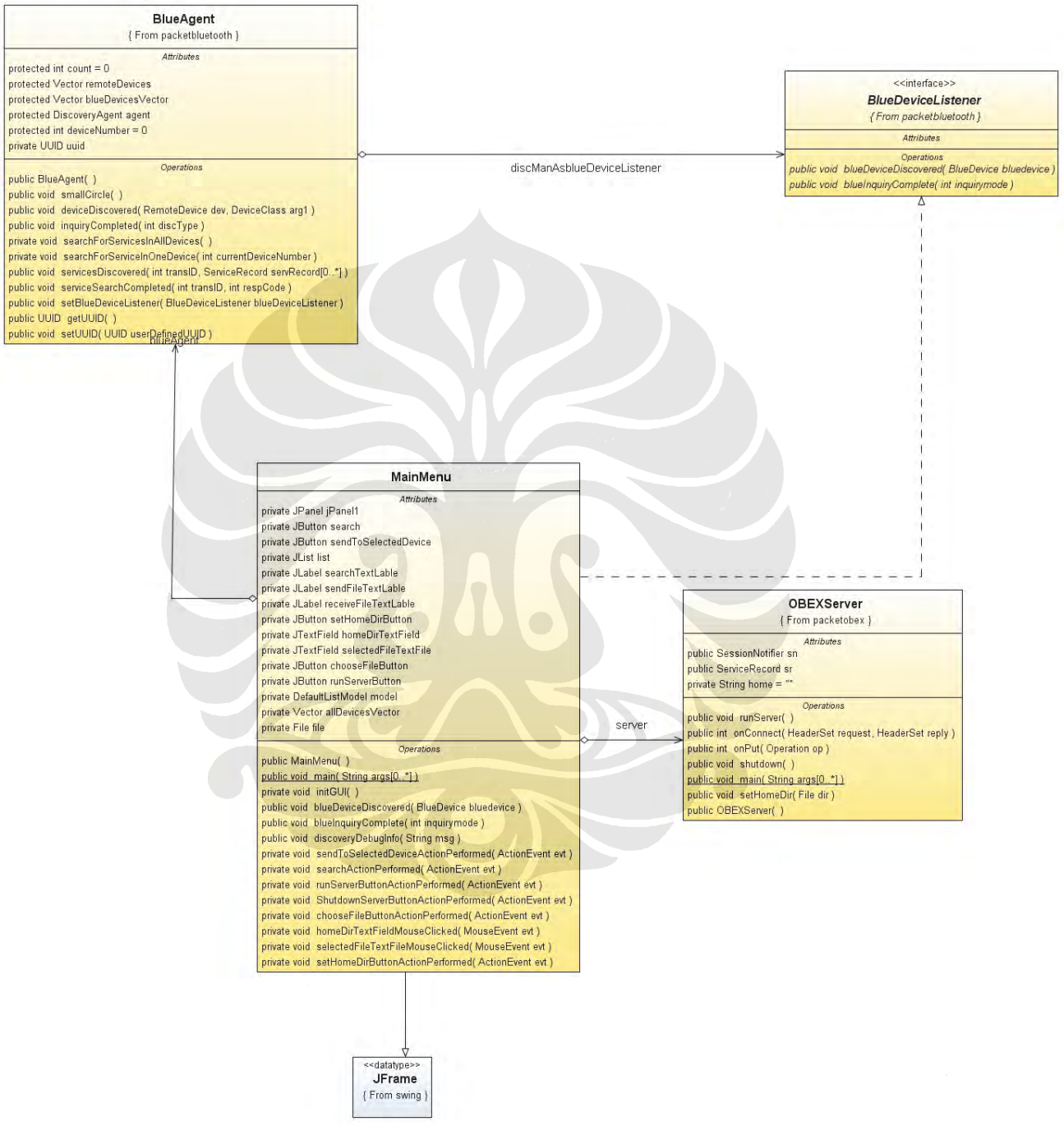
UseCase-00: Kirim File OBEX SERVER	
Penjelasan	Obex server mengirim file kepada device yang dipilih oleh user setelah divais
Prioriitas	Penting
Frekuensi Penggunaan	Sering
Actor	Obex server berupa sebuah PC
Skenario	1. Jalankan aplikasi OBEX Server 2. Cari divais Bluetooth 3. List dari divais Bluetooth muncul 4. Pilih divais 5. Pilih file yang mau dikirim 6. Kirim file Muncul pesan “file telah sukses terkirim”

UseCase-00: Connect RFCOMM Server	
Penjelasan	Setiap aktor mengaktifkan koneksi pada bluetoothnya, server pada PC & client pada HP.
Prioriitas	Penting
Frekuensi Penggunaan	Sering
Actor	Server & Client
Skenario	1. Menyediakan pilihan menu 2. Mengaktifkan koneksi 3. Informasi tentang video yang bisa di tampilkan 4. Video diload 5. Koneksi aktif pada Client 6. Informasi tentang video yang bisa ditampilkan 7. Client memilih video

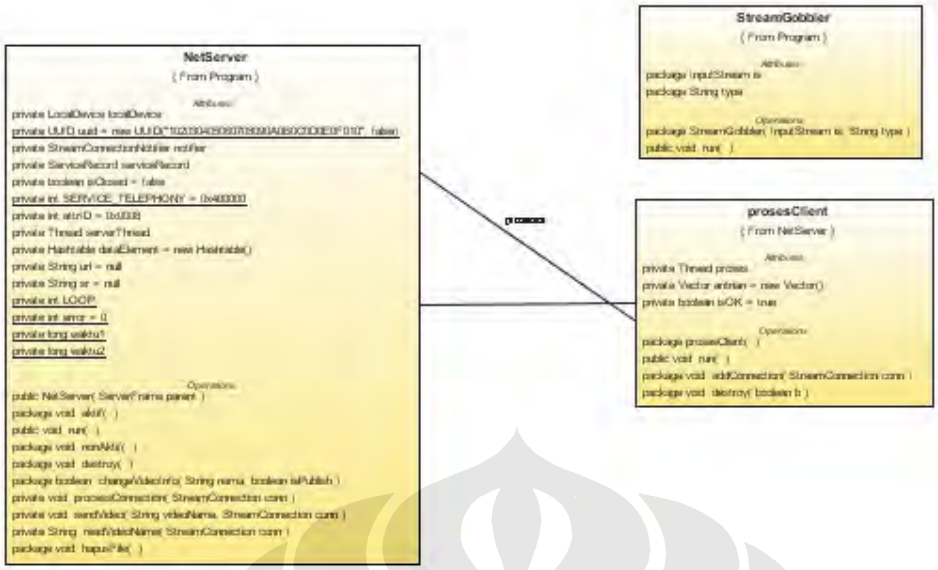
UseCase-00: Kirim video RFCOMM Server	
Penjelasan	Aktor mendapatkan info tentang video yang dapat diputar lalu memilih video pada HP
Prioriitas	Penting
Frekuensi Penggunaan	Sering
Actor	Client
Skenario	1. List video 2. Memilih video 3. Koneksi dengan serverVideo diload 4. Info bahwa video siap diputar 5. Play video 6. Server mengirim video. 7. Stop atau Pause

