

## PENGEMBANGAN INFRASTRUKTUR KOMUNIKASI MULTIGROUP PADA WI-FI DIRECT

Bagus Aji Pamungkas Setiyono<sup>1</sup>, Eko Sakti Pramukantoro<sup>2</sup>, Kasyful Amron<sup>3</sup>

<sup>1,2,3</sup>Fakultas Ilmu Komputer, Universitas Brawijaya

Email: <sup>1</sup>baguspamungkas094@gmail.com, <sup>2</sup>ekosakti@ub.ac.id, <sup>3</sup>kasyful@ub.ac.id

(Naskah masuk: 2 Februari 2017, diterima untuk diterbitkan: 3 Maret 2017)

### Abstrak

Wi-Fi Direct merupakan alternatif lain dari penerapan opportunistic network selain ad-hoc. Wi-Fi Direct dapat mudah ditemukan pada perangkat android keluaran terbaru. Hal ini dapat mengatasi permasalahan *ad-hoc* yang tidak bisa digunakan oleh perangkat android keluaran terbaru. Namun Wi-Fi Direct memiliki keterbatasan dikarenakan arsitektur dari Wi-Fi Direct yang terpusat sehingga jangkauan jaringannya tidak sama dengan *ad-hoc*. Oleh karena itu pada penelitian ini dilakukan pengembangan arsitektur komunikasi *multigroup* pada Wi-Fi Direct khususnya perangkat bergerak android dengan melibatkan jembatan antar *group* menggunakan *Legacy Client* serta menguji kinerja dari infrastruktur yang dibangun. Hasil dari pengujian kinerja arsitektur *multigroup* Wi-Fi Direct adalah berupa waktu *delay* penemuan perangkat, pembentukan *group*, pengiriman pesan *group* dan pengiriman pesan *multigroup*. Berdasarkan hasil penelitian berdasarkan skenario yang dilakukan dapat disimpulkan bahwa pemindaian perangkat mencatat waktu tercepat yaitu 4.041 detik dan waktu terlama 17.184 detik, pembentukan *group* mencatat waktu tercepat 1.524 detik dan waktu terlama 2.511 detik, pengiriman pesan *group* mencatat waktu tercepat 2.637 detik dan waktu terlama 6.126, dan untuk pengiriman pesan *multigroup* tidak dapat dilakukan. Komunikasi antar *Group Owner* tidak dapat dilakukan secara langsung karena terjadi konflik alamat IP.

**Kata kunci:** Wi-Fi Direct, *Group* Wi-Fi Direct, *Multigroup* Wi-Fi Direct, Opportunistic Network, Android.

### Abstract

Wi-Fi Direct is an alternative for the application of opportunistic network in addition to ad-hoc. Wi-Fi Direct can be easily found on the latest android device. It can overcome the problems of *ad-hoc* that can not be used by the latest android device. However Wi-Fi Direct have the limitations due to the architecture of Wi-Fi Direct centralized so that network coverage is not the same as the *ad-hoc*. Therefore, in this research, the development of communications architecture *Multigroup* on Wi-Fi Direct, especially mobile android involving bridges between groups using the *Legacy Client* and test the performance of the infrastructure. The results of the performance testing architecture *Multigroup* Wi-Fi Direct is a form of time delay device discovery, group formation, group messaging and message delivery *Multigroup*. Based on the results of research based on scenarios that can be concluded that the scanning device is recording the fastest time is 4.041 seconds and the longest time 17.184 seconds, the formation of the group recorded the fastest time of 1.524 seconds and the longest time 2.511 seconds, sending the message group recorded the fastest time of 2.637 seconds and the longest time 6.126, and for message delivery *Multigroup* can not be done. Communication between *Group Owner* can not be done directly because of an IP address conflict.

**Keywords:** Wi-Fi Direct, *Group* Wi-Fi Direct, *Multigroup* Wi-Fi Direct, Opportunistic Network, Android.

### 1. PENDAHULUAN

Interaksi antar perangkat bergerak sangat dibutuhkan untuk berkomunikasi jarak dekat ataupun jarak jauh. Namun pada saat ini komunikasi yang dilakukan antar perangkat bergerak dapat dikatakan sebagian komunikasinya bergantung pada paradigma berbasis *Access Point* (AP), jaringan seluler, *hotspot* *Wi-Fi* yang mengharuskan perangkat bergerak tersebut memiliki asosiasi dengan stasiun pangkalan umum atau bisa disebut *provider* untuk beroperasi. Tidak dapat disangkal bahwa paradigma tersebut

sampai saat ini nyaman digunakan. Namun pada saat yang sama dengan paradigma tersebut menyebabkan *overhead* yang rumit untuk berkomunikasi dan juga terdapat lokasi dimana paradigma tersebut tidak bisa digunakan pada perangkat bergerak

Perkembangan teknologi yang sangat cepat pada saat ini memungkinkan untuk dapat menyelesaikan permasalahan dalam hal hubungan antar perangkat satu sama lain. Pada teknologi *opportunistic network* memungkinkan untuk menciptakan beberapa perangkat untuk saling terhubung. Salah satu

implementasi dari *opportunistic network* adalah ad – hoc (Conti M.,2013). Namun permasalahan muncul ketika akan menerapkan ad – hoc dalam membangun komunikasi pada android. Salah satu permasalahannya adalah teknologi *ad – hoc* tidak mendukung keluaran terbaru perangkat android untuk kedepannya dan juga diperlukan adanya pengaturan pada perangkat yang tidak dapat dilakukan oleh pengguna biasa (Conti M., 2013). Perlu adanya alternatif lain sebagai penerapan *opportunistic network* untuk membangun jaringan antar perangkat ke perangkat pada perangkat android mengingat ad – hoc sudah memiliki beberapa kekurangan untuk menerapkan hal tersebut. Alternatif lain yaitu menggunakan teknologi Wi-Fi Direct.

Wi-Fi Direct merupakan teknologi yang mudah ditemukan pada perangkat bergerak android keluaran terbaru salah satu contohnya ialah pada Android 4.2 Jelly Bean. Berkembangnya perangkat bergerak khususnya android yang sangat pesat dan didukung dengan sifat dari android yang *open source* memudahkan para pengembang untuk menciptakan teknologi baru berupa aplikasi pada perangkat bergerak android tersebut.

Teknologi Wi-Fi Direct menggunakan arsitektur terpusat atau disebut juga dengan *centralized*. Ketika beberapa perangkat telah terhubung pada pusat tersebut menggunakan Wi-Fi Direct maka akan membentuk sebuah *group* dengan pusat tersebut sebagai *Group Owner* (GO) dan perangkat yang terhubung dengan *Group Owner* sebagai klien. Dalam *group* yang sama masing-masing perangkat dapat berkomunikasi satu sama lain. Namun pengiriman data antar perangkat harus diatur pengalamatannya dalam *group* tersebut.

