

Pembuatan Aplikasi Chat dengan Android Berbasis Protokol XMPP

Development of Chat Application with Android Based On XMPP Protocol

Bagus Anantavijaya, Giva Andriana Mutiara, Isa Puncuna

bagus.anantavijaya@gmail.com, giva.andriana@tass.telkomuniversity.ac.id, isapuncuna@yahoo.com

Prodi D3 Teknik Komputer, Fakultas Ilmu Terapan, Universitas Telkom

Abstrak

Salah satu teknologi yang sedang digemari oleh masyarakat adalah *mobile chatting*. Di dalam *chatting* dibutuhkan *server* sebagai penyedia layanan, *device* sebagai perangkat komunikasi, dan *end user* sebagai pengguna. Jika pengguna hanya membutuhkan aplikasi *chatting* pada area lokal, penggunaan *chat* dengan *internet* memakan banyak *resource*. Hal ini bisa diatasi dengan membuat *server chatting* sendiri dengan protokol XMPP yang mudah diimplementasikan. Salah satu aplikasi *server* berbasis XMPP adalah *Openfire* yang memiliki kemudahan dalam konfigurasinya. Keuntungan lainnya adalah privasi dari pengguna dapat dikontrol sendiri. Di era sekarang, pengguna *Android* jumlahnya bertambah secara signifikan sehingga ada banyak sekali pengguna *smartphone Android*, sehingga aplikasi *chatting* ini diimplementasikan pada *Android* agar pengguna tidak sibuk mengganti dan menggunakan *platform mobile* lainnya. Hasil dari proyek ini membuat pengguna dapat berkomunikasi satu ke satu dengan aplikasi *chatting Android*.

Kata kunci: Chatting, XMPP, Android

Abstract

One of technology that favored by people now is mobile chatting. Chatting needs a server, a device to communicate, and end user. If users only need a chatting application based on local area, by using chat with internet will take many resources. The solution is to make a local server with XMPP protocol that easy to implement. One of the most popular XMPP server is Openfire which has simplicity of its configuration. Another advantage is the user privacy can be controlled locally. In this era, Android users growth increase significantly, there are so may Android users, so the chatting application can easily implemented on Android in order to not hard to find another platform. The results of this project is to make users can communicate each other with Android chatting application.

Keywords: Chatting, XMPP, Android

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi telah berkembang dengan sangat pesat. Aplikasi chatting pada awalnya berbasis desktop, namun sekarang sudah bergeser berbasis perangkat bergerak (*mobile*).

Chatting tidak hanya populer kalangan remaja saja namun sekarang ini, sudah merambah kalangan dewasa bahkan dalam ruang lingkup kerja. Kepopuleran *mobile chatting* dapat dikatakan sudah menggeser kepopuleran SMS karena fitur chatting yang banyak dengan biaya yang murah. Dalam ruang lingkup kerja, chatting sangat berguna apabila orang-orang dalam satu gedung atau area lokal ingin berkomunikasi secara cepat namun tidak bisa bertatap muka antar satu dengan yang lainnya. Sebagai contoh, Kepala Bagian Personalia ingin berkomunikasi dengan Kepala Bagian Pemasaran untuk menanyakan apakah dibutuhkan seorang pramuniaga karena Bagian Pemasaran mendapatkan

surat lamaran kerja untuk pramuniaga. Kedua Kepala Bagian ini sibuk menjalankan aktivitas masing-masing, namun untuk berkomunikasi seperti kasus di atas, bertatap muka menjadi tidak efisien karena membutuhkan waktu untuk menemui yang bersangkutan secara langsung, maka aplikasi chat ini dapat membantu untuk mempersingkat waktu dalam komunikasi. Privasi masing-masing pengguna semakin terkontrol dengan aplikasi chatting yang berada dalam genggamannya perangkat bergerak (*mobile*).

Pada umumnya, aplikasi chatting terhubung dengan internet. Namun apabila kebutuhan hanya sebatas area lokal seperti kasus di atas, aplikasi chatting yang terhubung dengan internet akan menghabiskan banyak resources yaitu biaya hosting dan biaya internet. Solusinya adalah dengan membuat server sendiri dan membatasi ruang lingkup chatting hanya sebatas area lokal menjadikan chatting lebih aman dan efisien. Dengan membuat server sendiri, maka perusahaan mempunyai privasi penuh atas komunikasi internal perusahaan.

