

# Pembuatan Aplikasi *Messenger* Berbasis Android

Aldi Renaldi Gunawan<sup>1</sup>, Justinus Andjarwirawan<sup>2</sup>, Yulia<sup>3</sup>.

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra  
Siwalankerto 121-131 Surabaya

Telp.+6231 8439040, Fax. +6231 8436418

E-mail : aldirenaldigunawan@gmail.com<sup>1</sup>, justin@petra.ac.id<sup>2</sup>, yulia@petra.ac.id<sup>3</sup>

## ABSTRAK

Perkembangan teknologi informasi sekarang ini berkembang dengan sangat pesat. Salah satunya adalah penggunaan aplikasi *messenger* pada *mobile device*. Saat ini *messenger* merupakan salah satu aplikasi yang digunakan untuk berkomunikasi oleh masyarakat luas. Pembuatan skripsi ini, bertujuan untuk membuat aplikasi *messenger* yang dapat berkomunikasi antar *mobile device* khususnya *device* dengan sistem operasi Android.

Dalam pembuatan aplikasi *messenger*, terdapat *server side* yang terdiri dari *server* Openfire yang berfungsi sebagai *protocol* didalam pertukaran pesan dan Apache *server* yang berfungsi untuk menampung data-data pengguna. Serta *client side* yang terdiri dari tampilan *user interface* pada aplikasi. Aplikasi ini menyediakan fitur *login*, *registrasi*, *chatting*, *create group* dan *invite friends or group*.

Dari hasil pengujian terhadap beberapa Android *device* dengan versi dan merek yang berbeda-beda antara lain, LG G4, LG G2, Samsung Galaxy S2, Sony Xperia SP, Sony Xperia Z3 *compact*, aplikasi berjalan dengan baik pada semua *device*. Namun untuk tampilan, posisi dari *layout* seperti *textview*, *imageview*, *videoview* dan juga *listview* tidak berjalan dengan baik karena menggunakan satuan *density pixel* (DP).

Sehingga ukuran untuk setiap layar *device* yang memiliki *pixel per inch* (PPI) yang berbeda akan menampilkan hasil yang berbeda. Serta pada *device* Samsung Galaxy S2 dengan Android versi 4.1.2 tidak ditemukan *popup menu button* yang berada pada menu *friends*.

**Kata Kunci:** Android, *Messenger*, *Mobile Device*, *Openfire*.

## ABSTRACT

*The development of information technology is currently growing very rapidly. One of which is how to use messenger application on mobile device. Nowadays, messenger is one of the applications that are used to communicate by most people. This thesis aims to create a messenger application that can communicate between mobile devices especially devices with the Android operating system.*

*In the design and implementation of this application, there is a server side comprising: Openfire server that serves as a protocol in the exchange of messages and the Apache server that serves to accommodate the user data. As well as the client side comprising: user interface display applications. This application provides features such as login, registration, chat, create groups and invite friends or group.*

*From the results of tests on various Android devices with different versions and brands like LG G4, LG G2, Samsung Galaxy S2,*

*Sony Xperia SP and Sony Xperia Z3 Compact the application run well on all devices. But for the view, the position of the layout*

*as TextView, ImageView, videoview and also listview do not go well because of using density pixel (DP) so the size of each display device which has different pixel per inch (PPI) will show different results.*

*In addition on the device Samsung Galaxy S2 with Android 4.1.2 version popup menu button located on the menu friends is not found.*

**Keywords:** Android, *Messenger*, *Mobile Device*, *Openfire*.

## 1. PENDAHULUAN

Interaksi antar manusia semakin meningkat dan luas dari tahun ke tahun. Meningkatnya interaksi antar manusia ditandai dengan keharusan untuk memiliki *smartphone* untuk dapat membantu manusia agar dapat berinteraksi dengan manusia yang lainnya. Bentuk interaksi ini dapat berupa komunikasi dan juga pertukaran informasi. Pertukaran informasi ini dapat berupa pesan singkat, suara (telepon), video dan juga gambar yang dapat ditemukan pada beberapa *instant messenger* seperti *Line messenger*, *Blackberry messenger*, dan *WhatsApp messenger*.

