

BAB V

KESIMPULAN DAN SARAN

V.1 Kesimpulan

Dari hasil penelitian ini didapatkan kesimpulan sebagai berikut:

1. Untuk mendeteksi *bluetooth-enabled device* yang dituju yang jaraknya di luar jangkauan deteksi *bluetooth-enabled device* pengirim digunakan suatu *bluetooth-enabled device* sebagai perantara yang akan membantu mendeteksi *bluetooth-enabled device* lain disekitarnya. Tidak seperti *repeater* pada umumnya dengan menggunakan *hardware* khusus atau dengan pembangunan *tower-tower*, tetapi dalam pembangunan aplikasi *bluetooth repeater* ini digunakan suatu *bluetooth-enabled device* sebagai penguat sinyalnya.
2. Untuk mengirimkan teks ke *bluetooth-enabled device* yang dituju dengan menggunakan *bluetooth* digunakan metode *OutputStream* untuk mengirimkan pesan teks dalam bentuk *byte* dan *InputStream* untuk membaca pesan yang dikirimkan. Pesan dikirimkan menggunakan protokol *RFCOMM*.
3. Untuk membangun aplikasi penguat sinyal *bluetooth* ini digunakan produk *J2ME* yang dikeluarkan oleh *Sun Microsystems*. *J2ME* ini dipilih karena produknya yang bersifat *open source*, bebas platform, menyediakan dukungan penuh terhadap teknologi *bluetooth* dengan API-nya, yaitu *jsr-82*.

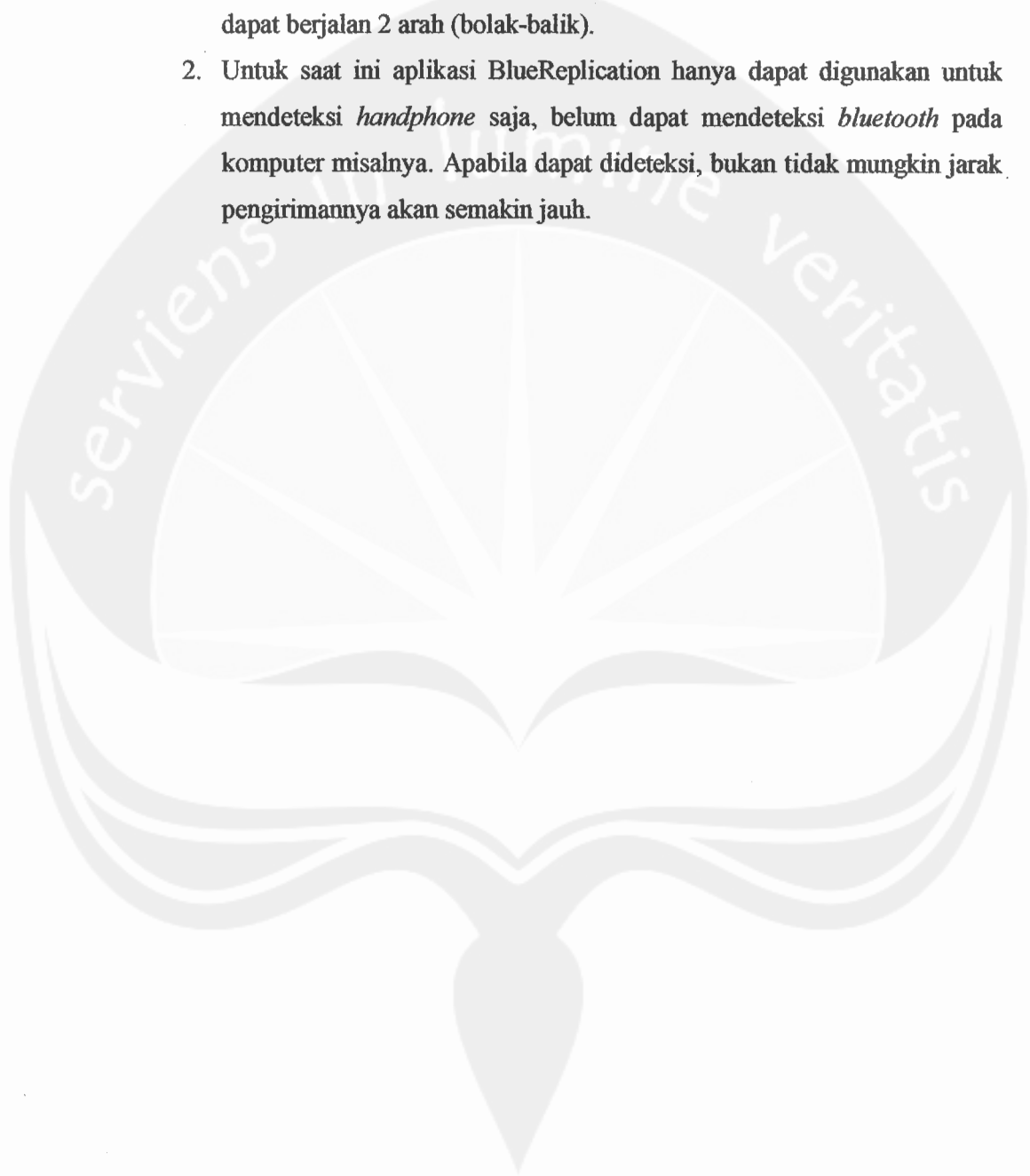
V.2 Saran

Aplikasi yang dibuat ini, masih terdapat banyak kekurangan, diantaranya:

1. Aplikasi *BlueReplication* masih bersifat *sequential* dalam pengiriman pesannya, artinya proses pengiriman pesannya harus berurutan dan

dimulai dari *client* mengirim pesan ke *server* kemudian *server* mengirim pesan ke *client* dan begitu seterusnya. Oleh sebab itu, aplikasi ini dapat dikembangkan lagi salah satunya adalah dengan menambahkan proses *Threading* didalamnya, sehingga dapat proses pengiriman pesannya dapat berjalan 2 arah (bolak-balik).

2. Untuk saat ini aplikasi BlueReplication hanya dapat digunakan untuk mendeteksi *handphone* saja, belum dapat mendeteksi *bluetooth* pada komputer misalnya. Apabila dapat dideteksi, bukan tidak mungkin jarak pengirimannya akan semakin jauh.



DAFTAR PUSTAKA

A Brief Introduction to MIDP Programming version 0.9.3, Forum Nokia, 7 Maret 2002.

A Brief Introduction to Networked MIDlets version 1.0, Forum Nokia, 18 November 2002.

Hartanto, Antonius Aditya. *Pemrograman Mobile Java dengan MIDP 2.0*. Andi Yogyakarta, 2004.

Hopkins, Bruce., Antony, Ranjith., *Bluetooth for Java*. Apress, 2004.

MIDP: Bluetooth API Developer's Guide, Forum Nokia version 2.0, 31 Oktober 2006.

Sari, Lestika. *Pengembangan Perangkat Lunak Aplikasi Chatting Berbasis Bluetooth Untuk Telepon Selular*. Yogyakarta, 2007.

<http://developer.sonyericsson.com/>

<http://developer.symbian.com/>

http://developer.symbian.com/main/downloads/papers/JABWT/JABWT_1_0.pdf

<http://developers.sun.com/mobility/getstart/articles/survey/>

<http://developers.sun.com/mobility/midp/articles/bluetooth1/>

<http://developers.sun.com/mobility/apis/articles/bluetoothintro/>

<http://developers.sun.com/mobility/midp/articles/bluetooth2/>

<http://developers.sun.com/mobility/midp/articles/wtoolkit/>

<http://developers.sun.com/mobility/midp/articles/tutorial2/>

<http://en.wikipedia.org/wiki/Repeater>

[http://forum.nokia.com/info/sw.nokia.com/id/98f61174-e3fc-499f-be81-](http://forum.nokia.com/info/sw.nokia.com/id/98f61174-e3fc-499f-be81-7ce66c0a99aa/Bluetooth_Technology_Overview_v1_0_en.pdf.html)

[7ce66c0a99aa/Bluetooth Technology Overview v1_0 en.pdf.html](http://forum.nokia.com/info/sw.nokia.com/id/98f61174-e3fc-499f-be81-7ce66c0a99aa/Bluetooth_Technology_Overview_v1_0_en.pdf.html)

www.oke.or.id/tutorial/jarkom.pdf

http://wireless.klings.org/klings_jabwt_master_thesis.pdf

www.bogor.net/idkf/bio2/indo-artikel/faisal-j2me.pdf

www.ilmukomputer.com/2006/09/04/pengantar-teknologi-bluetooth/ - 29k -

SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

BLUETOOTH REPEATER APPLICATION

(BlueRepeater)

Disusun oleh:


Marcellinus Willindra

03 07 03913

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta	Nomor Dokumen		Halaman
		<i>SKPL-BlueReplication</i>		<i>1/28</i>
		Revisi		<i>Tgl : 01-02-2007</i>

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	SKY							
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1	Pendahuluan.....	6
1.1	Tujuan.....	6
1.2	Lingkup Masalah	6
1.3	Definisi Istilah dan Singkatan.....	7
1.4	Referensi.....	7
1.5	Deskripsi Umum (Overview).....	7
2	Deskripsi Keseluruhan	8
2.1	Perspektif Produk.....	8
2.2	Kebutuhan Fungsionalitas Perangkat Lunak	9
2.2.1	Use Case : Memilih Device.....	9
2.2.2	Use case : Mendeteksi Device.....	10
2.2.3	Use Case : Memperkuat Sinyal.....	10
2.2.4	Use Case : Mengirimkan Pesan.....	10
2.2.5	Use Case : Koneksi Ke Server	10
2.2.6	Use Case : Start Server	10
2.2.7	Use Case : Start Repeater.....	11
2.2.8	Use Case : Pengiriman Pesan Biasa	11
2.2.9	Use Case : Pengiriman Pesan dengan Repeater	11
2.3	Karakteristik Sistem.....	11
2.3.1	User.....	11
3	Deskripsi Rinci Kebutuhan	12
3.1	Spesifikasi Kebutuhan Fungsionalitas	12
3.1.1	Spesifikasi Use Case : Memilih Device.....	12
3.1.2	Spesifikasi Use Case : Mendeteksi Device.....	13
3.1.3	Spesifikasi Use Case : Memperkuat Sinyal.....	14
3.1.4	Spesifikasi Use Case : Mengirimkan Pesan.....	15
3.1.5	Spesifikasi Use Case : Koneksi Ke Server.....	16
3.1.6	Spesifikasi Use Case : Start Server.....	17
3.1.7	Spesifikasi Use Case : Start Repeater.....	18
3.1.8	Spesifikasi Use Case : Pengiriman Pesan Biasa.....	19
3.1.9	Spesifikasi Use Case : Pengiriman Pesan dengan Repeater	20
3.2	Spesifikasi Kebutuhan Non-Fungsionalitas.....	21
3.2.1	Kebutuhan Antarmuka Eksternal	21
3.2.2	Antarmuka Pemakai.....	21
3.2.3	Antarmuka Perangkat Keras.....	21
3.2.4	Antarmuka Perangkat Lunak.....	21
3.2.5	Persistent Data.....	22
4	Realisasi Use Case	22
4.1	Static Structure Diagram.....	22
4.1.1	Analysis Class Diagram : Package Dependencies.....	22
4.1.2	Analysis Class Diagram : Package BlueReplication	23
4.1.3	Analysis Class Diagram : Use Case Mendeteksi Device.....	24
4.1.4	Analysis Class Diagram : Use Case Memilih Device	25
4.1.5	Analysis Class Diagram : Use Case Memperkuat Sinyal.....	25
4.1.6	Analysis Class Diagram : Use Case Mengirimkan Pesan.....	26
4.1.7	Analysis Class Diagram : Use Case Koneksi Ke Server	26
4.1.8	Analysis Class Diagram : Use Case Start Server	27
4.1.9	Analysis Class Diagram : Use Case Start Repeater.....	27

4.1.10	Analysis Class Diagram : Use Case Pengiriman Pesan Biasa	27
4.1.11	Analysis Class Diagram : Use Case Pengiriman Pesan dengan Repeater	28

Daftar Gambar

Gambar 2.1	Use Case Diagram.....	10
Gambar 4.1	Analysis Class Diagram : Package Dependencies.....	23
Gambar 4.2	Analysis Class Diagram : Package BlueReplication.....	24
Gambar 4.3	Analysis Class Diagram : Use Case Mendeteksi Device.....	25
Gambar 4.4	Analysis Class Diagram : Use Case Memilih Device.....	26
Gambar 4.5	Analysis Class Diagram : Use Case Memperkuat Sinyal.....	26
Gambar 4.6	Analysis Class Diagram : Use Case Mengirimkan Pesan.....	27
Gambar 4.7	Analysis Class Diagram : Use Case Koneksi Ke Server.....	27
Gambar 4.8	Analysis Class Diagram : Use Case Start Server.....	28
Gambar 4.9	Analysis Class Diagram : Use Case Start Repeater.....	28
Gambar 4.10	Analysis Class Diagram : Use Case Pengiriman Pesan Biasa.....	28
Gambar 4.11	Analysis Class Diagram : Use Case Pengiriman Pesan dengan Repeater.....	29

Daftar Tabel

Tabel 3.1	Spesifikasi Use Case : Memilih Device.....	14
Tabel 3.2	Spesifikasi Use Case : Mendeteksi Device.....	15
Tabel 3.3	Spesifikasi Use Case : Memperkuat Sinyal.....	16
Tabel 3.4	Spesifikasi Use Case : Mengirimkan Pesan.....	17
Tabel 3.5	Spesifikasi Use Case : Koneksi Ke Server.....	18
Tabel 3.6	Spesifikasi Use Case : Start Server.....	19
Tabel 3.7	Spesifikasi Use Case : Start Repeater.....	20
Tabel 3.8	Spesifikasi Use Case : Pengiriman Pesan Biasa.....	21
Tabel 3.9	Spesifikasi Use Case : Pengiriman Pesan dengan Repeater.....	21

PEMBANGUNAN APLIKASI PENGUAT SINYAL BLUETOOTH UNTUK MENGIRIMKAN TEKS ANTAR MOBILE DEVICE MENGGUNAKAN J2ME (BlueReplication)

1 Pendahuluan

1.1 Tujuan

Tujuan dari dokumen spesifikasi kebutuhan perangkat lunak (SKPL) dalam pengembangan perangkat lunak *Bluetooth Repeater Application* (BlueReplication), yaitu mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak, perangkat keras, dan pengguna), performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), atribut (fitur-fitur tambahan yang dimiliki sistem), dan mendefinisikan fungsi perangkat lunak serta batasan perancangan perangkat lunak. SKPL - BlueReplication ini juga digunakan oleh pengembang perangkat lunak sebagai acuan teknis untuk pengembangan perangkat lunak. Dalam SKPL ini akan dijelaskan mengenai kebutuhan-kebutuhan yang harus tersedia agar perangkat lunak yang diharapkan dapat terwujud. Dokumen SKPL ini menjadi dasar kesepakatan antara pihak pelanggan dengan pihak pengembang mengenai perangkat lunak yang akan dikembangkan.

1.2 Lingkup Masalah

Gambar di bawah ini menunjukkan gambaran mengenai jangkauan yang dapat dideteksi *bluetooth*

Bluetooth pada *mobile device* mampu mendeteksi lokasi *bluetooth-enabled device* lainnya, dalam radius sekitar 10 meter. Pertama kali, *bluetooth* pada *mobile device* diaktifkan, kemudian jalankan aplikasi BlueReplication. *Bluetooth* akan melakukan proses *device discovery* untuk menemukan keberadaan *bluetooth* lainnya dalam area yang masih dapat dideteksi. Hasilnya adalah berupa daftar nama-nama dari *bluetooth device* yang aktif dan terdeteksi. Pilih salah satu dari nama *bluetooth device* tersebut, sehingga proses berikutnya adalah *service discovery* untuk menemukan layanan berupa pengiriman teks dengan jangkauan sampai 2 kali pengiriman biasa.