3.2.2 Class Diagram

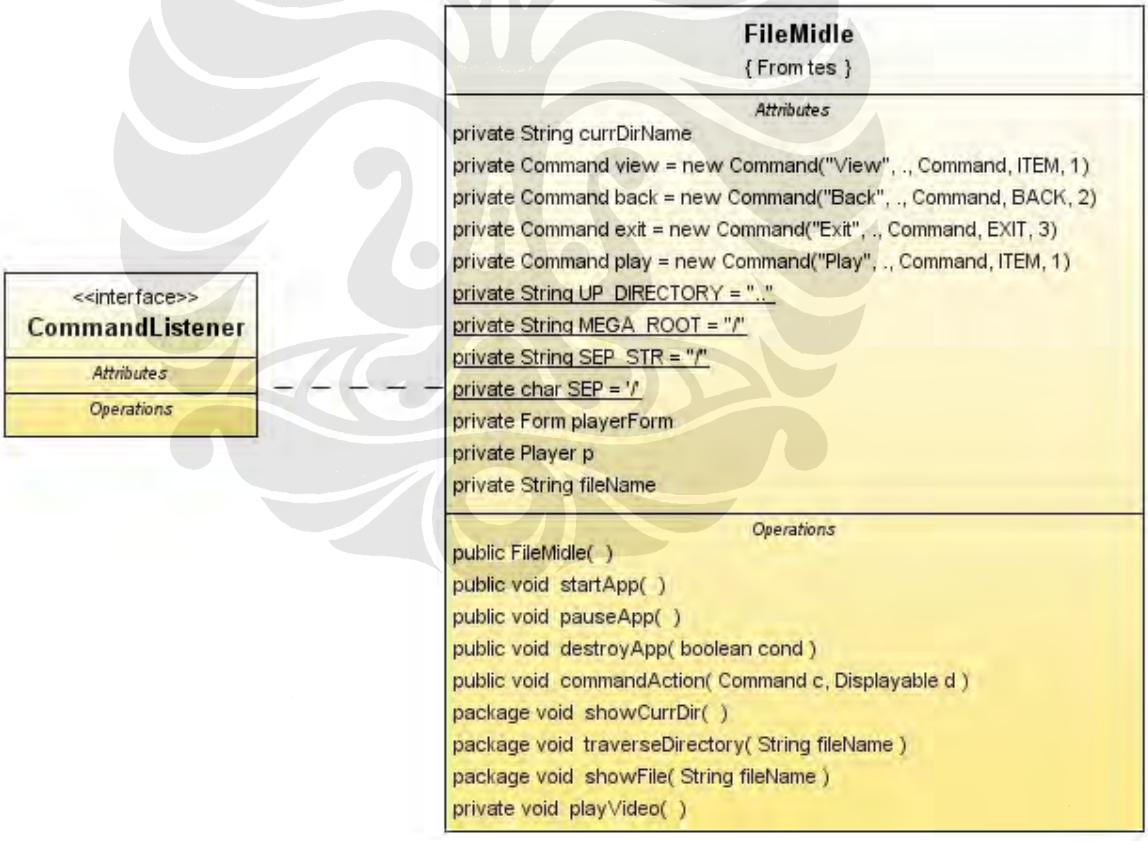
Memperlihatkan kumpulan kelas, antarmuka, dan kerja sama serta hubungan satu sama lainnya antar setiap kelas [7].



