

Aplikasi Pengiriman Pesan Secara Berantai pada Daerah Bencana Terisolasi Menggunakan Teknologi Manet

Tri Hadiyah Muliawati, Isbat Uzzin Nadhori, Yuliana Setiowati
Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.
Email: lerylee92@gmail.com, isbat@eepis-its.edu, yuliana@eepis-its.edu

Abstrak

Indonesia merupakan negara yang berpotensi besar untuk menjadi daerah terdampak bencana. Adanya bencana biasanya diikuti dengan beberapa kerusakan, salah satunya infrastruktur telekomunikasi. Padahal komunikasi merupakan salah satu faktor penting yang dapat digunakan untuk menunjang penanggulangan bencana yang baik.

Pada Tugas Akhir ini dibuat suatu aplikasi pengiriman pesan secara berantai yang menggunakan teknologi manet sebagai alternatif alat komunikasi baru di daerah bencana terisolasi. Adapun teknologi manet (mobile adhoc network) yang digunakan memanfaatkan bluetooth yang ada di HP.

Dari percobaan yang dilakukan diperoleh hasil bahwa banyaknya karakter yang dikirim berbanding lurus dengan waktu pengiriman pesan. Selain itu, terdapat perbedaan pengiriman pesan pada area tanpa dinding penghalang dan area dengan dinding penghalang. Pengiriman pesan pada area dengan dinding penghalang terbukti membutuhkan waktu lebih lama dan rentan akan putusya koneksi. Sehingga pengiriman pesan lebih efektif dilakukan pada area tanpa dinding penghalang.

Kata kunci : bencana, manet, bluetooth

1. Pendahuluan

Indonesia merupakan negara dengan tingkat potensi bencana alam yang tinggi. Adanya bencana pada suatu daerah biasanya diikuti dengan beberapa kerusakan infrastruktur, termasuk komunikasi. Padahal komunikasi merupakan salah satu faktor penting untuk menunjang efektifitas penanggulangan bencana.

Manet (Mobile Adhoc Network) merupakan salah satu solusi untuk menjawab tantangan tersebut. Dengan memanfaatkan perangkat jaringan, misalnya bluetooth yang terdapat pada handphone, jaringan ini pun dapat dibangun kapanpun dimanapun dibutuhkan. Untuk mengimplementasikan konsep teknologi manet pada daerah bencana, perlu dibuat sebuah aplikasi pengatur pengiriman pesan antar node yang ada pada jaringan adhoc tersebut.

2. Teori Penunjang

Bluetooth cenderung membentuk jaringan piconet, yakni jaringan berbentuk bintang dengan 1 *master* dan 7 *active slave*. Sehingga jangkauan koneksi antar node pun tetap 10 m. Jangkauan tersebut dapat diperluas dengan cara mengubah struktur jaringannya menjadi scatternet yang merupakan pengembangan dari jaringan piconet. Jaringan ini memungkinkan untuk memperbanyak jumlah node yang terlibat sekaligus memperluas jangkauan koneksi bluetooth. Akan tetapi pemeliharaan jaringan scatternet membutuhkan adanya usaha lebih. Seiring berkembangnya

teknologi, mulai dikembangkan beberapa algoritma untuk memelihara koneksi pada jaringan scatternet. Sebagaimana diujicobakan pada framework BEDnet yang digunakan untuk multiplayer game^[1]

2.1 Pemrograman J2ME

Merupakan kepanjangan dari Java 2 Micro Edition. J2ME biasa digunakan pada telepon seluler, pager, Personal Digital Assistants (PDA's) dan sejenisnya

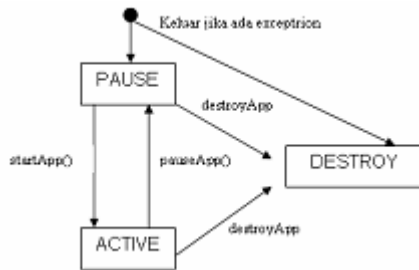
2.2 Mobile Information Device Profile (MIDP)