Setiap aplikasi chatting mempunyai protokol untuk bisa berkomunikasi. Protokol adalah aturan dan standar yang mengatur dan mengizinkan terjadinya sebuah hubungan komunikasi. Protokol chatting mempunyai dua jenis lisensi, proprietary dan open standard.

Extensible Messaging and Presence Protocol (XMPP) merupakan salah satu teknologi protokol yang bersifat open standard. Dengan protokol ini, aplikasi chat mempunyai standar yang baku dan sudah dipakai secara terbuka. Core protokol XMPP hanya mentransmisi aliran data dalam format XML namun dapat mengirim sejumlah payload yang disisipkan pada aliran XML protokol ini. XMPP ini akan dipakai sebagai protokol dalam aplikasi mobile chatting.

1.2 Rumusan Masalah

Bagaimana merancang dan membangun sebuah media chatting untuk mengatasi keterbatasan pada media komunikasi yang tidak memungkinkan orang bertatap muka pada area lokal saat ini.

1.3 Tujuan

Merancang dan membangun sebuah media chatting untuk mengatasi keterbatasan pada media yang tidak memungkinkan orang bertatap muka pada area lokal saat ini dengan protokol yang digunakan yakni *Extensible Messaging and Presence Protocol* (XMPP).

1.4 Batasan Masalah

1. Sistem operasi *mobile Android* yang digunakan untuk menjalankan aplikasi *client chatting* ini menggunakan *Android* versi 4.4.4 (*KitKat*).
2. Pemanfaatan *laptop* sebagai *server* dan *access point*.
3. Tidak membahas keamanan aplikasi.
4. Pengguna harus sama-sama mempunyai aplikasi *client chat*.
5. Diimplementasikan hanya sebatas *local area network*.
6. Aplikasi tidak menggunakan *video chat* dan *voice chat* karena tidak adanya *service* tersebut pada *OpenFire*.
7. Aplikasi tidak menggunakan fitur *attached file*.
8. Aplikasi tidak memiliki fitur verifikasi pengguna.

1.5 Metode Pengerjaan

Metode yang digunakan dalam mengerjakan proyek akhir ini adalah dengan menggunakan sistem *Waterfall*. Adapun tahapan yang dilakukan adalah sebagai berikut:

1. Pengumpulan Kebutuhan

Mempelajari tentang konsep pendukung, petunjuk-petunjuk, dan dokumentasi yang berkaitan dengan proyek akhir ini. Proses pembelajaran materi literatur melalui berbagai pustaka yang berkaitan dengan proyek ini yang terkait dengan *chatting*, XMPP, dan *openFire*.

2. Analisis Sistem

Proses menganalisa kinerja sistem sebelum dibuat, dapat ditarik kesimpulan sementara ketika rancangan sistem selesai dibuat. Setelah itu dilakukan perancangan secara fisik dan logik.

3. Perancangan Sistem

Pada tahap ini proses dilakukan dari segi perangkat keras dan perangkat lunak yang akan dirancang lebih lanjut dari perancangan logik berupa topologi yang akan digunakan dalam aplikasi *client-server*.

4. Coding

Pada tahap ini akan dilakukan *coding* atau pembuatan perangkat lunak dari aplikasi *chatting*. Yakni, pembuatan kode program yang didasarkan pada hasil analisis dan desain di tahapan sebelumnya.

5. Uji Coba

Melakukan pengujian kinerja terhadap *openFire* dan *Mobile chatting* yang sudah dibuat. Kemudian memastikan bahwa semua sudah terkoneksi dengan baik dan berjalan secara optimal.

6. Evaluasi dan Maintenance

Jika terdapat masalah yang diperoleh dari pengujian ini maka akan dilakukan evaluasi dan perbaikan.

2. Tinjauan Pustaka

2.1 Android

Android adalah sebuah sistem operasi untuk perangkat bergerak berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. *Platform Android* sangat lengkap, dapat dilihat dari segi sistem operasi, aplikasi, *tools* pengembangan, serta dukungan komunitas *open source*. [1]

2.2 Chatting

Dalam internet, *chatting* adalah obrolan kepada orang lain yang menggunakan *internet* dalam waktu yang bersamaan. Biasanya, obrolan ini merupakan pertukaran pesan teks yang membutuhkan sebuah *server* sebagai penyedia layanan dan sejumlah pengguna untuk terlibat dalam *chatting* tersebut. *Chatting* dapat dilakukan dengan menggunakan suara (*voice chat*) atau suara dan video. [3]

2.3 Extensible Messaging and Presence Protocol (XMPP)

“The Extensible Messaging and Presence Protocol (XMPP) is an open technology for real time communication, using the Extensible Markup Language (XML) as the base format for exchanging information. In essence, XMPP provides a way to send small pieces of XML from one entity to another in close to real time.” [4, p. 3]

Fitur-fitur yang terdapat dalam XMPP yaitu.

