

## MEMBANGUN APLIKASI *CHATTING* DENGAN PENERJEMAH OTOMATIS BERBASIS *MOBILE*

Firdaus Musyafi<sup>1</sup>, Irawan Afrianto<sup>2</sup>

Program Studi Teknik Informatika - Universitas Komputer Indonesia (UNIKOM)

Jl. Dipati Ukur No.114-116, Bandung 40132

Email : firdausmusyafi@gmail.com<sup>1</sup>, irawan\_afrianto@yahoo.com<sup>2</sup>

### ABSTRAK

Dalam berkomunikasi menggunakan aplikasi *chatting*, terkadang pengguna mengalami kendala saat berhadapan dengan orang asing dan menggunakan bahasa yang tidak mereka kuasai, mereka harus menerjemahkan terlebih dahulu setiap pesan yang diterima maupun pesan yang akan dikirim ke dalam bahasa yang digunakan oleh penerima.

Berdasarkan pada permasalahan tersebut, pada penelitian ini penulis membangun sebuah aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* untuk mengatasi kendala tersebut. Aplikasi ini secara otomatis mendeteksi bahasa dari setiap pesan yang masuk dan menerjemahkannya ke dalam bahasa yang digunakan oleh pengguna. Sistem ini dikembangkan pada perangkat Android dan menggunakan Microsoft Translator API sebagai alat penerjemah. Pengguna dapat menggunakan akun pada Google talk dan Facebook untuk melakukan *chatting* dengan daftar teman yang telah tersedia pada salah satu akun tersebut.

Setelah dilakukan beberapa pengujian *alpha* dan *betha* terhadap aplikasi *chatting* ini, maka dapat disimpulkan bahwa antarmuka pada aplikasi ini mudah digunakan, fasilitas penerjemah yang terdapat pada aplikasi ini dapat memberikan kemudahan bagi pengguna dalam mengatasi kesulitan berbahasa asing dengan lawan bicara dan fasilitas lain seperti *Group Chat* dan *History* telah dapat digunakan secara optimal.

Kata kunci : Aplikasi Chatting, Penerjemah, Mobile

### 1. PENDAHULUAN

Komunikasi merupakan suatu proses sosial yang sangat penting dalam kehidupan manusia. Saat ini, banyak alat komunikasi yang dapat digunakan untuk berinteraksi dan berkomunikasi dari jarak jauh, dan aplikasi *chatting* merupakan salah satu aplikasi yang sering digunakan dalam berkomunikasi.

Aplikasi *Chatting* memungkinkan penggunaanya mengirimkan pesan secara *realtime* ke pengguna lain yang terhubung ke internet. Beberapa penyedia

layanan *chatting* yang sering digunakan antara lain Yahoo Messenger (YM), Facebook chat, Google Talk, MSN. Menurut techcrunch 23 oktober 2012 [1], Facebook telah memiliki 1,01 milyar pengguna aktif dan 604 juta di antaranya menggunakan perangkat *mobile*. Hal ini menunjukkan penggunaan perangkat saat ini meningkat begitu pesat

Dalam berkomunikasi menggunakan aplikasi *chatting*, terkadang pengguna mengalami kendala saat berhadapan dengan orang asing dan menggunakan bahasa yang tidak mereka kuasai, mereka harus menerjemahkan terlebih dahulu setiap pesan yang diterima maupun pesan yang akan dikirim ke dalam bahasa yang digunakan oleh lawan bicara menggunakan kamus ataupun aplikasi penerjemah. Hal ini tentu sangat merepotkan dan menghambat dalam berkomunikasi, terlebih lagi jika lawan bicara lebih dari satu, seperti *chat conference* dan menggunakan bahasa yang berbeda antara satu dengan yang lainnya. Oleh karena itu dibutuhkan suatu sistem yang dapat mengatasi kendala tersebut.