Dalam satu *group* tidak memungkinkan terdapat lebih dari satu *Group Owner*, *Group Owner* tidak dapat terhubung menggunakan Wi-Fi Direct dengan *Group Owner* lainnya, dan *Group Owner* juga tidak dapat memposisikan dirinya sebagai klien Wi-Fi Direct pada *group* lainnya. Dengan peraturan Wi-Fi Direct tersebut maka satu *group* tidak dapat terhubung dengan kelompok lainnya. Namun menurut penelitian “Automatic Android – Based Wireless Mesh Network” oleh (Wong, dkk, 2014). Pada penelitian mengenai *Mesh Network* pada android dengan *wifi direct* oleh Wong, dkk ini menunjukkan bagaimana perangkat bergerak Android menggunakan Wi-Fi Direct dapat digunakan secara otomatis untuk membangun koneksi antar perangkat bergerak android secara langsung. Dengan menggunakan layanan jaringan untuk melakukan *broadcast* menggunakan Wi-Fi Direct, Wong, dkk dapat mengirimkan sebuah *soft AP's preshared key* yang dibuat oleh GO kepada perangkat lain. Kemudian perangkat lain tersebut dapat terhubung dengan *AP* yang telah dibuat oleh GO dengan

menggunakan layanan Wifi. Kemudian perangkat yang telah terhubung tersebut membuat sebuah *group* baru menggunakan Wi-Fi Direct.

Dengan demikian perlu dilakukan penelitian lebih lanjut mengenai pengiriman data dalam *group* Wi-Fi Direct dan membuat *group* Wi-Fi Direct dapat terhubung dengan *group* lainnya dan membentuk infrastruktur *multigroup* Wi-Fi Direct.

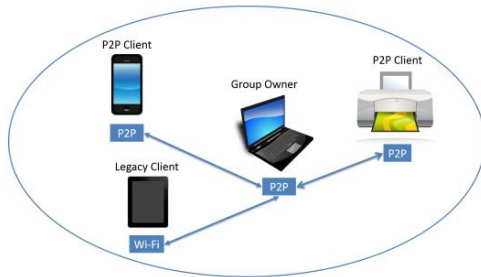
## 2. WI-FI DIRECT

Wi-Fi Direct adalah sebuah teknologi yang didefinisikan oleh Wi-Fi Alliance dengan tujuan perangkat dapat terhubung secara langsung *Device to Device* (D2D) dengan cepat untuk melakukan perintah – perintah seperti pengiriman data, sinkronisasi, dll. Wi-Fi Direct tidak memerlukan penggunaan jalur akses terpusat beda halnya dengan Wifi tradisional. Sebaliknya, perangkat menggunakan “Software Access Point” (Soft AP) (CAMPS, 2013).

Wi-Fi Direct dibuat dengan tujuan untuk komunikasi D2D(*Device to Device*) antara perangkat. Komunikasi antara perangkat pada Wi-Fi Direct terjadi dalam satu *group* dimana didalam *group* tersebut ada yang bertindak sebagai *Group Owner* (GO) sedangkan perangkat lainnya bertindak sebagai klien. Peran sebagai GO dibentuk oleh proses negosiasi pada saat pembentukan *group*. Setelah GO terpilih maka peran masing – masing klien tidak akan berubah selama sesi seluruh *group*. Ketika GO meninggalkan *group* maka klien akan terputus dan melakukan proses pencarian perangkat lagi untuk membentuk *group* baru.

*Group* pada Wi-Fi Direct bekerja mirip halnya dengan *Wi-Fi BSS* infrastruktur yang beroperasi pada saluran tunggal dimana setiap klien pada *group* dapat berkomunikasi. GO juga bertugas secara berkala untuk mengiklankan *group*nya sehingga memungkinkan perangkat lain untuk menemukan dan bergabung pada *group* tersebut. Pada Wi-Fi Direct GO merupakan perangkat yang menerapkan seperti fungsi *Access Point* (AP) dalam *group* P2P seperti bertanggung jawab atas *group*, termasuk pengendalian perangkat yang diizinkan untuk bergabung selama sesi *group* tersebut seperti yang ditunjukkan pada gambar 2.1. Perangkat yang bertugas menjadi GO akan membuat sebuah *Access Point* (AP). AP yang dibuat oleh GO akan memudahkan perangkat lain yang ingin terhubung pada GO tetapi tidak mendukung Wi-Fi Direct. Perangkat tersebut dapat dikatakan sebagai *Legacy Client* (LC). *Legacy Client* merupakan perangkat Wi-Fi konvensional yang tidak mendukung Wi-Fi Direct dan melihat GO sebagai *Wi-Fi Access Point*(AP) tradisional. Secara umum LC tidak termasuk dalam *group* Wi-Fi Direct dan tidak dapat ditingkatkan fungsionalitasnya. LC

hanya sekedar melihat GO sebagai AP tradisional biasa. Dalam koneksi antara P2P klien ataupun LC kepada GO, GO akan menyediakan sebuah DHCP *lease* dan IP Address layaknya sebuah *Access Point* biasa.



**Gambar 2.1 Arsitektur Group Wi-Fi Direct**  
Sumber : (Casetti, 2014)

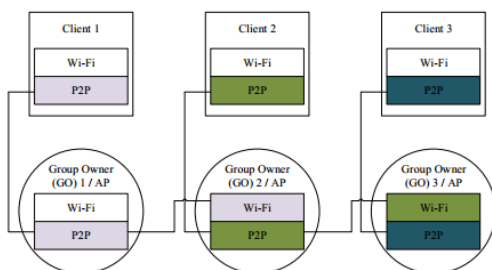
### 3. MULTIGROUP WI-FI DIRECT

Sebuah infrastruktur *multigroup* dapat diterapkan dengan membiarkan perangkat memiliki dua *interface* jaringan P2P secara *virtual*. Dengan cara ini perangkat dapat menjadi sebagai jembatan antar *group* menggunakan alamat MAC yang berbeda pada masing – masing *group*. Namun dalam perangkat android yang tidak di “root”, pengembang tidak dapat membuat hal tersebut.