1.3 Definisi Istilah dan Singkatan

Definisi, istilah dan singkatan yang digunakan dalam dokumen ini mengacu pada **Appendiks A : Daftar Istilah dan Singkatan.**

1.4 Referensi

Referensi yang digunakan dalam pembuatan dokumen ini adalah:

- GL-FINGERS-03, *Spesifikasi Kebutuhan Perangkat Lunak*, Jurusan Teknik Informatika, Universitas Atma Jaya Yogyakarta, 2005.
- Martin Fowler, Kendall Scott. *UML Distilled – Second Edition*. Addison Wesley. 1999.
- <http://ilmukomputer.com/umum/yanti-uml.php>

1.5 Deskripsi Umum (Overview)

Dokumen SKPL ini dibagi menjadi 4 bab. Bab pertama adalah **Pendahuluan**, yang berisi tentang deskripsi dokumen. Bab kedua adalah **Deskripsi Keseluruhan**, yang berisi penjelasan secara umum mengenai sistem yang akan dikembangkan meliputi fungsi-fungsi dari sistem, karakteristik pengguna, batasan dan asumsi yang diambil dalam

pengembangan perangkat lunak. Bab ketiga adalah **Spesifikasi Rinci Kebutuhan**, yang berisi penjelasan tentang kebutuhan sistem yang akan dikembangkan secara lebih rinci. Bab keempat adalah **Realisasi Use Case**, yang berisi realisasi use case dalam tahap analisis (konseptual), yang akan digunakan sebagai dasar realisasi use case pada tahap desain.

2 Deskripsi Keseluruhan

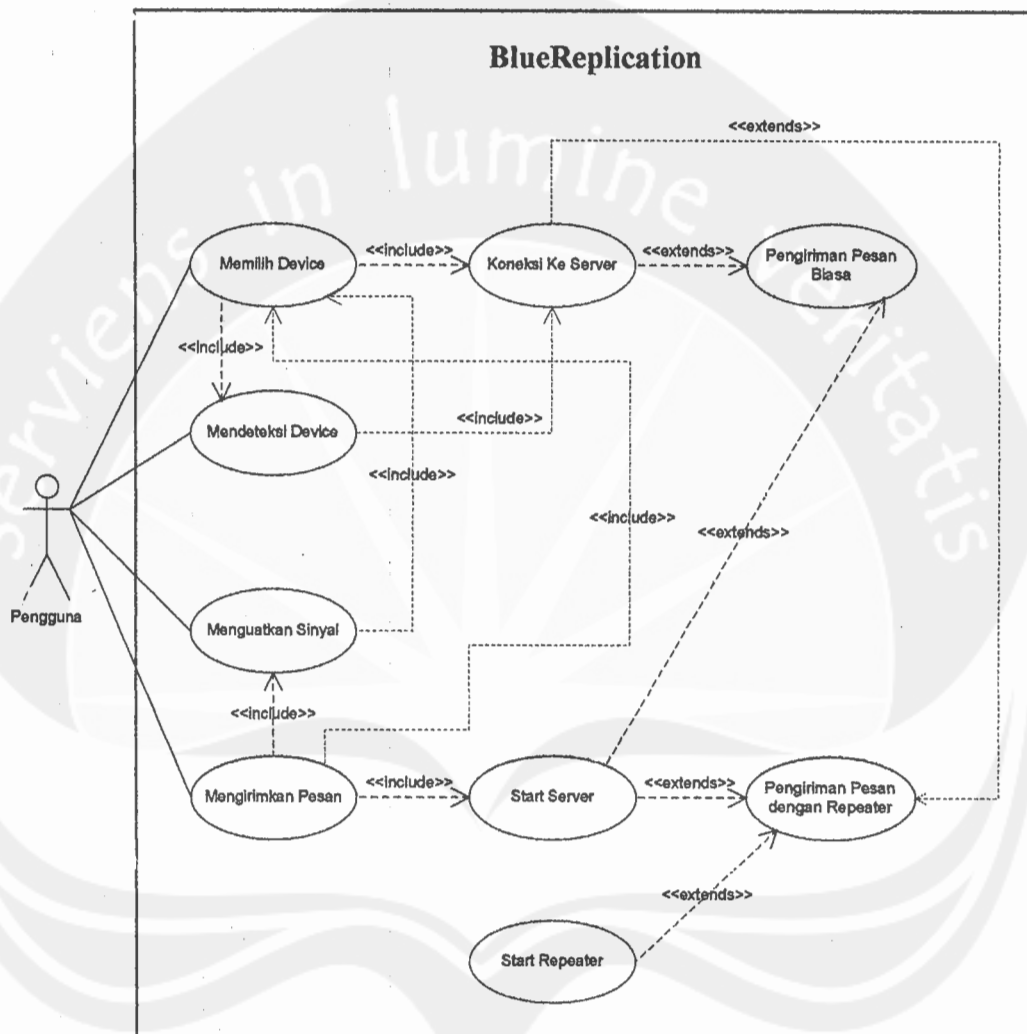
2.1 Perspektif Produk

BlueReplication adalah sebuah aplikasi *mobile* yang dibangun untuk mengirimkan pesan teks dengan jangkauan sampai 2 kali panjang deteksi normal *bluetooth mobile device* (*bluetooth* pada *handphone*). Aplikasi ini dapat membantu mengirimkan teks dengan jangkauan pengiriman mencapai 2 kali pengiriman semula dengan menggunakan perantara *bluetooth mobile device* lain yang dapat mendeteksi *bluetooth mobile device* yang dituju (penerima), agar sinyal dari *bluetooth mobile device* pengirim dapat diterima penerima.

Aplikasi ini akan dibangun dengan menggunakan bahasa pemrograman Java dengan J2ME sebagai toolsnya.

Pengguna akan berinteraksi dengan perangkat lunak melalui antarmuka berbasis GUI, sehingga piranti masukan yang digunakan adalah tombol keypad pada *handphone*.

2.2 Kebutuhan Fungsionalitas Perangkat Lunak



Gambar 2.1 Use Case Diagram

2.2.1 Use Case : Memilih Device

Use case ini digunakan oleh aktor untuk memilih nama dari *bluetooth-enabled device* yang akan dikirimkan pesan teks.

2.2.2 Use case : Mendeteksi Device

Use case ini digunakan oleh aktor untuk mendeteksi *bluetooth-enabled device* disekitarnya.

2.2.3 Use Case : Memperkuat Sinyal

Use case ini digunakan oleh aktor untuk memperkuat kembali sinyal *bluetooth* yang diterima sehingga dapat mendeteksi *bluetooth-enabled device* sampai 2 kali jarak deteksi normal.

2.2.4 Use Case : Mengirimkan Pesan

Use case ini digunakan oleh aktor untuk mengirimkan pesan teks yang dituliskan, ke *bluetooth-enabled device* yang dituju menggunakan *bluetooth*.

2.2.5 Use Case : Koneksi Ke Server

Use case ini digunakan oleh aktor untuk melakukan koneksi ke server caranya dengan melakukan proses *inquiry* untuk mendeteksi *bluetooth-enabled device* disekitarnya. Apabila tidak ada *bluetooth-enabled device* yang terdeteksi maka dapat dilakukan proses pendeteksian *bluetooth-enabled device* kembali, bila ada dipilih salah satu *bluetooth-enabled device* yang terdeteksi.

2.2.6 Use Case : Start Server

Use case ini digunakan oleh aktor untuk membuka layanan *chat*, sehingga *bluetooth-enabled device* yang melakukan proses ini dapat dideteksi oleh *bluetooth-enabled device* lain.

2.2.7 Use Case : Start Repeater

Use case ini digunakan oleh aktor untuk membuka layanan pendeteksian *bluetooth-enabled device* dan layanan untuk mengirimkan pesan ke suatu *bluetooth-enabled device*.

2.2.8 Use Case : Pengiriman Pesan Biasa

Use case ini digunakan oleh aktor untuk memilih pengiriman pesan tanpa menggunakan *bluetooth-enabled device* lain.

2.2.9 Use Case : Pengiriman Pesan dengan Repeater

Use case ini digunakan oleh aktor untuk memilih pengiriman pesan dengan menggunakan *bluetooth-enabled device* lain sebagai perantaranya.

2.3 Karakteristik Sistem

2.3.1 User

User harus memenuhi hal-hal berikut:

- 1) Mengerti tentang penggunaan aplikasi pada *mobile device* termasuk cara meng-*install*-nya.
- 2) Mengerti tentang cara mengirimkan pesan menggunakan *bluetooth* pada *mobile device*.

3 Deskripsi Rinci Kebutuhan

3.1 Spesifikasi Kebutuhan Fungsionalitas

3.1.1 Spesifikasi Use Case : Memilih Device

Use Case ID	UC-BlueReplication-01
Use Case Name	Memilih Device
Use Case Type	Essential
Priority	High
Actors	Pengguna
Description	Use case ini digunakan oleh aktor untuk memilih nama dari <i>bluetooth-enabled device</i> yang akan dikirimkan pesan teks.
Preconditions	<ol style="list-style-type: none">1. Use Case : Koneksi Ke Server sudah dilaksanakan dan paling tidak 1 <i>bluetooth device</i> terdeteksi sehingga tampil daftar nama-nama <i>bluetooth device</i> yang terdeteksi.2. Use Case : Mendeteksi Device sudah dilaksanakan dan paling tidak 1 <i>bluetooth device</i> terdeteksi sehingga tampil daftar nama-nama <i>bluetooth device</i> yang terdeteksi.
Basic Path	<ol style="list-style-type: none">1. Sistem menampilkan antarmuka nama-nama dari <i>bluetooth device</i> yang aktif.2. Pengguna memilih salah satu dari nama <i>bluetooth device</i> yang terdeteksi.
Alternative Paths	Pengguna menekan tombol Back untuk kembali ke tampilan sebelumnya, sehingga sistem menampilkan antarmuka pendeteksian kembali.
Postconditions	Diperoleh salah satu nama <i>bluetooth device</i> dari daftar nama <i>bluetooth device</i> yang tersedia.

Alternative Paths	Pengguna kembali ke menu sebelumnya dengan menekan tombol Back.
Postconditions	Sistem menampilkan pesan yang diterima dari <i>mobile device</i> lain, dan kita dapat menulis untuk dikirimkan ke <i>mobile device</i> lain dengan menekan tombol Send Message.
Exception Paths	-
Extends	Use Case : Pengiriman Pesan Biasa, Use Case : Pengiriman Pesan dengan Repeater
Includes	-
Bussiness Rules	-

Tabel 3.6 Spesifikasi Use Case : Start Server

3.1.7 Spesifikasi Use Case : Start Repeater

Use Case ID	UC-BlueReplication-07
Use Case Name	Start Repeater
Use Case Type	Essential
Priority	High
Actors	Pengguna
Description	Use case ini digunakan oleh aktor untuk membuka layanan pendeteksi <i>bluetooth-enabled device</i> dan layanan untuk mengirimkan pesan ke suatu <i>bluetooth-enabled device</i> .
Preconditions	Use Case : Pengiriman Pesan dengan Repeater sudah dilaksanakan.
Basic Path	Sistem menampilkan antarmuka layanan perantara (repeater) dan membuka layanan dengan nomor ID tertentu untuk membedakannya dengan layanan yang dibuka pada Use Case : Start Server.

Alternative Paths	Pengguna kembali ke menu sebelumnya dengan menekan tombol Back.
Postconditions	Sinyal yang diterima oleh <i>bluetooth mobile device</i> yang pengirim dikuatkan sehingga dapat mencapai <i>bluetooth mobile device</i> penerima.
Exception Paths	Apabila <i>bluetooth device</i> yang dituju tidak dalam keadaan aktif maka ditampilkan pesan kesalahan.
Extends	Use Case : Pengiriman Pesan dengan Repeater
Includes	-
Bussiness Rules	-

Tabel 3.7 Spesifikasi Use Case : Start Repeater

3.1.8 Spesifikasi Use Case : Pengiriman Pesan Biasa

Use Case ID	UC-BlueReplication-08
Use Case Name	Pengiriman Pesan Biasa
Use Case Type	Essential
Priority	High
Actors	Pengguna
Description	Use case ini digunakan oleh aktor untuk memilih pengiriman pesan tanpa menggunakan <i>bluetooth-enabled device</i> lain.
Preconditions	-
Basic Path	<ol style="list-style-type: none"> 1. Sistem menampilkan antarmuka pengiriman pesan biasa. 2. Pengguna memilih melakukan koneksi ke server atau start server.
Alternative Paths	-
Postconditions	Ditampilkan antarmuka, sesuai dengan pilihan yang dipilih pengguna.

Exception Paths	-
Extends	-
Includes	-
Bussiness Rules	-

Tabel 3.8 Spesifikasi Use Case : Pengiriman Pesan Biasa

3.1.9 Spesifikasi Use Case : Pengiriman Pesan dengan Repeater

Use Case ID	UC-BlueReplication-09
Use Case Name	Pengiriman Pesan dengan Repeater
Use Case Type	Essential
Priority	High
Actors	Pengguna
Description	Use case ini digunakan oleh aktor untuk memilih pengiriman pesan dengan menggunakan <i>bluetooth-enabled device</i> lain sebagai perantaranya.
Preconditions	-
Basic Path	1. Sistem menampilkan antarmuka pengiriman pesan dengan repeater. 2. Pengguna memilih melakukan koneksi ke server atau start server atau start repeater.
Alternative Paths	-
Postconditions	Ditampilkan antarmuka, sesuai dengan pilihan yang dipilih pengguna.
Exception Paths	-
Extends	-
Includes	-
Bussiness Rules	-

Tabel 3.9 Spesifikasi Use Case : Pengiriman Pesan dengan Repeater

3.2 Spesifikasi Kebutuhan Non-Fungsionalitas

3.2.1 Kebutuhan Antarmuka Eksternal

Kebutuhan antarmuka eksternal mencakup kebutuhan antarmuka pemakai, antarmuka perangkat keras dan antarmuka perangkat lunak.

3.2.2 Antarmuka Pemakai

Pemakai berinteraksi langsung dengan sistem BlueReplication melalui antarmuka berbasis GUI pada layar *mobile device*. Piranti masukan yang digunakan untuk menuliskan pesan dan memilih nama *bluetooth remote* dengan menggunakan *keypad*. Sedangkan keluaran dari sistem berupa isi pesan akan langsung ditampilkan ke layar *mobile device* bila proses pengiriman berhasil dan bila gagal akan ditampilkan pesan bahwa proses pengiriman gagal dan *error handling* lainnya.

3.2.3 Antarmuka Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan perangkat lunak ini memiliki spesifikasi sebagai berikut:

- 1) Handphone yang mendukung teknologi Java, Java MIDP.

3.2.4 Antarmuka Perangkat Lunak

Perangkat lunak yang digunakan untuk menjalankan perangkat lunak BlueReplication memiliki spesifikasi sebagai berikut:

- 1) Nama : Java MIDP
Sumber : Sun Microsystem
Fungsi : menyediakan landasan berjalannya aplikasi-aplikasi berbasis Java pada perangkat *mobile*.

3.2.5 Persistent Data

Data alamat tujuan pengiriman pesan (*bluetooth remote device name*) disimpan menggunakan Vector, suatu *collection* yang terdapat dalam Java..