Pada penelitian sebelumnya[3], telah dibangun sebuah aplikasi *chatting* yang dilengkapi dengan fasilitas penerjemah berbasis *web*. Aplikasi ini memungkinkan penggunaanya dapat berkomunikasi dengan pengguna lain menggunakan bahasa yang berbeda, misalnya pengguna dengan bahasa Indonesia berkomunikasi dengan pengguna lain yang menggunakan bahasa Inggris. Pesan teks yang dikirimkan dalam bahasa Indonesia akan diterjemahkan secara otomatis kedalam bahasa Inggris dan sebaliknya. Aplikasi ini berbasis web, dimana memiliki keterbatasan dalam pengaksesannya karena harus diakses dari komputer yang terhubung ke internet. Pengguna juga harus men-setting terlebih dahulu bahasa yang digunakan oleh lawan bicara setiap akan memulai percakapan, sehingga akan menyulitkan pengguna, terlebih jika pengguna belum mengetahui bahasa apa yang digunakan oleh lawan bicara. Penggunaan aplikasi ini hanya sebatas pada *private chatting*, sehingga pengguna akan merasa kesulitan jika akan membuat suatu *chat conference*, dan pada aplikasi ini juga belum tersedia fitur riwayat percakapan yang merupakan fitur dasar dari sebuah aplikasi *chatting*.

Berdasarkan pada uraian pemmasalahan yang telah dijelaskan, pada penelitian ini telah dibangun sebuah aplikasi *chatting* dengan fitur penerjemah otomatis dengan menambahkan fitur-fitur yang belum ada pada penelitian sebelumnya, seperti pendeteksian bahasa secara otomatis, *chat conference*, riwayat percakapan, dan berbasis *mobile*, karena perangkat *mobile* memiliki kelebihan dapat diakses kapan saja dan dimana saja. Pada aplikasi ini pengguna dapat berkomunikasi dengan pengguna lain yang menggunakan bahasa yang berbeda baik *private chat* maupun dalam *chat conference*. Pesan yang dikirimkan akan diterjemahkan terlebih dahulu ke dalam bahasa yang di gunakan oleh lawan bicara, begitu juga sebaliknya.

## 2. TEORI DAN ANALISIS

### 2.1 Instant Messenger (IM)

*Instant Messenger* (IM) atau yang biasa disebut aplikasi *chatting* merupakan perangkat lunak yang memfasilitasi pengiriman pesan singkat (*instant messaging*), suatu bentuk komunikasi secara langsung antara dua orang atau lebih menggunakan teks yang diketik. Teks dikirim melalui komputer yang terhubung melalui sebuah jaringan, misalnya internet. Setelah penggunaan *email* yang mengubah cara orang berkomunikasi dari cara konvensional untuk mengirimkan surat, teknologi pengiriman pesan singkat (*instant messaging*) diciptakan untuk menutupi kelemahan *email* yang kadang-kadang kurang cepat dan tidak waktu nyata (*real-time*).

### 2.2 Application Programming Interface (API)

*Application Programming Interface* (API) atau aplikasi antarmuka pemrograman adalah metode spesifik yang ditentukan oleh sistem operasi komputer atau program aplikasi dimana seorang *programmer* menulis sebuah program aplikasi dapat membuat permintaan dari sistem operasi atau aplikasi lain. API juga merupakan kumpulan fungsi-fungsi *eksternal* yang disediakan *library windows* untuk mengatur kemampuan dan tingkah laku setiap element di *Windows* (dari tampilan di *desktop* hingga alokasi *memory*) sehingga dapat dimanfaatkan suatu program untuk meningkatkan kemampuan program tersebut. API menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah untuk dipahami oleh *programmer* bila dibandingkan dengan *System Calls*, hal ini penting untuk aspek *editing* dan pengembangan, sehingga *programer* dapat mengembangkan sistem dengan mudah. API juga dapat digunakan pada Sistem Operasi mana saja asalkan sudah ada paket-paket API nya.

Keuntungan memprogram dengan menggunakan API di antaranya :

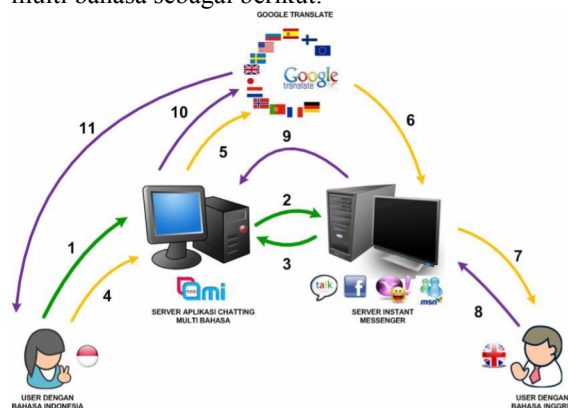
1. Portabilitas. *Programmer* yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter-

*install* API tersebut. Sedangkan *system call* berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.