Pertukaran informasi yang diinginkan oleh manusia ialah pertukaran yang dilakukan secara cepat baik dalam pengiriman maupun menerima sebuah informasi. Menurut *Juniper Research* pada tanggal 21 Juni 2011, pengguna *instant messenger* akan bertambah sebanyak tiga kali lipat menjadi 1.3 miliar pada tahun 2016. Banyaknya pengguna *instant messenger* saat ini membuat kebutuhan pengguna semakin meningkat. *Instant messenger* tidak hanya digunakan untuk melakukan *chatting* saja, tetapi diperlukan adanya fitur-fitur yang menarik sehingga mendukung komunikasi antar pengguna lebih baik lagi [2].

Dari permasalahan di atas, aplikasi *messenger* berbasis Android ini dibuat tidak hanya dapat melakukan pertukaran pesan singkat berbasis teks antar pengguna (*chatting*), namun juga terdapat beberapa fitur-fitur tambahan antara lain, pengiriman pesan antar pengguna dalam bentuk suara (*voice chat*), pengiriman pesan singkat dari satu pengguna terhadap beberapa pengguna (*multi unicast message*) yang berupa teks.

Adapun interaksi dari satu pengguna terhadap beberapa pengguna yang tergabung di dalam sebuah grup (*group chat*) berupa teks, pengguna juga dapat berbagi file gambar dan *video*. Aplikasi ini memiliki fitur *time line* yang menunjukkan sebuah aktifitas dari pengguna.

### 1.1 Perumusan Masalah

Perumusan masalah dalam skripsi ini adalah:

- Bagaimana mendesain tampilan yang menarik dan mudah untuk dioperasikan oleh pengguna.
- Bagaimana pengguna dapat mengakses aplikasi.

- Bagaimana membuat fitur multi unicast message agar dapat mengirim pesan berupa teks.
- Bagaimana pengguna dapat melakukan group chat di mana sebuah pengguna seorang pengguna akan bergabung di dalam suatu kelompok atau grup yang terdiri dari beberapa pengguna untuk melakukan interaksi yang berupa pengiriman pesan.

## 1.2 Tujuan Skripsi

Tujuan dari skripsi ini adalah membuat aplikasi *messenger* berbasis Android yang tidak hanya dapat melakukan pengiriman pesan berupa teks, namun juga pengguna dapat melakukan pengiriman pesan berupa suara (*voice chat*), gambar dan *video*.

## 1.3 Ruang Lingkup

Ruang lingkup dibatasi pada:

- Aplikasi dibuat menggunakan bahasa pemrograman Java atau native application.
- Fitur-fitur utama dari aplikasi ini:
  - Chatting, pengiriman pesan antar pengguna bisa berupa teks, gambar, video, free call, dan video call.
  - Multi unicast message, pengiriman pesan dari seorang pengguna kepada satu atau banyak pengguna-pengguna lain yang ingin dituju. Pengiriman pesan ini berupa teks.
  - Group chat, sebuah fitur di mana seorang pengguna yang bergabung di dalam sebuah grup yang dapat berinteraksi dengan beberapa pengguna yang ikut bergabung di dalam grup tersebut.
  - Time line, fitur di mana aktifitas dari seorang pengguna dapat dilihat oleh pengguna lain yang ada di dalam daftar friends.
  - Status pesan, fitur di mana pengguna dapat mengetahui apakah sebuah pesan yang dikirimkan sudah diterima, dibaca atau belum.
- Setiap pengguna diharuskan mendaftarkan diri dan juga melakukan login untuk mendapatkan akses ke aplikasi.
- Protocol yang digunakan adalah Extensible Messaging and Presence Protocol (XMPP) dan Smack API 4.1.
- Menggunakan encoding UTF-8 pada XML agar aplikasi dapat mengenali karakter sebuah huruf yang terkandung di dalam sebuah pesan.
- Software yang digunakan dalam pembuatan aplikasi:
  - Android Software Development Kit (SDK)
  - Android Development Tools (ADT)
  - Eclipse IDE
  - Database MySQL