4 Realisasi Use Case

4.1 Static Structure Diagram

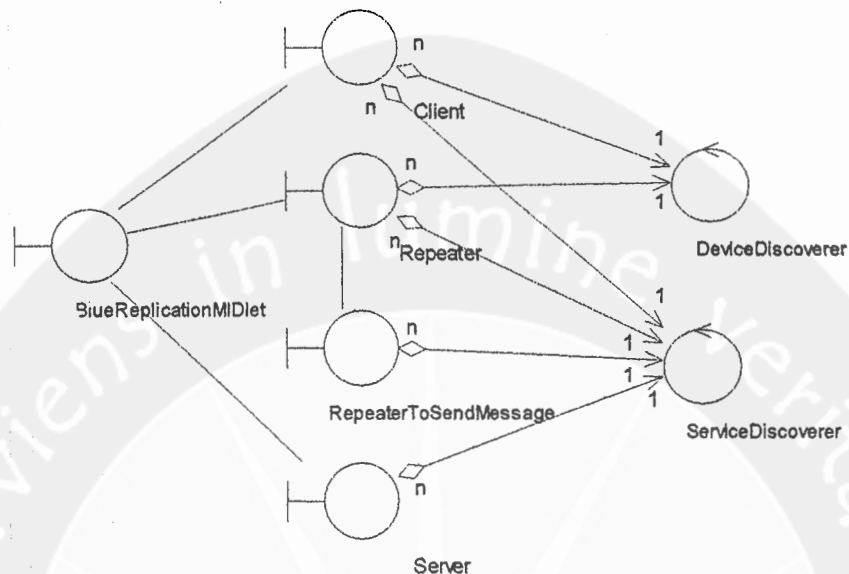
4.1.1 Analysis Class Diagram : Package Dependencies



Gambar 4.1 Analysis Class Diagram : Package Dependencies

Gambar diagram Package Dependencies menggambarkan paket dengan nama BlueReplication, dimana dalam *package* tersebut terdapat *file-file* yang berisi fungsionalitas dalam BlueReplication. *File-file* tersebut antara lain BlueReplicationMIDlet, Client, Server, Repeater, RepeaterToSendMessage, DeviceDiscovery, dan ServiceDiscovery.

4.1.2 Analysis Class Diagram : Package BlueReplication



Gambar 4.2 Analysis Class Diagram : Package BlueReplication

Gambar *package* BlueReplication di atas menjelaskan mengenai hubungan antar *class* dalam *package* BlueReplication.

1. BlueReplicationMIDlet

Class ini menyediakan *user interface* untuk menampilkan pilihan, yaitu *Server*, *Connect To Server*, dan *Repeater*.

2. Client

Class ini menyediakan *user interface* untuk menampilkan *bluetooth device* yang terdeteksi dengan menggunakan *bluetooth device* lain, menerima dan mengirim pesan ke *bluetooth device* tujuan. *Client* dapat menggunakan *DeviceDiscoverer* dan *ServerDiscoverer* lebih dari sekali, begitu pula sebaliknya, *DeviceDiscoverer* dan *ServiceDiscoverer* dapat digunakan oleh banyak *Client*.

3. Repeater

Class ini menyediakan *user interface* untuk mendeteksi *bluetooth device* dan mengirimkan hasil pendeteksiannya ke *bluetooth device* yang melakukan koneksi dengannya. *Repeater* dapat menggunakan *DeviceDiscoverer* dan

ServerDiscoverer lebih dari sekali, begitu pula sebaliknya, DeviceDiscoverer dan ServiceDiscoverer dapat digunakan oleh banyak Repeater.

4. RepeaterToSendMessage

Class ini menyediakan *user interface* fungsionalitas untuk membantu *bluetooth device* yang terkoneksi dengannya (pengirim) mengirimkan pesan ke *bluetooth device* dituju pengirim. RepeaterToSendMessage dapat menggunakan ServerDiscoverer lebih dari sekali, begitu pula sebaliknya, ServiceDiscoverer dapat digunakan oleh banyak RepeaterToSendMessage.

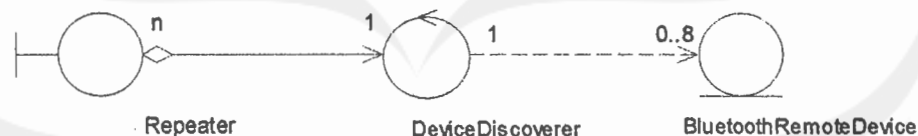
5. DeviceDiscoverer

Class ini menyediakan fungsionalitas untuk mendeteksi *bluetooth device* lain..

6. ServiceDiscoverer

Class ini menyediakan fungsionalitas untuk mencari layanan yang dibuka oleh *bluetooth device* lain.

4.1.3 Analysis Class Diagram : Use Case Mendeteksi Device



Gambar 4.3 Analysis Class Diagram : Use Case Mendeteksi Device

User interface Repeater mempunyai DeviceDiscoverer sebagai kontrol yang menyediakan fungsionalitas untuk mendeteksi *bluetooth remote device* disekitarnya. Repeater mampu mendeteksi *bluetooth remote device* sebanyak 8 *device* dalam keadaan aktif, bila tidak aktif maka tidak ada *bluetooth remote device* yang terdeteksi.

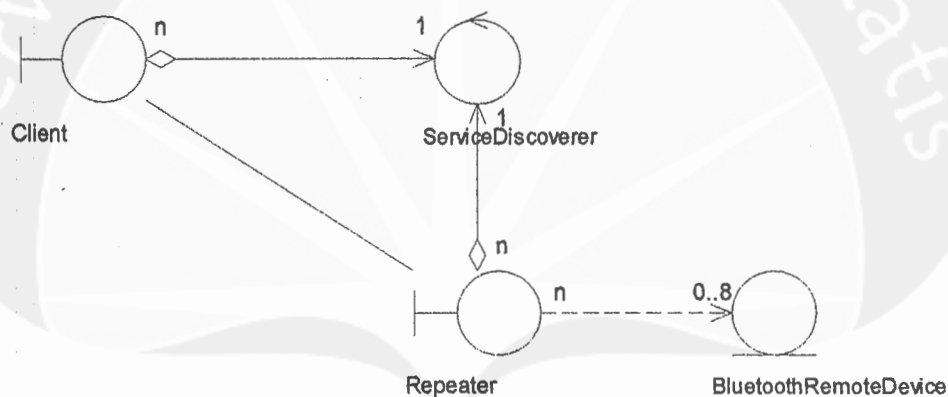
4.1.4 Analysis Class Diagram : Use Case Memilih Device



Gambar 4.4 Analysis Class Diagram : Use Case Memilih Device

Class *Client* sebagai *user interface*-nya dapat memilih satu *bluetooth remote device* yang terdeteksi.

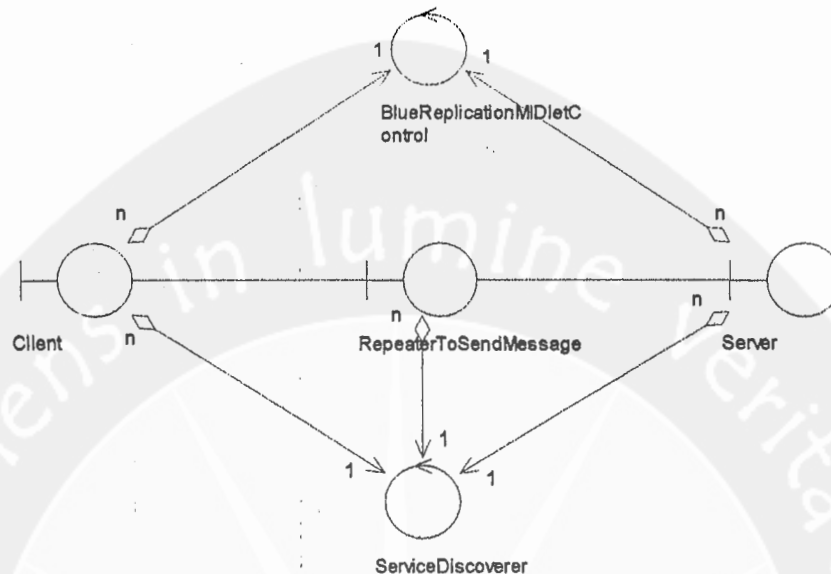
4.1.5 Analysis Class Diagram : Use Case Menguatkan Sinyal



Gambar 4.5 Analysis Class Diagram : Use Case Menguatkan Sinyal

Class *Client* sebagai *user interface*-nya mempunyai Class *ServiceDiscoverer* sebagai kontrol untuk mencari layanan yang sesuai dengan *class Repeater*, lalu *Repeater* akan mencari *bluetooth remote device* disekitarnya. Setelah itu, hasil pendeteksiannya dikirimkan ke *Client*.

4.1.6 Analysis Class Diagram : Use Case Mengirimkan Pesan



Gambar 4.6 Analysis Class Diagram : Use Case Mengirimkan Pesan

Class Client atau class Server dapat saling mengirimkan pesan dengan menggunakan fungsi `sendMessage()` yang terdapat pada *control* `BlueReplicationMIDletControl`. Client dan Server menggunakan `ServiceDiscoverer` untuk menemukan layanan pada `RepeaterToSendMessage` untuk saling mengirimkan pesan. Pesan dikirim melalui `RepeaterToSendMessage` baru kemudian diteruskan ke *device* tujuan.

4.1.7 Analysis Class Diagram : Use Case Koneksi Ke Server



Gambar 4.7 Analysis Class Diagram : Use Case Koneksi Ke Server

Class `BlueReplicationMIDlet` akan menjalankan class Client ketika menu Koneksi Ke Server dipilih. Setelah itu, Client melakukan proses *inquiry* untuk mendeteksi *bluetooth device* disekitarnya.

4.1.8 Analysis Class Diagram : Use Case Start Server



Gambar 4.8 Analysis Class Diagram : Use Case Start Server

Class BlueReplicationMIDlet akan menjalankan *class* Server ketika menu Start Server dipilih. Setelah itu, Server akan membuka layanan untuk mengirim dan menerima pesan.

4.1.9 Analysis Class Diagram : Use Case Start Repeater



Gambar 4.9 Analysis Class Diagram : Use Case Start Repeater

Class BlueReplicationMIDlet akan menjalankan *class* Repeater ketika menu Start Repeater dipilih. Setelah itu, Repeater akan membuka layanan untuk mendeteksi *bluetooth device* disekitarnya ketika terjadi koneksi dengan Client.

4.1.10 Analysis Class Diagram : Use Case Pengiriman Pesan Biasa



Gambar 4.10 Analysis Class Diagram : Use Case Pengiriman Pesan Biasa

Class BlueReplicationMIDlet ketika dijalankan akan menampilkan pilihan Chat dan Chat with Repeater. Pengguna memilih Chat sehingga akan tampil menu Start

Server dan Connect To Server maka pengiriman pesan dilakukan tanpa menggunakan *device* lain (Repeater).

4.1.11 Analysis Class Diagram : Use Case Pengiriman Pesan dengan Repeater



BlueReplicationMIDlet

Gambar 4.8 Analysis Class Diagram : Use Case Pengiriman Pesan dengan Repeater

Class BlueReplicationMIDlet ketika dijalankan akan menampilkan pilihan Chat dan Chat with Repeater. Pengguna memilih Chat with Repeater sehingga akan tampil menu Start Server, Connect To Server, dan Start Repeater maka pengiriman pesan dilakukan menggunakan *device* lain (Repeater).

DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

BLUETOOTH REPEATER APPLICATION

(BlueRepeater)

Disusun oleh:


Marcellinus Willindra

03 07 03913

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta	Nomor Dokumen		Halaman
		DPPL-BlueReplication		1/46
		Revisi		Tgl : 24-02-2007

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	SKY							
Diperiksa oleh								
Disetujui oleh								

Daftar Isi

1	Pendahuluan.....	5
1.1	Tujuan.....	5
1.2	Ruang Lingkup.....	5
1.3	Definisi dan Akronim.....	5
1.4	Referensi.....	5
1.5	Deskripsi Umum Dokumen.....	6
2	Deskripsi Perancangan Arsitektural.....	6
2.1	Deployment Diagram.....	6
2.1.1	Node Pengirim.....	7
2.1.2	Node Perantara.....	7
2.1.3	Node Penerima.....	7
2.2	Design Class.....	7
2.2.1	Pengantar.....	7
2.2.2	Package Dependencies.....	8
2.2.3	Package BlueReplication(BlueReplication.BlueReplication).....	8
2.2.3.1	Class Diagram Package BlueReplication.....	9
2.2.3.2	Class BlueReplication.BlueReplicationMIDlet.....	10
2.2.3.3	Class BlueReplication.Client.....	15
2.2.3.4	Class BlueReplication.DeviceDiscoverer.....	18
2.2.3.5	Class BlueReplication.Repeater.....	19
2.2.3.6	Class BlueReplication.RepeaterToSendMessage.....	21
2.2.3.7	Class BlueReplication.Server.....	23
2.2.3.8	Class BlueReplication.ServiceDiscoverer.....	25
2.3	Realisasi Use Case.....	27
2.3.1	Use Case Memilih Device.....	27
2.3.2	Mendeteksi Device.....	28
2.3.3	Use Case Memperkuat Sinyal.....	29
2.3.4	Use Case Mengirimkan Pesan.....	31
2.3.5	Use Case Koneksi Ke Server.....	33
2.3.6	Use Case Start Server.....	33
2.3.7	Use Case Start Repeater.....	34
2.3.8	Use Case Pengiriman Pesan Biasa.....	35
2.3.9	Use Case Pengiriman Pesan dengan Repeater.....	35
3	Deskripsi Perancangan Persistent Data.....	36
4	Deskripsi Perancangan Antarmuka.....	36
4.1	Use Case : Memilih Device.....	36
4.2	Use Case : Mendeteksi Device.....	38
4.3	Use Case : Memperkuat Sinyal.....	38
4.4	Use Case : Mengirimkan Pesan.....	39
4.5	Use Case : Koneksi Ke Server.....	40
4.6	Use Case : Start Server.....	42
4.7	Use Case : Start Repeater.....	43
4.8	Use Case : Pengiriman Pesan Biasa.....	44
4.9	Use Case : Pengiriman Pesan dengan Repeater.....	45

Daftar Gambar

Gambar 2.1. Deployment Diagram BlueReplication.....	6
Gambar 2.2. Package Dependencies.....	8
Gambar 2.3. Class Diagram Package BlueReplication.....	9
Gambar 2.4. Class BlueReplication.BlueReplicationMIDlet.....	10
Gambar 2.5. Class BlueReplication.Client	15
Gambar 2.6. Class BlueReplication.DeviceDiscoverer	18
Gambar 2.7. Class BlueReplication.Repeater.....	19
Gambar 2.8. Class BlueReplication.RepeaterToSendMessage.....	21
Gambar 2.9. Class BlueReplication Server.....	23
Gambar 2.10. Class BlueReplication.ServiceDiscoverer	25
Gambar 2.11. Sequence Diagram : Use Case Memilih Device.....	27
Gambar 2.12. Sequence Diagram : Use Case Mendeteksi Device.....	28
Gambar 2.13. Sequence Diagram : Use Case Memperkuat Sinyal.....	29
Gambar 2.14. Sequence Diagram : Use Case Mengirimkan Pesan.....	31
Gambar 2.15. Sequence Diagram : Use Case Koneksi Ke Server	33
Gambar 2.16. Sequence Diagram : Use Case Start Server	34
Gambar 2.17. Sequence Diagram : Use Case Start Repeater.....	34
Gambar 2.18. Sequence Diagram : Use Case Pengiriman Pesan Biasa	35
Gambar 2.19. Sequence Diagram : Use Case Pengiriman Pesan dengan Repeater	36
Gambar 4.1. Rancangan Antarmuka Memilih Device.....	37
Gambar 4.2. Rancangan Antarmuka Mendeteksi Device	38
Gambar 4.3. Rancangan Antarmuka Memperkuat Sinyal.....	39
Gambar 4.4. Rancangan Antarmuka Mengirimkan Pesan.....	40
Gambar 4.5. Rancangan Antarmuka Koneksi Ke Server	41
Gambar 4.6. Rancangan Antarmuka Start Server	42
Gambar 4.7. Rancangan Antarmuka Use Start Repeater.....	43
Gambar 4.8. Rancangan Antarmuka Pengiriman Pesan Biasa.....	44
Gambar 4.9. Rancangan Antarmuka Pengiriman Pesan dengan Repeater	45

Daftar Tabel

Tabel Appendix A : Daftar Istilah dan Singkatan	46
---	----

1 Pendahuluan

1.1 Tujuan

Dokumen BlueReplication ini dibuat untuk menyediakan deskripsi lengkap mengenai desain perangkat lunak *Bluetooth Repeater Application* (BlueReplication), versi 1.0. Dokumen ini khususnya ditujukan untuk pembangun perangkat lunak, dan orang lain yang tertarik untuk mengembangkan perangkat lunak ini lebih lanjut sebagai acuan teknis untuk pengembangan perangkat lunak BlueReplication.