2. Lebih mudah dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal *editing* dan pengembangan.

### 2.3 Analisis Prosedur yang Sedang Berjalan

Berdasarkan hasil penelitian, terdapat beberapa prosedur yang sedang berjalan pada aplikasi *chatting* multi bahasa sebagai berikut:



Gambar1. Alur Sistem Yang Sedang Berjalan

Adapun penjelasan dari gambaran umum di atas sebagai berikut :

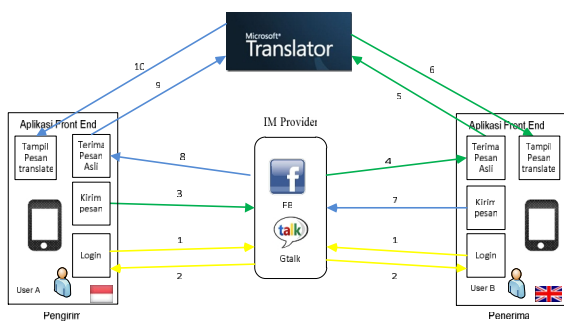
1. Pengguna dengan bahasa Indonesia melakukan login sesuai dengan instant messenger yang dipilih dan data login tersebut ( username dan password ) divalidasi oleh server aplikasi chatting multi bahasa ( AMI ).
2. Data login yang telah divalidasi oleh server AMI akan diotentifikasi oleh server instant messenger.
3. Ketika data login berhasil diotentifikasi oleh server instant messenger maka server instant messenger akan memberikan data-data yang berkaitan dengan user, diantaranya daftar teman dan status teman. Setelah dinyatakan berhasil login, maka server AMI akan menampilkan form chatting.
4. Pengguna dengan bahasa Indonesia mengirimkan pesan dalam bahasa Indonesia dan menetapkan bahasa Inggris (sesuai dengan bahasa yang digunakan lawan bicara) untuk menerjemahkan pesan yang akan dikirim.
5. Pesan dalam bahasa Indonesia diterjemahkan ke dalam bahasa Inggris oleh Google Translate.
6. Pesan yang telah diterjemahkan dikirimkan ke server instant messenger.
7. Pesan dalam bahasa Inggris tersebut dikirimkan oleh server instant messenger ke pihak lawan bicara.

8. Pihak lawan bicara mengirimkan pesan dalam bahasa Inggris yang kemudian diterima oleh server instant messenger.
9. Pesan dalam bahasa Inggris tersebut dikirimkan ke server AMI.
10. Oleh server AMI, pesan tersebut dikirimkan ke Google Translate untuk diterjemahkan kembali ke dalam bahasa Indonesia.
11. Setelah diterjemahkan, pesan dalam bahasa Indonesia tersebut akan dikirimkan ke pengguna.

**2.4 Analisis Sistem yang Akan Dibangun**

Sistem yang akan dibangun adalah suatu aplikasi mobile menggunakan platform Android. Sistem ini terdiri dari dua komponen utama, yaitu aplikasi frontend yang berjalan pada pihak client dan aplikasi backend yang berjalan pada server. Pada sistem ini, aplikasi frontend merupakan komponen yang aktif, dimana user dapat mengirim dan menerima pesan sedangkan aplikasi backend bertindak sebagai penyimpanan data pengguna.

Aplikasi frontend berfungsi sebagai client yang menghubungkan ke server yang dioperasikan oleh vendor jaringan IM (Instant Messenger), seperti Google Talk dan Facebook chat. Aplikasi frontend ini juga berfungsi untuk menghubungkan ke server Microsoft Translator untuk menerjemahkan pesan secara otomatis. Aplikasi backend merupakan aplikasi yang berfungsi untuk memonitoring sistem yang sedang berjalan dan menghapus data pengguna. Gambar 2 menunjukkan gambaran umum tentang sistem yang akan dibangun.



**Gambar 2.** Alur Kerja Sistem Yang Akan Dibangun

Berikut skenario alur kerja sistem yang akan dibangun :

1. User A dan User B adalah pengguna aplikasi. Pertama User A dan User B login terlebih dahulu menggunakan salah satu akun IM yang dimiliki dengan cara memasukkan email dan password untuk akun Gmail dan username dan password untuk akun facebook.
2. Ketika data login berhasil diotentikasi oleh server IM, maka server IM akan mengirimkan data-data yang berkaitan dengan akun user di antaranya daftar teman dan status. User dapat

memilih daftar teman yang sedang online untuk memulai chatting.