Merupakan spesifikasi untuk sebuah profil J2ME. MIDP User Interface API memiliki API level tinggi dan level rendah. API level rendah berbasiskan penggunaan dari kelas abstrak Canvas, sedangkan API kelas tinggi antara lain Alert, Form, List dan TextBox yang merupakan ekstensi dari kelas abstrak Screen. API level rendah lebih memberikan kemudahan kepada pengembang untuk memodifikasi sesuai kehendaknya, sedangkan API level tinggi biasanya hanya memberikan pengaksesan yang terbatas.

2.3 MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas `javax.microedition.midlet.MIDlet` yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan sub kelas dari bentuk dasar aplikasi sehingga antarmuka antara

J2ME dan aplikasi manajemen pada perangkat dapat terbentuk.



Gambar 1 Siklus Hidup MIDlet

Untuk mengaktifkannya diperlukan import package `javax.microedition.midlet`

2.4 RMS

Merupakan kumpulan record yang disimpan sebagai array dari byte dalam sebuah record store. RMS memiliki orientasi record basis data yang sederhana sehingga MIDlet dapat menyimpan informasi dan mengaksesnya. MIDlet yang berbeda dapat mengakses RMS yang sama.

Record yang disimpan dalam Record Store diakses berdasarkan `recordId` yang berupa integer. `RecordId` ini biasanya digunakan untuk mengakses record seperti fungsi indeks pada pengaksesan array.

Untuk mengaktifkannya diperlukan import package `javax.microedition.rms`

2.5 Jaringan Adhoc

Jaringan Adhoc merupakan jaringan wireless yang terdiri dari beberapa *mobile* node yang tidak memiliki router tetap. Node-node dalam jaringan ini berfungsi juga sebagai router yang bertanggung jawab untuk mencari dan menangani rute ke setiap node di dalam jaringan.

Node-node pada jaringan ini tidak hanya berperan sebagai pengirim dan penerima data, namun dapat berperan sebagai penunjang node yang lainnya, misalnya mempunyai kemampuan layaknya router. Dengan demikian diperlukan adanya routing protokol dalam jaringan ini untuk menunjang proses pengiriman data antar node-nya.

2.6 Bluetooth

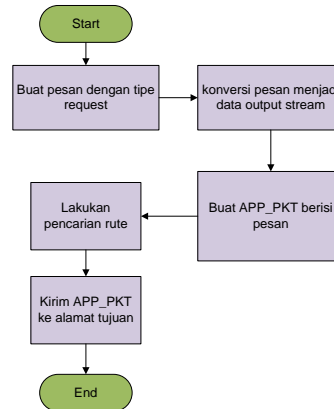
Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas.

3. Perancangan Sistem

Dalam aplikasi pengiriman pesan secara berantai ini terdapat beberapa proses inti, yakni :

3.1 Pengiriman Pesan

Merupakan proses pengiriman pesan dari node pengirim hingga menuju node tujuan. Berikut merupakan alur proses pengiriman pesan yang akan dilakukan :



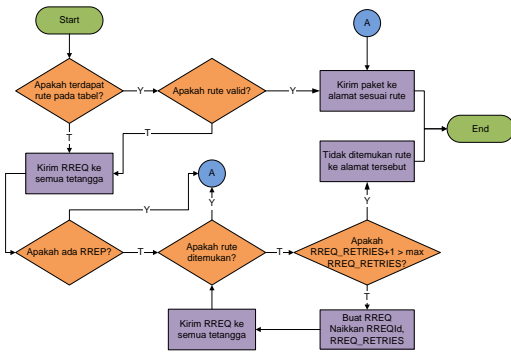
Gambar 2 Flowchart Pengiriman Pesan

Keterangan :