Berikut beberapa skenario membuat infrastruktur *multigroup* Wi-Fi Direct pada *android* yang tidak bisa dilakukan karena dilarang oleh standar :

1. Perangkat berperan sebagai klien pada satu *group* dan GO pada *group* lainnya
2. Perangkat menjadi GO lebih dari satu *group*
3. Perangkat menjadi klien lebih dari satu *group*

Dengan demikian untuk membentuk sebuah infrastruktur fisik *multigroup* dengan cara memanfaatkan *Legacy Client* pada *group* lain yang akan menjadi GO pada *group* tersebut seperti yang ditunjukkan pada gambar 3.1



**Gambar 3.1 Multigroup pada Wi-Fi Direct**

Sumber: (Casetti, 2014)

### 4. PEMROGRAMAN SOCKET

Sebuah aplikasi berbasis jaringan secara umum aplikasi terdiri dari satu pasang program yang berada pada dua end-system yang berbeda. Ketika aplikasi tersebut dijalankan proses klien dan server akan terbentuk dan saling berkomunikasi dengan membaca atau menulis pada sebuah *socket*. Begitu juga yang terjadi pada *group* Wi-Fi Direct. Setiap paket yang dikirimkan terdapat sumber dan tujuannya. Setiap *node* baik itu *Group Owner* ataupun klien pada Wi-Fi Direct dapat berkomunikasi dengan menggunakan *socket*. Setiap paket yang dikirimkan terdapat informasi berupa isi paket, sumber dan tujuan dari paket tersebut. Dengan demikian setiap pengembang dalam membangun aplikasi berbasis jaringan harus mempertimbangkan penulisan program pada sisi klien ataupun server supaya keduanya dapat saling berkomunikasi.

### 5. METODOLOGI

#### • Studi Literatur

Studi literatur yang dimaksud adalah menyusun kerangka teoritis untuk menyelesaikan permasalahan implementasi infrastruktur komunikasi *multigroup* Wi-Fi Direct pada android. Dengan demikian, pustaka dengan topik – topik di bawah ini perlu dipelajari untuk melaksanakan penelitian ini yaitu tentang Wi-Fi Direct, *Multigroup* Wi-Fi Direct, WifiP2pManager, dan pemrograman socket.

#### • Perancangan

Pada tahapan ini dilakukan perancangan untuk membangun infrastruktur komunikasi *multigroup* Wi-Fi Direct pada *android*. Adapun tahapan perancangan ini terdiri dari membangun infrastruktur komunikasi *multigroup* Wi-Fi Direct yaitu membangun *group* Wi-Fi Direct, membangun komunikasi *group* Wi-Fi Direct, membangun *multigroup* Wi-Fi Direct serta membangun komunikasi *multigroup* Wi-Fi Direct.

#### • Implementasi

Pada tahap implementasi ini, metode yang telah ditawarkan oleh penulis akan diimplementasikan menggunakan perangkat yang ada sesuai dengan kebutuhan sistem yang telah dijelaskan untuk mengembangkan infrastruktur *multigroup* Wi-Fi Direct pada android dimulai dari pemindaian perangkat, pembentukan *group*, komunikasi dalam *group*, dan komunikasi *multigroup*.

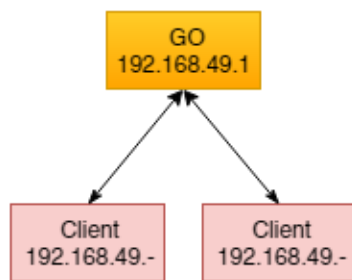
#### • Pengujian

Pengujian dilakukan untuk mengetahui kinerja dari infrastruktur *multigroup* Wi-Fi Direct berdasarkan waktu. Pengujian yang dilakukan sebagai berikut :

1. Pengujian *Delay* Pemindaian Perangkat
2. Pengujian *Delay* Pembentukan *Group*
3. Pengujian *Delay* Pengiriman Pesan dalam *Group*
4. Pengujian *Delay* Pengiriman Pesan *Multigroup*

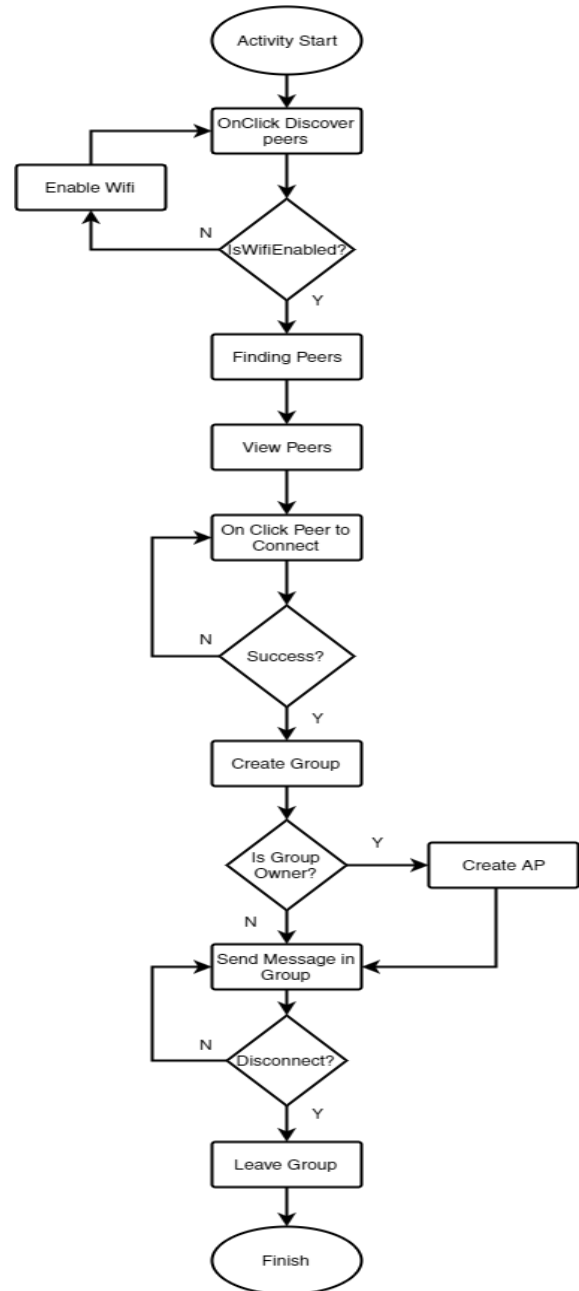
## 6. PERANCANGAN

### • Arsitektur *Group* Wi-Fi Direct



**Gambar 6.1 Rancangan Arsitektur *Group* Wi-Fi Direct**

Pada gambar 6.1 merupakan arsitektur *group* Wi-Fi Direct. Terdapat GO dan 2 klien. GO memiliki alamat IP final static yaitu 192.168.49.1. Dengan demikian setiap *group* yang terbentuk alamat IP GO selalu 192.168.49.1. Kemudian klien akan mendapatkan IP yang diberikan oleh GO yang juga sebagai DHCP server sehingga dalam satu *group* tersebut berada dalam satu jaringan yang sama. Karena arsitektur tersebut berbentuk *centralized* atau terpusat dimana GO akan menjadi pusat dari klien maka ketika klien ingin berkomunikasi dengan klien lainnya akan melalui GO.



