

APLIKASI GROUP CHAT DENGAN BLUETOOTH PADA TELEPON SELUER

David Johanes Pasaribu, Budi Susanto, Andronicus Riyono
Fakultas Teknik Program Studi Teknik Informatika
Universitas Kristen Duta Wacana Yogyakarta
Email: davejupjup@gmail.com, budsus@ukdw.ac.id, riyono@gmail.com

Abstrak:

Telepon seluler yang telah dilengkapi dengan Bluetooth membuat komunikasi dapat dilakukan dengan banyak pilihan cara, yaitu dengan menggunakan jaringan GSM, jaringan CDMA, atau jaringan Bluetooth. Salah satu aplikasi telepon seluler untuk berkomunikasi dengan menggunakan jaringan Bluetooth adalah Aplikasi Group Chat. Keuntungan berkomunikasi dengan jaringan Bluetooth adalah tidak adanya biaya komunikasi yang dikeluarkan. Aplikasi ini dikembangkan dengan menggunakan lingkungan J2ME. J2ME adalah teknologi Java yang didesain untuk membangun aplikasi yang dapat berjalan pada telepon seluler. Java juga menyediakan Java API untuk Bluetooth (JSR-82) yang menyediakan paket untuk membantu dalam pemrograman pengembangan aplikasi Bluetooth.

Aplikasi Group Chat yang dibangun menyediakan komunikasi 2 atau lebih telepon seluler, satu telepon seluler berfungsi sebagai *server* yang menyediakan layanan *chatting* dan telepon seluler yang lain bertindak sebagai *client* yang melakukan koneksi ke *server*. Protokol Bluetooth yang digunakan untuk berkomunikasi antara *client* dan *server* adalah protokol L2CAP.

Kata Kunci: Bluetooth, J2ME, Aplikasi Group Chat, L2CAP, JSR-82

1. Pendahuluan

Bluetooth adalah suatu teknologi komunikasi *wireless* yang menggunakan frekuensi radio ISM 2.4 GHz. Dengan teknologi Bluetooth, komunikasi data dan suara dapat dilakukan secara *wireless* pada jarak tertentu. Salah satu perangkat elektronik *portable* yang banyak memanfaatkan kemampuan Bluetooth adalah telepon seluler. Berdasarkan fungsinya, telepon seluler digunakan untuk berkomunikasi jarak jauh. Seperti panggilan suara dan mengirim pesan. Dalam penerapannya, komunikasi dengan menggunakan telepon seluler dinilai masih terlalu mahal dan kurang efisien ketika komunikasi yang akan kita lakukan berada dalam radius kurang dari 10 meter. Hal itu dikarenakan ada biaya yang harus dikeluarkan, yaitu biaya pulsa.

Dengan memanfaatkan Bluetooth pada telepon seluler dimungkinkan membuat suatu jaringan lokal untuk dapat berkomunikasi tanpa mengeluarkan biaya.

Untuk itu, penulis dalam penelitian ini membuat suatu aplikasi Java berbasis J2ME (*Java 2 Micro Edition*) yang dapat membantu manusia dalam berkomunikasi dengan sesama tanpa mengeluarkan biaya dengan menggunakan telepon seluler yang telah terpasang Bluetooth. Aplikasi ini memungkinkan dua atau lebih pemakai dapat saling berkomunikasi dengan cara mengirimkan pesan secara *personal* atau *broadcast* melalui protokol komunikasi Bluetooth yang sudah tersedia pada perangkat telepon selular.

2. Landasan Teori

2.1. Bluetooth

Bluetooth adalah sebuah teknologi komunikasi tanpa kabel (*wireless*) yang beroperasi pada pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)*. Bluetooth menerapkan *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara perangkat-perangkat Bluetooth dengan jarak jangkauan layanan yang terbatas. Bluetooth sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk **WLAN (Wireless Local Area Network)** dimana menggunakan frekuensi radio standar **IEEE 802.11**, hanya saja pada Bluetooth mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan *transfer* data yang lebih rendah.