1. *Presence* sebagai status kehadiran pengguna dalam jaringan (*online*).
2. Koneksi dapat dibangun melalui TLS (*Transport Layer Security*) atau tanpa TLS.
3. Terdapat berbagai *extensions* untuk menjalankan *services*.
4. Desentralisasi arsitektur, di mana aplikasi *chat* dapat terhubung dengan *chat* yang berbeda namun sama-sama menggunakan protokol XMPP. Jaringan ini mirip seperti jaringan *e-mail*, di mana paket data dapat diteruskan kepada *server* XMPP lain yang menjangkau tujuannya.

2.4 Extensible Markup Language (XML)

XML adalah sebuah *meta-language* untuk mendeskripsikan data. XML merupakan sebuah cara mempresentasikan data tanpa tergantung kepada sistem. XML berbasis teks, sehingga dapat dengan mudah dipindahkan dari satu sistem komputer ke sistem yang lain. Dengan XML, data dipresentasikan dalam sebuah dokumen yang terstruktur dan telah menjadi sebuah standar. [2]

2.5 Java

Java menurut definisi dari SUN adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer personal ataupun pada lingkungan jaringan. *Java* terdiri atas sebuah mesin interpreter yang diberi nama *Java Virtual Machine (JVM)*. *Java Virtual Machine* inilah yang akan membaca *bytecode* dalam *file*. *Class* dari suatu program sebagai sebuah representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa *java* disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat *Java Virtual Machine*. [6]

3. Analisis dan Perancangan

3.1 Gambaran Sistem

Dalam komunikasi area lokal, sebuah perusahaan masih menggunakan komunikasi tatap muka langsung yang berakibat pada tidak efisiennya waktu dalam informasi yang dikirim dan diterima. Penggunaan *chat* lokal ini membantu mengatasi

masalah tersebut. Setiap pengguna yang memiliki ponsel *Android*, dapat memasang dan menggunakan aplikasi ini untuk berkomunikasi dengan pengguna lain dalam area lokal.



Gambar 3.1 Gambaran Sistem

3.2 Analisis Pengguna

Berikut ini adalah kriteria yang akan dibangun untuk user:

1. User hanya bisa membuat satu akun.
2. Satu user hanya dapat berkomunikasi melalui chatting dengan satu user lainnya.

3.3 Kebutuhan Perangkat Keras dan Perangkat Lunak

3.3.1 Kebutuhan Perangkat Keras

Dalam pembuatan aplikasi ini, pengerjaan aplikasi dengan menggunakan perangkat keras antara lain sebagai berikut:

1. Satu komputer *server* dengan spesifikasi:
 - a. *Processor Intel quad core 2.4 GHz*
 - b. *RAM 2GB*
 - c. *Hard disk 320GB*
 - d. *VGA 512MB*
2. Dua smartphone *Android* sebagai client dengan spesifikasi minimum:
 - a. *Processor 800 MHz*
 - b. *RAM 1 GB*
 - c. *Memory space 20 MB*

3.3.2 Kebutuhan Perangkat Lunak

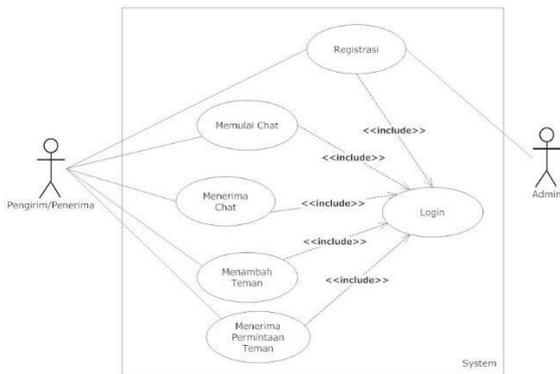
Dalam pembuatan aplikasi ini menggunakan perangkat keras antara lain sebagai berikut:

1. Sistem operasi *Microsoft Windows 7*
2. *Openfire*
3. Sistem operasi *mobile Android 4.4.4*
4. *Android Studio*
5. *Smack JAR*