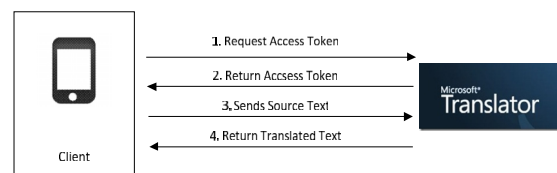
3. User A sebagai pengirim dan User B sebagai penerima. User A mengirim pesan teks ke User B menggunakan bahasa Indonesia
4. Kemudian User B menerima pesan teks dari user A masih dalam bahasa Indonesia.
5. Sistem akan mengecek kode bahasa yang digunakan oleh user B, kemudian pesan dan kode bahasa Inggris akan dikirimkan Microsoft Translator untuk diterjemahkan ke dalam bahasa Inggris.
6. Pesan yang sudah diterjemahkan ke dalam bahasa Inggris akan ditampilkan ke User B.
7. User B sebagai pengirim dan User A sebagai penerima. User B mengirim pesan teks ke User A menggunakan bahasa Inggris
8. Kemudian User A menerima pesan teks dari user B masih dalam bahasa Inggris.
9. Sistem akan mengecek kode bahasa yang digunakan oleh User A, kemudian pesan dan kode bahasa Indonesia akan dikirimkan Microsoft Translator untuk diterjemahkan ke dalam bahasa Indonesia.
10. Pesan yang sudah diterjemahkan ke dalam bahasa Inggris akan ditampilkan ke User B.

**2.5 Analisis Microsoft Translator**

Microsoft Translator API merupakan API yang digunakan fitur penerjemah yang terdapat pada sistem. Secara default API ini akan menerjemahkan semua pesan yang diterima ke dalam bahasa yang digunakan oleh user.

Berikut Langkah-langkah menerjemahkan pesan menggunakan Microsoft Translator API :

1. Client Meminta Access Token ke server
2. Server merespon dengan memberikan Access Token
3. Client mengirim teks sumber untuk diterjemahkan
4. Server membalas dengan teks sumber yang telah diterjemahkan



**Gambar 3.** Alur Penggunaan Microsoft Translator API

1. Akses Token

Client memulai dengan meminta *access token* ke server, kemudian server akan merespon dengan memberikan *access token* ke Client. Parameter untuk permintaan token menggunakan *URL-encoded* dan dikirimkan melalui *HTTP request body*. Berikut parameter-parameter yang digunakan dalam meminta *access token*

**Tabel 1.** *Token Request Input Parameters*

Parameter	Deskripsi
<i>client_id</i>	<i>Client id</i> yang didapatkan ketika mendaftarkan aplikasi ke <i>Azure DataMarket</i>
<i>client_secret</i>	<i>Client secret</i> yang didapatkan ketika mendaftarkan aplikasi ke <i>Azure DataMarket</i>
<i>scope</i>	Gunakan <a href="http://API.microsofttranslator.com">http://API.microsofttranslator.com</a> URL sebagai nilai lingkup untuk Penerjemah <i>API Microsoft</i> .
<i>grant_type</i>	Gunakan " <i>client_credentials</i> " sebagai nilai <i>grant_type</i> untuk Penerjemah <i>API Microsoft</i> .

Respon untuk permintaan akses token berisi akses token yang dapat Anda gunakan untuk mengakses *Translator API Microsoft*. Respon berupa *JSON-encoded* dengan beberapa atribut *output* sebagai berikut.

**Tabel 2.** *Token Request Output Properties*

Parameter	Deskripsi
<i>access_token</i>	Akses token yang dapat Anda gunakan untuk otentikasi Anda akses ke <i>Microsoft Translator API</i> .
<i>token_type</i>	Format akses token.
<i>expires_in</i>	Jumlah detik dimana akses token valid.
<i>scope</i>	<i>Domain</i> dimana token ini berlaku. Untuk <i>Microsoft Translator API</i> , domain adalah <a href="http://API.microsofttranslator.com">http://API.microsofttranslator.com</a> .

## 2. Menerjemahkan Pesan

Setelah *client* mendapatkan akses token, *client* dapat memulai menerjemahkan pesan teks dari suatu bahasa ke bahasa lain. Uri yang digunakan untuk mengirimkan pesan memiliki format sebagai berikut :

<http://API.microsofttranslator.com/V2/Ajax.svc/Translate>

Parameter-parameter *query* yang dapat digunakan pada *API Microsoft Translator* di cantumkan pada table berikut.