1. Terdapat 2 tipe pesan yang dikenali sistem, yakni response dan request. Response digunakan untuk pesan yang berisi notifikasi yang menerangkan bahwa pesan telah sampai ditujuan. Sedangkan Request digunakan untuk tipe pesan yang berisi informasi yang akan dikirim. Oleh karenanya digunakan tipe Request.
2. Agar pesan bisa dikirim, maka pesan harus dikonversi kedalam `OutputStream` dengan tipe data `Byte`.
3. Selanjutnya dibuat paket data `APP_PKT` yang berisi pesan tersebut. `APP_PKT` merupakan tipe paket data yang dikenali oleh AODV sebagai paket berisi pesan.
4. Dilakukan pencarian rute untuk menemukan rute menuju alamat tujuan, termasuk di dalamnya pengecekan rute pada tabel rute node asal.
5. Apabila rute ditemukan maka `APP_PKT` akan dikirimkan menuju node tujuan.

3.2 Pencarian Rute

Sebagaimana telah disinggung di poin sebelumnya untuk mengirimkan pesan ke alamat tujuan perlu dilakukan pencarian rute. Pencarian rute yang dimaksud disini termasuk pengecekan pada tabel rute yang dimiliki node. Berikut merupakan flowchart untuk menggambarkan proses dalam pencarian rute :



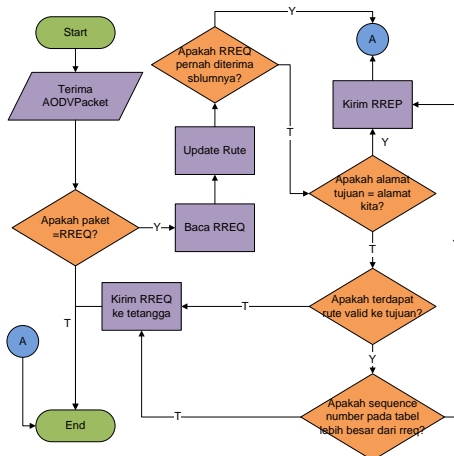
Gambar 3 Flowchart Pencarian Rute

Keterangan:

1. Pengecekan ketersediaan rute pada tabel rute milik node asal.
2. Bila terdapat rute yang valid maka paket data berisi pesan akan dikirimkan melalui rute tersebut
3. Bila rute yang valis tidak ditemukan maka node asal akan mengirimkan RREQ ke semua node tetangga.
4. Selanjutnya node pengirim RREQ hanya bisa menunggu balasan RREP.
5. Apabila sampai jangka waktu tertentu node belum mendapatkan RREP atas RREQ yang dikirimnya maka node bisa mengirimkan ulang RREQ dengan RREQid yang berbeda selama belum melampaui RREQ_RETRIES.
6. Apabila rute tetap tidak ditemukan hingga maksimal RREQ_RETRIES, maka tmemang idak terdapat rute menuju alamat tersebut.

3.3 Pengelolaan RREQ

Berikut alur proses yang terjadi apabila sebuah node menerima RREQ.



Gambar 4 Flowchart Pengelolaan RREQ yang diterima

Keterangan :

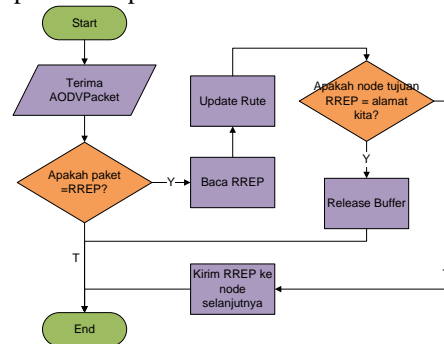
1. Mengingat terdapat beberapa paket data yang dikirimkan pada jaringan, maka tiap

kali terdapat paket masuk akan dilakukan pengecekan tipe paket tersebut.