1.2 Ruang Lingkup

Dokumen DPPL ini menyediakan deskripsi lengkap perancangan perangkat lunak untuk BlueReplication, versi 1.0. Perancangan ini merupakan arsitektur sistem yang dijelaskan melalui perancangan *class/modul*, detail operasi apa yang akan dilakukan oleh masing-masing *class/modul*, dan layout *database*. Metodologi yang digunakan dalam perancangan adalah metode USDP (*Unified Software Development Process*) dari Rational Software.

1.3 Definisi dan Akronim

Untuk definisi istilah dan singkatan yang digunakan dalam dokumen ini dapat mengacu pada **Apendiks A :Daftar Istilah dan Singkatan**.

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. DPPL-JPAM Grading Tool, *Deskripsi Perancangan Perangkat Lunak*, Program Studi Teknik Informatika – UAJY.
2. DPPL-FINGERS, *Deskripsi Perancangan Perangkat Lunak*, Program Studi Teknik Informatika – UAJY.

3. SKPL-BlueReplication, *Spesifikasi Kebutuhan Perangkat Lunak*, Program Studi Teknik Informatika – UAJY.
4. Martin Fowler, Kendall Scott. *UML Distilled – Second Edition*, Addison Wesley, 1999.
5. <http://ilmukomputer.com/umum/yanti-uml.php>.

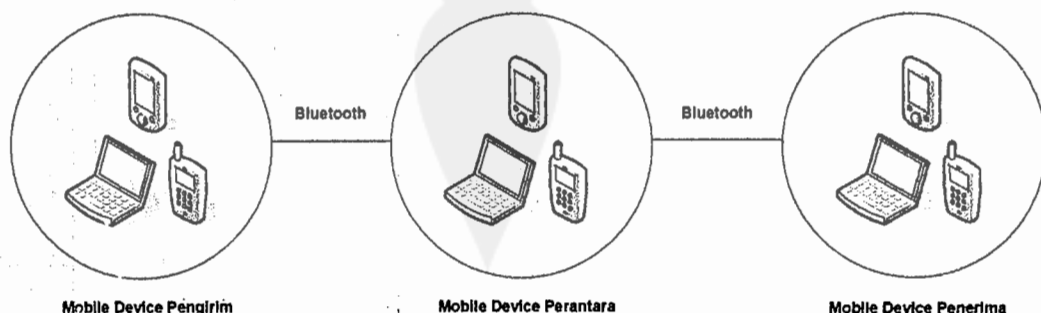
1.5 Deskripsi Umum Dokumen

Dokumen ini terdiri dari empat bab. Bab pertama adalah **Pendahuluan**, yang berisi deskripsi dokumen. Bab kedua adalah **Deskripsi Perancangan Arsitektural**, yang berisi deskripsi arsitektur sistem. Bab ketiga adalah **Deskripsi Perancangan Persistent Data**, yang berisi deskripsi data-data yang akan disimpan pada *persistent storage*. Bab keempat adalah **Deskripsi Perancangan Antarmuka**, yang berisi deskripsi rancangan GUI yang digunakan sistem untuk berinteraksi dengan *user*.

2 Deskripsi Perancangan Arsitektural

2.1 Deployment Diagram

Deployment diagram ini dibuat untuk menunjukkan semua node (lihat **Apendiks A : Daftar Istilah dan Singkatan**) pada sistem, hubungan di antara mereka, dan proses yang akan dijalankan di masing-masing node.



Gambar 2.1 Deployment Diagram BlueReplication

2.1.1 Node Pengirim

Node pengirim merupakan suatu *bluetooth device* yang digunakan untuk mengirimkan pesan teks ke node penerima dengan atau tanpa perantara. Apabila node penerima tidak terdeteksi node pengirim tetapi dapat terdeteksi oleh node perantara, maka digunakan node perantara dalam pengirimannya.

2.1.2 Node Perantara

Node perantara digunakan oleh node pengirim untuk membantu pendeteksian dan pengiriman pesan ke node penerima.

2.1.3 Node Penerima

Node penerima digunakan untuk menerima pesan yang dikirimkan oleh node pengirim melalui atau tanpa node perantara.

2.2 Design Class

2.2.1 Pengantar

Nama *class* yang digunakan dalam *design class* adalah nama *class* yang valid, termasuk nama *package*-nya. Untuk *class-class* yang berasal dari *package* Java juga digunakan nama *class* dengan *package* lengkap, misalnya *java.microedition.midlet*. Untuk penjelasan tipe data yang utuh dapat dilihat pada bagian deskripsi *class*, sedangkan gambar *design class* tidak akan menggunakan nama *package* yang lengkap.

Stereotype yang digunakan dalam *design class* adalah :

- << boundary >>

Boundary class merupakan *class* yang berfungsi untuk menghubungkan sistem dengan *user* di luar sistem.

- << control >>

Control class adalah suatu *class* yang objeknya melakukan interaksi antar sekelompok objek lain. *Control class* biasanya memiliki karakteristik yang spesifik untuk satu use case, dan objek *class* ini biasanya hanya aktif pada realisasi use case.

- << entity >>

Entity class adalah *class* yang bersifat pasif, dalam arti *class* tersebut tidak memulai interaksi dengan *class* lain. *Entity class* ini biasanya merepresentasikan suatu objek yang disimpan dalam *persistent storage*.

Untuk hirarki *design class* berdasarkan *package*, dapat dilihat pada **Apendiks B : Hirarki Design Class berdasarkan Package.**

2.2.2 Package Dependencies

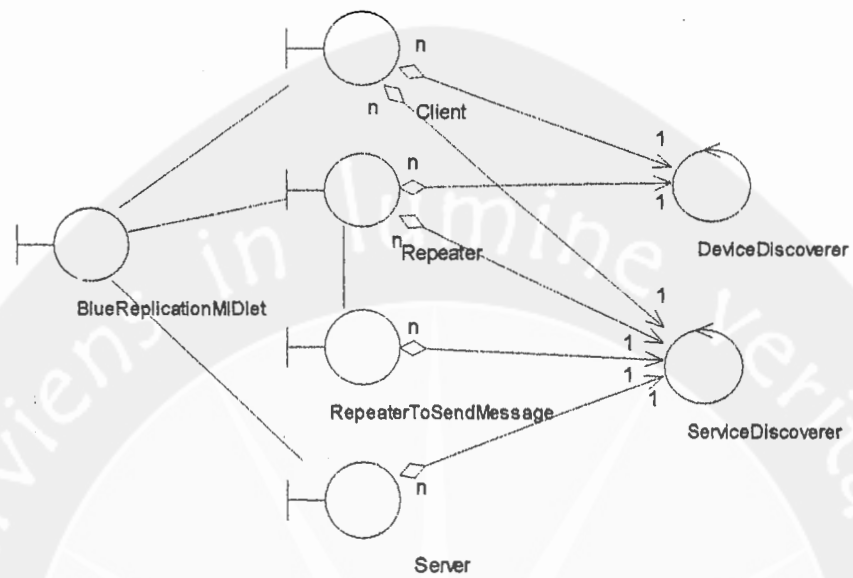


Gambar 2.2 Package Dependencies

2.2.3 Package BlueReplication(BlueReplication.BlueReplication)

Package ini menyediakan class-class yang digunakan oleh program untuk untuk mendeteksi dan mengirimkan pesan ke pengirim melalui atau tanpa perantara.

2.2.3.1 Class Diagram Package BlueReplication



Gambar 2.3 Class Diagram Package BlueReplication

2.2.3.2 Class *BlueReplication.BlueReplicationMIDlet*

BlueReplicationMIDlet
<pre> +UUID_STRING : string = "11112233445566778899AABBCCDDEEFF" -FIRST_MENU[] : string = {"Chat", "Chat with Repeater"} -SECOND_MENU[] : string = {"Server", "Connect"} -THIRD_MENU[] : string = {"Server", "Connect", "Repeater"} -localDevice : LocalDevice -discoveryAgent : DiscoveryAgent -conn : L2CAP Connection -remoteDevice[] : RemoteDevice -status : StringItem -chatUI : Form -welcomeList : List -roleListWithRepeater : List -devices : ChoiceGroup -backCommand : Command -exitCommand : Command -startCommand : Command -connectCommand : Command -selectCommand : Command -sendCommand : Command -incoming : TextField -outgoing : TextField +BlueReplicationMIDlet() +startApp() : void +pauseApp() : void +destroyApp() : void +commandAction(in c : Command, in d : Displayable) : void +exit() : void +getDisplayUI() : Display +errorTime() : int +releaseResource() : void +getCode() : string +getBluetoothReceiver() : string +getMessage() : string +makeRemoteDevice(in bluetoothReceiver : string) : RemoteDevice +readMessage(in conn : StreamConnection) : string +sendMessage(in message : string, in serviceURL : string) : void +doAlert(in msg : string, in timeOut : int) : void +startServer() : void +startReceiver() : void +sendMessage() : void +deviceInquiryFinished(in remoteDevices : RemoteDevice, in message : string) : void +serviceSearchFinished(in serviceRecord : ServiceRecord, in message : string) : void </pre>

Gambar 2.4 Class *BlueReplication.BlueReplicationMIDlet*

Deskripsi

Class ini merepresentasikan *class* yang berisi fungsionalitas yang akan dijalankan aplikasi.

Atribut

- + UUID_STRING : String
Merepresentasikan nomor layanan *bluetooth*
- - FIRST_MENU[] : String

Merepresentasikan menu pilihan

- - SECOND_MENU : String

Merepresentasikan menu pilihan

- - THIRD_MENU : String

Merepresentasikan menu pilihan

- - localDevice : LocalDevice

Merepresentasikan *instance class* dari LocalDevice

- - discoveryAgent : DiscoveryAgent

Merepresentasikan *instance class* dari DiscoveryAgent

- - conn : L2CAPConnection

Merepresentasikan koneksi dengan jenis protokol L2CAP

- - remoteDevice[] : RemoteDevice

Merepresentasikan *instance class* dari RemoteDevice

- - status : StringItem

Merepresentasikan string item

- - chatUI : Form

Merepresentasikan form untuk chat

- - welcomeList : List

Merepresentasikan menu pilihan pembuka saat aplikasi BlueReplication dijalankan

- - roleListWithRepeater : List

Merepresentasikan menu pilihan menu yang dipilih adalah menu chat with repeater

- - devices : ChoiceGroup

Merepresentasikan *device-device* yang ditemukan saat proses inquiry selesai

- - backCommand : Command

Merepresentasikan command untuk kembali ke menu sebelumnya

- - exitCommand : Command

Merepresentasikan command untuk keluar dari aplikasi

- - startCommand : Command

Merepresentasikan command untuk membuka layanan dengan nomor id tertentu.

- - connectCommand : Command

Merepresentasikan command untuk melakukan koneksi ke *device* tujuan

- - selectCommand : Command

Merepresentasikan command untuk memilih pilihan

- - sendCommand : Command

Merepresentasikan command untuk mengirimkan pesan

- - incoming : TextField

Merepresentasikan field untuk menuliskan pesan yang masuk

- - outgoing : TextField

Merepresentasikan field untuk menuliskan pesan yang akan dikirimkan

Method

- + BlueReplicationMIDlet()

Konstruktor class BlueReplication.BlueReplicationMIDlet. Buat instance baru BlueReplication.BlueReplicationMIDlet tanpa atribut terdefinisi.

- + startApp()

Method untuk menjalankan aplikasi MIDlet.

- + pauseApp()

Method untuk menghentikan aplikasi MIDlet.

- + destroyApp()

Method untuk menghancurkan aplikasi MIDlet.

- + commandAction(Command c, Displayable d)

Method untuk melakukan aksi saat penekanan tombol dilakukan.

Parameters:

c – tombol yang dipilih untuk melakukan aksi tertentu.

d – tampilan saat suatu tombol dipilih

- + exit()

Method untuk memanggil method destroyApp().

- + getDisplayUI()

Method untuk mengeset tampilan.

- + `errorTime()` : Integer

Method untuk mengeset lamanya tampilan error.

Returns:

Waktu tampilan error (durasi).

- + `releaseResource()`

Method untuk membebaskan semua resources aplikasi (membersihkan *memory*).

- + `getCode()` : String

Method untuk mengambil kode dari pesan yang dikirimkan.

Returns:

Kode pengiriman pesan.

- + `getBluetoothReceiver()` : String

Method untuk mengambil alamat *bluetooth device* tujuan.

Returns:

Alamat *bluetooth device* penerima atau yang dituju.

- + `getMessage()` : String

Method untuk mengambil isi pesan dari pesan yang dikirimkan.

Returns:

Isi pesan yang dikirimkan.

- + `makeRemoteDevice(String bluetoothReceiver)` : RemoteDevice

Method untuk membuat RemoteDevice dari sebuah alamat *bluetooth device*.

Parameters:

`bluetoothReceiver` – alamat dari *bluetooth device* yang dituju.

Returns:

Menghasilkan alamat *bluetooth device* yang bertipe RemoteDevice.

- + `readMessage(StreamConnection conn)` : String

Method untuk membaca pesan yang dikirimkan lewat StreamConnection.

Parameters:

`conn` – berisi koneksi yang terjadi antar device.

Returns:

Pesan yang dikirimkan lewat StreamConnection.

- + sendMessage(String message, String url)

Method untuk mengirimkan pesan ke alamat url terdefinisi.

Parameters:

message – pesan yang akan dikirimkan.

url – alamat dari *bluetooth device* yang akan dikirimkan pesan.

- + showAlert()

Method untuk menampilkan pesan kesalahan.

- + startServer()

Method untuk menjalankan server.

- + startReceiver()

Method untuk menjalankan tampilan penerimaan dan pengiriman pesan.

- + sendMessage()

Method untuk mengirimkan pesan.

- + deviceInquiryFinished()

Method untuk menampilkan *bluetooth device* yang terdeteksi.

- + serviceSearchFinished(ServiceRecord serviceRecord, String message)

Method untuk melakukan koneksi terhadap layanan yang ditemukan.