2.2. Protokol L2CAP

L2CAP ini merupakan salah satu lapisan protokol inti dari sistem Bluetooth. Fungsinya adalah untuk mengatur aspek tingkat tinggi dari masing-masing koneksi, misalnya siapa sedang terhubung dengan siapa, apakah koneksi tersebut menggunakan enkripsi atau tidak, tingkat *performansi* apa yang dibutuhkan dan sebagainya. Selain itu, L2CAP juga bertanggung jawab terhadap proses konversi format data yang timbul antara berbagai API di atasnya dengan protokol Bluetooth yang lebih rendah. L2CAP ini diimplementasikan dalam bentuk *software* dan dapat dieksekusi baik dari sistem *host* maupun oleh *prosesor local* dalam sistem Bluetooth.

2.3. Jaringan Bluetooth

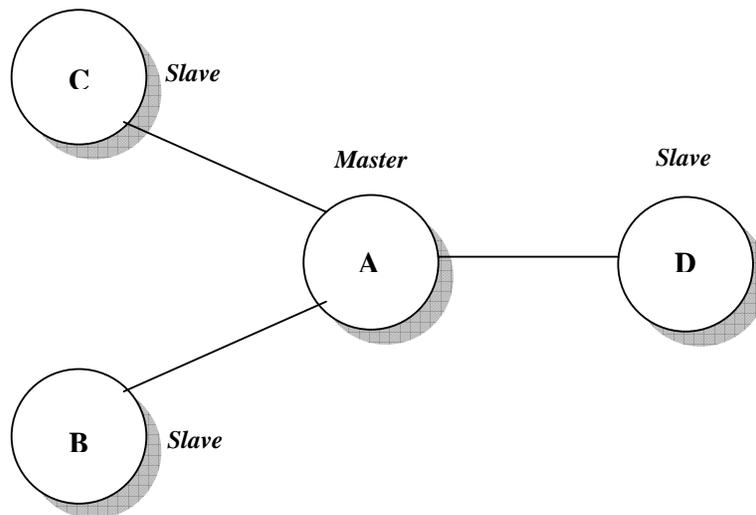
Jaringan Bluetooth menggunakan prinsip *master-slave*. Perangkat yang bertindak sebagai *master* adalah perangkat yang mengawali atau memulai terbentuknya koneksi. Sebagai contoh misalnya, terdapat dua buah perangkat Bluetooth (A dan B) yang hendak berkomunikasi. Perangkat A akan mengawali koneksi dengan menghubungi perangkat B, hal ini berarti perangkat A akan bertindak sebagai *master* dan perangkat B bertindak sebagai *slave*

dan komunikasi seperti ini sering dikenal dengan istilah *point to point communication*, seperti pada gambar 1.



Gambar 1. Point to Point Communication

Selain komunikasi *point to point*, bluetooth juga memungkinkan dibentuknya jaringan yang terdiri lebih dari dua perangkat. Jaringan seperti ini dikenal dengan nama *piconet*. *Piconet* adalah bentuk lazim dari jaringan Bluetooth yang terbentuk dari sebuah *master* dan 1 atau lebih *slave*. Sebuah *piconet* dapat terdiri dari sebuah *master* dan maksimal tujuh buah *slave* yang aktif pada suatu waktu, seperti gambar 2.



Gambar 2. Piconet

Untuk membentuk sebuah hubungan antara 2 atau lebih perangkat Bluetooth, terdapat beberapa proses yang harus dilakukan, yaitu:

1. *Inquiry* adalah suatu proses bagaimana perangkat Bluetooth belajar tentang perangkat Bluetooth lain yang ada di dalam jangkauannya.
2. *Pairing* adalah proses dimana akan dibangun hubungan antar perangkat Bluetooth (antara *master* dengan sebuah *slave*).

2.4. Gambaran J2ME

Java 2 Micro Edition atau yang biasa disebut J2ME adalah teknologi Java yang didesain untuk membuat perangkat lunak supaya dapat berjalan pada perangkat-perangkat kecil *consumer device*, terutama *wireless*. Beberapa perangkat *wireless* tersebut antara lain PC, PDA, *communicator*, *embedded device*, *smart card* dan telepon seluler. Secara umum perangkat-perangkat tersebut mempunyai karakteristik yang terbatas dibandingkan dengan komputer biasa, antara lain adalah layar *display* yang kecil, menghabiskan sedikit daya dari baterai dan memiliki memori yang terbatas.