**Gambar 6.2 Flowchart Pembentukan *Group***

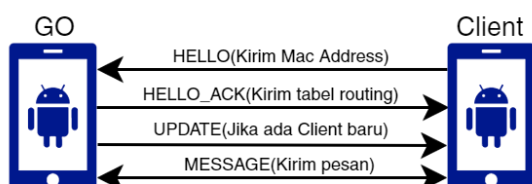
Pada gambar 6.2 merupakan flowchart dari pembentukan *group* pada Wi-Fi Direct. Proses dimulai dari pemindaian perangkat hingga terbentuk sebuah *group* dan perangkat meninggalkan sebuah *group*.

- **Komunikasi dalam Group**

**Tabel 6.1 Informasi Routing Table**

Key	Value		
Mac Address	AllEncompasingP2PClient		
Mac Address	IP Address	GO Mac Address	Client Name

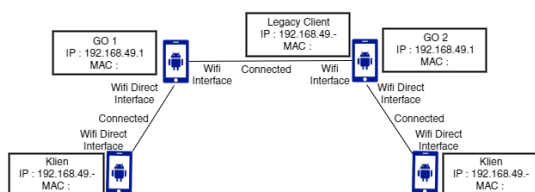
Pada table 6.1 merupakan table informasi *routing table*. Kelas *AllEncompasingP2PClient* akan berisi semua nilai informasi perangkat yang terhubung dengan *Mac Address* yang akan menjadi *key* sedangkan *IP Address*, *GO Mac Address*, dan *Client Name* merupakan nilai yang lain.



**Gambar 6.3 Distribusi Routing Table dan Komunikasi dalam Group**

Pada gambar 6.3 merupakan distribusi *routing table* dan komunikasi dalam *group*. Pengiriman pesan akan menggunakan pengiriman TCP. Pesan merupakan salah satu bentuk paket yang dikirimkan dan diberi nama *MESSAGE*. Selain itu terdapat paket lain yang dikirimkan kepada perangkat yang terhubung dalam satu *group*. Paket tersebut adalah *HELLO*, *HELLO\_ACK*, dan *UPDATE*. Paket *HELLO* berisi *Mac Address* yang akan dikirimkan oleh klien kepada GO. Kemudian GO akan mengambil informasi *Mac Address*, *IP Address*, dan *Client Name* melalui *TCP Socket* yang dijalankan dan memasukkan informasi tersebut pada *routing table*. *HELLO\_ACK* berisi *routing table* yang akan dikirimkan pada klien yang baru terhubung pada GO. *UPDATE* berisi tentang informasi jika ada klien baru yang terhubung dan GO akan mengirimkan informasi tersebut kepada kliennya masing – masing.

- **Arsitektur Multigroup Wi-Fi Direct**

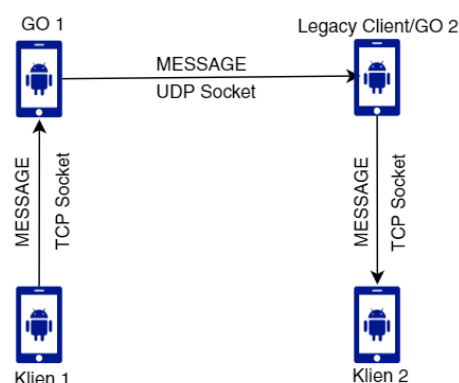


**Gambar 6.4 Rancangan Arsitektur Multigroup Wi-Fi Direct**

Pada gambar 6.4 merupakan rancangan arsitektur *multigroup* yang akan dibuat. Setiap perangkat mempunyai identitas berupa peran perangkat apakah sebagai Klien, *Group Owner* atau *Legacy Client*. Dan juga informasi interface jaringan yang digunakan oleh masing – masing perangkat yaitu *Wi-Direct Interface* atau *Wifi Interface*. Masing – masing peran akan mempunyai informasi yang akan ditampilkan pada aplikasi berupa alamat IP dan alamat MAC. Dengan demikian penulis dapat mengetahui bahwa perangkat yang ada dalam *multigroup* sudah terhubung atau belum.

- **Komunikasi Multigroup**

Pada saat klien 1 mengirimkan paket bertipe *MESSAGE* dalam *group* maka paket tersebut akan diterima oleh GO 1. Ketika GO 1 menerima paket tersebut, nilai dari paket tersebut akan disimpan dalam sebuah variable bertipe *string*. Kemudian variable tersebut akan dikirimkan menggunakan *UDP Socket* dengan tujuan alamat IP dan port yang diatur pada *Legacy Client*. *Socket* dari *Legacy Client* akan terbuka selama perangkat tersebut masih menjadi *Legacy Client*. Paket yang telah diterima oleh *Legacy Client* akan diambil nilainya kemudian akan disimpan pada variable. Nilai dari variable tersebut akan diteruskan oleh GO 2 dan dibuat sebuah paket bertipe *MESSAGE* yang akan dikirimkan secara broadcast pada klien 2. Komunikasi *multigroup* dapat dilihat pada gambar 6.5.



**Gambar 6.5 Rancangan Komunikasi Multigroup Wi-Fi Direct**

- **Pengujian**

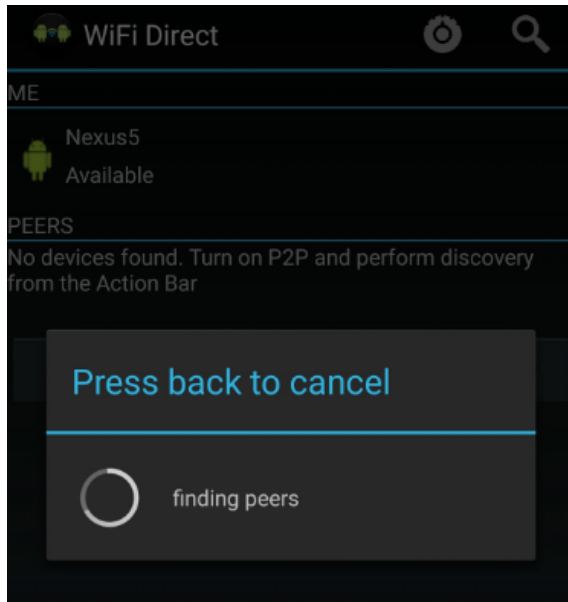
Pada penelitian ini pengujian yang dilakukan yaitu menguji kinerja dari Wi-Fi Direct. Pengujian tersebut meliputi waktu delay pemindaian perangkat, waktu delay pembentukan *group*, waktu delay pengiriman pesan dalam *group*, dan waktu delay pengiriman pesan *multigroup*. Percobaan dilakukan masing – masing sebanyak dua puluh kali percobaan. Waktu yang diperoleh akan ditampilkan dalam bentuk grafik *Cumulative Distribution Function*.

(CDF). CDF merupakan penjumlahan dari setiap kejadian hingga mencapai nilai satu pada akhirnya.