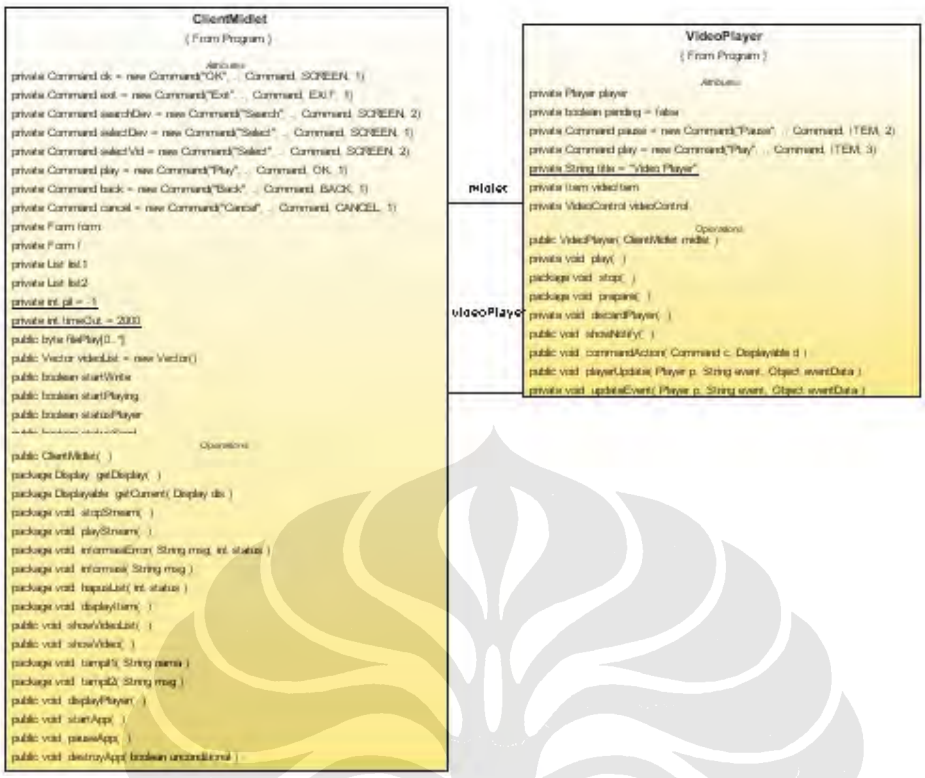
Gambar 3.9 Diagram Class Server Obex



Gambar 3.10 Diagram Class Server RFCOMM



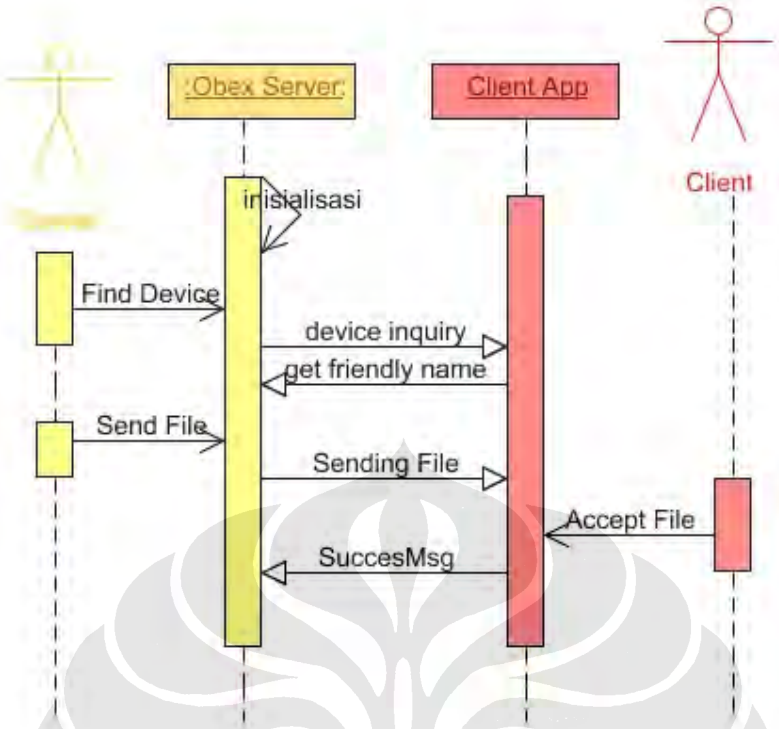
Gambar 3.11 Diagram Class Client Obex



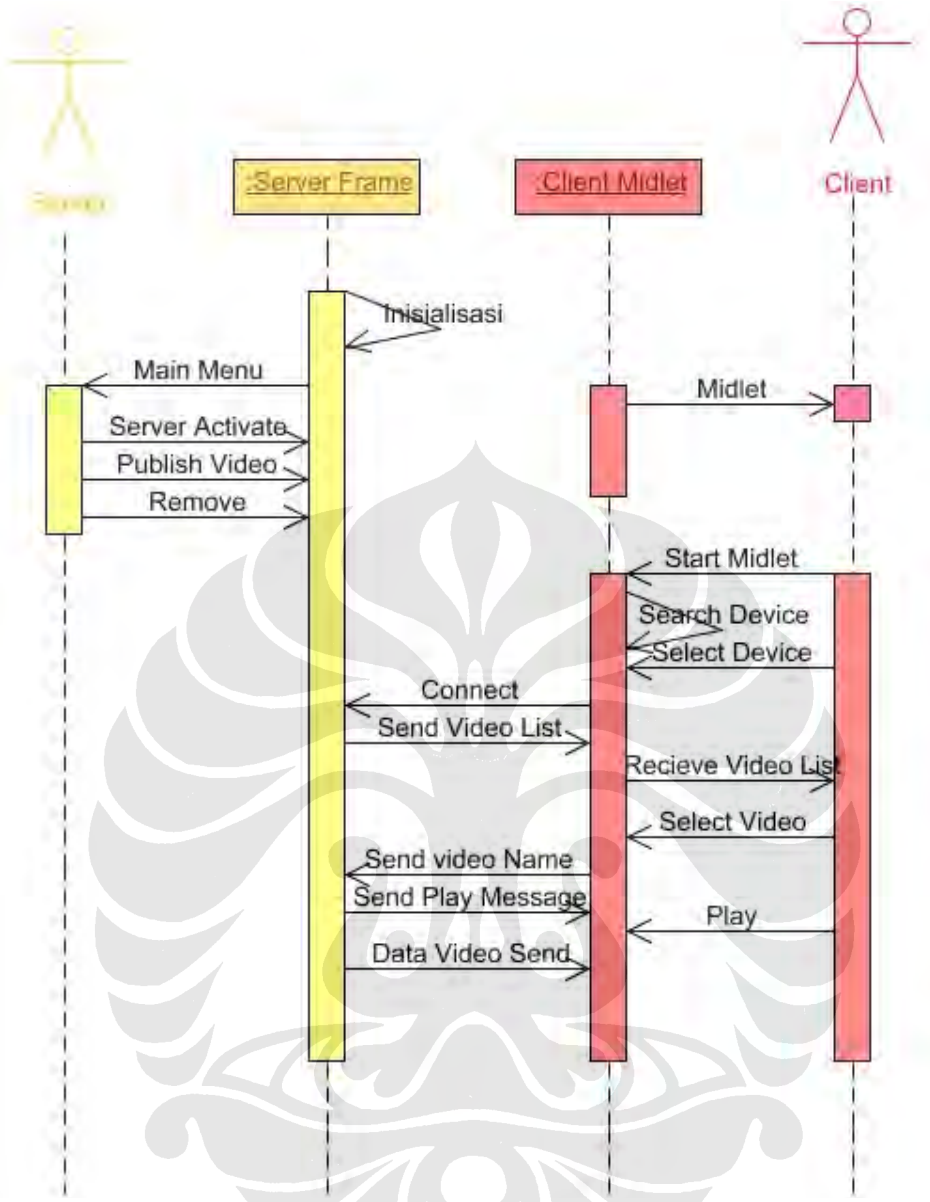
Gambar 3.12 Diagram Class Client RFCOMM

3.2.3 Sequence Diagram

Sequence diagram digunakan untuk menyatakan keseluruhan rincian skenario dan pesan yang mengalir diantara objek-objek setiap saat [7].



Gambar 3.13 Sequence Diagram Obex



Gambar 3.14 Sequence Diagram RfCOMM

BAB IV
UJI COBA IMPLEMENTASI SISTEM

4.1 KONEKSI PADA BLUETOOTH

Koneksi dengan menggunakan Bluetooth dapat digolongkan menjadi dua tipe yaitu :

4.1.1 Koneksi Statik

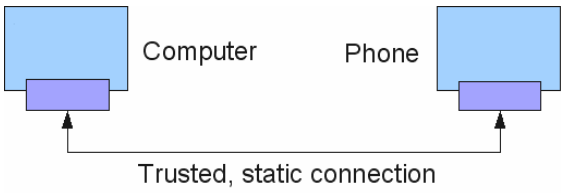
Salah satu kegunaan umum dari komunikasi menggunakan Bluetooth adalah ketika seorang user ingin melakukan synkronisasi data atau mengirimkan file antara PC dengan sebuah HP. Misalnya melakukan synkronisasi kalender dan melakukan instalasi aplikasi ke sebuah HP.

Ketika seorang User ingin melakukan komunikasi yang intensif (frekuensinya berulang) diantara dua divais yang sudah saling dikenal dan diperlukan sebuah auntentifikasi, kedua divais ini biasanya akan melakukan pairing (hanya satu kali). Pairing ini harus dilakukan melalui sebuah HP dan PC UI dan perlu mengetikan passkey yang sama untuk kedua divais

Apablal proses autorisasi dibutuhkan, user harus menerima setiap koneksi Bluetooth yang akan diterima. Akan tetapi, koneksinya dapat kita buat menjadi trusted, sehingga pada koneksi yang berikutnya secara otomatis langsung diterima tanpa perlu autentifikasi.

Catatan : autentifikasi dan autorisasi hanya dapat di set On atau Off pada MIDlet dari client atau end divais.

Berikut ini merupakan gambar 4.1 koneksi static:



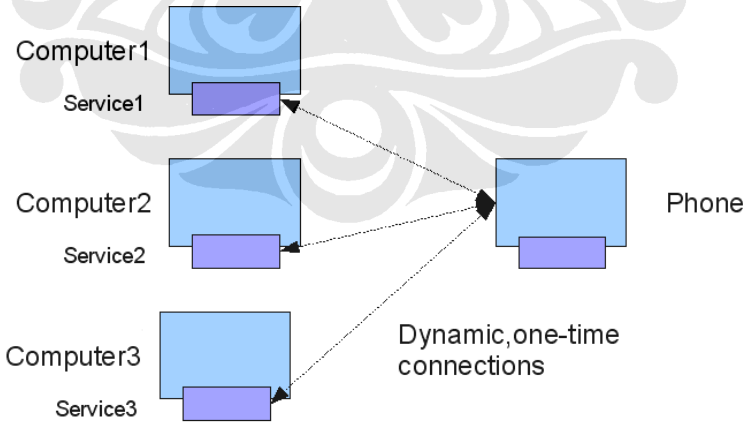
Gambar 4.1 Koneksi Statik

4.1.2 Koneksi Dinamis

Prinsip yang kedua dalam koneksi Bluetooth adalah mencari divais berdasarkan layanan yang ditampilkan. Tidak penting divais mana yang terkoneksi akan tetapi lebih kepada ketepatan layanan.

Salah satu contoh menarik dari tipe koneksi ini adalah layanan komersial, sebagai contoh, mencetak gambar dari HP. User masuk ke dalam kios foto dan mulai menjalankan software printer di HP-nya. Software ini lalu mencari layanan bluetooth yang menyediakan layanan transfer gambar. Ketika layanannya sudah ditemukan maka koneksi akan dibuat dan HP akan meminta user untuk memilih foto yang akan dicetak. Begitu foto sudah dikirm maka koneksi ditutup. Ilustrasi pada gambar 4.2.

Pada prinsip ini, computer dan HP tidak melakukan autentifikasi. Yang pertama kali dilakukan adalah bagaimana computer dan HP saling menemukan koneksi masing lalu melakukan pairing kemudian melakukan proses transfer dan koneksi pun selesai. Pada kasus sebelumnya umumnya dalam menemukan koneksi Bluetooth diinisisasi secara manual oleh user pada PC atau HP. Namun pada koneksi dinamis ini semua prinsip diatas bias dilakukan secara otomatis, sehingga menyederhanakan aksi yang harus dilakukan user. Sehingga sinkronisasi dapat dilakukan secara otomatis dan user hanya diberitahu bahwa datanya telah sinkron.