3.4 Perancangan Unified Modelling Language (UML)

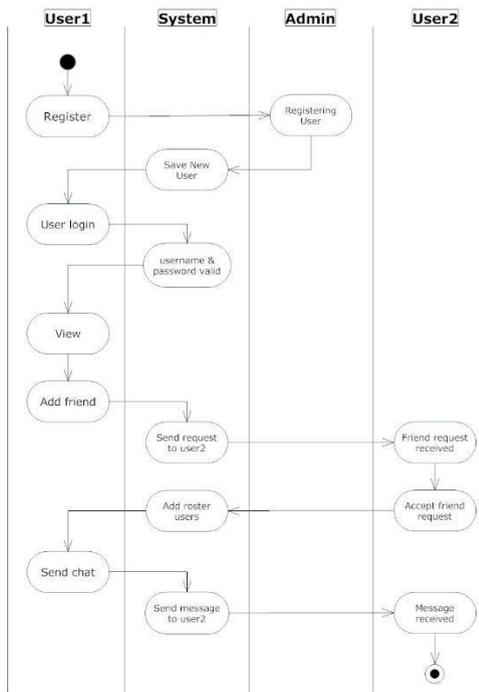
1. Use Case Diagram

Usecase yang dirancang melibatkan ada tiga *actor* beserta perannya masing-masing *actor* tersebut, diantaranya ada *actor* penerima, admin dan pengirim. *Actor* pengirim dan penerima bisa mengakses menu Tambah Teman, Mulai Chat, Terima Permintaan Teman, Terima Chat, dan juga Registrasi. Untuk actor admin memiliki fitur kelola data *user* (pengirim / penerima). Ketiga *actor* tersebut harus memulai aktivitasnya dengan login ke sistem terlebih dahulu.



Gambar 3.2 Use Case Diagram

2. Activity Diagram

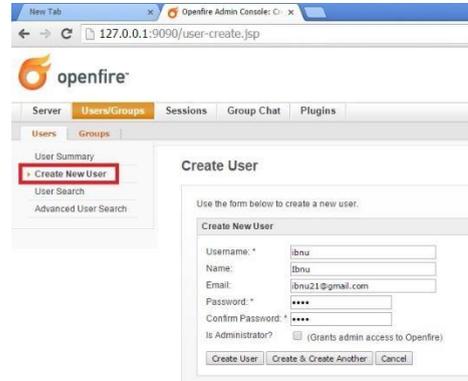


Gambar 3.3 Activity Diagram

4. Implementasi dan Pengujian

4.1 Implementasi

Create New User, isikan *username*, *name*, *email*, *password*, dan *confirm password*.



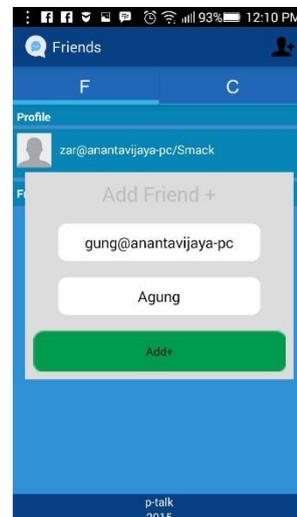
Gambar 4.5 Create User

Login ke aplikasi.



Gambar 4.6 Login Aplikasi

Add Friend.



Gambar 4.7 Add Friend

Chatting



Gambar 4.8 Chatting

4.2 Pengujian

Pengujian dilakukan dengan metode *blackbox testing*[7], pengujian terdiri dari pengujian registrasi, pengujian *login*, pengujian *add friend*, pengujian *chatting*, dan pengujian jangkauan *chatting*.

4.2.1 Pengujian Registrasi

Dalam pengujian registrasi, parameter yang diuji adalah nilai benar atau salah ketika *user* memasukkan *username*. *Username* yang benar adalah *username* yang baru dan tidak pernah terdaftar dalam sistem sebelumnya.

Hasil dari pengujian ini adalah sistem memberikan peringatan kepada *user* ketika *user* memasukkan *username* yang sudah terdaftar pada sistem dan *user* harus mengisi *username* baru sampai sistem menerima *username* tersebut dan mendaftarkan *user* pada sistem.

4.2.2 Pengujian Login

Pengujian *login* bertujuan untuk mengetahui bagaimana sistem *login* bekerja dan hasil yang dicapai ketika *user* memasukkan masukan yang salah.

Form login muncul ketika aplikasi belum mengenali *user* siapa yang menggunakan aplikasi chat. *Form login* terdiri atas dua masukan yaitu *username* dan *password*.