## 7. IMPLEMENTASI

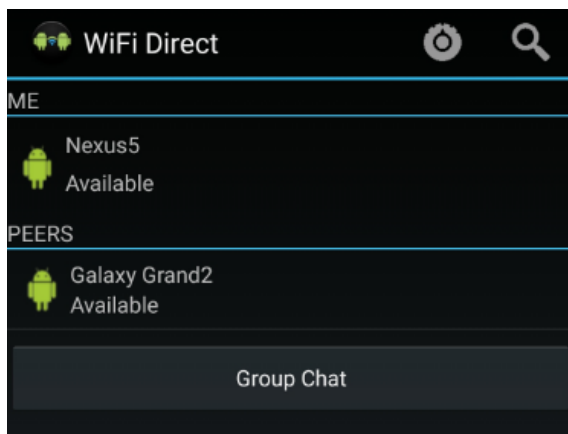
### • Pemindaian Perangkat

Pada gambar 7.1 merupakan tampilan aplikasi ketika tombol *discover* ditekan. Akan ada keterangan *Discovery Initiated* yang menandakan bahwa proses pemindaian berhasil dijalankan.

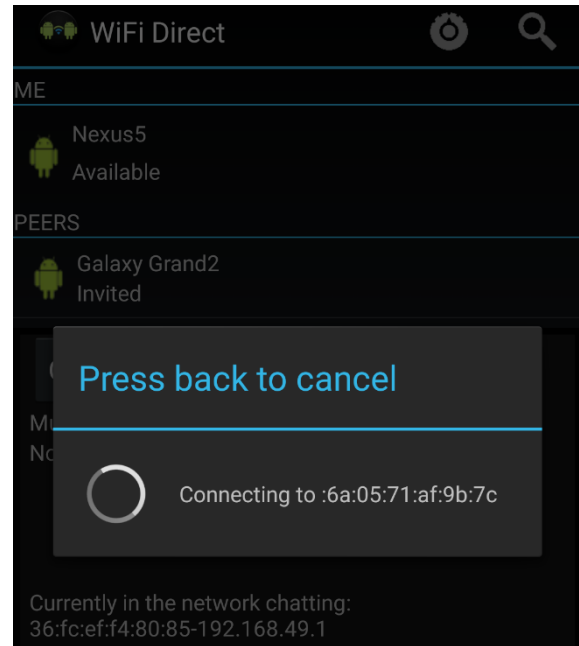


**Gambar 7.1 Proses Pemindaian Perangkat**

Pada gambar 7.2 merupakan tampilan aplikasi ketika proses pemindaian telah selesai dan hasil penemuan akan ditampilkan. Dapat dilihat perangkat tersebut berhasil menemukan perangkat lain dengan nama *Galaxy Grand2* yang statusnya *available*. Itu berarti perangkat *Galaxy Grand2* tidak terhubung dengan perangkat lain menggunakan Wi-Fi Direct.



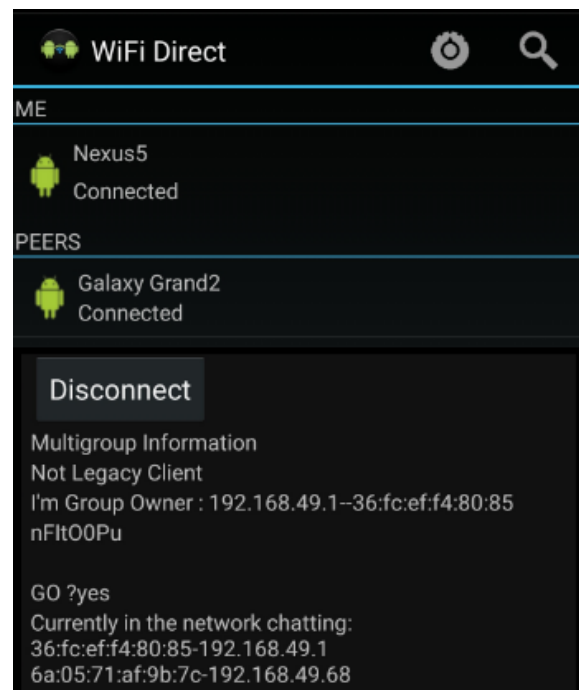
**Gambar 7.2 Menampilkan Hasil Pemindaian**



**Gambar 7.3 Melakukan Proses Permintaan Koneksi dan Pembentukan Group**

### • Permintaan Koneksi dan Membentuk Group

Pada gambar 7.3 merupakan tampilan aplikasi ketika perangkat menekan tombol *connect* untuk terhubung pada perangkat yang telah ditemukan. Dapat dilihat perangkat tersebut mencoba terhubung dengan perangkat *Galaxy Grand2* dengan alamat MAC 6a:05:71:af:9b:7c

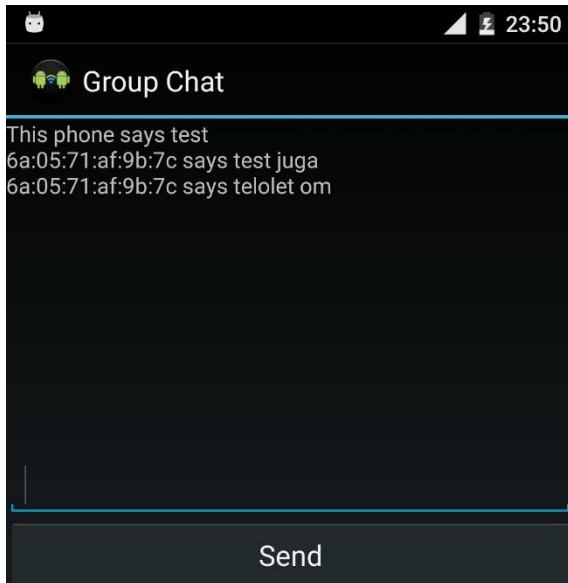


**Gambar 7.4 Terbentuk Group**

Pada gambar 7.4 merupakan tampilan pada aplikasi ketika terbentuk *group* dan perangkat



tersebut menjadi *Group Owner*. Terdapat informasi berupa keterangan perangkat sebagai *Group Owner* dengan tanda *GO ?yes* dan alamat MAC serta alamat IP dari semua perangkat yang ada didalam *group*. Dari informasi tersebut dapat membantu penulis untuk mengetahui perangkat mana saja yang ada dalam *group* tersebut.