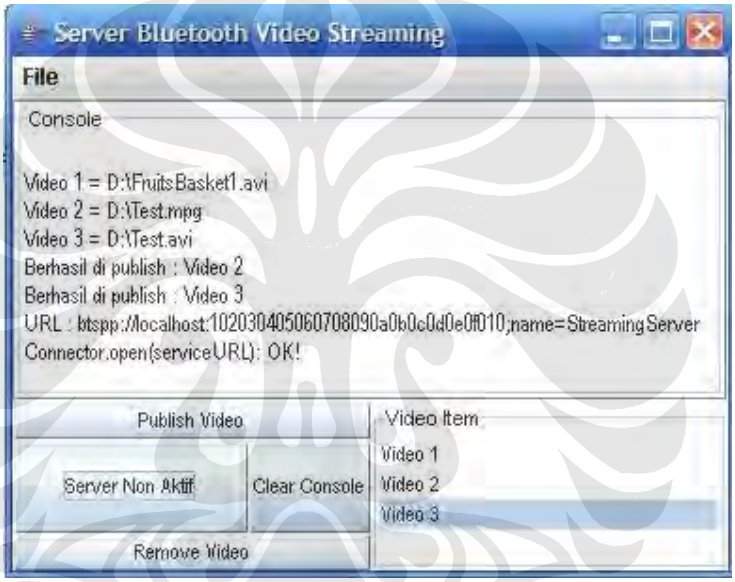
Gambar 4.2 Koneksi dinamis

4.2 IMPLEMENTASI

4.2.1 *User Interface* pada *server*

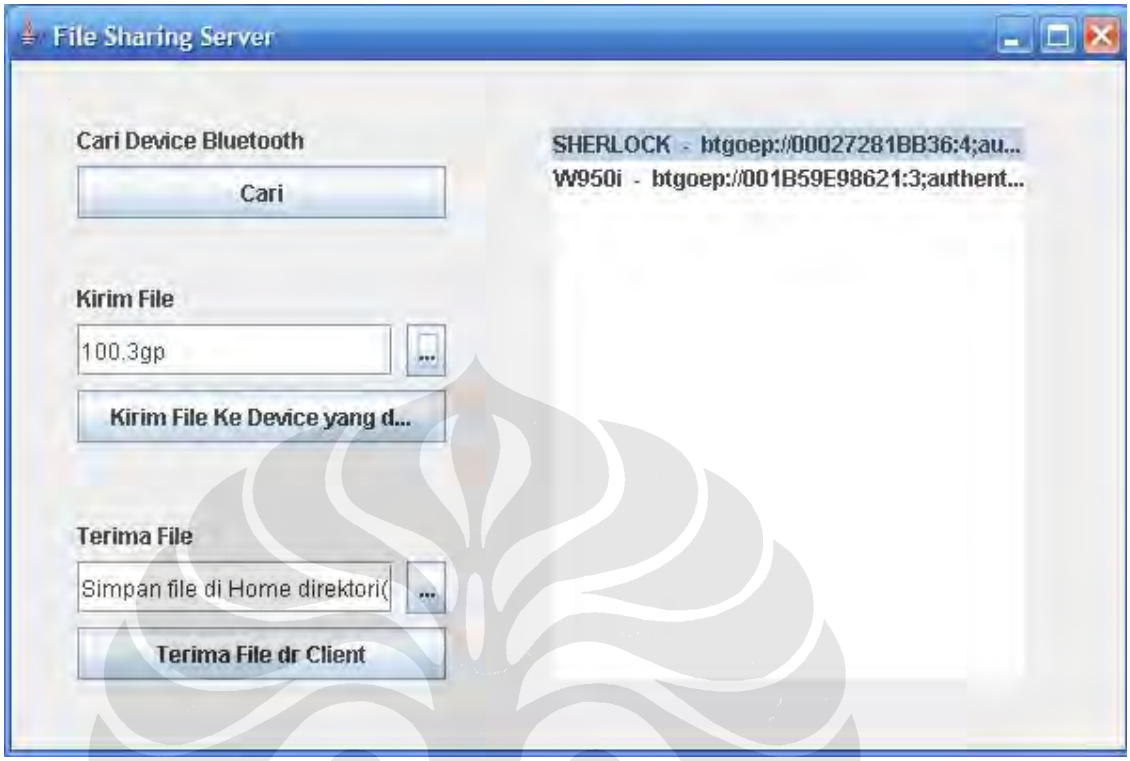
Pada sistem server RFCOMM akan dilakukan pengaktifan bluetooth agar dikenali oleh client, dan akan mem-*publish* video yang nantinya dapat dimainkan oleh client. Server akan menunggu hingga client melakukan koneksi kepada server, setelah client me-*request* video, server melakukan proses *encoding* data video yakni dengan membaginya menjadi beberapa paket yang kemudian tiap paket tersebut dikirim secara berurutan ke client.

Berikut merupakan contoh tampilan dari sistem server pada gambar 4.3



Gambar 4.3 User Interface RFCOMM

Pada sistem server Obex akan dilakukan pengaktifan bluetooth agar dikenali oleh client, dan kemudian akan mencari divais Bluetooth yang ada di sekitarnya lalu akan memilih divias kemudian mengirmkan file pada divais pada ditunjuk, client kemudian dapat melihat isi file yang terkirim setelah file selesai ditransfer. Pada gambar 4.4 contoh tampilan dari user interface server Obex.



Gambar 4.4 User Interface Obex

4.3 ANALISA DAN UJI COBA SISTEM

Pengujian dilakukan dengan menggunakan *personal computer* seperti yang telah dispesifikasikan pada BAB III, sebuah device bluetooth dongle, sebuah handphone W950i, dan sebuah *stopwatch*. Dalam pengujian ini akan dianalisa lamanya sebuah divais menemukan divais dan servis bluetooth, yaitu dengan menggunakan protokol bluetooth Obex dengan stack Bluetooth yang berbeda yakni Widcomm dan Winsock (Microsoft bluetooth Stack). Pengujian kedua akan dianalisa lamanya sebuah divais menemukan divais dan servis bluetooth, yaitu dengan menggunakan protokol bluetooth RFCOMM dengan stack Bluetooth yang berbeda yakni Widcomm dan Winsock Pengujian ketiga yakni pengiriman file dari server ke client dengan menggunakan tipe koneksi yang berbeda dengan RFCOMM dan Obex.

4.3.1 Pengujian *service discovery* berdasarkan setting Obex.

Pengujian dilakukan sebanyak 40 kali, setiap stack baik widcomm dan winsock dilakukan perubahan berdasarkan perubahan jarak yang masing-masing jarak atau jangkauan dilakukan pengujian sebanyak sepuluh kali dengan *server obex*, dan setiap *perubahan jarak* yang sama akan dilakukan pengujian sebanyak dua kali. Dari hasil pengujian tersebut didapat tabel 4.1 dan 4.2 seperti di bawah.

	Search Discovery Aplikasi via OBEX	
Data ke-n	jarak 0,3 m (s)	pada jarak 2,4 m (s)
1	25	25.15
2	24.07	25.2
3	22.15	25.6
4	24.95	26.23
5	24.39	25.38
6	21.89	25.42
7	26.7	26.07
8	25.31	25.71
9	24.98	26.12
10	25.21	25.05
Total	244.65	255.93
Mean	24.465	25.593

Tabel 4.1 Waktu ***client*** untuk ***device discovery*** Obex ***Widcomm Bluetooth stack***

	Device Discovery Aplikasi via OBEX	
Data ke-n	jarak 0,3 m (detik)	jarak 2,4 m (detik)
1	24.95	25.15
2	24.76	25.2
3	25.17	25.6
4	25.24	26.23
5	24.73	25.38
6	24.89	25.42
7	25.01	26.07
8	24.8	25.71
9	24.98	26.12
10	25.21	25.05
Total	249.74	255.93
Mean	24.974	25.593

Tabel 4.2 Waktu ***client*** untuk ***device discovery*** Obex ***Winsock Bluetooth stack***

4.3.2 Pengujian *service discovery* berdasarkan setting RFCOMM.

Pengujian dilakukan sebanyak 40 kali, setiap stack baik widcomm dan winsock dilakukan perubahan berdasarkan perubahan jarak yang masing-masing jarak atau jangkauan dilakukan pengujian sebanyak sepuluh kali dengan *server RFCOMM*, dan setiap *perubahan jarak* yang sama akan dilakukan pengujian sebanyak dua kali. Dari hasil pengujian tersebut didapat tabel 4.3 dan 4.4 seperti di bawah.

	Search Discovery Aplikasi via RFCOMM	
Data ke-n	arak 0,3 m (detik)	jarak 2,4 m (detik)
1	12.89	13.54
2	12.97	13.05
3	13.03	13.1
4	12.8	13.38
5	13.06	13.03
6	13.27	13.97
7	12.9	13.2
8	12.95	13.09
9	13.05	13.41
10	12.96	13.1
Total	129.88	132.87
Mean	12.988	13.287

Tabel 4.3 Waktu ***client*** untuk ***device discovery*** RFCOMM ***Widcomm stack***

	Device Discovery Aplikasi via RFCOMM	
Data ke-n	jarak 0,3 m (detik)	jarak 2,4 m (detik)
1	12.63	13.03
2	12.76	13.4
3	12.73	13
4	12.84	13.37
5	12.82	13.07
6	12.76	13.65
7	12.89	13.04
8	12.75	13.09
9	12.85	13.04
10	12.96	13.1
Total	127.99	131.79
Mean	12.799	13.179

Tabel 4.4 Waktu ***client*** untuk ***device discovery*** RFCOMM ***Winsock stack***

4.3.3 Pengujian *transfer file* berdasarkan setting Obex.

Pengujian dilakukan sebanyak 14 kali, setiap stack baik widcomm dan winsock dilakukan perubahan berdasarkan perubahan ukuran file yang masing-

masing ukuran file dilakukan pengujian sebanyak tujuh kali dengan *server obex*i. Dari hasil pengujian tersebut didapat tabel 4.5 dan 4.6 seperti di bawah.