Sistem akan mengecek masukan *username* dan *password* dan membandingkannya dengan *database* sistem. Jika *user* terdaftar dan *user* memasukkan *username* dan *password* dengan benar sesuai data registrasi maka sistem akan mengizinkan *user* untuk masuk ke dalam antarmuka menu utama *chatting*.

Sistem akan memberi peringatan ketika *username* dan/atau *password* yang dimasukkan salah.

4.2.3 Pengujian Add Friend

Dalam memulai *chatting*, *user* harus memilih salah satu *friend* untuk *chatting*. Ketika *user* tidak mempunyai *friend* atau belum mempunyai *friend* yang hendak dilakukan *chatting*, maka *user* harus melakukan proses *add friend*. Syarat bagi *user* yang melakukan *add friend* yaitu mempunyai *username* teman yang hendak ditambah.

Pengujian *add friend* bertujuan untuk mengetahui hasil yang dicapai ketika *user* memasukkan *username* yang salah dalam proses *add friend*.

Hasil yang dicapai adalah sistem akan tetap memasukkan *username* yang salah pada *friend list user*.

4.2.4 Pengujian Chatting

Dalam pengujian *chatting*, parameter yang diuji adalah keberhasilan dalam mengirimkan karakter-karakter yang dikirimkan *user*.

Karakter-karakter yang dikirimkan pada pengujian ini adalah: abcdefghijklmnopqrstuvwxyz 1234567890 ~`!@#\$%^&*)(-_+ =\|][‘;” :/?.>< ©® €£ ™

Hasil dari pengujian ini adalah sistem dapat mengirim karakter-karakter tersebut dan aplikasi penerima dapat menampilkan seluruh karakter-karakter tersebut.

4.2.5 Pengujian Jangkauan Chatting

Dalam pengujian jangkauan *chatting*, dilakukan pada tempat terbuka. Dua *user* melakukan *chatting* dari titik nol terhadap access point lalu *user* menjauh tiap meter.

Berdasarkan hasil dari pengukuran *chatting*, diperoleh jarak terjauh yang terkecil untuk melakukan *chatting* yaitu 24 meter dan jarak terjauh yang terbesar untuk melakukan *chatting* yaitu 37 meter. *Chatting* dimungkinkan ketika *smartphone* menangkap sinyal *WiFi* dengan baik. Apabila sinyal *WiFi* lemah, peluang *chatting* dapat terkirim semakin kecil karena *WiFi* memiliki cakupan jarak (*coverage*).

5. Kesimpulan

Dari hasil pembangunan sistem ini serta dari hasil uji coba yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

- a. Aplikasi dapat digunakan untuk melakukan *chatting* pada area lokal.

- b. Aplikasi menggunakan *WiFi* untuk berkomunikasi dengan *server*.
- c. Registrasi *user* harus menggunakan *username* baru yang belum terdaftar.
- d. Agar dapat masuk ke dalam *menu* utama aplikasi, *user* harus *login* dengan benar sesuai dengan *username* dan *password* yang telah terdaftar dan didapat.
- e. Dalam *menu add friend*, terdapat kekurangan yaitu tidak adanya verifikasi *username* yang akan ditambah oleh *user* apakah terdapat pada data registrasi *server* atau tidak.
- f. Terdapat batas jarak *chatting* dari *access point* yang disebabkan oleh batas cakupan (*coverage*) *WiFi*.
- g. *Xabber* memiliki fitur *chatting* yang lebih lengkap daripada aplikasi XMPP lokal.

6. Daftar Pustaka

- [1] W. F. Ableson, *Android in Action – Second Edition*, Stamford: Manning Publications Co, 2011.
- [2] N. Gramlich, *Android Programming*, Chicago: andbook, 2009.
- [3] Yuhefizar, *10 Jam Menguasai Internet, Teknologi & Aplikasinya*, Jakarta: Elex Media Komputindo, 2008.
- [4] P. S. Andre, *XMPP: The Definitive Guide*, Sebastopol: O'Reilly Media Inc, 2009.
- [5] M. Sharma, *Openfire Administration: A practical step-by-step guide to rolling out a secure*, Brimingham: Packt Publishing Ltd., 2008.
- [6] A. Kadir, *Dasar Pemrograman Java 2*, Yogyakarta: Andi Publisher, 2008.
- [7] R. S. Pressman, *Software engineering: a practitioner's approach*, New York: McGraw-Hill, 2010.