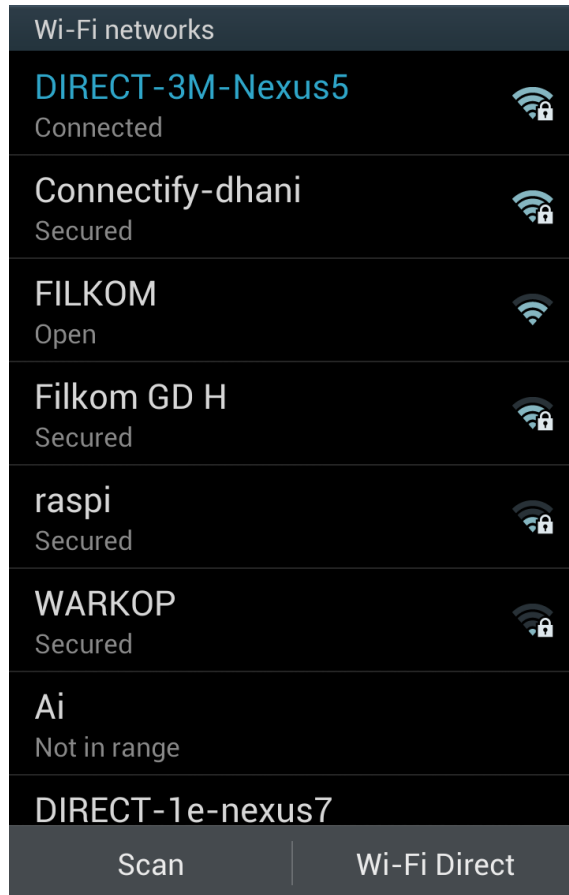


**Gambar 7.5** Aktivitas Perpesanan

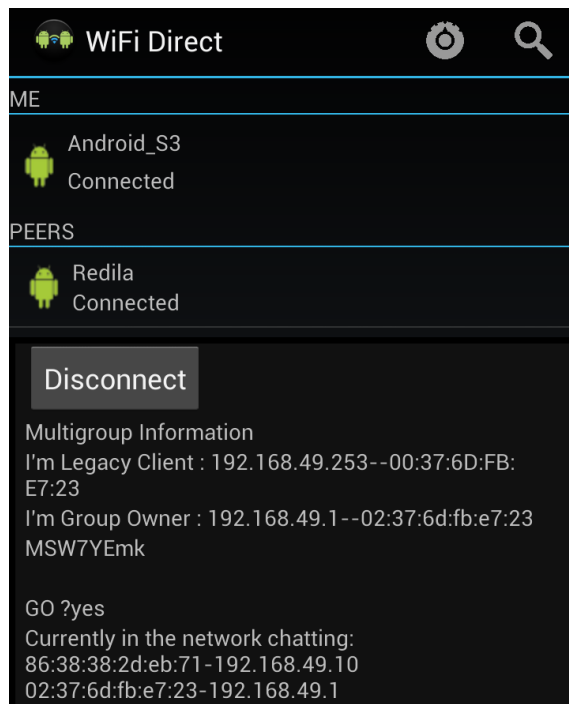
Pada gambar 7.5 merupakan tampilan aplikasi saat aktivitas perpesanan pada perangkat sebagai *Group Owner*. Pengguna dapat mengetahui pesan yang dikirimkan dan juga pesan yang masuk dengan identitas berupa alamat MAC pengirim pesan tersebut.

- **Koneksi *Legacy Client* dan Membentuk *Multigroup***

Pada gambar 7.6 merupakan tampilan intent Wi-Fi pada perangkat. Perangkat tersebut berhasil terhubung dengan *Access Point Group Owner* dengan nama *DIRECT-3M-Nexus5*.

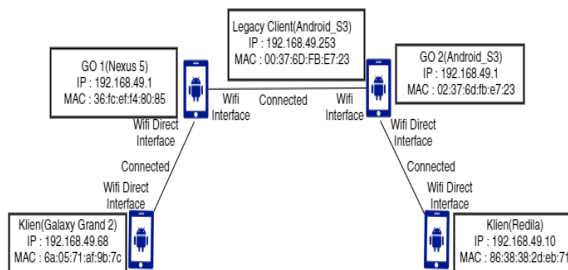


**Gambar 7.6** *Legacy Client* terhubung dengan *Access Point Group Owner* pertama



**Gambar 7.7** *Legacy Client* menjadi *Group Owner* *group* kedua

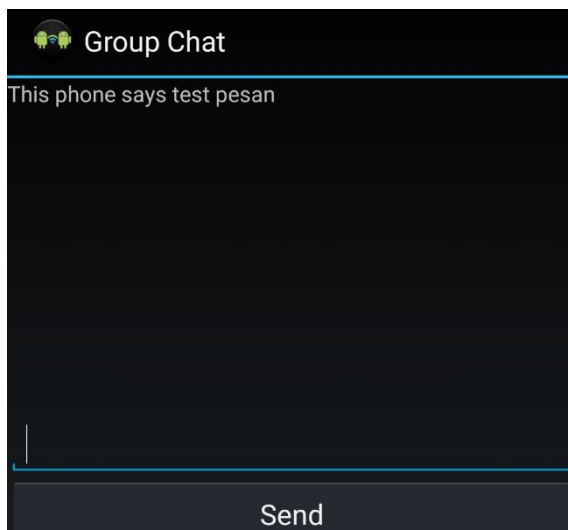
Pada gambar 7.7 merupakan tampilan aplikasi pada perangkat *Legacy Client* yang terhubung dengan *Group Owner group* pertama dan mendapatkan alamat IP 192.168.49.253 dengan menggunakan alamat MAC yaitu 00:37:6D:FB:E7:23. Selain itu *Legacy Client* juga membentuk *group* dan menjadi *Group Owner* pada *group* baru yang dibentuk dengan alamat IP 192.168.49.1 dengan menggunakan alamat MAC yaitu 02:37:6d:fb:e7:23.



**Gambar 7.8 Arsitektur Multigroup**

Pada gambar 7.8 merupakan arsitektur *multigroup* yang berhasil dibangun sesuai dengan perancangan yang dibuat. Terdapat empat perangkat dan dua *group*. Masing – masing *group* terdiri dari dua perangkat. Informasi mengenai alamat IP dan alamat MAC berhasil ditambahkan pada gambar beserta keterangan mengenai *interface* mana yang dipakai oleh perangkat untuk terhubung pada perangkat lainnya.

- **Komunikasi Multigroup**



**Gambar 7.9 Group Owner Mengirim Pesan kepada Legacy Client**

Pada gambar 7.9 merupakan tampilan aplikasi saat aktifitas perpesanan. Tampilan tersebut merupakan tampilan perangkat *Group Owner group* 1 dan mengirimkan pesan teks yaitu “test pesan”. Pesan tersebut akan dikirimkan pada klien *group* 1 dan juga *Legacy Client*.

```
26230-26230/com.ecse414.android.echo E/masuk thread: udp
26230-26598/com.ecse414.android.echo E/socket: dibuat
26230-26598/com.ecse414.android.echo E/kirim pesan: test pesan
```