Data ke-n	Ukuran file (KB)	Lama pengiriman (s)		Kecepatan transfer (KB/s)	
		Pada Winsock stack	Pada Widcomm stack	Pada Winsock stack	Pada Widcomm stack
1	100	3.04	4.28	32.89	23.36
2	200	6.92	8.86	28.90	22.57
3	300	10.07	12.21	29.79	24.57
4	400	13.74	16.32	29.11	24.51
5	500	16.41	20.36	30.47	24.56
6	600	20.04	24.32	29.94	24.67
7	700	21.56	25.42	32.47	23.36

Tabel 4.5 Waktu ***transfer file pada*** Obex ***server Widcomm dan Winsock Bluetooth stack***

4.3.4 Analisa *service discovery* berdasarkan setting Obex dan RFCOMM.

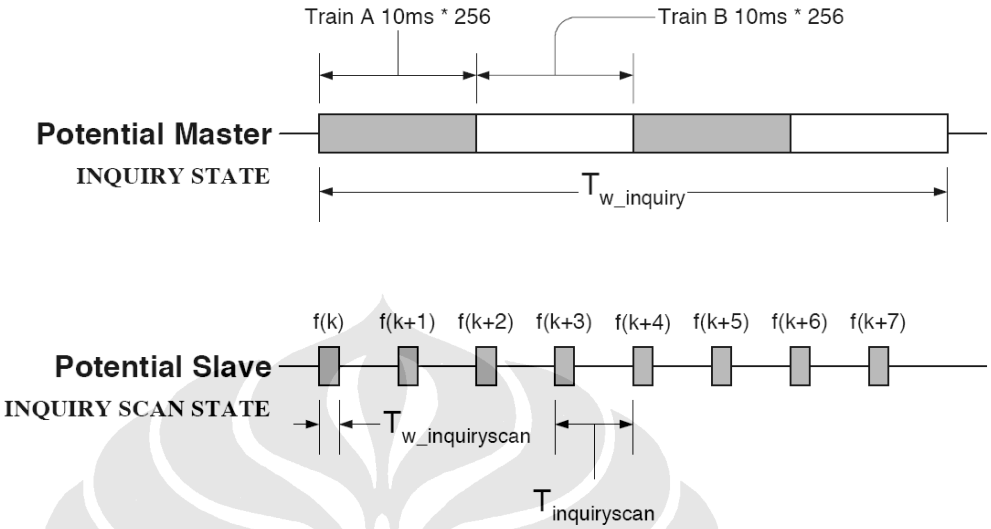
Ketika kita melakukan koneksi pada Bluetooth maka kita akan melalui dua tahap yaitu proses inquiry dan paging. Pada proses inquiry , sebuah divais mencoba untuk menemukan divais Bluetooth lain. Ketika sudah ditemukan maka proses paging digunakan untuk membuat koneksi antar kedua divais tadi. Pihak yang aktif dalam koneksi itu disebut "sender" dan akan menjadi sebuah master dalam sebuah formasi piconet. Pihak yang pasif disebut "receiver"

Sender Inquiry

Pihak sender yang ingin membuat sebuah koneksi dengan pasangan Bluetooth lainnya membutuhkan alamat dan juga informasi mengenai frekuensi asal dari divais tadi. Jika informasi ini tidak ada, pada umumnya dalam membentuk suatu formasi piconet maka dia harus melalui proses inquiry.

Sebuah divais dalam melakukan proses inquiry mengirimkan beberapa packet inquiry pada 32 jenis frekuensi carier dalam Bluetooth frekuensi band. Frekuensi ini diabgi menjadi dua, ke dalam16 frekuensi yang berurutan, yang kita misalkan A dan B. Divais yang melakukan inquiry ini melakukan hop ke semua 16 deretan frekunsi tadi setiap 10 ms. Ini diulang-ulang selama 256 kali (2,56 s) sebelum frekuensi ini melakukan hop pada deretan selanjutnya. Untuk menemukan semua divais yang dalam sebuah ruangan yang bebas noise, inquier harus bergantian antara A dan B sekurang-kurangnya empat kali. Dengan kata lain

inquiry memakan waktu 10,24 s pada kondisi ideall seperti yang terlihat pada gambar 4.5



Gambar 4.5 Prosedur Inquiry pada bluetooth

Receiver Behaviour

Divais yang tidak melakukan inquiry dan sudah berpartispasi dalam foramsi piconet berada pada mode standby, yang secara periodic mengamati proses inquiry. Dalam mode standby. Dalam mode stand by melakukan hop ke frekuensi pembawa, mendengarkan frekeunsi sender namunlebih lambat dari pada frekensi yang ditransmisikan sender. Hop terjadi setiap 1,28 detik dan receiver mendengar 11,25 ms untuk setiap frekuensi.

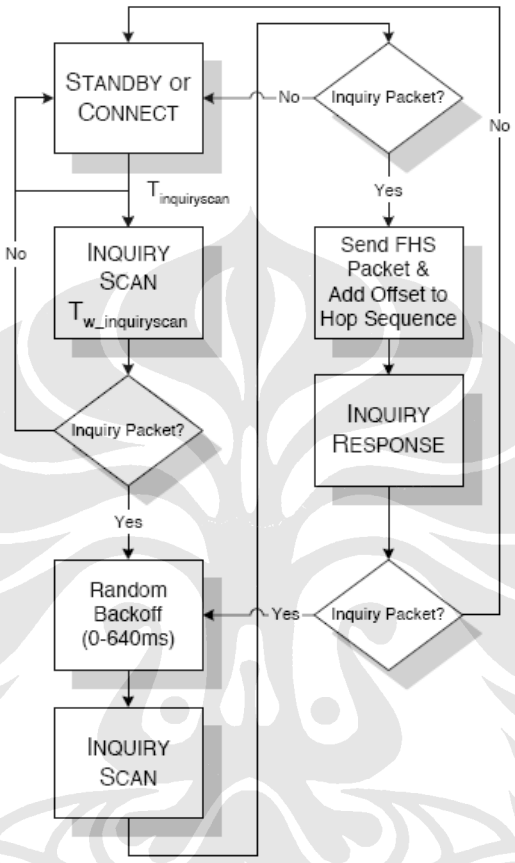
Jika receiver menagkap sebuah paket inqury mengurangi waktu sebanyak 0 sampai 640 ms, lalu mendengar paket inquiry pada frekuensi sama Hal ini bertujuan untuk mengurangi collision antar receiver. Jika paket inquiry ke dua diterima maka ia akan mengrimkan alamat dan informasi frekuensinya. Lalu kemudian diterima paket paging maka ia akan membuat proses koneksi.

Sender Page

Ketika sender memiliki alamat dan informasi frekuensi dari divias terdekat, Ia bisa membuat koneksi dengan divais yang memiliki paket paging tersebut. Paging ditujukan pada divais tertentu, menggunakan kode akses tertentu.

Data waktu untuk pencarian *server* (*device discovery*) ketika tidak ada device yang ditemukan dapat dikatakan sebagai waktu yang dibutuhkan aplikasi untuk melakukan *device discovery*. Waktu 13,287 detik ini lebih lambat

dibandingkan dengan data pada *bluetooth specification* yang menyatakan bahwa waktu minimum untuk melakukan device discovery pada *bluetooth* adalah 10,24 detik . Logika ringkas dalam bentuk state diagram ada pada gambar 4.6.