## 2. LANDASAN TEORI

### 2.1 Android

Android adalah sistem operasi ponsel berbasis Linux yang dikembangkan oleh Google. Android adalah unik karena Google

secara aktif mengembangkan *platform* tetapi memberikan secara gratis untuk produsen *hardware* dan operator telepon yang ingin menggunakan Android pada perangkat mereka. Google membentuk kelompok *hardware*, *software*, dan perusahaan telekomunikasi yang disebut *Open Handset Alliance* dengan tujuan memberikan kontribusi bagi pembangunan Android. Sebagian besar anggota juga memiliki tujuan untuk membuat uang dari Android, baik dari menjual telepon, layanan telepon, atau aplikasi *mobile* [4].

### 2.2 Eclipse

Eclipse adalah sebuah *Integrated Development Environment* (IDE) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform*. Bahasa pemrograman utama dari Eclipse sendiri adalah Java, namun dapat juga mendukung pengembangan aplikasi dengan bahasa pemrograman lainnya, seperti C/C++, Perl, dan PHP.

### 2.3 Extensible Messaging and Presence Protocol (XMPP)

*Extensible Messaging and Presence Protocol* (XMPP) merupakan aturan yang dibuat untuk berkomunikasi secara *real-time*. XMPP diciptakan oleh Jeremie Miller pada tahun 1998, disempurnakan dalam komunitas Jabber *open-source* pada tahun 1999 dan 2000, dan diresmikan oleh *Internet Engineering Task Force* (IETF) pada tahun 2002 dan 2003, sehingga publikasi XMPP RFC 3920 dan RFC 3921 pada tahun 2004 yang mewakili definisi yang paling lengkap dari XMPP pada waktu itu dan pada tahun 2011 RFC dari XMPP diperbaharui menjadi RFC 6120 [1].

### 2.4 Smack API

Smack merupakan XMPP (Jabber) *client library* bersifat *opensource* yang ditujukan untuk *instant messaging*. Dapat digunakan untuk mengintegrasikan XMPP kedalam sebuah aplikasi. *Library* ini dapat diintegrasikan dengan pemrograman berbasis Java. Smack diproduksi oleh *developer* yang sama dengan *server openfire* dan lisensi yang dimiliki ada di bawah naungan Apache. Pada pengembangannya, muncul Android Smack API (aSmack) dimana yang ditujukan kepada pemrograman berbasis Android pada versi-versi awal. Namun pada Smack versi 4.1.0 dikembangkan dan digabung dengan *library* aSmack sehingga pada versi ini Java *native* untuk pemrograman Android dapat menggunakan satu *library* saja.

### 2.5 Openfire

Openfire adalah *real-time collaboration* (RTC) *server* berlisensi di bawah naungan Apache. Openfire merupakan protocol yang bersifat terbuka sehingga banyak digunakan untuk *instant messaging*, XMPP (juga disebut Jabber). Openfire sangat mudah untuk digunakan dan diatur, tapi menawarkan keamanan dan kinerja yang sangat baik [3].

### 2.6 Sinch

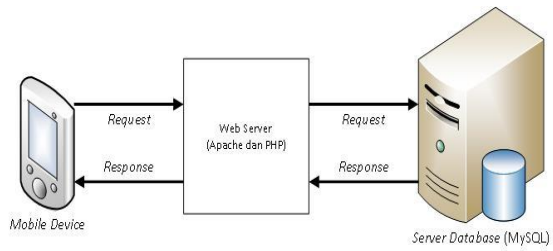
Sinch merupakan sebuah layanan berbasis *cloud*, *platform* komunikasi *mobile* yang dapat melakukan panggilan, pengiriman pesan, dan verifikasi ke aplikasi lebih mudah. Sinch didirikan pada Mei 2014 oleh Andreas Bernström di Stockholm dan San Francisco, tim Sinch memiliki lebih dari 100 tahun pengalaman pembangunan di industri suara.