2.5. JSR-82

JSR-82 adalah sebuah standar yang ditetapkan oleh **JCP (Java Community Process)** untuk menyediakan acuan bersama dan terbuka untuk pengembangan aplikasi Bluetooth di Java. JSR-82 API menyembunyikan kompleksitas dari *Bluetooth Protocol Stack* untuk pembuatan aplikasi Bluetooth. JSR-82 memiliki 2 paket yang berdiri sendiri untuk membantu pemrograman aplikasi Bluetooth lebih mudah, yaitu `javax.bluetooth` dan `javax.obex`. Dalam pembuatan Aplikasi Group Chat ini, paket yang akan digunakan adalah `javax.bluetooth`.

Tabel 1

Kelas-Kelas Dalam Paket `javax.Bluetooth`

Nama Kelas	Gambaran
L2CAPConnection	<i>Interface</i> yang mewakili sebuah koneksi yang berorientasi saluran L2CAP
Discovery Listener	<i>Interface</i> yang membiarkan sebuah aplikasi untuk menerima pencarian perangkat dan pencarian layanan
L2CAPConnectionNotifier	<i>Interface</i> yang menyediakan sebuah koneksi L2CAP <i>notifier</i>
ServiceRecord	<i>Interface</i> yang menggambarkan karakteristik dari sebuah layanan Bluetooth

Tabel 1 LanjutanKelas-Kelas dalam Paket *javax.Bluetooth*

Nama Kelas	Gambaran
UUID	Kelas UUID mendefinisikan <i>universally unique identifiers</i>
BluetoothConnectionException	BluetoothConnectionException ini dilemparkan saat sambungan Bluetooth (L2CAP, RFCOMM, atau OBEX) tidak dapat dibangun dengan sukses.
BluetoothStateException	BluetoothStateException dilemparkan ketika permintaan yang dibuat untuk sistem Bluetooth, sistem tidak dapat mendukung keadaan pada saat permintaan tersebut.
ServiceRegistrationException	ServiceRegistrationException dilemparkan ketika ada kegagalan untuk menambahkan catatan layanan kepada Layanan lokal Discovery Database (SDDB) atau untuk memodifikasi layanan yang sudah ada catatan di SDDB.
DeviceClass	kelas DeviceClass mewakili catatan kelas perangkat seperti yang didefinisikan oleh Bluetooth spesifikasi.
DiscoveryAgent	DiscoveryAgent kelas yang menyediakan metode untuk melakukan penemuan perangkat dan layanan.
LocalDevice	Kelas yang mewakili perangkat lokal Bluetooth
RemoteDevice	Kelas yang mewakili proses remoteDevice Bluetooth.
DataElement	Kelas DataElement mendefinisikan berbagai data jenis layanan <i>attribute</i> Bluetooth yang dipunya

3. Implementasi Sistem

Aplikasi Group Chat ini akan menanyakan nama identitas lokal pemakai harus dimasukkan ke sistem untuk kemudian pemakai dapat memilih apakah menjadi *server* atau *client*. Jika bertindak sebagai *server*, maka aplikasi akan menampilkan pesan tunggu, sampai ada *client* yang terhubung ke *server*. Pada sisi *client*, aplikasi Group Chat ini akan melakukan proses *inquiry* untuk mencari telepon selular mana yang bertindak sebagai *server*. Aplikasi akan menampilkan daftar alamat telepon selular yang bertindak sebagai *server* pada sisi *client*, kemudian pemakai dapat memilih *server* di telepon selular mana yang akan dihubungi.