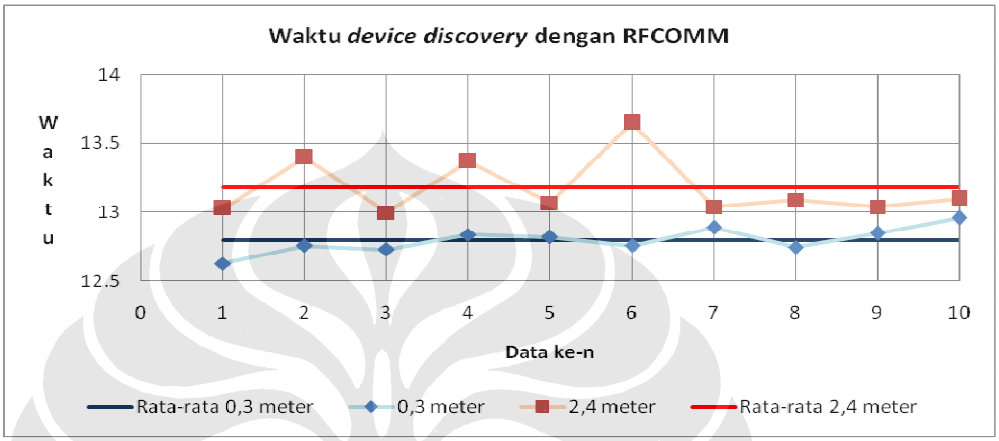


Gambar 4.6 State diagram proses inquiry

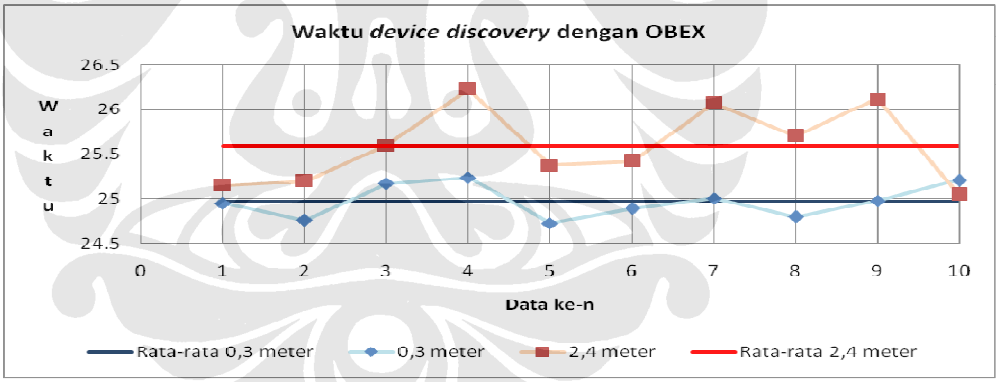
Pada perbandingan uji coba diatas prokol RFCOMM yang lebih baik daripada protokol Obex dalam hal service discovery seperti yang kita lihat perbandingannya dalam gambar garafik 4.7 dan 4.8. Hasil uji coba pada protokol RFCOMM lebih mendekati nilai data literatur dibandingkan protokol Obex. Nilai rata-rata uji coba service discovery pada RFCOMM dengan jangkauan nilai antara 12,799 – 13,287 s, hampir mendekati nilai literatur service discovery pada kondisi ideal senilai 10,24s. Untuk selisih nilai uji coba dengan nilai literatur dikarenakan banyak hop yang harus dilakukan. Sesuai dengan rumus pada tabel 4.6 berikut. Sedangkan device discovery pada Obex lebih lama dikarenakan layer Obex terletak di atas layer RFCOMM.

standalone Bluetooth	worst case	10.24 + *n* 1.28
	typical case	10.24 + *n* 0.64
	best case	10.24

Tabel 4.6 Tabel Service discovery waktu yang dibutuhkan



Gambar 4.7 Grafik perbandingan waktu *device discovery* ***Winsock stack***

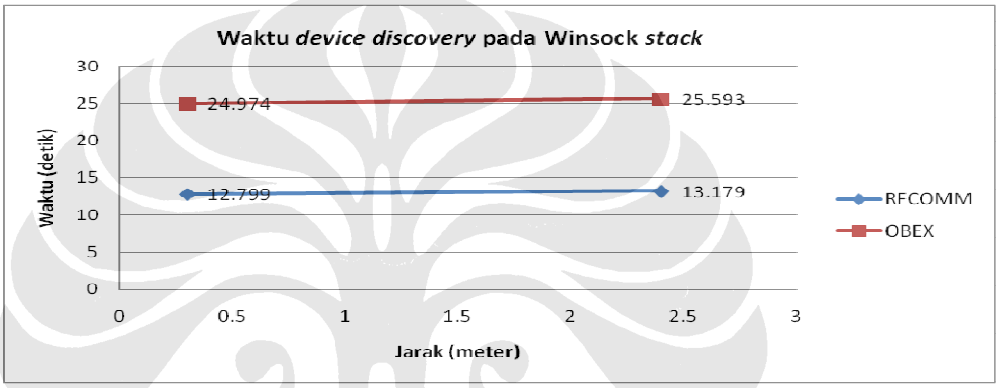


Gambar 4.8 Grafik waktu *device discovery* menggunakan OBEX

4.3.5 Analisa jarak berdasarkan setting Obex dan RFCOMM pada *service discovery*.

Dari waktu rata-rata untuk keempat tabel diatas, terlihat bahwa semakin jauh jarak antara *server* dengan *handheld*, semakin lama waktu yang dibutuhkan untuk mengadakan koneksi, hal tersebut dapat kita lihat dengan gambar grafik 4.9. Dapat disimpulkan bahwa jarak antara *server* dan *handheld* mempengaruhi waktu yang dibutuhkan *handheld* untuk mencari *server* dan melakukan koneksi. Hal ini dikarenakan daya yang dipancarkan oleh server semakin berkurang. Daya yang dipancarkan oleh bluetooth kelas 1 dengan jangkauan 100m sebesar 1mW dengan

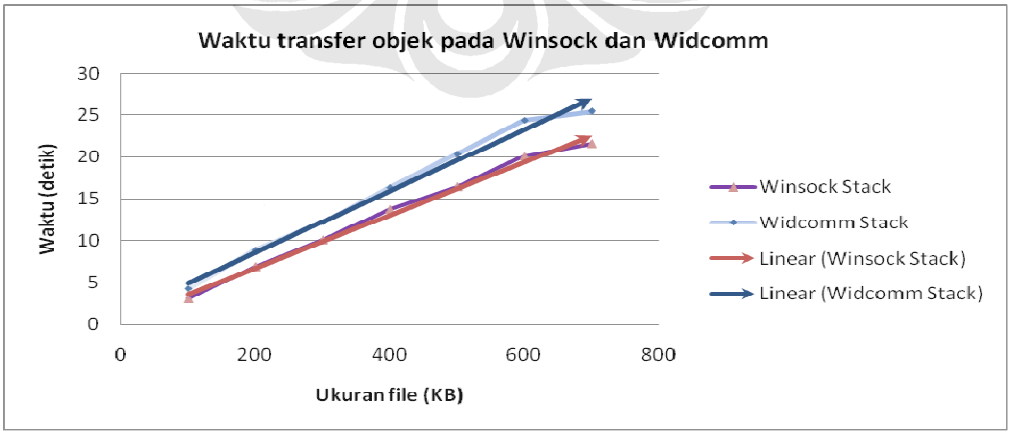
intensitas sebesar 20dB. Daya yang dipancarkan bluetooth ini sangat kecil sehingga senakin jauh jarak server dengan client maka akan senakin banyak loss daya. Sehingga waktu yang dibutuhkan semakin lama. Adanya fluktuasi daya di penerima menyebabkan terjadinya fading. Fading terjadi karena interferensi atau superposisi gelombang multipath yang memiliki amplitudo dan fasa yang berbeda-beda. Sinyal multipath juga akan menyebabkan distorsi sinyal / cacat sinyal. Problem ini secara khusus berkaitan dengan bandwidth sinyal yang digunakan dalam komunikasi mobile, dan juga karena respon pulsa yang berbeda dari sinyal multipath



Gambar 4.9 Grafik perbandingan waktu *device discovery*

4.3.6 Analisa Transfer file berdasarkan setting Obex pada *Bluetooth Stack Widcomm dan Winsock*.

Pada penelitian ini, dilakukan pengukuran terhadap kecepatan transfer file video melalui koneksi OBEX menggunakan *stack* Winsock dan Widcomm yang dibandingkan hasilnya pada gambar grafik 4.10

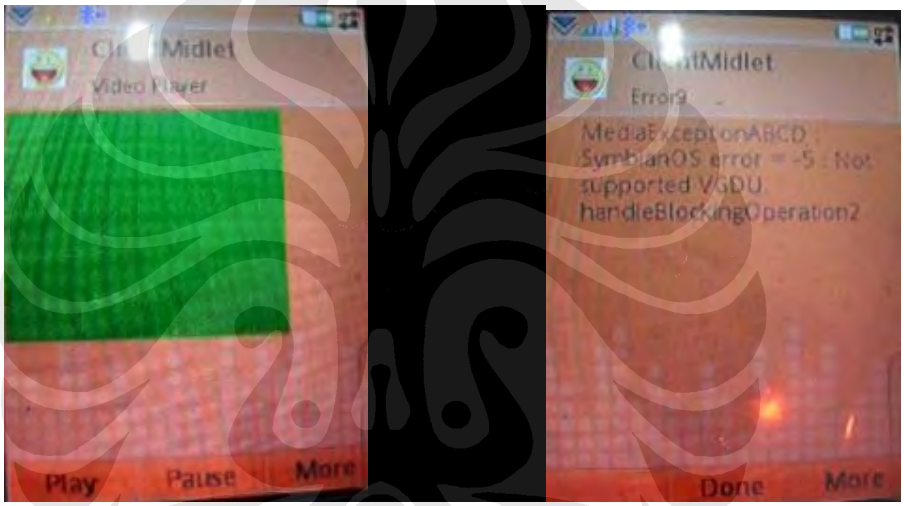


Gambar 4.10 Grafik waktu transfer pada Winsock dan Widcomm

Dari gambar diatas dapat dilihat bahwa perbedaan kecepatan tranfer object antara widcomm dan winsock cenderung stabil marginnya sekitar 1 s. Winsock melakukan transfer lebih cepat karena dia bekerja pada OS windows, bukan pada driver atau aplikasi tambahan.