2.2.3.3 Class BlueReplication.Client

Client
<pre> +UUID_STRING : string = "F0E0D0C0B0A000908070605040302010" -info : string = "Use Longest Inquiry\nto detect bluetooth devices\ntthat out of your range\nusing one of bluetooth devices\ntthat you selected" -bluetoothDevices : string -bluetoothReceiver : string -midlet : BlueReplicationMIDlet -deviceDiscoverer : DeviceDiscoverer -deviceVector : Vector -localDevice : LocalDevice -discoveryAgent : DiscoveryAgent -remoteDevice[] : RemoteDevice -uuidSet[] : UUID -conn : StreamConnection -devices : ChoiceGroup -devicesLongestInquiry : ChoiceGroup -exitCommand : Command -inquiryCommand : Command -backCommand : Command -nextCommand : Command -infoCommand : Command -longestInquiryCommand : Command -selectCommand : Command -sendCommand : Command -clientUI : Form -detectionUI : Form -devicesUI : Form -infoUI : Form -longestInquiryUI : Form -incoming : TextField -outgoing : TextField +Client(in midlet : BlueReplicationMIDlet) +commandAction(in c : Command, in d : Displayable) : void +doInquiry(in frm : Form) : void +deviceInquiryFinished(in deviceVector : Vector, in message : string) : void +serviceSearchFinished(in message : string, in serviceURL : string) : void </pre>

Gambar 2.5 Class BlueReplication.Client

Deskripsi

Class ini digunakan untuk melakukan koneksi ke server

Atribut

- + UUID_STRING : String
Merepresentasikan nomor layanan *bluetooth*
- - info : String
Merepresentasikan informasi penggunaan tombol Longest Inquiry
- - bluetoothDevices : String
Merepresentasikan penyimpanan alamat *bluetooth device*
- - bluetoothReceiver : String
Merepresentasikan penyimpanan alamat *bluetooth device* penerima
- - midlet : BlueReplicationMIDlet
Merepresentasikan *instance class* dari BlueReplicationMIDlet
- - deviceDiscoverer : DeviceDiscoverer
Merepresentasikan *instance class* dari DeviceDiscoverer

- - deviceVector : Vector

Merepresentasikan penyimpanan alamat *bluetooth device* bertipe Vector

- - localDevice : LocalDevice

Merepresentasikan *instance class* dari LocalDevice

- - discoveryAgent : DiscoveryAgent

Merepresentasikan *instance class* dari DiscoveryAgent

- - remoteDevice[] : RemoteDevice

Merepresentasikan *instance class* dari RemoteDevice

- - uuidSet[] : UUID

Merepresentasikan *instance class* dari UUID

- - conn : StreamConnection

Merepresentasikan koneksi dengan jenis protokol RFCOMM

- - devices : ChoiceGroup

Merepresentasikan *device-device* yang ditemukan saat proses inquiry selesai

- - devicesLongestInquiry : ChoiceGroup

Merepresentasikan *device-device* yang ditemukan saat proses longest inquiry selesai

- - exitCommand : Command

Merepresentasikan command untuk keluar dari aplikasi

- - inquiryCommand : Command

Merepresentasikan command untuk melakukan proses *inquiry*

- - backCommand : Command

Merepresentasikan command untuk kembali ke menu sebelumnya

- - nextCommand : Command

Merepresentasikan command untuk memproses proses selanjutnya

- - infoCommand : Command

Merepresentasikan command untuk menampilkan menu info

- - longestInquiryCommand : Command

Merepresentasikan command untuk melakukan proses *longest inquiry*

- - selectCommand : Command

Merepresentasikan command untuk memilih pilihan pada menu

- - `sendCommand : Command`
Merepresentasikan command untuk mengirimkan pesan
- - `clientUI : Form`
Merepresentasikan tampilan menu client
- - `detectionUI : Form`
Merepresentasikan tampilan menu pendeteksian
- - `devicesUI : Form`
Merepresentasikan tampilan menu *bluetooth device* yang terdeteksi
- - `infoUI : Form`
Merepresentasikan tampilan menu info
- - `longestInquiryUI : Form`
Merepresentasikan tampilan menu longest inquiry
- - `incoming : TextField`
Merepresentasikan field untuk menuliskan pesan yang masuk
- - `outgoing : TextField`
Merepresentasikan field untuk menuliskan pesan yang akan dikirimkan

Method

- + `Client(BlueReplicationMIDlet midlet)`
Konstruktor kelas `BlueReplication.Client`. Buat instance baru `BlueReplication.Client` dengan atribut terdefinisi.
- + `commandAction(Command c, Displayable d)`
Method untuk melakukan aksi saat penekanan tombol dilakukan.

Parameters:

- c – tombol yang dipilih untuk melakukan aksi tertentu.
- d – tampilan saat suatu tombol dipilih

- + `doInquiry(Form frm)`
Method untuk melakukan proses inquiry.

Parameters:

frm – form mana yang akan ditampilkan

- + `deviceInquiryFinished(Vector deviceVector, String message)`

Method untuk menampilkan hasil pendeteksian.

Parameters:

deviceVector – berisi alamat-alamat dari bluetooth device terdeteksi yang disimpan dalam Vector.

message – keterangan mengenai hasil inquiry (sukses atau gagal).

- + serviceSearchFinishes(String message, String serviceURL)

Method untuk mengirimkan pesan ke *bluetooth device* yang dituju.

Parameters:

message – keterangan mengenai hasil pencarian layanan (sukses atau gagal).

serviceURL – alamat service yang ditemukan.

2.2.3.4 Class BlueReplication.DeviceDiscoverer

DeviceDiscoverer
-deviceVector : Vector -midlet : BlueReplicationMIDlet -client : Client -repeater : Repeater
+DeviceDiscoverer(in midlet : BlueReplicationMIDlet) +DeviceDiscoverer(in client : Client, in repeater : Repeater) +deviceDiscovered(in remoteDevice : RemoteDevice, in deviceClass : DeviceClass) : void +inquiryCompleted(in discType : int) : void +servicesDiscovered(in transID : int, in serviceRecord[] : ServiceRecord) : void +serviceSearchCompleted(in transID : int, in respCode : int) : void

Gambar 2.6 Class BlueReplication.DeviceDiscoverer

Deskripsi

Class ini menyediakan fungsionalitas yang digunakan untuk mendeteksi *bluetooth device*.

Atribut

- - deviceVector : Vector

Merepresentasikan *instance class* dari Vector

- - midlet : BlueReplicationMIDlet

Merepresentasikan *instance class* dari BlueReplicationMIDlet

- - client : Client

Merepresentasikan *instance class* dari Client

- - repeater : Repeater

Merepresentasikan *instance class* dari Repeater

Method

- + DeviceDiscoverer (BlueReplicationMIDlet midlet)
Konstruktor class BlueReplication.DeviceDiscoverer. Buat instance baru BlueReplication.DeviceDiscoverer dengan atribut terdefinisi.
- + DeviceDiscoverer (Client client, Repeater repeater)
Konstruktor class BlueReplication.DeviceDiscoverer. Buat instance baru BlueReplication.DeviceDiscoverer dengan atribut terdefinisi.
- + deviceDiscovered (RemoteDevice remoteDevice, DeviceClass deviceClass)

Method ini untuk menyimpan *bluetooth device* yang terdeteksi ke dalam Vector.

- + inquiryCompleted (int discType)

Method ini untuk memanggil fungsi deviceInquiryFinished().

- + serviceDiscovered (int transID, ServiceRecord[] serviceRecord)

Method ini sebagai pengimplementasian DiscoveryListener.

- + serviceSearchCompleted (int transID, int respCode)

Method ini sebagai pengimplementasian DiscoveryListener.

2.2.3.5 Class BlueReplication.Repeater

Repeater
+UUID_STRING : string = A1A2A3A4A5A6A7A8A9A0B1B2B3B4B5B6 -senderName : string -bluetoothDevices : string -midlet : BlueReplicationMIDlet -localDevice : LocalDevice -discoveryAgent : DeviceDiscoverer -remoteDevice : RemoteDevice -uuidSet : UUID
+Repeater(in midlet : BlueReplicationMIDlet) +commandAction(in c : Command, in d : Displayable) : void +startRepeater() : void +doInquiry() : void +deviceInquiryFinished(in deviceVector : Vector, in message : string) : void +serviceSearchFinished(in message : string, in serviceURL : string) : void

Gambar 2.7 Class BlueReplication.Repeater

Deskripsi

Class ini digunakan untuk memperkuat sinyal yang dikirimkan *bluetooth device* pengirim agar sampai ke *bluetooth device* penerima

Atribut

- + UUID_STRING : String
Merepresentasikan nomor layanan *bluetooth*
- + senderName : String
Merepresentasikan alamat *bluetooth device* pengirim
- - bluetoothDevices : String
Merepresentasikan penyimpanan alamat *bluetooth device*
- - midlet : BlueReplicationMIDlet
Merepresentasikan *instance class* dari BlueReplicationMIDlet
- - localDevice : LocalDevice
Merepresentasikan *instance class* dari LocalDevice
- - discoveryAgent : DiscoveryAgent
Merepresentasikan *instance class* dari DiscoveryAgent
- - remoteDevice : RemoteDevice
Merepresentasikan *instance class* dari RemoteDevice
- - uuidSet[] : UUID
Merepresentasikan *instance class* dari UUID

Method

- + Repeater(BlueReplicationMIDlet midlet)
Konstruktor kelas BlueReplication.Repeater. Buat instance baru BlueReplication.Repeater dengan atribut terdefinisi.
- + commandAction(Command c, Displayable d)
Method untuk melakukan aksi saat penekanan tombol dilakukan.

Parameters:

- c – tombol yang dipilih untuk melakukan aksi tertentu.
- d – tampilan saat suatu tombol dipilih

- + doInquiry()
Method untuk melakukan proses inquiry.

- + deviceInquiryFinished(Vector deviceVector, String message)

Method untuk menampilkan hasil pendeteksian.

Parameters:

deviceVector – berisi alamat-alamat dari bluetooth device terdeteksi yang disimpan dalam Vector.

message – keterangan mengenai hasil inquiry (sukses atau gagal).

- + serviceSearchFinishes(String message, String serviceURL)

Method untuk menampilkan layanan yang ditemukan pada saat proses pencarian layanan selesai.

Parameters:

message – keterangan mengenai hasil pencarian layanan (sukses atau gagal).

serviceURL – alamat service yang ditemukan.

2.2.3.6 Class BlueReplication.RepeaterToSendMessage

RepeaterToSendMessage
+UUID_STRING : string = "B1B2B3B4B5B6B7B8B9B0C1C2C3C4C5C6"
-bluetoothReceiver : string
-bluetoothSender : string
-text : string
-code : string
-midlet : BlueReplicationMIDlet
-serviceDiscoverer : ServiceDiscoverer
-localDevice : LocalDevice
-discoveryAgent : DiscoveryAgent
-uuidSet[] : UUID
-conn : StreamConnection
-exitCommand : Command
-repeaterToSendMessageUI : Form
+repeaterToSendMessage(in midlet : BlueReplicationMIDlet)
+commandAction(in c : Command, in d : Displayable) : void
+sendMessage() : void
+serviceSearchFinished(in message : string, in serviceURL : string) : void

Gambar 2.8 Class BlueReplication.RepeaterToSendMessage

Deskripsi

Class ini digunakan untuk membantu pengiriman pesan ke *bluetooth device* tujuan

Atribut

- + UUID_STRING : String

Merepresentasikan nomor layanan *bluetooth*

- - `bluetoothReceiver` : String

Merepresentasikan penyimpanan alamat *bluetooth device* penerima

- - `bluetoothSender` : String

Merepresentasikan penyimpanan alamat *bluetooth device* pengirim

- - `text` : String

Merepresentasikan isi pesan yang dikirim

- - `code` : String

Merepresentasikan code pengiriman pesan

- - `midlet` : BlueReplicationMIDlet

Merepresentasikan *instance class* dari BlueReplicationMIDlet

- - `serviceDiscoverer` : ServiceDiscoverer

Merepresentasikan *instance class* dari ServiceDiscoverer

- - `localDevice` : LocalDevice

Merepresentasikan *instance class* dari LocalDevice

- - `discoveryAgent` : DiscoveryAgent

Merepresentasikan *instance class* dari DiscoveryAgent

- - `uuidSet[]` : UUID

Merepresentasikan *instance class* dari UUID

- - `conn` : StreamConnection

Merepresentasikan koneksi dengan jenis protokol RFCOMM

- - `exitCommand` : Command

Merepresentasikan command untuk keluar dari aplikasi

- - `repeaterToSendMessageUI` : Form

Merepresentasikan tampilan menu repeater untuk pengiriman pesan

Method

- + `RepeaterToSendMessage(BlueReplicationMIDlet midlet)`

Konstruktor kelas `BlueReplication.Client`. Buat instance baru `BlueReplication.RepeaterToSendMessage` dengan atribut terdefinisi.

- + `commandAction(Command c, Displayable d)`

Method untuk melakukan aksi saat penekanan tombol dilakukan.

Parameters:

c – tombol yang dipilih untuk melakukan aksi tertentu.

d – tampilan saat suatu tombol dipilih

- + sendMessage ()

Method untuk mengirimkan pesan.

- + serviceSearchFinished(String message, String serviceURL)

Method untuk melakukan koneksi terhadap layanan yang ditemukan.

2.2.3.7 Class BlueReplication.Server

Server
+UUID_STRING : string = "123456789ABCDEF"
-bluetoothReceiver : string
-midlet : BlueReplicationMIDlet
-localDevice : LocalDevice
-discoveryAgent : DiscoveryAgent
-uuidSet[] : UUID
-conn : StreamConnection
-mainUI : Form
-exitCommand : Command
-sendCommand : Command
-incoming : TextField
-outgoing : TextField
+Server(in midlet : BlueReplicationMIDlet)
+commandAction(in c : Command, in d : Displayable) : void
+startServer() : void
+startReceiver() : void
+serviceSearchFinished(in message : string, in serviceURL : string) : void

Gambar 2.9 Class BlueReplication.Server

Deskripsi

Class ini digunakan untuk.

Atribut

- + UUID_STRING : String

Merepresentasikan nomor layanan *bluetooth*

- - bluetoothReceiver : String

Merepresentasikan penyimpanan alamat *bluetooth device* penerima

- - midlet : BlueReplicationMIDlet

Merepresentasikan *instance class* dari BlueReplicationMIDlet

- - localDevice : LocalDevice

Merepresentasikan *instance class* dari LocalDevice

- - discoveryAgent : DiscoveryAgent

Merepresentasikan *instance class* dari DiscoveryAgent

- - uuidSet[] : UUID

Merepresentasikan *instance class* dari UUID

- - mainUI : Form

Merepresentasikan tampilan menu repeater untuk pengiriman pesan

- - exitCommand : Command

Merepresentasikan command untuk keluar dari aplikasi

- - sendCommand : Command

Merepresentasikan command untuk mengirimkan pesan

- - incoming : TextField

Merepresentasikan field untuk menuliskan pesan yang masuk

- - outgoing : TextField

Merepresentasikan field untuk menuliskan pesan yang akan dikirimkan

Method

- + Server(BlueReplicationMIDlet midlet)

Konstruktor kelas BlueReplication. Server. Buat instance baru BlueReplication. Server dengan atribut terdefinisi.

- + commandAction(Command c, Displayable d)

Method untuk melakukan aksi saat penekanan tombol dilakukan.

Parameters:

c -- tombol yang dipilih untuk melakukan aksi tertentu.

d -- tampilan saat suatu tombol dipilih

- + startServer()

Method untuk menjalankan server.

- + startReceiver()

Method untuk menjalankan tampilan penerimaan dan pengiriman pesan.

- + serviceSearchFinished(String message, String serviceURL)

Method untuk melakukan koneksi terhadap layanan yang ditemukan.

2.2.3.8 Class *BlueReplication.ServiceDiscoverer*

ServiceDiscoverer
+SERVICE_NAME : string -serviceURL : string -serviceRecord : ServiceRecord -midlet : BlueReplicationMIDlet -repeater : Repeater -repeaterToSendMessage : RepeaterToSendMessage -server : Server +serviceDiscoverer(In midlet : BlueReplicationMIDlet) +serviceDiscoverer(In client : Client, In repeater : Repeater, In repeaterToSendMessage : RepeaterToSendMessage, In server : Server) +deviceDiscovered(In remoteDevice : RemoteDevice, In deviceClass : DeviceClass) : void +InquiryCompleted(In discType : Int) : void +serviceDiscovered(In transID : Int, In servRecord : ServiceRecord) : void +serviceSearchCompleted(In transID : Int, In responseCode : Int) : void

Gambar 2.10 Class *BlueReplication.ServiceDiscoverer*

Deskripsi

Class ini menyediakan fungsionalitas yang digunakan untuk mencari layanan yang dibuka oleh *bluetooth device*.

Atribut

- - SERVICE_NAME : String
Merepresentasikan nama layanan yang digunakan untuk *chat* tanpa repeater
- - serviceURL: String
Merepresentasikan penyimpanan alamat URL *bluetooth device*.
- - serviceRecord : ServiceRecord
Merepresentasikan *instance class* dari ServiceRecord
- - midlet : BlueReplicationMIDlet
Merepresentasikan *instance class* dari BlueReplicationMIDlet
- - repeater : Repeater
Merepresentasikan *instance class* dari Repeater
- - repeaterToSendMessage : RepeaterToSendMessage
Merepresentasikan *instance class* dari RepeaterToSendMessage
- - server : Server
Merepresentasikan *instance class* dari Server

Method

- + ServiceDiscoverer (BlueReplicationMIDlet midlet)

Konstruktor class BlueReplication. ServiceDiscoverer. Buat instance baru BlueReplication. ServiceDiscoverer dengan atribut terdefinisi.