Misi dari Sinch adalah untuk memudahkan bagi pengembang untuk menambahkan panggilan suara, pengiriman pesan, dan verifikasi untuk aplikasi. Untuk pengembang, Sinch mempertahankan infrastruktur komunikasi yang kompleks, untuk mengintegrasikan SDK yang digunakan [5].

### 3. DESAIN SISTEM

#### 3.1 Sistem Arsitektur

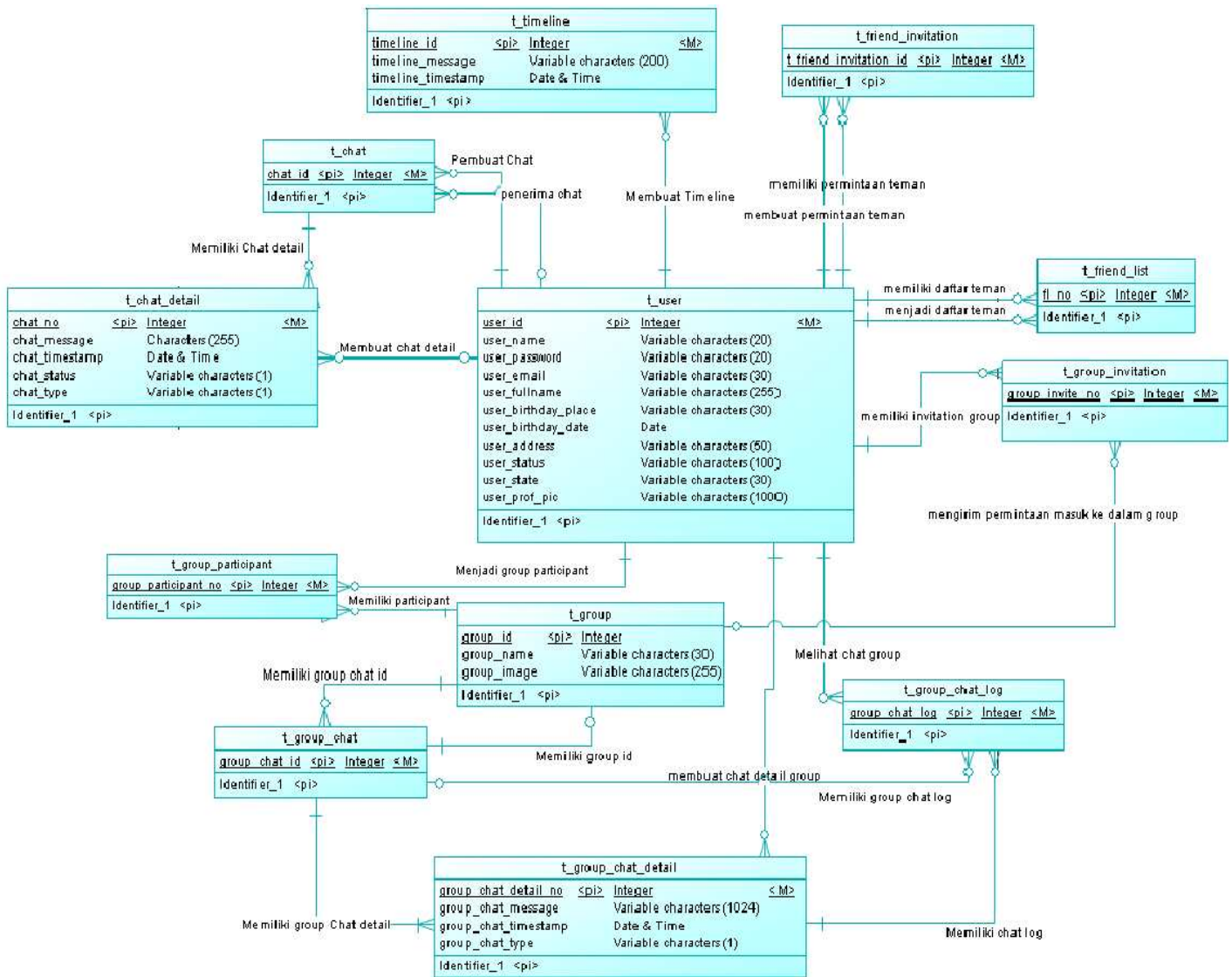
Sistem arsitektur yang digunakan dalam aplikasi ini yaitu *device* yang merupakan *client side* akan melakukan *request* menggunakan metode GET dan POST pada *file php* yang terdapat pada *web server* dan *web server* akan melakukan *request* terhadap *database* untuk mengirim *data* kembali pada *client* yang melakukan *request*. Sistem arsitektur aplikasi dapat dilihat pada Gambar 1.