4.3.7 Pengujian *transfer file* pada rfcomm.

Dari percobaan yang dilakukan untuk kondisi ini, terjadi kegagalan dalam pengiriman file dari server pc ke mobile client. Hasil pada file bisa kita lihat pada gambar 4.11.

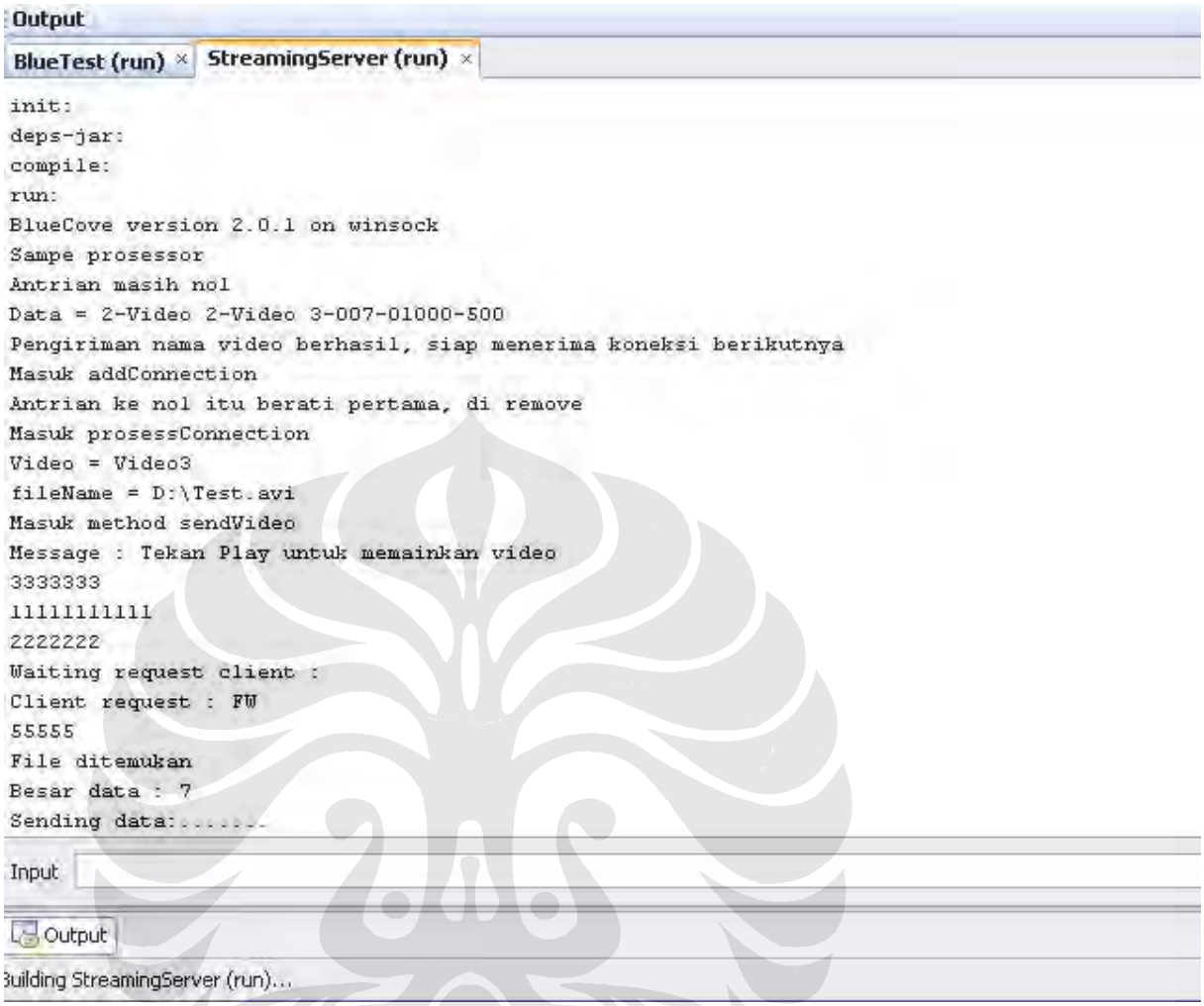


Gambar 4.11 File Error pada waktu dikirim ke client

Terjadinya MediaException Hal ini bisa dikarenakan dua hal :

1. Packet data yang dikirim coruppt yang merupakan kesalahan transmisi atau protokol.
2. video player pada level aplikasi tidak mensupport codec video dalam hal ini MMAPI JSR-135

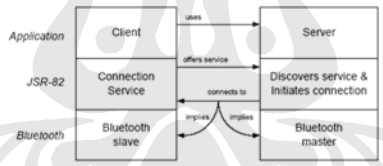
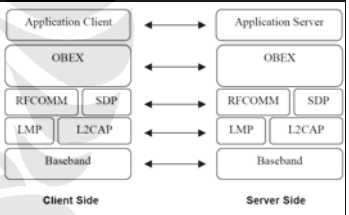
Kemungkinan kedua telah dipatahkan karena kami berhasil membuat video player dari source code yang sama dan memutar file dengan code yang sama pada w950i. Lalu kemudian kita melakukan analisa paket. Yang dikirim berikut ini adalah contoh analisa nya



Gambar 4.12 console debugging video streaming server

Pada gambar 4.8 terlihat bahwa console mengirmkan data sebanyak 7kB dan data di slice ke dalam ByteArrayOutputStream dengan besar slice sebesar 1024b. dan setiap file data yang berhasil dikirm maka client memberikan pesan “Client Request FW” yang menandakan bahwa data telah diterima client. Dan ketika data diterima dan coba di lihat maka akan tampak hasilnya pada gambar4.11 diatas

Tabel 4.7 Tabel Perbandingan RFCOMM dan OBEX

	RFCOMM	**OBEX**
OSI *Layer*	Transport	Sesion
Kompleksitas implementasi	Tinggi	Menengah
Bagian Termodifikasi	Perangkat Lunak	Perangkat lunak
Dukungan point-to-multipoint	Ya	Ya
Profil Bluetooth	LAN Acces Protocol	Object Push, File Transfer, Synchronization.
Paket dalam JSR-API82	Javax.bluetooth	Javax.obex
Koneksi	Stream / binary (DataInputStream, DataOutputStream)	Object (*get, put)*
URL koneksi	btspp:// localhost:" + uuid;**authenticate**=true;**authorize**=true;**encrypt**=true	"btgoep://localhost:" + uuid +;authenticate=false;master=false;encrypt=false";
Diagram Blok		
Aplikasi yg cocok	Bluetooth Chat (pesan dengan atribut string)	Bluetooth File Transfer (objek berupa file)
Resume berdasarkan Hasil Percobaan		
Waktu Service Discovery	12,799 – 13,287 s Mendekati nilai literatur 10,24s	24.465 - 25.593 s Lebih lama karena satu layer diatasnya butuh enkapsulapsi data lebih banyak.
Transfer File	FileInputStream, hasilnya file coruppt, data yang ditransmisikan melalui	Get & put, hasil file lengkap.

	ByteArrayOutputStream mempunyai batasan maksimal 80 b	
Header File	Tidak bisa dimodifikasi	Bisa dimodifikasi (nama, panjang dan deskripsi file).
Waktu Service Discovery dipengaruhi jarak	Ya	Ya
Waktu Service Discovery dipengaruhi Stack Bluetooth (widcomm/ winsock)	Ya	Ya
Header File	Tidak bisa dimodifikasi	Bisa dimodifikasi (nama, panjang dan deskripsi file).