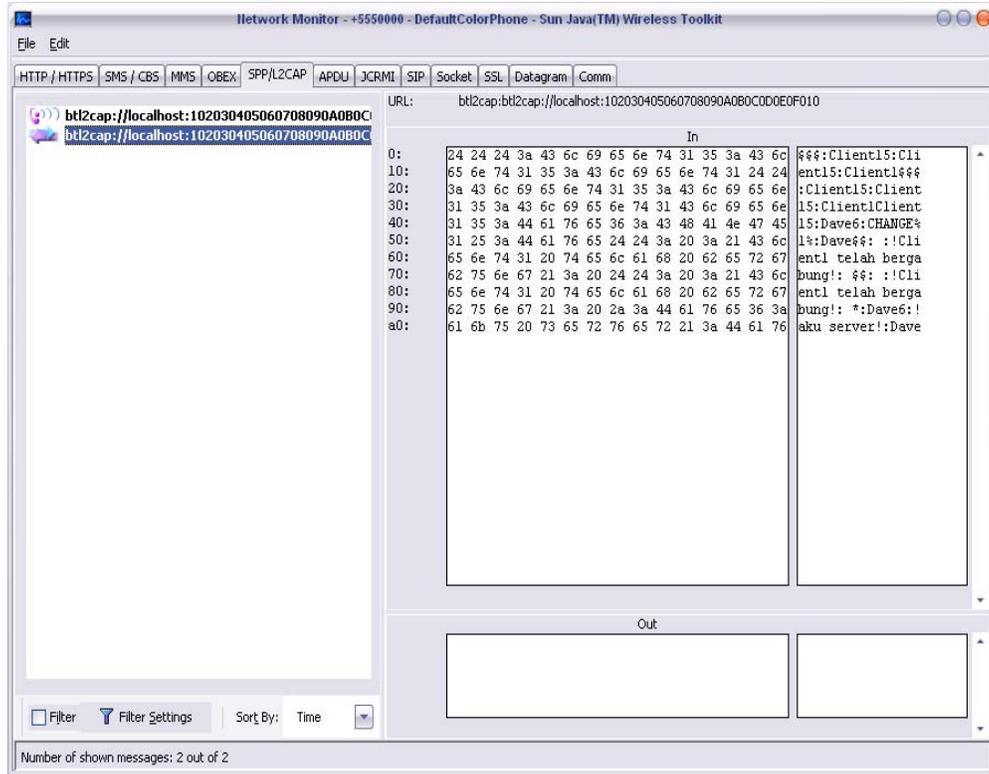


Gambar 3. Tampilan Device Discovered pada Telepon Seluler

Jika sudah disetujui oleh sisi *server*, sebagai pihak penerima permintaan koneksi dari *client*, selanjutnya aplikasi akan membuka form yang memungkinkan aktifitas chat berjalan antara *server* dan *client* melalui protokol bluetooth.

4. Analisa Sistem

Pesan yang telah dituliskan dalam *text box* memiliki tipe *String*. Untuk melakukan proses pengiriman paket dengan menggunakan protokol L2CAP, pesan harus diubah dahulu ke dalam bentuk *array Byte*. Proses pengubahan bentuk *String* menjadi *array Byte* dilakukan dengan memanggil *method* *getBytes()* pada *method* *SendMessage()* untuk pengiriman *global* dan *method* *SendPrivate()* pada pengiriman *personal* pada aplikasi ini. Selanjutnya proses pengiriman pesan yang sebenarnya dilakukan melalui protokol L2CAP dengan memanggil *method* *send()* yang telah disediakan pada *class* *L2CAPconnection* yang merupakan bagian dari paket *javax.bluetooth*.