**Gambar 7.10 Tampilan Log Group Owner saat Mengirim Pesan kepada Legacy Client**

Pada gambar 7.10 merupakan tampilan *Log* pada *android studio monitor* yang menunjukkan bahwa *Group Owner group* 1 saat mengirimkan pesan. *Group Owner group* 1 berhasil membuat *udp socket* dan mengirimkan pesan kepada *Legacy Client*.

```
26195-26261/com.ecse414.android.echo E/UDPReceiver: Dijalankan
26195-26261/com.ecse414.android.echo E/UDPReceiver: socket dibuat
26195-26261/com.ecse414.android.echo E/UDPReceiver: siap menerima pesan
26195-26261/com.ecse414.android.echo E/UDPReceiver: terima pesan
26195-26261/com.ecse414.android.echo E/UDPReceiver: test pesan
```

**Gambar 7.11 Tampilan Log Legacy Client Saat Menerima Pesan dari Group Owner**

Pada gambar 7.11 merupakan tampilan *Log* pada *android studio monitor* yang menunjukkan bahwa *udp socket* telah dibuka pada *Legacy Client* dan siap menerima pesan yang dikirimkan oleh *Group Owner group* 1. Pada kondisi ini *Legacy Client* belum membentuk *group* dan tidak menjadi *Group Owner* dari *group* yang lain. Dapat dilihat pesan yang dicetak pada *Log* merupakan pesan yang berhasil masuk pada *Legacy Client* sesuai dengan yang dikirimkan oleh *Group Owner group* 1.

```
28890-28931/com.ecse414.android.echo E/UDPReceiver: Dijalankan
28890-28931/com.ecse414.android.echo E/UDPReceiver: socket dibuat
28890-28931/com.ecse414.android.echo E/UDPReceiver: siap menerima pesan
```

**Gambar 7.12 Tampilan Log Legacy Client/Group Owner Group Kedua Saat Menerima Pesan dari Group Owner Group Pertama**

Pada gambar 7.12 merupakan tampilan *log legacy client/group owner group kedua* saat menerima pesan dari *group owner group pertama*. Pada kondisi ini *Legacy Client* merupakan *Group Owner* dari *group* 2. *Legacy Client* dapat membuka *UDP socket* dan siap menerima pesan. Tetapi dapat ditunjukkan pada gambar bahwa pesan tidak masuk pada *Legacy Client* sedangkan *UDP Socket* sudah dibuka. Proses menerima pesan tidak berjalan dan menyebabkan pesan tidak masuk pada *Legacy Client* yang juga sebagai *Group Owner*.

Pada gambar 7.13 merupakan tampilan *Log* pada *android studio monitor* ketika *Legacy Client* yang sebagai *Group Owner group* 2 akan mengirimkan pesan pada *Group Owner group* 1(192.168.49.1). Dapat dilihat bahwa perangkat *Legacy Client* yang sebagai *Group Owner group* 2 tidak dapat menjalankan *UDP Socket*. Pada saat akan membuat



socket akan terjadi kesalahan dengan beberapa pesan kesalahan diantaranya yaitu *Address already in use*.

```
01-16 17:07:21.260 24530-24530/com.esee44.android.echo E/pesan: pesan dari legacy client
01-16 17:07:21.265 24530-24530/com.esee44.android.echo W/initialisasi: thread UDP
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: java.net.BindException: bind failed: EADDRINUSE (Address already in use)
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: at libcore.io.IoBridge.bind(IoBridge.java:19)
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: at java.net.PlainDatagramSocketImpl.bind(PlainDatagramSocketImpl.java:68)
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: at java.net.DatagramSocket.createSocket(DatagramSocket.java:133)
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: at java.net.DatagramSocket.<init>(DatagramSocket.java:78)
01-16 17:07:21.265 24530-27215/com.esee44.android.echo W/System.err: at com.esee44.android.echo.receiver.udp.UDPSender.run(UDPSender.java:49)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: at java.lang.Thread.run(Thread.java:760)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: Caused by: libcore.io.IOException: bind failed: EADDRINUSE (Address already in use)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: at libcore.io.Poll$bind(Native Method)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: at libcore.io.ForwardingIo$bind(ForwardingIo.java:39)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: at libcore.io.IoBridge.bind(IoBridge.java:17)
01-16 17:07:21.270 24530-27215/com.esee44.android.echo W/System.err: ... 5 more
```

**Gambar 7.13 Tampilan Log Legacy Client/Group Owner Group Kedua Saat Mengirim Pesan kepada Group Owner Group Pertama**

## 8. PENGUJIAN

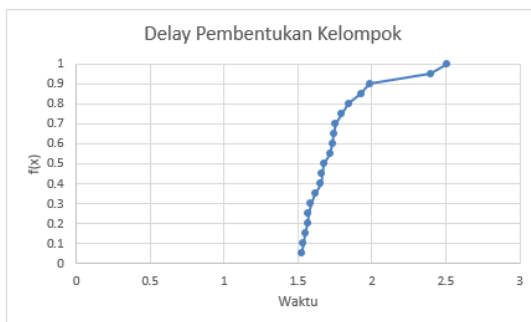
### • Pengujian Delay Pemindaian Perangkat



**Gambar 8.1 Grafik CDF Delay Pemindaian Perangkat**

Pada gambar 8.1 dapat disimpulkan waktu pemindaian perangkat dimulai dari paling cepat 4.041 detik dan waktu terlama 17.184 detik. Garis grafik diatas semakin melebar menandakan bahwa dalam dua puluh kali percobaan terjadi ketidakstabilan waktu pemindaian perangkat mulai dari  $f(x) = 0.9$ .

### • Pengujian Delay Pembentukan Group

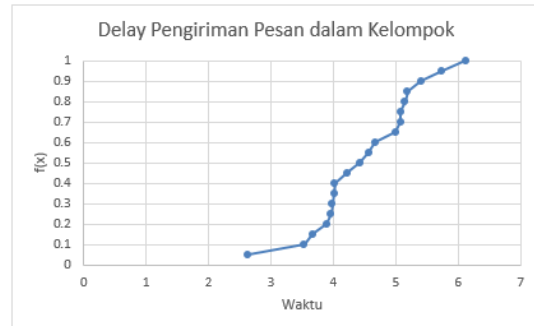


**Gambar 8.2 Grafik CDF Delay Pembentukan Group**

Pada gambar 8.2 dapat disimpulkan waktu pembentukan group tercepat didapatkan waktu 1.524 detik. Kemudian dalam dua puluh kali percobaan yang dilakukan didapatkan waktu terlama dalam

pembentukan group adalah 2.511 detik. Garis pada grafik tersebut mulai melebar pada  $f(x) = 0.1$  dan terus melebar hingga nilai  $f(x) = 1$ . Ini menandakan dalam dua puluh kali percobaan terdapat ketidakstabilan pada persentase awal. Namun lebar waktu tercepat dan terlambat tidak terlalu besar hanya terpaut sekitar satu detik.