2. Apabila node menerima RREQ, maka informasi rute menuju node pengirim RREQ akan diupdate.
3. Selanjutnya, lakukan pengecekan apakah node pernah menerima RREQ tersebut sebelumnya. Bila node pernah menerima RREQ yang sama maka RREQ akan diabaikan.
4. Lakukan pengecekan alamat tujuan RREQ.
 - a. Apabila node merupakan node tujuan RREQ, maka node akan mengirimkan paket RREP
 - b. Apabila node memiliki rute yang valid ke node tujuan RREQ dengan sequence number yang lebih besar atau sama maka node akan mengirimkan paket RREP.
 - c. Bila tidak memenuhi salah satu kondisi diatas dan TTL RREQ > 1, RREQ akan dikirimkan ke node lainnya.

3.4 Pengelolaan RREP

Secara garis besar gambaran alur pengelolaan RREP dapat dilihat pada flowchart berikut:



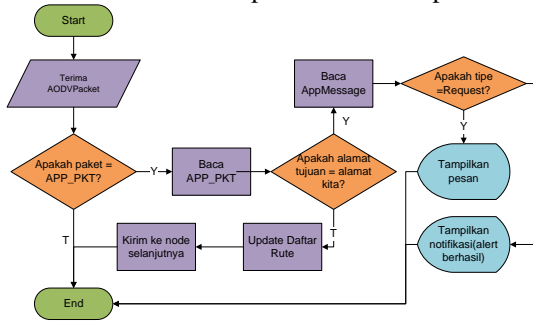
Gambar 5 Flowchart Pengelolaan RREP

Keterangan :

1. Apabila node menerima RREP, maka node akan mengupdate informasi rute pada tabel rutanya. Informasi yang diupdate adalah rute menuju node sebelumnya yang meneruskan RREP.
2. Selain itu informasi rute mengenai node tujuan RREP juga di update.
3. Lakukan pengecekan informasi node asal RREP
4. Apabila node asal RREP merupakan perangkat bluetooth kita, maka rute ditemukan dan kirimkan pesan yang disimpan. Apabila node asal RREP bukan merupakan perangkat bluetooth kita, maka RREP akan diteruskan ke node selanjutnya.

3.5 Penerimaan Paket Pesan

Berikut merupakan gambaran proses yang terjadi saat node menerima paket data berisi pesan.



Gambar 6 Flowchart Penerimaan Paket Pesan

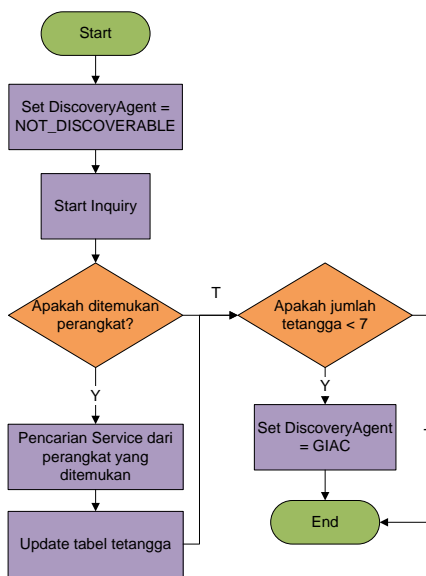
Keterangan :

1. Lakukan pengecekan informasi pada paket pesan masuk tersebut.
2. Apabila alamat tujuan paket bukan merupakan alamat node maka update tabel rute node lalu kirimkan paket pesan ke node selanjutnya.
3. Apabila alamat tujuan paket merupakan alamat node akan dilakukan pengecekan lagi untuk menentukan apakah pesan tersebut hanyalah notifikasi atau merupakan pesan baru dari node lainnya.

3.6 Koneksi dengan Perangkat Bluetooth lain

3.6.1 Pencarian Perangkat Lain dan Service yang Ditawarkan (Device and Service Discovery)

Berikut merupakan gambaran proses yang terjadi saat node melakukan pencarian perangkat dan service.