Gambar 1. Diagram Arsitektur Sistem

#### 3.2 Conceptual Model Diagram (CMD)

Gambar 2, merupakan *conceptual model diagram* yang menunjukkan relasi antar data yang digunakan didalam aplikasi yang telah dibuat.



Gambar 2. Entity Relationship Diagram (ERD)

## 4. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan secara lengkap tampilan-tampilan layar pada program aplikasi *messenger*.

### 4.1 Pengujian Halaman Login

Halaman *login* ini merupakan halaman yang pertama kali ditampilkan oleh sistem pertama kali aplikasi dijalankan.



Gambar 3. Halaman Login

Pada Gambar 3, pengguna diwajibkan memasukkan *username* dan *password* ketika melakukan *login*. Sistem akan melakukan pengecekan ketika *input* yang dimasukkan oleh pengguna benar, maka pengguna akan menuju *menu* utama. Ketika gagal, *system* akan menampilkan pesan gagal.

### 4.2 Pengujian Halaman Sign Up

Halaman *signup* akan ditampilkan ketika tombol *sign up* ditekan, dengan tujuan untuk mendaftarkan *username* yang nantinya digunakan untuk melakukan *login*. Pengguna yang ingin mendaftarkan *username* akan dihadapkan pada tampilan *form sign up* yang hanya menampilkan *form username*, di mana pengguna diwajibkan memasukkan *username* dan kemudian sistem akan melakukan pengecekan terhadap *input* yang dilakukan oleh *user*.

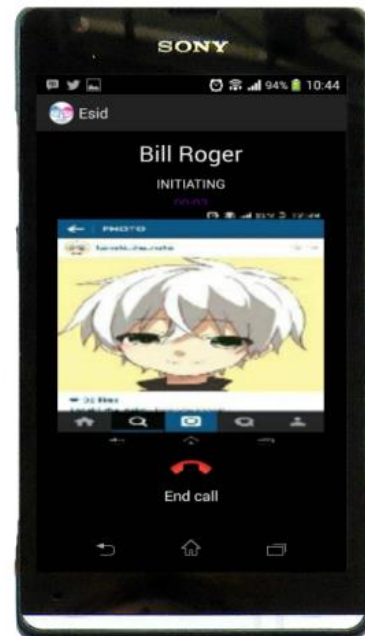
Jika *username* yang dimasukkan pengguna sudah terdaftar pada *database* maka sistem akan menampilkan pesan “username already used” dengan adanya pesan ini, pengguna diharapkan untuk menggunakan *username* lain yang belum terdaftar. Sedangkan *username* yang belum terdaftar pada *database* maka sistem akan menampilkan *form password* dan *re-password*. Ketika *password* dan *re-password* yang diisi memiliki nilai yang sama maka tombol untuk mendaftar akan ditampilkan oleh sistem untuk melakukan proses pendaftaran *username*. Tampilan *signup* pengecekan *username* dan *input password* dapat dilihat pada Gambar 4.