**Tabel 3.** Parameter *Query Microsoft Translator API*

Parameter	Deskripsi
<i>appid</i>	Diperlukan. Sebuah <i>string</i> yang berisi " <i>Bearer</i> " + "" + akses token.
<i>text</i>	Diperlukan. Sebuah <i>string</i> yang mewakili teks untuk diterjemahkan. Ukuran teks harus tidak melebihi 10000 karakter.
<i>from</i>	Opsional. Sebuah <i>string</i> yang mewakili kode bahasa teks terjemahan. Jika dibiarkan kosong respon akan memberikan hasil dari

	<i>language auto-detection</i>
<i>to</i>	Diperlukan. Sebuah <i>string</i> yang mewakili kode bahasa untuk menerjemahkan teks ke bahasa target.
<i>contentType</i>	Diperlukan. Format teks yang diterjemahkan. Format yang didukung adalah " <i>text / plain</i> " dan " <i>text / html</i> ".
<i>category</i>	Opsional. Sebuah <i>string</i> yang berisi kategori ( <i>domain</i> ) terjemahan. Defaultnya adalah "umum".

Jika *request* sukses, maka *server* akan memberikan sebuah nilai *string* yang mewakili teks yang diterjemahkan.

## 3. IMPLEMENTASI

### 3.1 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan Aplikasi *Chatting* dengan Penerjemah Otomatis Berbasis *Mobile* dapat dilihat pada tabel 4.

**Tabel 4.** Implementasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	<i>Windows 7, Android OS 2.2 Froyo</i>
Bahasa Pemrograman	<i>PHP,Java</i>
<i>Web Server</i>	<i>XMPP, Localhost</i>
<i>Database server</i>	<i>MySQL</i>
<i>Web Browser</i>	<i>Mozilla Firefox, Google Chrome</i>
<i>Tools</i> pembangun	<i>Eclipse, Android ADT</i>

### 3.2 Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan Aplikasi *Chatting* dengan Penerjemah Otomatis Berbasis *Mobile* adalah seperti yang tercantum pada tabel 5 Dan Spesifikasi perangkat keras *mobile* yang digunakan dalam pembangunan Aplikasi tercantum pada tabel 6.

**Tabel 5.** Implementasi Perangkat Keras

Perangkat Keras	Spesifikasi
<i>Processor</i>	<i>Intel Pentium Core 2 Duo 2.93Ghz</i>
<i>Monitor</i>	CRT Standar
<i>RAM</i>	2 GB
<i>VGA</i>	512 MB
<i>Hardisk</i>	250 GB

**Tabel 6.** Implementasi Perangkat Keras *Mobile*

Perangkat Keras	Spesifikasi
<i>CPU</i>	600 MHz ARMv6
<i>Display</i>	<i>TFT capacitive touchscreen</i>
<i>Storage</i>	160 MB <i>storage</i>
RAM	384MB RAM



#### 4. PENGUJIAN

Pengujian terhadap aplikasi ini menggunakan metode *Alpha* dan *Beta*. Proses dan hasil pengujian sebagai berikut :

##### a. Pengujian *Alpha*

Pengujian fungsional yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak.

Pengujian dilakukan dengan menguji setiap proses use case diagram dan kemungkinan kesalahan yang terjadi untuk setiap proses. Pengujian ini dilakukan secara *Black Box*, yaitu dilakukan dengan memperhatikan masukan ke sistem dan keluaran dari sistem. Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 7.

Tabel 7. Rencana Pengujian Aplikasi

Menu yang diuji	Detail pengujian	Jenis Uji
Login ke sistem	Verifikasi data login user ke database server	<i>Black box</i>
Register	Membuat sebuah akun baru pada sistem	<i>Black box</i>
Login Via Google Talk	Verifikasi data login user ke server Google Talk	<i>Black box</i>
Login Via Facebook	Verifikasi data login user ke server Facebook	<i>Black box</i>
Add Account	Menambah akun IM yang digunakan	<i>Black box</i>
Mengirim Pesan	Memilih teman pada daftar teman yang tersedia	<i>Black box</i>
	Mengirimkan pesan menggunakan bahasa yang dikuasai	
Menerima Pesan	Menerima pesan dan menerjemahkannya	<i>Black box</i>
Group Chat	Membuat sebuah chat <i>converence</i>	<i>Black box</i>

Berdasarkan hasil pengujian dapat disimpulkan bahwa semua fungsionalitas pada aplikasi *chatting* dengan penerjemah otomatis ini dapat berfungsi dengan baik.