Gambar 7 Flowchart Pencarian Perangkat dan Service

Keterangan :

1. Sebelum perangkat bluetooth mencari perangkat lain, terlebih dahulu perangkat bluetooth akan mengeset dirinya menjadi NOT_DISCOVERABLE sehingga tidak bisa ditemukan oleh perangkat lainnya.
2. Selanjutnya terjadi proses inquiry.
3. Apabila perangkat lain berhasil ditemukan maka akan dilakukan pencarian service yang ditawarkan oleh perangkat tersebut. Lalu tabel tetangga akan diupdate.
4. Cek jumlah tetangga yang dimiliki perangkat tersebut, apabila jumlah tetangga kurang dari 7 maka status perangkat bluetooth akan dikembalikan menjadi GIAC.

3.6.2 Request Koneksi dari Perangkat Lain

Selama suatu perangkat memiliki jumlah perangkat < 7 maka ia akan membuka koneksi L2CAP sehingga memungkinkan untuk menerima tawaran perangkat lain untuk melakukan koneksi. Tiap kali perangkat menerima tawaran koneksi dari perangkat lain maka nilai tabel tetangga akan diupdate.

3.7 Pemeliharaan Rute

Pemeliharaan rute dilakukan dengan cara pengiriman paket hello. Paket hello ini dijalankan sejak aplikasi pertama kali dijalankan dan dikirimkan secara periodik kepada node tetangga masing-masing node.

Apabila suatu node(A) tidak menerima paket hello dari node tetangganya(B) selama jangka waktu tertentu maka akan node tersebut (A) akan mengirimkan RERR kepada node lain mengenai hilangnya node (B).

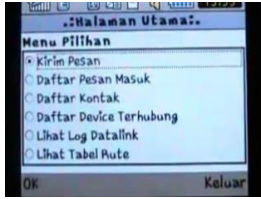
3.8 Record Management System

Pada aplikasi ini digunakan 3 buah RMS untuk penyimpanan data yang digunakan pada aplikasi. Adapun RMS yang digunakan antara lain :

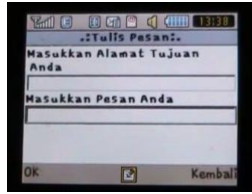
- RMS kontak
Digunakan untuk menyimpan data kontak yang terdiri dari nama dan alamat bluetooth
- RMS history
Digunakan untuk menyimpan data history, yakni nama dan alamat perangkat lain yang pernah mengirimkan pesan.
- RMS pesan
Digunakan untuk menyimpan data pesan, yakni nama, alamat bluetooth, dan pesan yang diterima.

4. Hasil dan Analisa

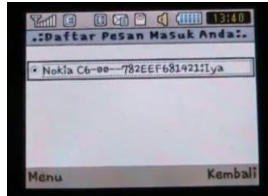
Berikut merupakan beberapa tampilan output program.



Gambar 8
Tampilan menu utama



Gambar 9
Tampilan menu kirim pesan

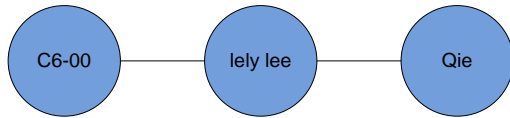


Gambar 10
Tampilan menu daftar pesan masuk

Dari percobaan yang dilakukan diperoleh hasil yang bervariasi terhadap beberapa kondisi yang diberikan.

4.1 Percobaan 1

Adapun pada percobaan 1 dilakukan uji coba menggunakan 3 handphone pada area tanpa dinding penghalang. Koneksi antara ketiga handphone tersebut dapat dilihat pada gambar berikut:



Gambar 11
Diagram koneksi antar perangkat pada percobaan 1

Berikut hasil uji coba pengiriman pesan pada percobaan 1:

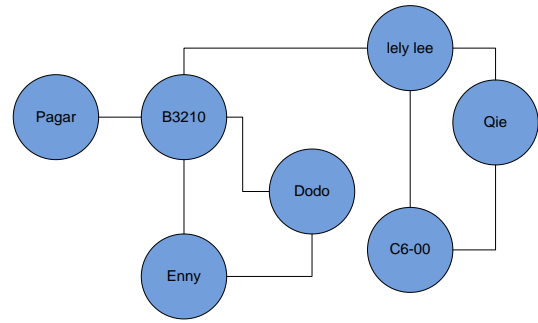
Tabel 1 Tabel Hasil Percobaan 1

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	Qie	C6-00	24	24	1 detik
2	Qie	C6-00	46	46	1 detik
3	Qie	C6-00	84	84	1 detik
4	Qie	C6-00	160	160	2 detik

Data di atas menunjukkan bahwa semua karakter pesan yang dikirim berhasil diterima oleh handphone tujuan. Akan tetapi, uji coba nomor 4 dengan jumlah karakter 160 menunjukkan waktu pengiriman yang lebih lama dibandingkan uji coba nomor 1, 2, dan 3.

4.2 Percobaan 2

Adapun pada percobaan 2 dilakukan uji coba pada area tanpa dinding penghalang dengan menggunakan 7 handphone. Koneksi antara ketujuh handphone tersebut dan hasil uji coba pengiriman pesan pada percobaan ini dapat dilihat pada gambar dan tabel berikut:



Gambar 12
Diagram koneksi antar perangkat pada percobaan 2

Tabel 2 Tabel Hasil Percobaan 2

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	Pagar	Dodo	24	24	1 detik
2	Pagar	Enny	24	24	1 detik
3	C6-00	B3210	24	24	2 detik
4	C6-00	Dodo	24	24	5 detik
5	Dodo	Qie	24	24	5 detik
6	C6-00	Dodo	84	84	6 detik
7	Dodo	Qie	84	84	7 detik
8	C6-00	Dodo	113	113	8 detik
9	Dodo	Qie	113	113	7 detik
10	C6-00	Dodo	160	160	8 detik
11	Dodo	Qie	160	160	8 detik

Sebagaimana percobaan sebelumnya, data di atas menunjukkan bahwa semua karakter pesan yang dikirim berhasil diterima oleh handphone tujuan. Selain itu, uji coba juga menunjukkan bahwa pengiriman pesan dengan karakter yang lebih banyak membutuhkan waktu lebih lama, sebagaimana bisa dilihat pada uji coba nomor 10, 11 bila dibandingkan dengan uji coba nomor 4-9.

Selain itu juga diperoleh hasil bahwa jumlah lompatan sebanding dengan waktu yang dibutuhkan. Pada uji coba nomor 1 dan 2 dimana dibutuhkan 2 lompatan untuk mencapai handphone tujuan hanya dibutuhkan waktu 1 detik. Sedangkan pada ujicoba nomor 3, 4, dan 5 dimana terjadi lebih dari 2 lompatan membutuhkan waktu lebih dari 1 detik untuk mencapai handphone tujuan.

4.3 Percobaan 3 dan 4

Adapun pada percobaan 3 dan 4 dilakukan uji coba pada area dengan dinding penghalang dengan menggunakan 3 handphone. Akan tetapi, pada percobaan 3 jarak antar handphone lebih lebar dibandingkan pada percobaan 4. Koneksi antara ketiga handphone tersebut dapat dilihat pada gambar berikut:



Gambar 13
Diagram koneksi antar perangkat pada percobaan 3 dan 4

Adapun data hasil uji coba pada percobaan 3 dan 4 dapat dilihat pada tabel berikut:

Tabel 3 Tabel Hasil Percobaan 3

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	leylee	C6-00	24	24	1 detik
2	leylee	C6-00	46	46	2 detik
3	C6-00	leylee	84	Putus	Putus