- + ServiceDiscoverer (Client client, Repeater repeater, RepeaterToSendMessage repeaterToSendMessage, Server server)

Konstruktor class BlueReplication. ServiceDiscoverer. Buat instance baru BlueReplication. ServiceDiscoverer dengan atribut terdefinisi.

- + deviceDiscovered(RemoteDevice remoteDevice, DeviceClass deviceClass)

Method ini sebagai pengimplementasian DiscoveryListener.

- + inquiryCompleted(int discType)

Method ini sebagai pengimplementasian DiscoveryListener.

- + serviceDiscovered(int transID, ServiceRecord[] serviceRecord)

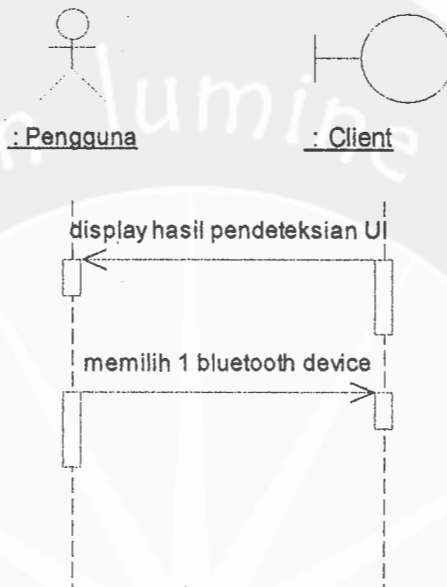
Method ini digunakan untuk mencari layanan yang dibuka *bluetooth device*.

- + serviceSearchCompleted(int transID, int respCode)

Method ini digunakan untuk memanggil method `serviceSearchFinished()`.

2.3 Realisasi Use Case

2.3.1 Use Case Memilih Device

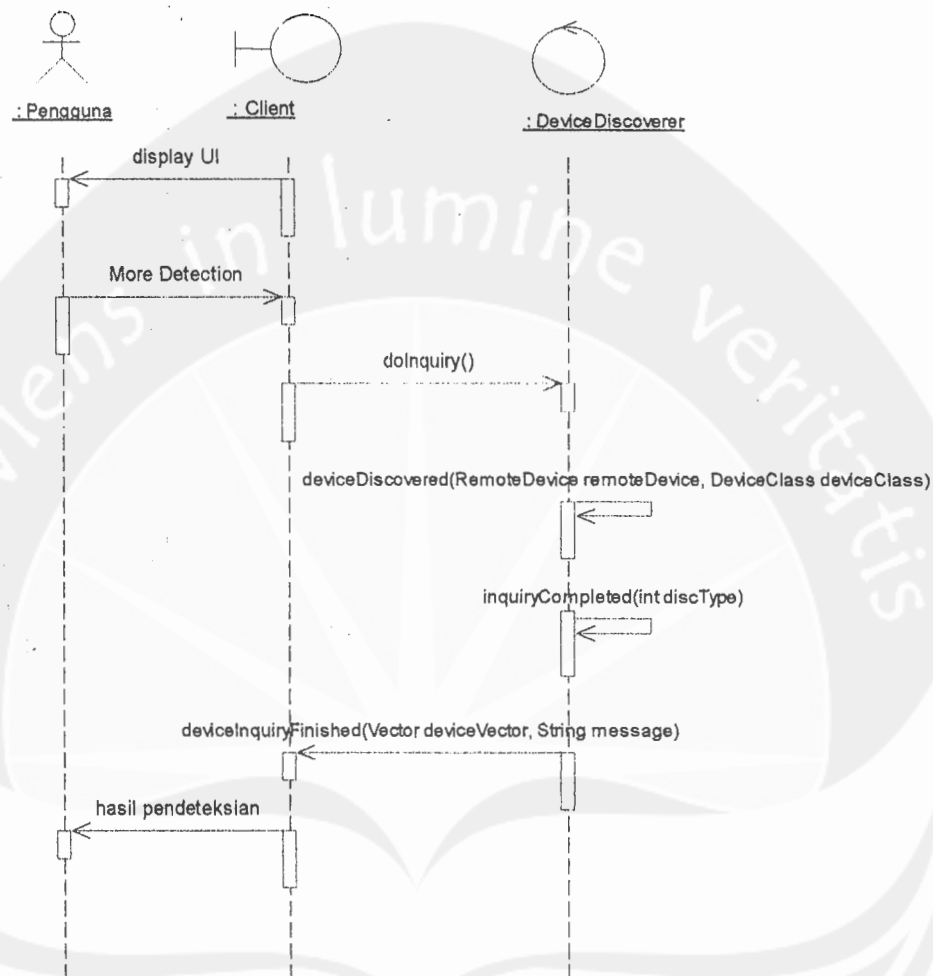


Gambar 2.11 Sequence Diagram : Use Case Memilih Device

Flow of events :

1. Sistem menampilkan hasil pendeteksian *bluetooth device* ke pengguna.
2. Pengguna memilih salah satu *bluetooth device* yang terdeteksi.

2.3.2 Mendeteksi Device



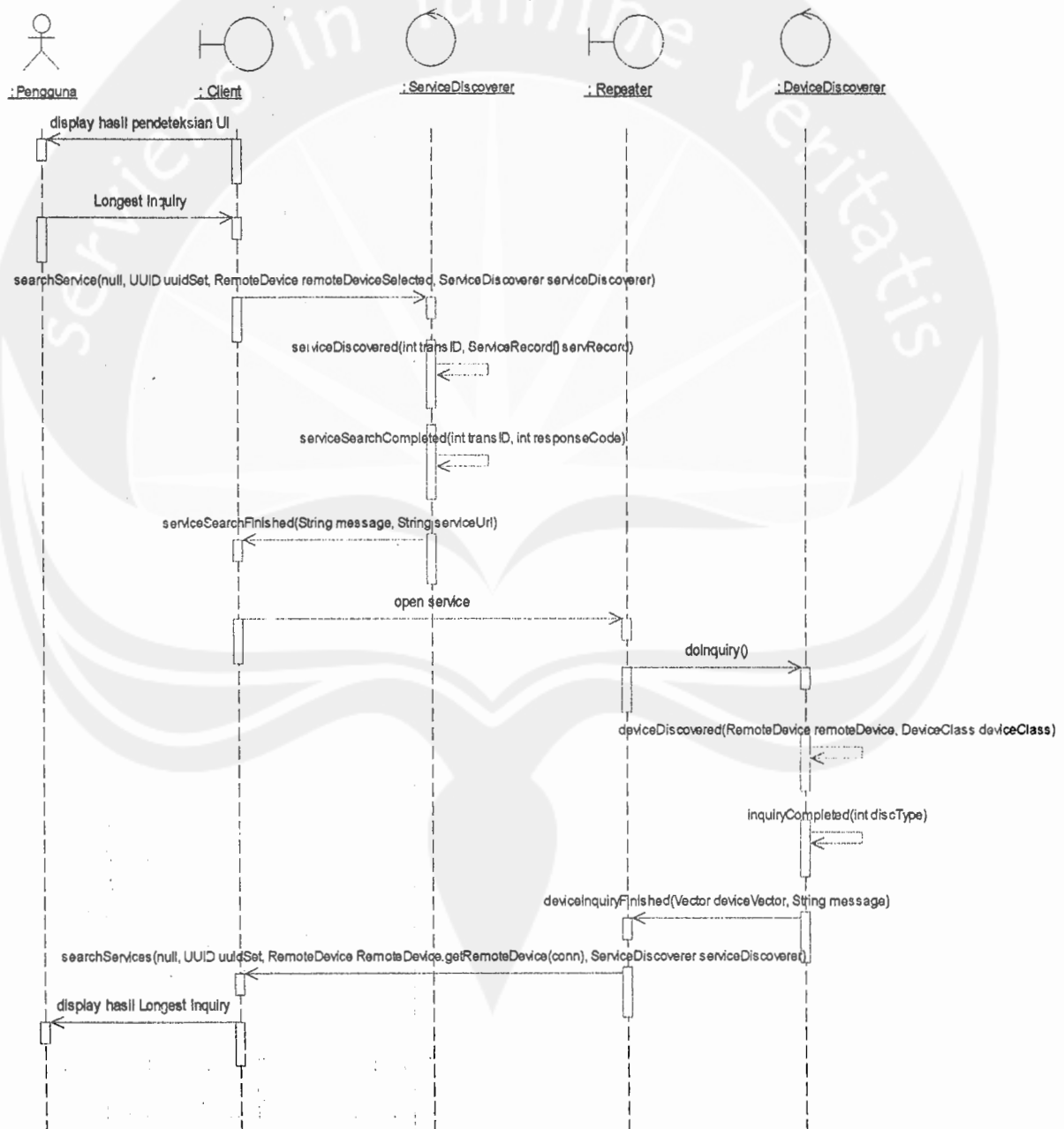
Gambar 2.12 Sequence Diagram : Use Case Mendeteksi Device

Flow of events :

1. Sistem menampilkan menu pendeteksian kembali.
2. Pengguna menekan tombol More Detection.
3. Sistem memanggil fungsi `doInquiry()` untuk mendeteksi Device menggunakan `DeviceDiscoverer`.
4. Pemanggilan fungsi `doInquiry()` memicu sistem untuk memanggil `deviceDiscovered(RemoteDevice remoteDevice, DeviceClass deviceClass)` dan `inquiryCompleted(int discType)` untuk mendeteksi device.

5. Sistem memanggil fungsi `deviceInquiryFinished(Vector deviceVector, String message)` untuk mengirimkan hasil pendeteksian.
6. Sistem menampilkan hasil pendeteksiannya ke pengguna.

2.3.3 Use Case Menguatkan Sinyal

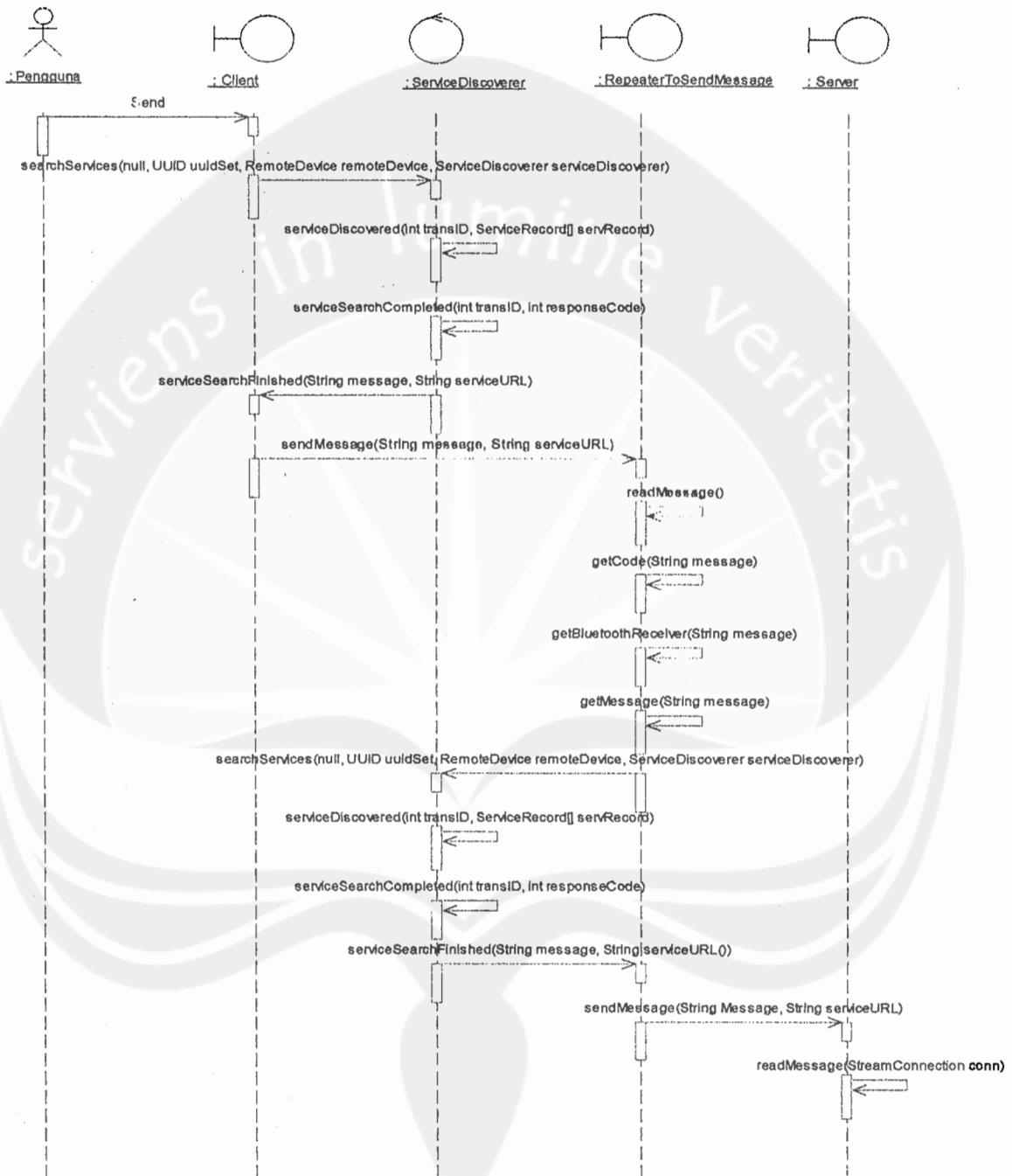


Gambar 2.13 Sequence Diagram : Use Case Menguatkan Sinyal

Flow of events :

1. Sistem menampilkan hasil pendeteksian *bluetooth device* ke pengguna.
2. Pengguna menekan tombol Longest Inquiry.
3. Sistem mencari layanan dengan fungsi `searchService` yang sesuai dengan nomor ID yang masukkan dalam pencarian layanan menggunakan `ServiceDiscoverer`, dalam hal ini Repeater yang membuka layanan yang dicari.
4. Sistem memanggil fungsi `serviceDiscovered()` dan `serviceSearchCompleted()` untuk mencari layanan.
5. Setelah selesai, sistem memanggil `serviceSearchFinished()`. Dalam fungsi ini terkandung perintah untuk membuka layanan yang dibuka Repeater.
6. Client membuka layanan dengan nomor ID tertentu untuk menerima hasil pendeteksian *bluetooth device* menggunakan Longest Inquiry
7. Layanan yang dibuka pada Repeater, memicu sistem untuk melakukan pendeteksian dengan memanggil fungsi `doInquiry()` menggunakan `DeviceDiscoverer` untuk membantu menemukan *bluetooth device*.
8. Sistem memanggil fungsi `deviceDiscovered(RemoteDevice remoteDevice, DeviceClass deviceClass)` dan `inquiryCompleted(int discType)` untuk mendeteksi device.
9. Sistem memanggil fungsi `deviceInquiryFinished(Vector deviceVector, String message)` untuk mengirimkan hasil pendeteksian.
10. Setelah pencarian *bluetooth device* selesai, Repeater mencari layanan yang dibuka oleh Client untuk mengirimkan hasil pendeteksian.
11. Sistem menampilkan hasil pendeteksian menggunakan Longest Inquiry.