##### b. Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas sistem. Pada pengujian *beta* ini, pengujian akan dilakukan secara objektif dimana aplikasi diuji secara langsung kepada beberapa pengguna. Para calon pengguna akan diberikan kuesioner. Sampel yang diambil sebanyak 20 orang.

Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk nantinya dapat diambil kesimpulan penilaian dari para pengguna terhadap aplikasi *chatting* dengan penerjemah otomatis ini.

Kuesioner akan terdiri dari 7 pertanyaan dan memiliki tiga pilihan yang mewakili dari tujuan akhir yang ingin dicapai dalam pembangunan aplikasi ini,

Kesimpulan pengujian *beta* adalah sebagai berikut :

- 1) 85% dari yang diharapkan 100%, penilaian terhadap aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan menarik.
- 2) 84% dari yang diharapkan 100%, penilaian terhadap antarmuka aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan mudah digunakan.
- 3) 87% dari yang diharapkan 100%, penilaian terhadap aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan membantu dalam berkomunikasi lewat *chatting*.
- 4) 92% dari yang diharapkan 100%, penilaian terhadap penerjemah pada aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan sangat membantu dalam mengatasi kesulitan penggunaan bahasa asing.
- 5) 78% dari yang diharapkan 100%, penilaian terhadap akses login ke Google Talk dan Facebook pada aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan membantu.
- 6) 80% dari yang diharapkan 100%, penilaian terhadap fasilitas penerjemah pada aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan dapat digunakan secara optimal.
- 7) 81% dari yang diharapkan 100%, penilaian terhadap aplikasi *chatting* dengan penerjemah otomatis berbasis *mobile* dapat dikategorikan perlu dikembangkan.

#### 5. KESIMPULAN DAN SARAN

##### 5.1 Kesimpulan

Setelah melakukan analisis, perancangan, dan pengujian, maka dapat diperoleh kesimpulan sebagai berikut :

1. Aplikasi *chatting* dengan penerjemah otomatis ini dapat memberikan solusi atas kendala pengguna dalam penggunaan bahasa asing dengan lawan bicaranya karena dilengkapi fitur penerjemah bahasa otomatis
2. Aplikasi *chatting* dengan penerjemah otomatis ini memberikan kemudahan dalam peaksesannya, karena aplikasi ini berbasis *mobile* dapat diakses dimana saja dan kapan saja.
3. Aplikasi *chatting* dengan penerjemah otomatis ini memberikan kemudahan pada penggunaannya dalam menentukan bahasa yang digunakan oleh lawan bicara karena dilengkapi dengan fitur pendeteksi bahasa.
4. Aplikasi *chatting* dengan penerjemah otomatis ini memberikan kemudahan dalam

berkomunikasi *chatting* ke beberapa orang yang berbeda bahasa karena dilengkapi dengan fitur *Group Chat*

5. Aplikasi *chatting* dengan penerjemah otomatis ini memberikan kemudahan dalam melihat riwayat percakapan, karena riwayat percakapan pesan asli dan pesan yang diterjemahkan disimpan kedalam *database*.

### 5.2 Saran

Untuk pengembangan aplikasi *chatting* dengan penerjemah otomatis ini, terdapat beberapa saran yang dapat dilakukan, antara lain:

1. Penambahan Fitur *Add Friend*, dimana user dapat menambah daftar teman yang menggunakan aplikasi *chatting* dengan penerjemah otomatis.
2. Penambahan Fitur Cari teman, dengan fitur ini pengguna bisa mencari teman yang berada di berbagai tempat sehingga pengguna dapat lebih mudah dalam mencari teman yang menggunakan bahasa asing
3. Menambahkan lebih banyak IM Provider, sehingga pengguna dapat melakukan *chatting* dengan daftar teman dari berbagai account IM.
4. Menambahkan fitur multimedia seperti gambar, audio, video, emoticon agar suasana komunikasi *chatting* lebih hidup.

### DAFTAR PUSTAKA

- [1] *Facebook Announces Monthly Active Users Were At 1.01 Billion As Of September 30th, An Increase Of 26% Year-Over-Year*, <http://techcrunch.com/2012/10/23/facebook-announces-monthly-active-users-were-at-1-01-billion-as-of-september-30th/>, diakses tanggal 13 juli 2013
- [2] *Translator Languages Code*, <http://msdn.microsoft.com/en-us/library/hh456380.aspx>, diakses 19 juni 2013
- [3] Wulandari Nila. (2013). "*Rancang Bangun Aplikasi Chatting Multi Bahasa*". Bandung: Universitas Komputer Indonesia