Gambar 4. Halaman Signup Input Password

### 4.3 Pengujian Fitur Free Call

Pengguna dapat menggunakan fitur *voice call* dengan menekan tombol *voice call* yang ada maka secara otomatis sistem akan menampilkan *call initiation state* dimana sistem melakukan panggilan dengan menampilkan *profile picture* dan *display name* dari pengguna yang dituju, serta sebuah tombol untuk mengakhiri panggilan yang sedang dilakukan. Tampilan *call initiation state* dapat dilihat pada Gambar 5.



Gambar 5. Tampilan Call Initiation State

Sistem akan menampilkan *display name* dan *profile picture* dari yang melakukan panggilan kepada pengguna yang dituju serta dua buah tombol yang digunakan untuk pengguna yang dituju apakah

panggilan tersebut ingin diterima atau ditolak. Tampilan *opponent call state* dapat dilihat pada Gambar 6.



Gambar 6. Tampilan *Opponent Call State*

Ketika pengguna yang dituju menerima panggilan yang berlangsung, sistem akan menampilkan tampilan *establish call state* dimana pengguna dapat melakukan percakapan melalui *device* masing-masing. Tampilan *establish call state* dapat dilihat pada Gambar 7.



Gambar 7. Tampilan *Establish Call State*

#### 4.4 Pengujian Fitur *Video Call*

Pengguna dapat menggunakan fitur *video call* dengan menekan tombol *video call* yang ada maka secara otomatis sistem akan menampilkan *call initiation state* dimana sistem melakukan panggilan dengan menampilkan *display name* dari pengguna yang

dituju, serta sebuah tombol untuk mengakhiri panggilan yang sedang dilakukan. Tampilan *call initiation state* dapat dilihat pada Gambar 8.

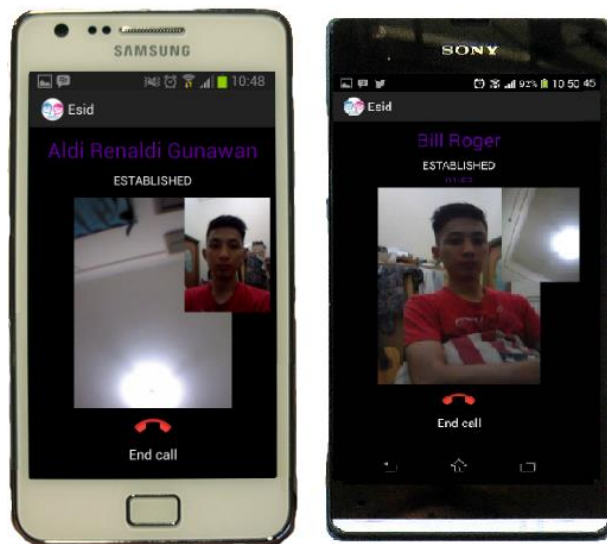


Gambar 8. Tampilan *VideoCall Initiation State*

Sistem akan menampilkan *display name* dan *profile picture* dari yang melakukan panggilan kepada pengguna yang dituju serta dua buah tombol yang digunakan untuk pengguna yang dituju apakah panggilan tersebut ingin diterima atau ditolak.

Ketika pengguna yang dituju menerima panggilan *video* yang berlangsung, sistem akan menampilkan tampilan *establish video call state* yang memiliki dua buah *frame layout* besar dan kecil.

*Frame layout* berukuran besar digunakan untuk menampilkan hasil *video call* dari lawan bicara pengguna dan *frame layout* berukuran kecil digunakan untuk menampilkan *video* yang akan ditampilkan kepada lawan bicara. Tampilan *establish video call state* dapat dilihat pada Gambar 9.



Gambar 9. Tampilan *Establish VideoCall State*

## 4.5 Pengujian Penggunaan *Bandwidth*

Pengujian penggunaan *bandwidth* dilakukan untuk mengetahui besarnya data yang mengalir pada aplikasi ini ketika aplikasi sedang berjalan.