2.3.4 Use Case Mengirimkan Pesan



Gambar 2.14 Sequence Diagram : Use Case Mengirimkan Pesan

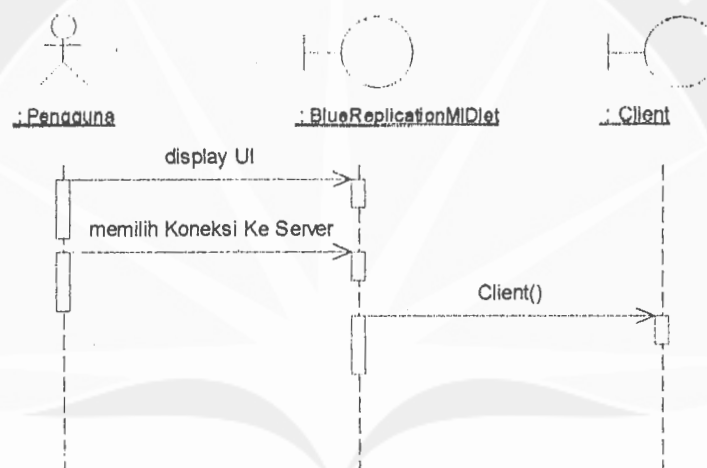
Flow of events :

1. Pengguna menekan tombol Send.

2. Sistem mencari layanan dengan fungsi `searchService()` yang sesuai dengan nomor ID yang masukkan dalam pencarian layanan menggunakan `ServiceDiscoverer`, dalam hal ini `RepeaterToSendMessage` yang membuka layanan yang dicari.
3. Sistem memanggil fungsi `serviceDiscovered()` dan `serviceSearchCompleted()` untuk mencari layanan.
4. Setelah selesai, sistem memanggil `serviceSearchFinished()` untuk memberitahukan ke Client alamat url dari layanan yang dibuka `RepeaterToSendMessage`.
5. Client memanggil fungsi `sendMessage` untuk mengirimkan pesan ke Server. Dalam fungsi ini, terkandung perintah untuk membuka layanan yang dibuka `RepeaterToSendMessage`.
6. Setelah Client dan `RepeaterToSendMessage` terkoneksi, memicu sistem untuk memanggil fungsi `readMessage()` untuk membaca pesan yang dikirimkan Client.
7. Sistem menggunakan fungsi `getCode()` untuk mengambil kode yang dikirimkan Client lewat pesan.
8. Sistem menggunakan fungsi `getBluetoothReceiver()` untuk mengambil alamat bluetooth device yang dituju.
9. Sistem menggunakan fungsi `getMessage()` untuk mengambil isi pesan yang dikirimkan Client.
10. Sistem mencari layanan dengan fungsi `searchService()` yang sesuai dengan nomor ID yang masukkan dalam pencarian layanan menggunakan `ServiceDiscoverer`, dalam hal ini Server yang membuka layanan yang dicari.
11. Sistem memanggil fungsi `serviceDiscovered()` dan `serviceSearchCompleted()` untuk mencari layanan.
12. Setelah selesai, sistem memanggil `serviceSearchFinished()` untuk memberitahukan ke `RepeaterToSendMessage` alamat url dari layanan yang dibuka Server.

13. RepeaterToSendMessage memanggil fungsi sendMessage untuk mengirimkan pesan ke Server. Dalam fungsi ini, terkandung perintah untuk membuka layanan yang dibuka Server.
14. Setelah RepeaterToSendMessage dan Server terkoneksi, memicu sistem untuk memanggil fungsi readMessage() untuk membaca pesan yang dikirimkan RepeaterToSendMessage.

2.3.5 Use Case Koneksi Ke Server

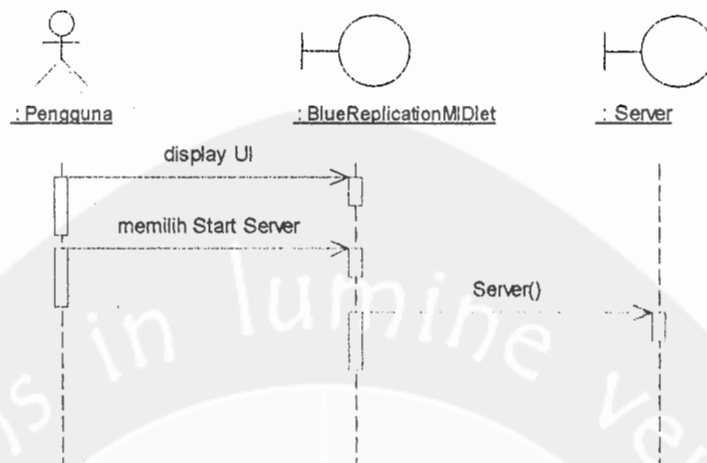


Gambar 2.15 Sequence Diagram : Use Case Koneksi Ke Server

Flow of events :

1. Sistem menampilkan antarmuka ke pengguna.
2. Pengguna memilih pilihan Connect.
3. Sistem akan menampilkan antarmuka Client dengan memanggil konstruktor class Client().

2.3.6 Use Case Start Server

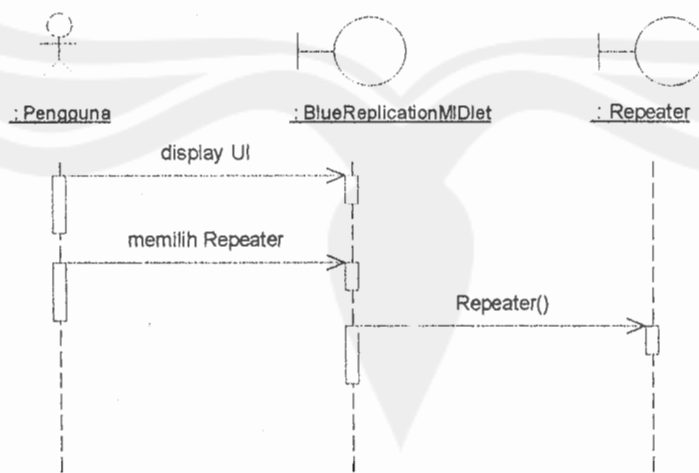


Gambar 2.16 Sequence Diagram : Use Case Start Server

Flow of events :

1. Sistem menampilkan antarmuka ke pengguna.
2. Pengguna memilih pilihan Server.
3. Sistem akan menampilkan antarmuka Server dengan memanggil konstruktor class Server().

2.3.7 Use Case Start Repeater



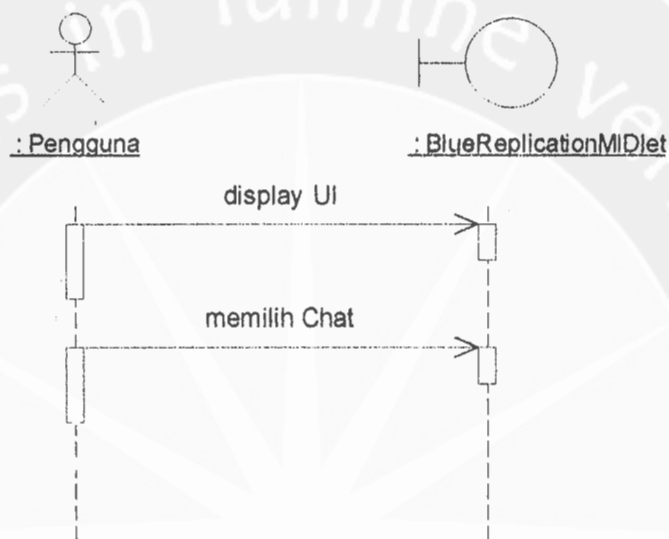
Gambar 2.17 Sequence Diagram : Use Case Start Repeater

Flow of events :

1. Sistem menampilkan antarmuka ke pengguna.

2. Pengguna memilih pilihan Repeater.
3. Sistem akan menampilkan antarmuka Repeater dengan memanggil konstruktor class Repeater ().

2.3.8 Use Case Pengiriman Pesan Biasa

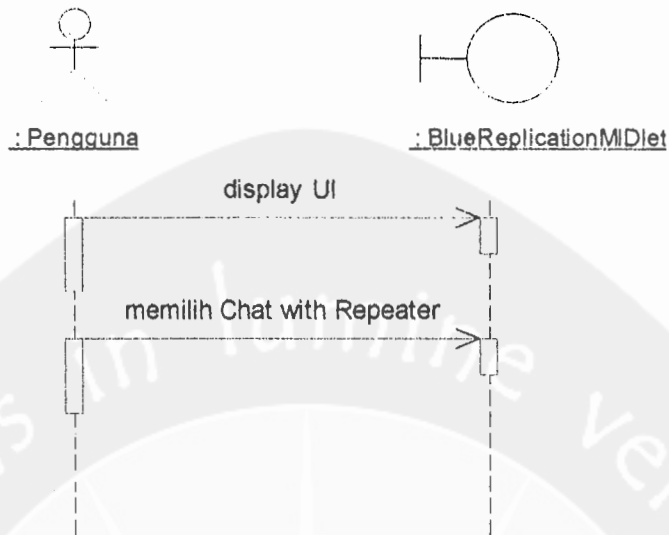


Gambar 2.18 Sequence Diagram : Use Case Pengiriman Pesan Biasa

Flow of events :

1. Sistem menampilkan antarmuka ke pengguna.
2. Pengguna memilih pilihan Chat.

2.3.9 Use Case Pengiriman Pesan dengan Repeater



Gambar 2.19 Sequence Diagram : Use Case Pengiriman Pesan dengan Repeater

Flow of events :

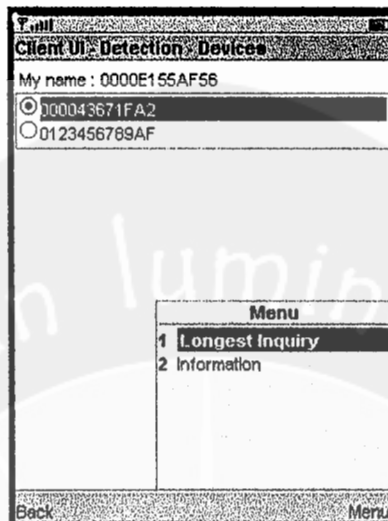
1. Sistem menampilkan antarmuka ke pengguna.
2. Pengguna memilih pilihan Chat with Repeater.

3 Deskripsi Perancangan Persistent Data

Aplikasi BlueReplication ini tidak menggunakan database untuk penyimpanan data.

4 Deskripsi Perancangan Antarmuka

4.1 Use Case : Memilih Device



Gambar 4.1 Rancangan Antarmuka Memilih Device

Deskripsi

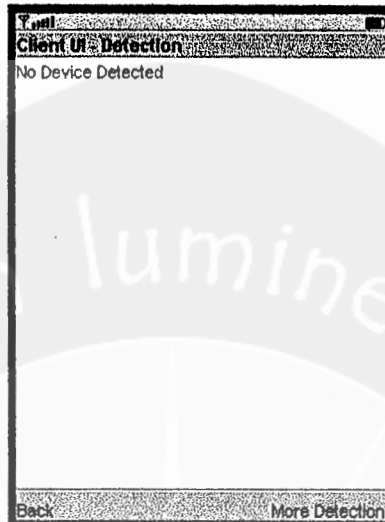
- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.Client**.
- ChoiceGroup digunakan agar pengguna dapat memilih salah satu *device* yang ditampilkan.
- Tombol Menu digunakan memilih aksi apa yang akan dilakukan, apakah akan melakukan Longest Inquiry dengan memilih pilihan Longest Inquiry atau melihat informasi penggunaan tombol Longest Inquiry dengan memilih pilihan Information.

Event

Urutan aksi yang terjadi :

1. Sistem menampilkan *bluetooth device* yang terdeteksi.
2. Pengguna memilih salah satu *bluetooth device* yang ditampilkan.
3. Pengguna menekan tombol Menu.
4. Jika pengguna memilih Longest Inquiry pada Menu maka dilakukan proses Longest Inquiry pada *bluetooth device* yang dipilih, jika pengguna memilih Information maka akan ditampilkan informasi mengenai penggunaan tombol Longest Inquiry.

4.2 Use Case : Mendeteksi Device



Gambar 4.2 Rancangan Antarmuka Mendeteksi Device

Deskripsi

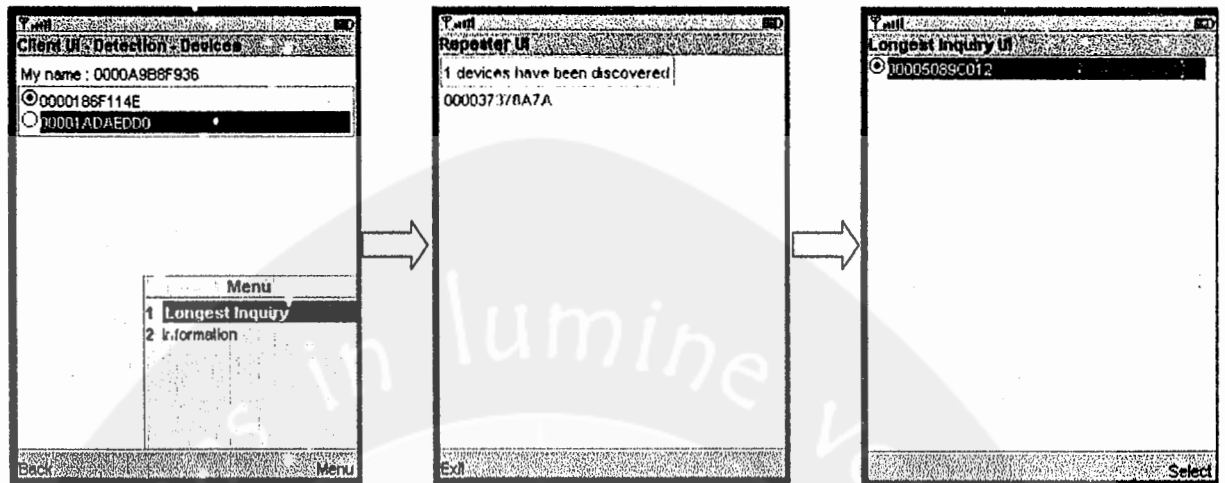
- Rancangan antarmuka ini diimplementasikan pada **Class BlueReplication.Cliet**.
- Tombol back digunakan untuk kembali ke halaman sebelumnya
- Tombol More Detection digunakan untuk mendeteksi kembali bluetooth device

Event

Urutan aksi yang terjadi :

1. Setelah proses pencarian *bluetooth device* selesai, sistem akan menampilkan hasil pendeteksiannya.
2. Pengguna menekan tombol More Detection untuk mendeteksi *bluetooth device* kembali.

4.3 Use Case : Memperkuat Sinyal



Gambar 4.5 Rancangan Antarmuka Menguatkan Sinyal

Deskripsi

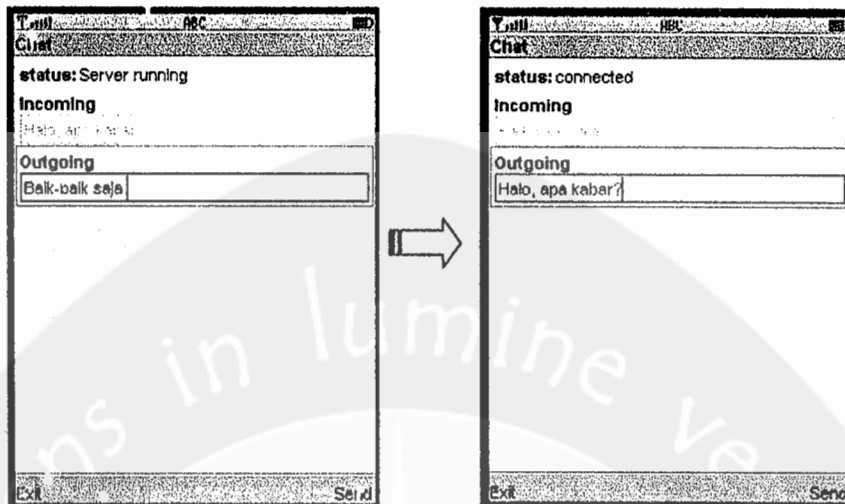
- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.Client** dan **BlueReplication.Repeater**.
- Tombol Longest Inquiry digunakan untuk melakukan proses Longest Inquiry.
- Tombol Back digunakan untuk kemali ke halaman sebelumnya.

Event

Urutan aksi yang terjadi :

1. Pengguna memilih Longest Inquiry pada device pertama (Gambar 4.5 paling kiri).
2. Hal tersebut menyebabkan repeater bereaksi untuk mendeteksi bluetooth device disekitarnya (Gambar 4.5 tengah).
3. Hasil pendeteksian repeater dikirim ke pengirim (Gambar 4.5 paling kanan).