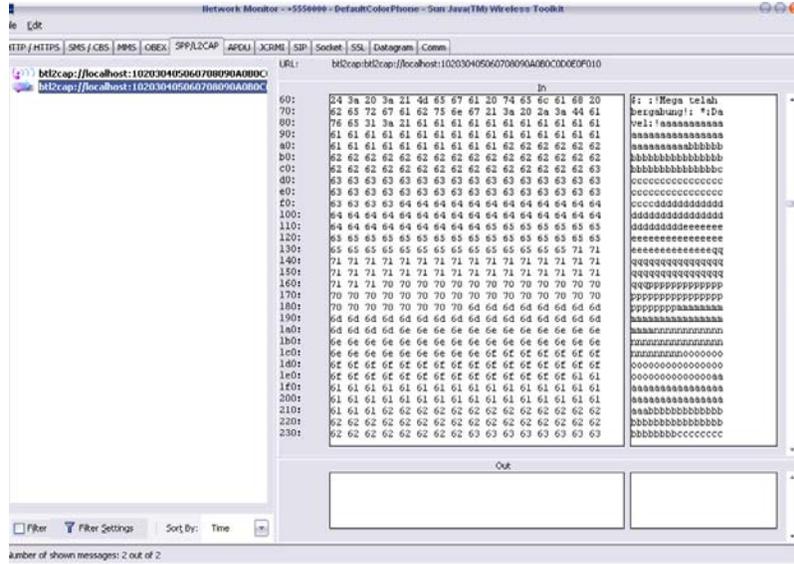


Gambar 4. Tampilan Network Monitor pada Proses Pengiriman dan Penerimaan Pesan

Pada Gambar 4 diberikan suatu gambar network monitoring dari server yang menjelaskan bagaimana pesan dikirimkan dengan menggunakan protokol L2CAP. Pada kolom bagian paling kanan menjelaskan tentang pesan yang dikirimkan oleh server. Kolom bagian tengah merupakan bagian yang menjelaskan pesan yang telah dikirim dalam bentuk *String* telah diubah kedalam bentuk *array Bytes*. Pada baris **a0**: menjelaskan bahwa server mengirim pesan "**aku server**" dan kolom bagian tengah menjabarkan tiap karakter *String* "**aku server**" menjadi *array Bytes* yaitu, nilai *hexadesimal* dari tiap karakter.

Dalam proses penerimaan pesan, untuk menerima pesan yang telah dikirimkan. Harus dipastikan terlebih dahulu apakah pesan telah siap untuk dibaca, yaitu dengan cara memanggil *method ready()*. *Method ready()* akan mengembalikan nilai "*true*" jika ada data yang diterima sudah siap. Ketika data sudah siap, sistem akan mengambil data pada *variabel receiveMTU*. Selanjutnya dilakukan proses pengecekan besar data yang masuk.

Untuk menampilkan data yang telah diterima. Data yang berbentuk *array byte* harus diubah kembali menjadi berbentuk *String* yang disimpan dalam *variable received*.



Gambar 5. Tampilan Analisis dari Pengiriman Pesan dengan 672 karakter

Ukuran data yang dapat dikirimkan melalui protocol L2CAP sangat tergantung pada ukuran MTU (Maximum Transmission Unit). Sebagai contoh pada Gambar 5 terlihat bahwa pesan sebanyak 672 karakter yang dikirimkan oleh *server* tidak dapat diterima oleh *client*. Hal ini dikarenakan besaran baku maksimum untuk MTU pada protokol L2CAP adalah 672 bytes dan besaran minimum MTU adalah 48 bytes. Jika pesan yang diterima atau yang dikirimkan melebihi besar MTU, maka data yang dikirimkan hilang atau tidak dapat diterima oleh *client*. Besar data 672 bytes yang maksimal dikirimkan sudah termasuk dengan *header paket* dan pesan.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisa yang telah dilakukan, maka dapat ditarik kesimpulan bahwa dengan ditunjang Java API untuk Bluetooth (JSR-82), pembuatan Aplikasi Group Chat pada telepon seluler dengan Bluetooth dapat dilakukan dan berjalan dengan baik. Proses pertukaran data antara *client* dan *server* juga dapat dilakukan dengan baik.

6. Daftar Pustaka

- [1] C. Bala Kumar, Paul J. Kline, & Timothy J. Thompson. (2004). *Bluetooth Application Programming With Java APIs*. Elsevier Inc, USA.
- [2] Harta Dwijaksana, Made. (2006). *Studi dan Implementasi Kriptografi Kunci-Public Untuk Otentikasi Perangkat dan Pengguna pada Komunikasi Bluetooth*. Jurnal Informatika ITB, Bandung.
- [3] Hopkins, Bruce & Antony, Ranjith. (2003). *Bluetooth for Java*. Appress Inc, USA.
- [4] Kacmarova, Alena. *Internet Chatting Inside Out*, 13 agustus 2009, <www.skase.sk/Volumes/JTL02/DOC/KacmarovaFINAL.doc>.
- [5] Nurhadi. Interaksi *dan Identitas Diri Pemakai Internet Relay Chat (IRC)*, 13 agustus 2009, <[http://arc.ugm.ac.id/files/\(2706-H-2004\).pdf](http://arc.ugm.ac.id/files/(2706-H-2004).pdf)>.
- [6] Shalahudding, M & A.S, Rosa. (2008). *Pemrograman J2ME*. Informatika Bandung, Bandung.
- [7] Yulia & Willyanto Santoso, Leo. (2004). *Studi dan Uji Coba Teknologi Bluetooth Sebagai Alternatif Komunikasi Data Nirkabel*. Jurnal Informatika Universitas Petra, Surabaya.