### • Pengujian Delay Pengiriman Pesan dalam Group



**Gambar 8.3 Grafik CDF Delay Pengiriman Pesan dalam Group**

Pada gambar 8.3 dapat disimpulkan yaitu waktu pengiriman pesan dalam group tercepat 2.637 detik kemudian dilanjutkan dengan jarak waktu kedua yang cukup jauh yaitu 3.353 detik dan terus melebar hingga waktu terlama yang didapatkan adalah 6.126 detik. Dengan demikian pengiriman pesan berupa text menggunakan protocol TCP dalam group Wi-Fi Direct waktu yang dibutuhkan cukup lama dan kurang stabil dari dua puluh kali percobaan yang dilakukan.

### • Pengujian Delay Pengiriman Pesan Multigroup

Pada implementasi yang dilakukan pengiriman pesan Multigroup Wi-Fi Direct tidak dapat dilakukan. Pada percobaan Group Owner group 1 mengirimkan pesan pada Legacy Client/Group Owner group 2, pesan berhasil dikirimkan oleh Group Owner group 1 menggunakan UDP Socket namun pada saat Legacy Client/Group Owner group 2 telah membuat socket dan siap menerima pesan, pesan tidak bisa diterima. Hal ini dikarenakan dua Group Owner tidak dapat berkomunikasi secara langsung karena terjadi konflik alamat IP. Group Owner group 2 (192.168.49.253) yang juga sebagai Legacy Client group 1 akan membuang pesan yang masuk dikarenakan paket yang dikirimkan mempunyai sumber alamat IP yaitu 192.168.49.1 yang sama dengan alamat IP perangkat itu sendiri. Jika dijalankan fungsi `getlocalIPaddress` maka perangkat Legacy Client yang juga sebagai Group Owner group 2 akan memiliki dua alamat IP yaitu 192.168.49.1 dan 192.168.49.253. Hal ini terjadi juga dalam percobaan selanjutnya yaitu ketika Legacy Client/Group Owner group 2 akan

mengirimkan pesan kepada *Group Owner group 1*. Ketika akan menjalankan *UDP Socket* dengan alamat IP tujuan pesan yaitu *Group Owner group 1* (192.168.49.1) akan tampil pesan kesalahan pada *adroid monitor* yang salah satunya adalah *Address already in use*. Pesan gagal dikirimkan dikarenakan pesan dikirimkan pada *local loop* dan tidak kepada *Wifi Interface*.

## 9. KESIMPULAN

1. Dalam *delay* pemindaian perangkat tercatat waktu tercepat yaitu 4.401 detik dan waktu terlambat yaitu 17.184 detik. Dalam grafik CDF pada  $f(X) = 0$  hingga 0.7 waktu yang dihasilkan antara 4 sampai 5 detik kemudian pada  $f(x) = 0.8$  hingga 1 waktunya melebar hingga waktu terlambat yaitu 17.184 detik.
2. Dalam *delay* pembentukan *group Wi-Fi Direct* tercatat waktu tercepat yaitu 1.524 detik dan waktu terlambat yaitu 2.511 detik. Dalam grafik CDF dapat diketahui waktu yang dibutuhkan untuk pembentukan *group* dengan scenario yang dilakukan cukup stabil dari  $f(x)$  0 hingga 1 dengan waktu delay dari waktu tercepat dan terlambat hanya 1 detik.
3. Dalam *delay* pengiriman pesan dalam *group Wi-Fi Direct* tercatat waktu 2.637 detik dan waktu terlambat 6.126 detik. Dalam grafik CDF dapat diketahui waktu mulai melebar pada  $f(x) = 0.1$  dan terus bergerak melebar hingga  $F(x) = 1$  sampai waktu terlambat.
4. Dalam pengujian *delay* pengiriman pesan multigroup Wi-Fi Direct scenario yang dirancang tidak dapat dilakukan dikarenakan komunikasi antar *Group Owner* tidak dapat dilakukan. *Group Owner* tidak dapat berkomunikasi secara langsung karena terjadinya konflik alamat IP antar *Group Owner*. Pada saat *Legacy Client/Group Owner group 2* membuang pesan yang dikirimkan oleh *Group Owner group 1* karena sumber alamat IP adalah 192.168.49.1 yang sama dengan alamat IP *Group Owner group 2*. Perlu adanya mekanisme routing baru yang dapat mengatur permasalahan sumber alamat IP paket pesan.

## 10. SARAN

1. Perlu diadakan penelitian mengenai cara pengiriman data yang berbeda dalam *group Wi-Fi Direct* atau membandingkan cara pengiriman yang telah dibuat oleh penulis dengan cara pengiriman lainnya.
2. Perlu diadakan penelitian lebih lanjut tentang komunikasi *multigroup Wi-Fi Direct* dengan memanfaatkan infrastruktur komunikasi yang telah dibuat oleh penulis sehingga dapat menyelesaikan permasalahan pengiriman pesan

*multigroup* pada penelitian ini. Dengan menerapkan mekanisme *routing* baru yang dapat mengatasi konflik alamat IP antar *Group Owner* dapat menciptakan sebuah komunikasi *multigroup Wi-Fi Direct*. Mekanisme *routing* tersebut adalah *Content Centric Network (CCN)*. Penelitian mengenai *Content Centric Network (CCN)* dapat dilakukan pada penelitian selanjutnya guna mengembangkan dari penelitian ini.

## 11. DAFTAR PUSTAKA

- Camps-Mur, D., Garcia-Saavedra, A., & Serrano, P. (2013). *Device to device communications with WiFi Direct: overview and experimentation*. Madrid: IEEE.
- Casetti, C., Chiasserini, C., Pelle, L. C., Valle, C. D., Duan, Y., & Giaccone, P. (2014). *Content-centric Routing in Wi-Fi Direct Multigroup Networks*. Torino: Department of Electronics and Telecommunications, Politecnico di Torino.
- Conti, M., Delmastro, F., Minutiello, G., & Paris, R. (2013). *Experimenting opportunistic networks with WiFi Direct*. Pisa: Institute of Informatics and Telematics (IIT).
- Funai, C., Tapparello, C., & Heinzelman, W. (2015). *Supporting Multi-hop Device-to-Device Networks Through WiFi Direct Multi-group Networking*. Rochester, NY: Department of Electrical and Computer Engineering, University of Rochester.
- Jung, W.-S., Ahn, H., & Ko, Y.-B. (2014). *Designing Content-Centric Multi-hop Networking over Wi-Fi Direct on Smartphones*. Korea: Graduate School of Computer Engineering, Graduate School of NCW, Ajou University.
- Marinho, R. P., Menegato, U. B., & de Oliveira, R. A. (2015). *Mobile Devices Routing Using Wi-Fi Direct Technology*. MG - Brazil: Laboratório Imobilis, Universidade Federal de Ouro Preto, Ouro Preto.
- Wong, P., Varikota, V., Nguyen, D., & Abukmail, A. (2014). *Automatic Android-based Wireless Mesh Networks*. Clear Lake, Houston: School of Science and Computer Engineering, University of Houston.