4.4 Use Case : Mengirimkan Pesan



(a) Server

(b) Client

Gambar 4.6 Rancangan Antarmuka Mengirimkan Pesan

Deskripsi

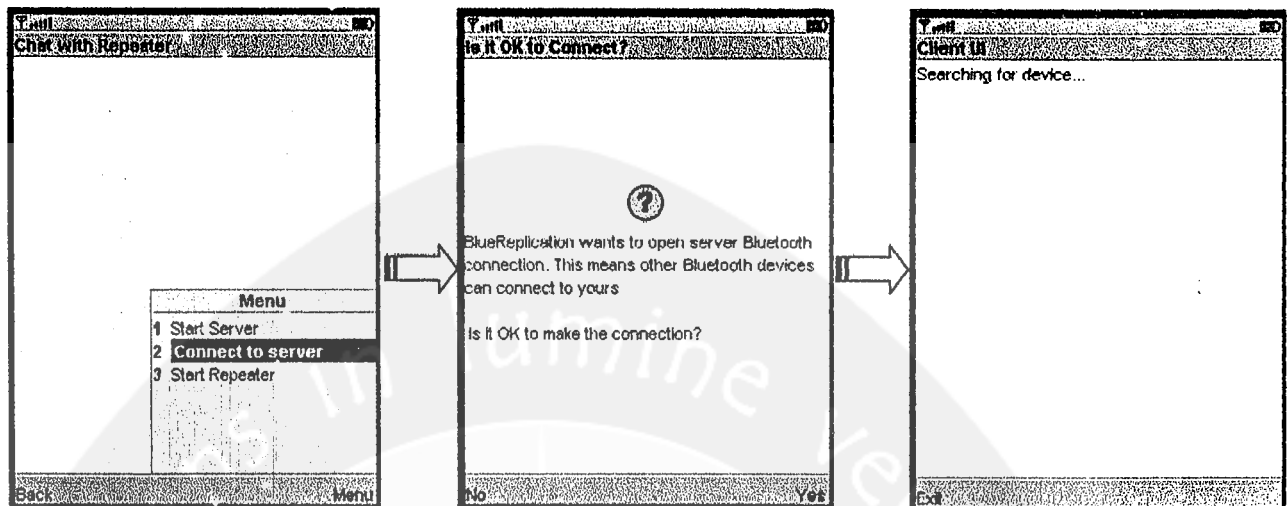
- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.Client** dan **BlueReplication.Server**.
- Kolom incoming untuk menerima pesan yang masuk.
- Kolom outgoing untuk mengirimkan pesan.
- Tombol Send digurakan untuk mengirimkan pesan.
- Tombol Exit digunakan untuk keluar dari aplikasi.

Event

Urutan aksi yang terjadi :

1. Client atau Server yang akan mengirimkan pesan, menuliskan pesan yang akan dikirimkan pada kolom outgoing.
2. Pesan akan dikirimkan melalui repeater.
3. Pesan yang sampai akan tampil pada kolom incoming.

4.5 Use Case : Koneksi Ke Server



Gambar 4.7 Rancangan Antarmuka Koneksi Ke Server

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.BlueReplicationMIDlet** dan **BlueReplication.Client**.
- Menu pilihan Server untuk menjadikan device tersebut sebagai server dari aplikasi.
- Menu pilihan Connect untuk menjadikan device tersebut sebagai client dari aplikasi.
- Menu pilihan Repeater untuk menjadikan device tersebut sebagai repeater (perantara) dari aplikasi.
- Tombol Back digunakan untuk kembali ke halaman sebelumnya.

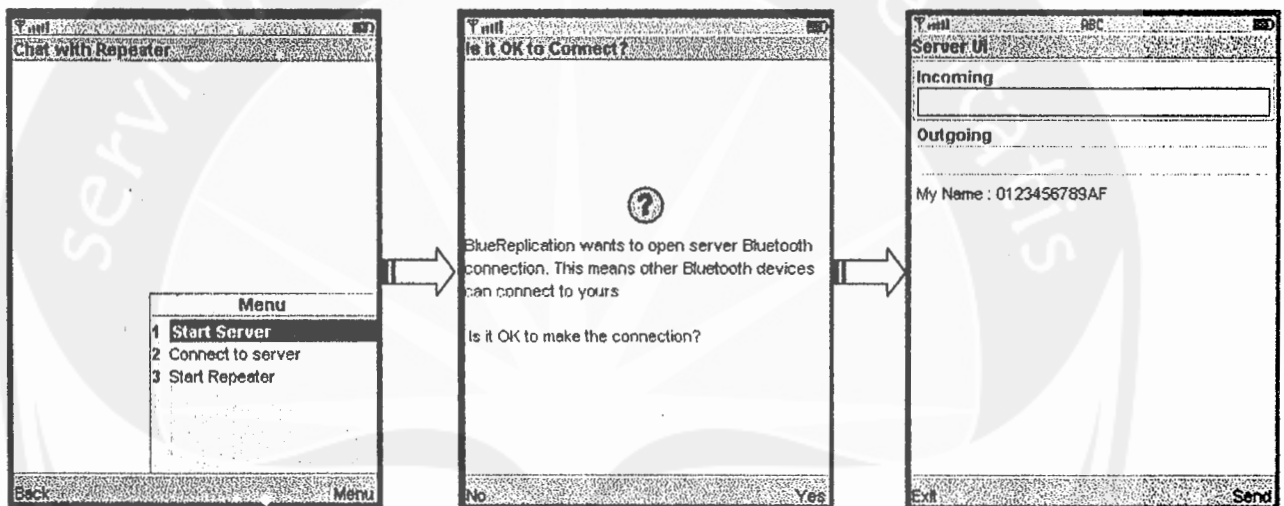
Event

Urutan aksi yang terjadi :

1. Pengguna memilih menu Connect.
2. Akan tampil pesan apakah akan membuka koneksi *bluetooth device* atau tidak sehingga dapat dideteksi oleh *bluetooth device* lain
3. Pilih Yes maka sistem akan membuka koneksi *bluetooth*.
4. Sistem akan melakukan proses inquiry untuk mendeteksi *bluetooth device* disekitarnya.

5. Bila memilih Server, maka *device* tersebut otomatis akan menjadi server dari aplikasi ini dan sistem akan membuka layanan server dengan nomor ID tertentu.
6. Bila memilih Repeater, maka *device* tersebut otomatis akan menjadi perantara dari aplikasi ini dan sistem akan membuka layanan repeater dengan nomor ID tertentu

4.6 Use Case : Start Server



Gambar 4.8 Rancangan Antarmuka Start Server

Deskripsi

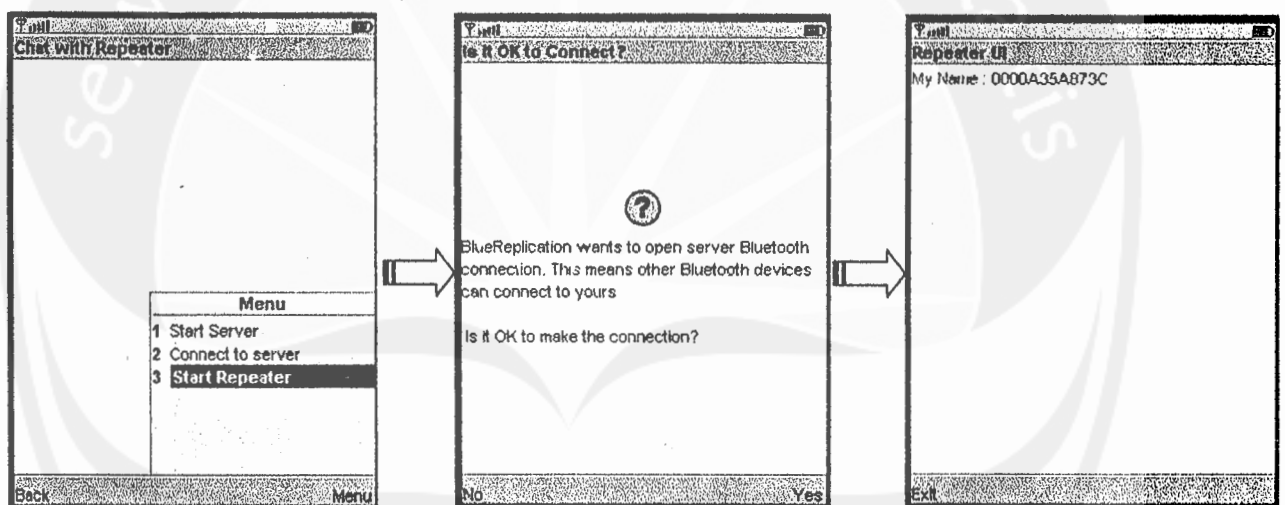
- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.BlueReplicationMIDlet** dan **BlueReplication.Server**.
- Menu pilihan Server untuk menjadikan device tersebut sebagai server dari aplikasi.
- Menu pilihan Connect untuk menjadikan device tersebut sebagai client dari aplikasi.
- Menu pilihan Repeater untuk menjadikan device tersebut sebagai repeater (perantara) dari aplikasi.
- Tombol Back digunakan untuk kembali ke halaman sebelumnya.
- Tombol Send digunakan untuk mengirimkan pesan.
- Tombol Exit digunakan untuk keluar dari aplikasi.

Event

Urutan aksi yang terjadi :

1. Pengguna memilih menu Server.
2. Akan tampil pesan apakah akan membuka koneksi *bluetooth device* atau tidak sehingga dapat dideteksi oleh *bluetooth device* lain
3. Pilih Yes maka sistem akan membuka koneksi *bluetooth*.
4. Server siap untuk melakukan penerimaan dan pengiriman pesan dengan membuka layanan dengan nomor ID tertentu

4.7 Use Case : Start Repeater



Gambar 4.9 Rancangan Antarmuka Use Start Repeater

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication**, **BlueReplicationMIDlet** dan **BlueReplication.Repeater..**
- Menu pilihan Server untuk menjadikan device tersebut sebagai server dari aplikasi.
- Menu pilihan Connect untuk menjadikan device tersebut sebagai client dari aplikasi.
- Menu pilihan Repeater untuk menjadikan device tersebut sebagai repeater (perantara) dari aplikasi.

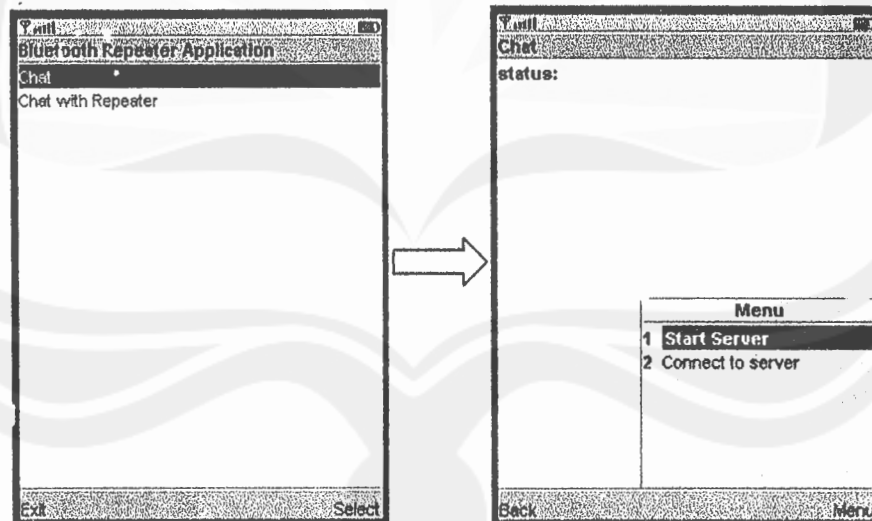
- Tombol Back digunakan untuk kembali ke halaman sebelumnya.
- Tombol Exit digunakan untuk keluar dari aplikasi.

Event

Urutan aksi yang terjadi :

1. Pengguna memilih menu Repeater.
2. Akan tampil pesan apakah akan membuka koneksi *bluetooth device* atau tidak sehingga dapat dideteksi oleh *bluetooth device* lain.
3. Pilih Yes maka sistem akan membuka koneksi *bluetooth*.
4. Repeater siap untuk melakukan penguatan sinyal dan membantu pengiriman pesan dengan membuka layanan dengan nomor ID tertentu.

4.8 Use Case : Pengiriman Pesan Biasa



Gambar 4.10 Rancangan Antarmuka Pengiriman Pesan Biasa

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.BlueReplicationMIDlet**.
- Menu Chat untuk memilih pengiriman pesan tanpa perantara (Repeater).
- Menu Chat with Repeater untuk memilih pengiriman pesan menggunakan perantara (Repeater).

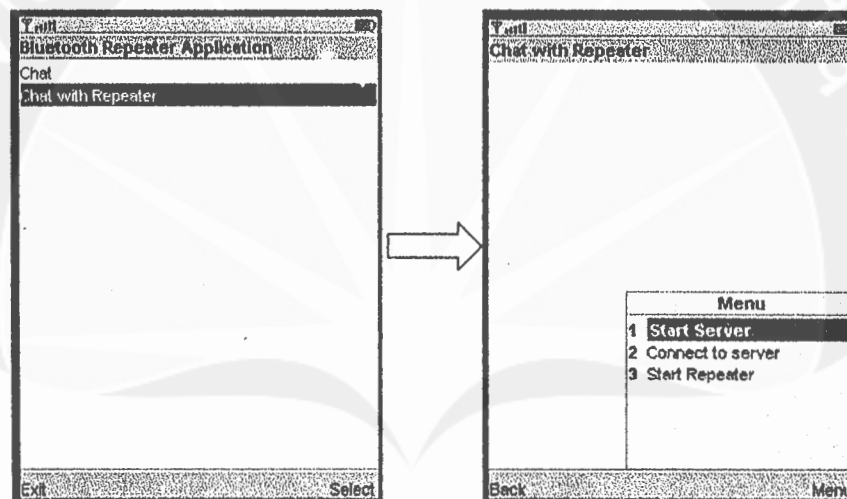
- Tombol Back digunakan untuk kembali ke halaman sebelumnya.
- Tombol Exit digunakan untuk keluar dari aplikasi.

Event

Urutan aksi yang terjadi :

1. Pengguna memilih menu Chat.
2. Sistem akan menampilkan pilihan Start Server dan Connect To Server.
3. Pengguna memilih salah satu dari pilihan-pilihan tersebut.

4.9 Use Case : Pengiriman Pesan dengan Repeater



Gambar 4.11 Rancangan Antarmuka Pengiriman Pesan dengan Repeater

Deskripsi

- Rancangan antarmuka ini diimplementasikan pada Class **BlueReplication.BlueReplicationMIDlet**.
- Menu Chat untuk memilih pengiriman pesan tanpa perantara (Repeater).
- Menu Chat with Repeater untuk memilih pengiriman pesan menggunakan perantara (Repeater).
- Menu pilihan Server untuk menjadikan device tersebut sebagai server dari aplikasi.
- Menu pilihan Connect untuk menjadikan device tersebut sebagai client dari aplikasi.

- Menu pilihan Repeater untuk menjadikan device tersebut sebagai repeater (perantara) dari aplikasi.
- Tombol Back digunakan untuk kembali ke halaman sebelumnya.
- Tombol Exit digunakan untuk keluar dari aplikasi.

Event

Urutan aksi yang terjadi :

1. Pengguna memilih menu Chat with Repeater.
2. Sistem akan menampilkan pilihan Server, Connect dan Repeater.
3. Pengguna memilih salah satu dari pilihan-pilihan tersebut.

Apendiks A : Daftar Istilah dan Singkatan

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga Software Design Description (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
SKPL	Spesifikasi Kebutuhan Perangkat Lunak disebut juga Software Requirement Specification (SRS) merupakan deskripsi dari kebutuhan perangkat lunak yang akan dikembangkan.
BlueReplication	Perangkat lunak yang membantu mengirimkan pesan dengan jangkauan mencapai 2 kali pengiriman semula dengan menggunakan suatu <i>bluetooth device</i> lain sebagai perantara (Repeater).