BAB 5

KESIMPULAN

Kesimpulan yang bisa didapat dari tugas akhir ini, yaitu :

1. Ketika kita melakukan koneksi pada Bluetooth maka kita akan melalui dua tahap yaitu proses inquiry dan paging. Pada proses inquiry , sebuah divais mencoba untuk menemukan divais Bluetooth lain. Ketika sudah ditemukan maka proses paging digunakan untuk membuat koneksi antar kedua divais tadi
2. Proses inquiry secara ideal hanya memakan waktu 10,24s terjadinya penambahan waktu pada service discovery dikarenakan paket data inquiry belum diterima oleh bluetooth device.
3. Nilai rata-rata uji coba service discovery pada RFCOMM dengan jangkauan nilai antara 12,799 – 13,287 s, hampir mendekati nilai literatur service discovery pada kondisi ideal senilai 10,24s. Sedangkan device discovery pada Obex lebih lama dikarenakan layer Obex terletak di atas layer RFCOMM.
4. Jarak antara *server* dan *handheld* mempengaruhi waktu yang dibutuhkan *handheld* untuk mencari *server* dan melakukan koneksi. Hal ini dikarenakan daya 1mW yang dipancarkan oleh server semakin berkurang. Sehingga menyebabkan fading pada divais penerima.
5. Perbedaan kecepatan tranfer object antara widcomm dan winsock cenderung stabil marginnya sekitar 1 s. Winsock melakukan transfer lebih cepat karena dia bekerja pada OS windows, bukan pada driver atau aplikasi tambahan.
6. Pengirman file lebih baik menggunakan protokol obex karena terdapat profil GEOP (general object push) , object push dan sinkronisasi. Tetapi memiliki kekurangan karena
    a. Bukan merupakan protocol komunikasi maka dia hanya mengirim atau menerima dalam satu kesempatan / half duplex.
    b. Dan memerlukan authentifikasi oleh client untuk setipa pengiriman file.

DAFTAR ACUAN

[1] John G. Apostolopoulos, Wai-tian Tan, Susie J. Wee, "Video Streaming : Concepts, Algorithms, and Systems". Mobile and Media Systems Laboratory, Hewlett-Packard Paolo Alto, USA. Diakses 20 Oktober 2007 dari Hewlett-Packard Labs.
www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf

[2] Donny B.U., "Streaming: Membuat File Besar Serasa Kecil", 2002. Diakses 10 Oktober 2007.
www.free.vlsm.org/v17/com/ictwatch/paper/paper018.htm

[3] Onno W. Purbo, "Kebutuhan Infrastruktur Untuk Video Conference". Diakses 30 Oktober 2007 dari situs Dr. Onno W Purbo.
www.onno.vlsm.org/v11/onno-ind-3/network/kebutuhan-infrastruktur-untuk-**video**-conference-3-2003.rtf

[4] ITU-T. *Line Transmission of Non-Telephone Signals.* ITU-T Recommendation H.261 (03/93). 1994. Diakses 22 November 2007 dari situs ITU-T.
www.itu.int/rec/T-REC-H.261-199303-I/en

[5] Tesi di Laurea, "Confronto tra due protocolli per reti Wireless: IEEE 802.11 e Bluetooth", 2001. Diakses 30 November 2007
http://fly.isti.cnr.it/didattica/tesi/Tallarico/tesi

[6] Keven Boyett, "Protocol Tests Lay Groundwork for Bluetooth Success", Tektronix, 2001. Diakses 30 November 2007
http://archive.e.valuationengineering.com/archive/articles/0801blue.htm

[7] Budi Raharjo, Imam Haryanto, Arif Haryono, *Tuntunan Pemrograman Java untuk Handphone.* (Bandung : Informatika, 2007), hal. 1-2.

[8] Sun Microsystems, "JDK 5.0 Documentation". Diakses 9 November 2007 dari Sun Microsystem.
http://java.sun.com/j2se/1.5.0/docs/index.html

[9] Salahudin, M dan Rosa A.S, *Pemrograman J2ME.* (Bandung : Informatika, 2006).

[10] Martin de Jode, *Programming Java 2 Micro Edition on Symbian OS*, A developer's guide to MIDP 2.0. (Inggris: John Wiley & Sons Ltd, 2004), hal 4.

[11] Rohit Kapoor, Matteo Cesana, Mario Gerla , "Link Layer Support for Streaming MPEG Video over Wireless Links", International Conference on

Computer Communications and Networks (ICCCN 2003), 2003. Diakses 20 Oktober 2007 dari UCLA Computer Science Department.
http://www.cs.ucla.edu/ST/docs/icccn2003.pdf

[12] Hirotsugu Okura, M. Kato, dan S. Tasaka, "A Media Synchronization Experiment on Continuous Media Transmission in Bluetooth LAN Access", 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001.
http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=20847&isYear=2001&count=11&page=4&ResultStart=100

[13] Wang Xiaohang, "Video Streaming over Bluetooth: A Survey". Institute for Infocomm Research (I2R), Singapura. Diakses 20 Agustus 2007 dari CiteSeer.IST
http://www.citeseer.ist.psu.edu/708770.html

[14] C. Hooi Chia, M. Salim Beg, "MPEG-4 Video Transmission over Bluetooth Link", Proc. IEEE International Conf. on Personal Wireless Communications, New Delhi, Dec 2002.
ieeexplore.ieee.org/iel5/30/28185/01261191.pdf

DAFTAR PUSTAKA

Apostolopoulos, John G., Wai-tian Tan, Susie J. Wee, "Video Streaming : Concepts, Algorithms, and Systems". Mobile and Media Systems Laboratory, Hewlett-Packard Paolo Alto, USA. Diakses 20 Oktober 2007 dari Hewlett-Packard Labs. www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf

Austerberry, David. *The Technology of Video and Audio Streaming* (Inggris : Elsevier, 2005).

Boyett, Keven, "Protocol Tests Lay Groundwork for Bluetooth Success", Tektronix, 2001. Diakses 30 November 2007 http://archive.e.valuationengineering.com/archive/articles/0801blue.htm

de Jod, Martin. *Programming Java 2 Micro Edition on Symbian OS.* A developer's guide to MIDP 2.0. (Inggris: John Wiley & Sons Ltd, 2004).

Demetriades, Gregory C. *Streaming Media*, Building and Implementing a Complete Streaming System. (USA : Wiley Publishing Inc., 2003)

Hopkins, Bruce, Ranjith Antony. *Bluetooth for Java.* (USA : Apress, 2003).

ITU-T. *Line Transmission of Non-Telephone Signals.* ITU-T Recommendation H.261 (03/93). 1994. Diakses 22 November 2007 dari situs ITU-T. www.itu.int/rec/T-REC-H.261-199303-I/en

Kapoor, R., Matteo Cesana, Mario Gerla , "Link Layer Support for Streaming MPEG Video over Wireless Links", International Conference on Computer Communications and Networks (ICCCN 2003), 2003. Diakses 20 Oktober 2007 dari UCLA Computer Science Department.

Klingsheim, Andre N. "J2ME Bluetooth Programming." Tesis, Program Pasca Sarjana Departemen Informatika Universitas Bergensis, Norwegia, 2004.

M, Salahudin, dan Rosa A.S. *Pemrograman J2ME.* (Bandung : Informatika, 2006).

Purbo, Onno W. "Kebutuhan Infrastruktur Untuk Video Conference". Diakses 30 Oktober 2007 dari situs Dr. Onno W Purbo. www.onno.vlsm.org/v11/onno-ind-3/network/kebutuhan-infrastruktur-untuk-**video**-conference-3-2003.rtf

Raharjo, B., Imam Haryanto, Arif Haryono. *Tuntunan Pemrograman Java untuk Handphone.* (Bandung : Informatika, 2007).

Sun Microsystems, “JDK 5.0 Documentation”. Diakses 9 November 2007 dari Sun Microsystem.
http://java.sun.com/j2se/1.5.0/docs/index.html

Xiaohang, Wang “Video Streaming over Bluetooth: A Survey”. Institute for Infocomm Research (I2R), Singapura. Diakses 20 Agustus 2007 dari CiteSeer.IST
http://www.citeseer.ist.psu.edu/708770.html