### 4.5.1 Pengujian Penggunaan *Bandwidth* Pada Fitur *Chatting*

Penggunaan *bandwidth* yang digunakan ketika menerima pesan masuk kurang lebih sebesar 200 *byte*, sedangkan ketika mengirim pesan berupa teks *bandwidth* yang digunakan kurang lebih sebesar 1 *kilobyte*. Tampilan penggunaan *bandwidth* ketika pesan masuk dapat dilihat pada Gambar 10.



Gambar 10. Penggunaan *Bandwidth* Ketika Pesan Masuk

Untuk mengirim pesan yang berupa gambar dan *video*, menggunakan *bandwidth* sebesar ukuran asli dari gambar dan *video* yang dikirim.

### 4.5.2 Pengujian Penggunaan *Bandwidth* Pada Fitur *Voice Call*

Penggunaan *bandwidth* untuk fitur *voice call* baik ketika pengguna melakukan panggilan atau menerima panggilan masuk kurang lebih sebesar 23 *kilobyte*. Tampilan penggunaan *bandwidth* untuk fitur *voice call* dapat dilihat pada Gambar 11.



Gambar 11. Penggunaan *Bandwidth* Untuk Fitur *Voice Call*

### 4.5.3 Pengujian Penggunaan *Bandwidth* Pada Fitur *Voice Call*

Penggunaan *bandwidth* untuk fitur *video call* baik ketika pengguna melakukan panggilan atau menerima panggilan masuk kurang lebih sebesar 28 *kilobyte*. Tampilan penggunaan *bandwidth* untuk fitur *videocall* dapat dilihat pada Gambar 12.



Gambar 12. Penggunaan *Bandwidth* Untuk Fitur *Video Call*

## 5. KESIMPULAN

Dari proses yang dilakukan mulai dari perancangan sistem sampai pada tahap pengujian aplikasi yang telah dibuat, dapat diambil kesimpulan sebagai berikut.

- Dari hasil pengujian terhadap beberapa Android *device* dengan versi dan merek yang berbeda-beda antara lain, LG G4, LG G2, Samsung Galaxy S2, Sony Xperia SP, Sony Xperia Z3 Compact, aplikasi berjalan dengan baik pada semua *device*. Namun untuk tampilan, posisi dari layout seperti *textview*, *imageview*, *videoview* dan juga *listview* tidak berjalan dengan baik karena menggunakan satuan *density pixel* (DP).
- Pada *device* Samsung Galaxy S2 dengan Android versi 4.1.2 tidak ditemukan *popup menu button* yang berada pada *menu friends*.
- Dari hasil kuisioner yang telah dibagikan dapat disimpulkan bahwa aplikasi yang telah dibuat mudah dan nyaman untuk digunakan baik dalam proses *login* dan registrasi *username*, proses pertukaran pesan (*chatting*), panggilan suara dan *video* (*Voice call* dan *video call*).

- Bandwidth yang digunakan oleh fitur-fitur pada aplikasi yang telah dibuat dapat bersaing dengan aplikasi-aplikasi serupa.

## 6. DAFTAR PUSTAKA

- [1] Andre, S. 2011. *XMPP protocol*. IETF.  
URI=<https://tools.ietf.org/html/rfc6120#ref-IMP-REQS>
- [2] Ashdown, D. 2011. *Instant Messaging To Account For 75% of Mobile Messaging Traffic By 2018*. Juniper Research Ltd.  
URI=<http://www.juniperresearch.com/viewpressrelease.php?pr=248>
- [3] Herzmann, D. 2014. *Openfire 3.9.3 Has Been Released*. Jive Software Community.  
URI=<https://community.igniterealtime.org/blogs/ignite/2014/05/06/openfire-393-has-been-released>
- [4] Karch, M. 2015. *What is Google Android?*. Google.  
URI=[http://google.about.com/od/socialtoolsfromgoogle/p/android\\_what\\_is.htm](http://google.about.com/od/socialtoolsfromgoogle/p/android_what_is.htm)
- [5] Sinch. 2014. *About Us*. URI=<https://www.sinch.com/about-us/company/>