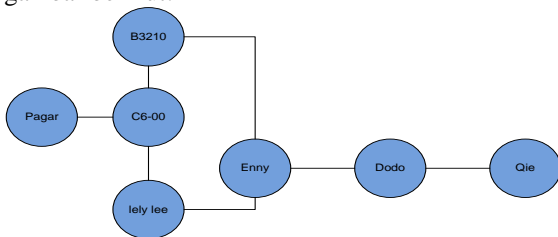
Tabel 4 Tabel Hasil Percobaan 4

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	Lelylee	C6-00	24	24	1 detik
2	C6-00	leylee	84	84	3 detik
3	C6-00	leylee	160	160	3 detik

Data dari hasil percobaan menunjukkan bahwa terdapat putusnya koneksi saat uji coba nomor 3 pada percobaan 3. Akan tetapi, saat dilakukan uji coba ulang dengan jarak yang lebih pendek sebagaimana pada uji coba nomor 1 percobaan 4 didapatkan hasil yang berbeda. Pada uji coba nomor 1 percobaan 4 pesan berhasil dikirimkan.

4.4 Percobaan 5 dan 6

Adapun pada percobaan 5 dan 6 dilakukan uji coba pada area dengan dinding penghalang dengan menggunakan 7 handphone. Akan tetapi, pada percobaan 5 jarak antar handphone lebih lebar dibandingkan pada percobaan 6. Koneksi antara ketujuh handphone tersebut dapat dilihat pada gambar berikut:



Gambar 14

Diagram koneksi antar perangkat pada percobaan 5 dan 6

Adapun data hasil uji coba pada percobaan 5 dan 6 dapat dilihat pada tabel berikut:

Tabel 5 Tabel Hasil Percobaan 5

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	Pagar	B3210	24	24	1 detik
2	Pagar	Lelylee	24	24	1 detik
3	Qie	Enny	24	24	1 detik
4	C6-00	Dodo	24	Putus	Putus
5	Qie	B3210	24	Putus	Putus
6	leylee	Dodo	24	Putus	Putus
7	leylee	B3210	24	24	1 detik

Tabel 6 Tabel Hasil Percobaan 6

No	Pengirim	Penerima	Jumlah Karakter dikirim	Jumlah Karakter diterima	Waktu Pengiriman Pesan
1	C6-00	Dodo	24	24	7 detik
2	Qie	B3210	24	24	5 detik
3	C6-00	Dodo	84	84	8 detik
4	Qie	B3210	84	84	8 detik
5	C6-00	Dodo	113	113	9 detik
6	Qie	B3210	113	113	8 detik
7	C6-00	Dodo	160	160	10 detik
8	Qie	B3210	160	160	10 detik
9	Qie	C6-00	160	160	12 detik

Data hasil percobaan menunjukkan bahwa terdapat beberapa kali putusnya koneksi pada percobaan 5 sebagaimana terdapat pada percobaan 3. Akan tetapi, pada percobaan 6 dengan jarak antar handphone yang lebih pendek didapatkan hasil yang berbeda. Pada percobaan 6, semua pesan berhasil dikirim.

5. Kesimpulan

Dari hasil uji coba dan analisis dapat ditarik kesimpulan sbb:

- Pengiriman pesan secara berantai lebih efektif dilakukan pada area tanpa dinding penghalang.
- Pengiriman pesan secara berantai pada area dengan dinding penghalang rentan akan putusnya koneksi.
- Adanya dinding penghalang menyebabkan jarak koneksi antar handphone menjadi lebih pendek.
- Banyaknya karakter yang dikirimkan berbanding lurus dengan waktu pengiriman pesan.
- Banyaknya lompatan yang dilewati berbanding lurus dengan waktu pengiriman pesan

6. Daftar Pustaka

- Nielsen, Michael, Glenstrup, Arne John, dkk. 2008. *Real-world Bluetooth Manet Java Middleware*
- Thompson, Timothy J, dkk. 2008. *Bluetooth Application Programming with the JAVA APIs*. Burlington : Morgan Kaufmann Publishers
- Wicaksono, Ady. 2002. *Pemrograman Aplikasi Wireless dengan Java*. Jakarta : PT. Elex Media Komputindo